



**UNIVERSITÀ
DEGLI STUDI
DI TRIESTE**

UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXXV CICLO DEL DOTTORATO DI RICERCA IN

FISICA

**RELATIVISTIC EFFECTS IN COSMOLOGICAL
SIMULATIONS**

Settore scientifico-disciplinare: **Astrofisica**

**DOTTORANDO / A
Eduardo Quintana Miranda**

Eduardo Quintana

**COORDINATORE
PROF. Francesco Longo**

Francesco Longo

**SUPERVISORE DI TESI
PROF. Pierluigi Monaco**

Pierluigi Monaco

**CO-SUPERVISORE DI TESI
PROF. Luca Tornatore**

Luca Tornatore

ANNO ACCADEMICO 2021/2022

Abstract

General relativity (GR) is currently our best theory for gravitation. It has been confirmed by experiments from solar system tests like planetary orbits, to the recent gravitational wave detection and the imaging of the event horizon of M87* and SgrA* black holes. In cosmology, GR is confirmed at very large scales by observational tests like strong gravitational lensing or redshift-space distortions. The Λ CDM model (Λ to indicate a cosmological constant in Einstein equations and CDM for Cold Dark Matter) is indeed based on GR plus the cosmological principle and an early inflation to create primordial perturbations.

However, GR is not able to explain the nature of two of the most prominent ingredients of the cosmological model: Dark Matter (DM) and Dark Energy (DE). The first is responsible for the formation of cosmological structures such as the DM halos that host galaxies, while the second causes an accelerated expansion of the universe at the present age. Seeking to find an explanation for DM and DE is what drives the research for alternatives theories to GR. The next generation of large scale surveys, such as Euclid, Rubin-LSST, SKAO, DESI, Roman and Spherex will provide more data about the properties of DM and DE, that will hopefully help us constrain those theories.

Starting from a theoretical model, one common approach to produce predictions that later can be compared to observations consists in running numerical simulations. However, simulations of the formation of cosmological large scale structure have historically neglected the relativistic nature of the gravitational interaction, by fixing a homogeneous Friedman-Lemaitre-Robertson-Walker (FLRW) space-time metric and letting matter particles interact through a Newtonian gravitational potential. This Newtonian approach dismisses 5 out of 6 degrees of freedom of the relativistic gravitational field (these degrees of freedom are the components of the space-time metric after a system of coordinates is fixed), and thus some little signals may be washed out by the Newtonian approximation in the first place, that would otherwise help draw a line between different theories of gravity.

Recent developments have brought forward general relativistic codes such as GEVOLUTION and GRAMSES, that do take into account the space-time metric components as a dynamic field sourced by inhomogeneities in the energy-momentum tensor. With these tools one can study relativistic effects from first principles that have no Newtonian counterpart, such as the motion of relativistic species (e.g. relativistic neutrinos), weak lensing and integrated Sachs-Wolf (ISW) effect, that paired with the data from a detailed galaxy survey (like the upcoming Euclid mission) can help to test GR on cosmological scales.

In this PhD thesis I present the implementation of a method to study the evolution

of the large scale structure in a weak-field approximation of GR. Our starting point is the assumption that we can reproduce the classical evolution of the universe with an N -body code in which small scales—below 1 Mpc—interactions are particle-to-particle in the Newtonian limit, while in the large scales these interaction are mediated through the relativistic gravitational field in the weak-field approximation.

We have implemented this idea by first redesigning `GEVOLUTION` code from a monolithic application into a relativistic Particle-Mesh (PM) library `LIBGEVOLUTION`. Then we have modified `GADGET-4`, a Newtonian `TREEPM` code, adding `LIBGEVOLUTION` as a plug-in replacement to its original Newtonian PM. This project goes by the name of `GRGADGET`.

The advantage of a combined Tree+PM approach with respect to a pure PM method lies in the fact that we can simulate huge cubic boxes, with a length of the order of the $2\text{ Gpc}/h$, representing a portion of the visible universe and still be able to resolve structures where dark matter halos hosting galaxies are formed on scales below $100\text{ kpc}/h$ using reasonable computational resources. Furthermore one can scale the size of the box without necessarily needing to increase the memory requirements or sacrificing the small scale resolution. Running `GRGADGET` we have also realized that failing to resolve the non-linear structures, for example in the case of a pure PM code, would underestimate the amplitude of the highest Fourier modes of the relativistic fields by more that a 30% factor.

This thesis is organized in 6 chapters. Chapter 1 will present the state of the art of modern cosmology, the main questions about the nature of dark energy and dark matter, how different theories of gravity might provide the answers and the need for cosmological simulations that take into account relativistic effects. Chapter 2 will give a detailed review of the theory of perturbations in general relativity. Chapter 3 will describe the numerical methods used in this thesis to simulate the evolution of the universe, the concept behind N -body simulations, the Particle-Mesh and `TREEPM` methods. Some technical characteristics of `GEVOLUTION` and `GADGET-4` codes that are relevant for this work will be addressed in this chapter as well. In chapter 4 there will be an exhaustive documentation of `GRGADGET`: its procedural and memory models, its advantages over `GEVOLUTION` and limitations. Chapter 5 will be dedicated to the validation of the code through medium size runs, testing the properties of the matter distribution through the two-point correlation function compared to the Newtonian `GADGET-4` and `GEVOLUTION` simulations. Chapter 6 will be dedicated to the conclusions of the work.

Chapters 4 and 5 present the original work produced by the author, already published in ArXiv (Quintana-Miranda, Monaco, and Tornatore 2023) and it is waiting for review on the MNRAS journal.

Notation

The notation convention in this thesis is as follows:

- No natural units are adopted. The work in this thesis is highly technical in simulation codes, and computer programs are better implemented to work on variables which do not retain physical dimensions. However, whenever one must translate simulation results into physical predictions units must be recovered and very often the constants that are not visible when natural units are in place constitute the conversion factor from code units.
- The metric signature is $(-, +, +, +)$ as it is customary in cosmology.
- Einstein's summation convention is used where repeated greek indexes indicate a sum over the values 0, 1, 2 and 3, while repeated latin indexes indicate a sum over the values 1, 2 and 3.
- An index after a comma indicate partial derivative, while an index after a semi-colon indicate covariant derivative with respect to the affine connection $(\Gamma^\mu_{\alpha\beta})$ of the space-time metric $(g_{\mu\nu})$ unless otherwise stated. Also a vertical bar $|$ is used to indicate covariant derivatives with respect to the affine connection $(\Gamma^{(3)k}_{ij})$ related from the spatial metric (γ_{ij}) .
- The Riemann tensor for a connection Γ is computed as:

$$R^\mu_{\alpha\nu\beta} = \Gamma^\mu_{\alpha\beta,\nu} - \Gamma^\mu_{\alpha\nu,\beta} + \Gamma^\mu_{\lambda\nu} \Gamma^\lambda_{\alpha\beta} - \Gamma^\mu_{\lambda\beta} \Gamma^\lambda_{\alpha\nu}. \quad (1)$$

- The Ricci tensor is computed from the Riemann tensor as:

$$R_{\alpha\beta} = R^\nu_{\alpha\nu\beta}. \quad (2)$$

- In our convention Einstein equations are written as:

$$R^\mu_{\nu} - \frac{R}{2} g^\mu_{\nu} + \Lambda g^\mu_{\nu} = -\frac{8\pi G}{c^4} T^\mu_{\nu}. \quad (3)$$

- Our sign convention picks an extra ‘-’ sign in Einstein equations because we define the sign of the Energy-Momentum tensor such that the component T^0_0 corresponds to energy density and it is positive for an ordinary set of particles.

Contents

Contents	iv
1 Introduction	1
1.1 Cosmology as the ultimate test for GR	1
1.2 GR in Cosmological Simulations	3
2 Theory of Cosmological Perturbations	5
2.1 Cosmology in the homogeneous universe	5
2.2 Perturbation theory in differential geometry	10
2.3 Gauges	15
2.4 Field Equations	16
2.5 Geodesics	18
2.6 Energy Momentum Tensor	20
3 Numerical methods in cosmology	23
3.1 <i>N</i> -body cosmological simulations	23
3.2 Particle-Mesh	28
3.3 TREEPM	35
3.4 GADGET-4	36
3.5 Gevolution	39
4 GrGadget	41
4.1 LATFIELD2	42
4.2 Domain decomposition	43
4.3 Concept and design rules	44
4.4 Classes and memory model	46
4.5 Boosting Gevolution	47
4.6 GrGadget code	49
5 Code validation	53
5.1 GEVOLUTION and GADGET-4 original codes	54
5.2 Newtonian forces	56
5.3 Relativistic Cosmology with GRGADGET.	57
5.4 Performance	64
6 Conclusions	71

CONTENTS

v

A Perturbations of the Ricci tensor	75
B Nyquist-Shannon Theorem	81
Bibliography	85

Introduction

1.1 Cosmology as the ultimate test for GR

Einstein’s theory of General Relativity (GR) is said to be the most elegant theory in physics. It changed our understanding of space-time from an invariant flat perspective to a dynamic and mutable geometric entity that bends with the presence of matter and energy. In GR, gravity is no longer a distant interaction force between massive objects, but a manifestation of curvature in the geometry of space-time. Regardless of its beauty, it is our best theory of gravity confirmed by experiments. GR has been successful in predicting all kinds of phenomena like the precession of planetary orbits, the deflection of light and gravitational redshift in solar system tests, weak and strong gravitational lensing, and the more recent confirmations with the detection of gravitational waves (Abbott et al. 2016) and the direct imaging of a black hole event horizon (Event Horizon Telescope Collaboration et al. 2019).

Shortly after the discovery of GR, Friedmann and later Lemaître, Robertson and Walker, derived a metric for spacetime that solves Einstein equations in the context of an homogeneous and isotropic universe; the so called FLRW metric. Today the standard model of cosmology, Λ CDM, uses that metric and it has been successful in predicting and explaining a whole range of observations from the Hubble expansion, the Big Bang Nucleosynthesis (BBN) to the Cosmic Microwave Background (CMB).

Observations of the redshift of type Ia supernovae combined with the measurements of the Baryonic Acoustic Oscillations (BAO) by the Sloan Digital Sky Survey (SDSS) and the temperature map of the CMB by the Planck satellite provide evidence of the geometric flatness of the universe and its accelerated expansion (Riess et al. 1998; Perlmutter et al. 1999; Percival et al. 2010; Suzuki et al. 2012; Planck Collaboration et al. 2020). Λ CDM model explains these observations establishing that: (i) only the 15% of the matter contents of the universe is in the form of baryonic matter—i.e. ordinary matter composed by protons, neutrons and electrons; that it is found in the form of gas, dust, stars and planets—while the other 85% is in the form of some Dark Matter (DM) that does not interact with photons and its presence can only be measured through its gravitational effect, and (ii) there is some mysterious form of energy called Dark Energy (DE) that has negative pressure and drives the accelerated expansion of the universe. Figure 1.1 shows the *cosmic triangle* of evidence for the geometric structure and matter/energy contents of the Universe. The most recent results from Planck’s satellite, have confirmed these conclusions with higher confidence (Planck Collaboration et al. 2020) see table 1.1.

Even though there is compelling evidence for the existence of DM and DE, there are

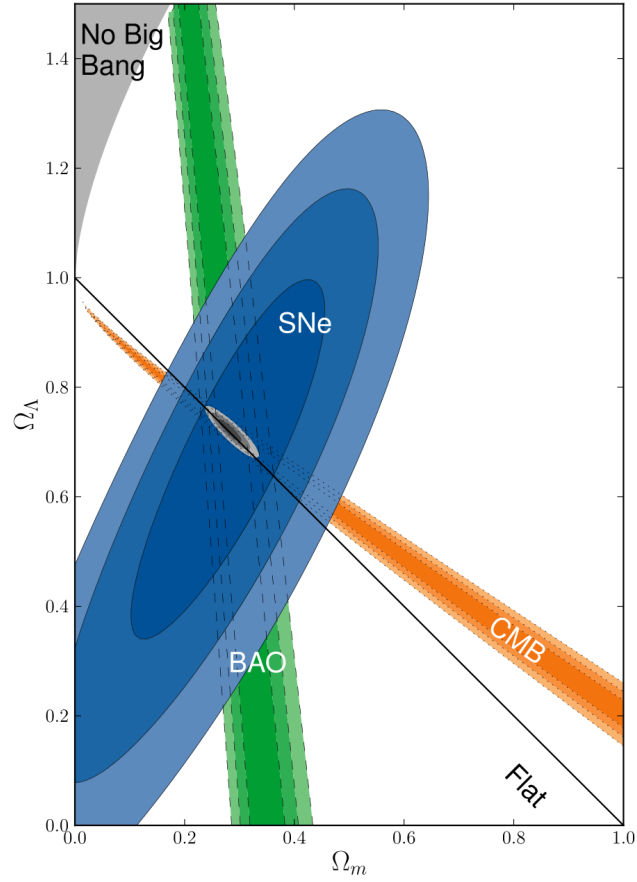


Figure 1.1: Fit on the $(\Omega_m, \Omega_\Lambda)$ plane of the Λ CDM model from SNe Ia combined with the constraints from BAO and CMB (Suzuki et al. 2012).

Parameter	Value
$\Omega_b h^2$	0.0224 ± 0.0001
$\Omega_c h^2$	0.1193 ± 0.0009
Ω_Λ	0.688 ± 0.005
Ω_m	0.311 ± 0.005

Table 1.1: Cosmological parameters extracted from Planck Collaboration et al. 2020.

no answers about their physical nature within the standard model of particle physics and GR itself unless we introduce the cosmological constant Λ into Einstein equations. The need for explanations drives the research towards models of modified GR introducing modifications to the action in the form a function of the Riemann scalar, $f(R)$ theories, or by proposing the existence of additional gravitational degrees of freedom (Clifton et al. 2012).

Any good candidate for a modified GR theory of gravity must produce the same predictions of GR at many scales, like Solar system tests, but there are still regimes in the large scales in which GR has not been tested with enough accuracy and that leaves room for whole classes of alternatives (Alam et al. 2020).

Upcoming surveys will provide more data about the observed properties of DM and DE by probing the large-scale distribution of galaxy clustering. The Dark Energy Survey (DES¹), for instance, is an optical and near infrared survey from the ground that will measure the distribution of an estimated number of 300 million galaxies in the redshift range $0.1 < z < 1$ and from their angular power spectrum it will be able to measure the DE parameters in its equation of state (Krause et al. 2017). Euclid² satellite, expected to be launched in 2023, will measure the shape of over 10^9 galaxies to map the DM distribution in the Universe through weak gravitational lensing effects, it will carry out a spectroscopic redshift survey of $\sim 50 \times 10^6$ galaxies over 75% of the lifetime of the Universe; with this data the theory group aims to measure the equation of state of DE and the growth factor as a function of redshift to constraint modified gravity models (Laureijs et al. 2011).

On the scales of the order of the Hubble radius relativistic effects start to become detectable on the datasets coming from galaxy surveys. They include: gravitational redshift (also known as Sachs-Wolfe and Integrated Sachs-Wolfe effects), Shapiro time delay and weak gravitational lensing (Borzyszkowski, Bertacca, and Porciani 2017; Yoo, Fitzpatrick, and Zaldarriaga 2009). These arise from the fact that we observe far away galaxies in our past lightcone, and the photons that reach our telescope travel through space-time following geodesic paths that differ from straight lines because of the presence of matter, energy or gravitational waves that change the shape of the space-time. If not taken into account, these effects become the source of systematic errors in the estimated observables and a missed opportunity to test GR on cosmological scales.

1.2 GR in Cosmological Simulations

Computer simulation are used to fill the gap between theory and observations. For the very complex process of non-linear clustering of matter under gravitational interaction in cosmology, there is no analytical description, and simulations are our only predictive tool. From these simulations, one can obtain virtual realizations of galaxy catalogs that try to resemble the observations in order to calibrate the data processing pipeline of the surveys, obtain biases and covariance matrices of estimators, and they are used also as forecasting tools to select observables that maximize the signal to noise ratio or the discriminative power over a particular theory of interest.

Until recently, most simulations were using Newton's law of gravitation. The effects of GR can be added *a posteriori* using ray tracing techniques on the lightcone assuming the light travels in straight lines with respect to the FLRW Universe; for example LIGER method is based on this approach (Borzyszkowski, Bertacca, and Porciani 2017). In the post-processing, gravity is represented by the Newtonian potential that satisfies Poisson

¹www.darkenergysurvey.org

²sci.esa.int/web/euclid

equation, and from the knowledge that there exists a map from the Newtonian simulation to a GR state of the Universe in Poisson gauge (Chisari and Zaldarriaga 2011) one can justify and obtain all kinds of purely GR effects that have no Newtonian explanation such as redshift-space distortions and weak gravitational lensing.

However, it is desirable to be able to obtain those effects from first principles and self consistent fully relativistic simulations. It wouldn't be possible to study the evolution and imprint of relativistic species, in the large scale structure by using Newtonian simulations in a self consistent way. The non relativistic setups are also missing 5 out of 6 degrees of freedom of the gravitational field that could produce high order corrections to the observable quantities; these if not taken into account in future precision observations, might become the source of systematic errors. Newtonian simulation also lack the power to evaluate the effects of GR in the clustering of matter in the large scales, which is relevant to test modified gravity theories that predict an enhancement of the clustering with respect to GR.

The most recent efforts to study relativistic effects in N -body simulations have produced two codes called GRAMSES (Barrera-Hinojosa and Li 2020) and GEVOLUTION (Adamek et al. 2016).

GRAMSES uses the Arnowitt-Deser-Misner formalism of GR in a background independent framework. The code uses Gauss-Seidel relaxation method to solve the partial differential equations of the gravitational sector in an adaptive mesh, and it is parallelized for Message Passing Interface (MPI) that allows it to run massive simulation sizes in supercomputers. GRAMSES has been used to study the weak lensing effects of the vector gravitomagnetic field (frame dragging) (Barrera-Hinojosa, Li, and Cai 2022).

GEVOLUTION is also parallelized with the MPI paradigm, but unlike GRAMSES, GR is solved through space-time metric perturbations with respect to a FLRW background, and Einstein equations are solved using Fourier methods on a fixed size mesh. In the few years since its publication, GEVOLUTION has been used to address all kind of problems that are inherent to GR such as the effects of early radiation and massive neutrinos in the clustering at large scales (Adamek et al. 2017; Adamek, Durrer, and Kunz 2017), the generation of vorticity and its relation with the frame dragging potential (Jelic-Cizmek et al. 2018), and the backreaction problem (Adamek et al. 2019).

The work described in this thesis seeks to extend GEVOLUTION code capabilities beyond the limitations imposed by the static resolution of the mesh. We propose a TREEPM approach to numerical GR in cosmology using N -body simulations in which the dynamics of the particles are driven by a relativistic mesh for the large scales and a Newtonian treatment for short range interactions (Quintana-Miranda, Monaco, and Tornatore 2023). This method is based on the assumption that in the context of cosmology, where matter is diluted in huge volumes of space and non-relativistic species are predominant, the Newtonian limit approximation works very well for scales well below the Hubble horizon. In our implementation we use GADGET-4 (Springel et al. 2021) code as the main driver of the simulation and GEVOLUTION code is modified to work as a Particle-Mesh library substituting GADGET-4's Newtonian mesh. This approach has the advantage of producing accurate non-linear clustering, which is challenging for pure mesh codes.

Theory of Cosmological Perturbations

In 1964 Arno Penzias and Robert Wilson discovered a radio signal whose origin could not be attributed to anthropic activity or a galactic source. It was coming from all directions in the sky. Later, detailed measurements by COBE satellite confirmed that this signal, now named Cosmic Microwave Background (CMB), has an almost perfect thermal black body spectral distribution with a temperature of $2.7259 \pm 0.0006\text{K}$ (Fixsen 2009), suggesting that the CMB is remnant of the of past state of the universe when it was hotter and denser than it is today. The tiny anisotropy of the CMB temperature T field, $\Delta T/T \approx 10^{-5}$, is one of the many observational confirmations that as seen from Earth the universe appears to be isotropic.

One might think that the Earth occupies a special position in the universe from which it looks spherically symmetric. But since the time of Copernicus, we have learned that the Sun and not Earth is in the center of the Solar system. The Solar system itself is located in some ordinary average location in the Milky Way galaxy, and the Milky Way is just one of the many neighboring spiral galaxies. There is no reason to believe that the isotropy observed is due because the Milky Way occupies a special location in the universe. The *cosmological principle* summarizes this Copernican assumption about the structure of the universe:

Viewed on a sufficiently large scale, the properties of the Universe are the same for all observers and in all directions.

Modern cosmology is built upon this assumption and this is enough to provide a fair understanding of the universe at large and its history.

However, there are inhomogeneities in the universe, some regions are denser than others, there are structures in the large scales that resemble 3 dimensional dense networks and there are almost empty regions of space in between. To properly model the dynamics of such complex universe and the propagation of light signals in it one must develop a general relativistic mathematical description that goes beyond the homogeneity assumption. This chapter is dedicated to explaining the fundamental concepts of the theory of perturbations in cosmology and obtain the main equations that we will use in the rest of the thesis.

2.1 Cosmology in the homogeneous universe

Before we dive into the particular aspects of the perturbations themselves we need to properly define the *background*, that is the idealized description of the homogeneous universe

on large scales. The general relativistic description of the background begins by writing an ansatz for a Lorentzian metric for space-time in some coordinate system. Equation (2.1) gives the line element of the so called Friedman-Lemaître-Robertson-Walker metric, that is without losing generality, the metric that describes in general relativity a spherically symmetric and translational invariant space-time—following the recipe of the cosmological principle.

$$ds^2 = a^2(\tau)(-c^2 d\tau^2 + \gamma_{ij} dx^i dx^j). \quad (2.1)$$

We use the Einstein summation convention where repeated indexes denote summation and latin indexes i, j take the values 1, 2, 3; greek indexes take the values 0, 1, 2, 3. We use c to denote the speed of light in the vacuum. Here τ (also denoted x^0 in covariant expressions) is the coordinate that represents time and it is conventionally called *conformal time*. While (x^1, x^2, x^3) are the spatial coordinates which we haven't specified because our formulation does not require it, in fact they can be either cartesian coordinates or spherical, and the spatial metric γ_{ij} (which is only dependent on the spatial coordinates x^i) could represent a flat or a curved space, but it must be translational and rotational invariant. The variable $a(\tau)$, called *scale factor*, is a function of τ alone.

This line element corresponds to class of coordinate systems in which time and space are separated—there are no mixed terms $d\tau dx^i$ in the metric—which is convenient for simplicity and it allows to introduce the concept of *comoving coordinates* (x^1, x^2, x^3) and *comoving observers* (those observers at rest with respect to these coordinates). In the FLRW universe, space-time can be understood as a time-ordered sequence of three-dimensional spatial slices. Comoving observers worldlines help physically define the meaning of those space slices, because all comoving observers see the same universe that evolves in time. They can synchronize their clocks at a certain stage of the universe, for example when the temperature T of the universe had a certain value T_0 . Then space slices can be time ordered by the comoving observers clocks and identified as the hypersurfaces of constant time (Wald 1984).

Comoving observers move in the FLRW space-time with constant spatial coordinates, that means their 4-velocity is of the form $u^\mu = (\frac{d\tau}{dt_o}, 0, 0, 0)$, where t_o is the proper time of the observer. Using eq. (2.1) we can deduce that for comoving observers there is a direct relation between τ and their clock t_o :

$$dt_o = \frac{1}{c} \sqrt{-ds^2} = a d\tau, \quad (2.2)$$

their 4-velocity take the form

$$u^\mu = (a^{-1}, 0, 0, 0). \quad (2.3)$$

Using the expression of the metric (2.1) we can compute the affine connection of g with components:

$$\begin{aligned} \Gamma^0_{00} &= \mathcal{H} \\ \Gamma^i_{00} &= 0 \\ \Gamma^0_{0i} &= \Gamma^0_{i0} = 0 \\ \Gamma^i_{0j} &= \Gamma^i_{j0} = \delta^i_j \mathcal{H} \\ \Gamma^0_{ij} &= \mathcal{H} \frac{\gamma^{ij}}{c^2} \\ \Gamma^k_{ij} &= \Gamma^{(3)k}_{ij} \end{aligned} \quad (2.4)$$

where $\Gamma^{(3)}$ is the affine connection of the 3-dimensional metric γ

$$\Gamma^{(3)k}_{ij} = \frac{\gamma^{kl}}{2}(-\gamma_{ij,l} + \gamma_{il,j} + \gamma_{jl,i}), \quad (2.5)$$

γ^{ij} denotes the inverse of γ_{ij} and $\mathcal{H} = \frac{\dot{a}}{a}$ is the *conformal Hubble parameter*.

Using (2.4) we can verify that the comoving observers are freefalling, i.e. their 4-velocity satisfy the geodesic equation:

$$u^\mu{}_{;\nu} u^\nu = 0. \quad (2.6)$$

We use the notation convention in which commas represent partial derivatives, e.g. $T_{,\mu} \equiv \frac{\partial T}{\partial x^\mu}$, semicolons represent covariant derivatives, e.g. $V^\mu{}_{;\nu} \equiv V^\mu{}_{,\nu} + \Gamma^\mu{}_{\sigma\nu} V^\sigma$ and a vertical bar represent covariant derivatives in the spatial hypersurface, e.g. $v^i{}_{|j} \equiv v^i{}_{,j} + \Gamma^{(3)i}{}_{kj} v^k$.

Light propagation

Light propagation of individual photons in general relativity can be described their *wave-vector* \mathbf{k} , that we define as:

$$\mathbf{k} \equiv d\alpha, \quad (2.7)$$

which is a 1-form that encodes the wave propagation in space-time. Here α is the angular phase function of the wave. The individual components of \mathbf{k} depend on the choice of the coordinates x^μ :

$$k_\mu = \frac{\partial \alpha}{\partial x^\mu}. \quad (2.8)$$

One can imagine the 2D hypersurfaces of constant α as the wave front, for example in the case of an isotropic source in flat space those surface would be concentric spheres. The 4-momentum carried by photons \mathbf{P} is proportional to \mathbf{k}

$$P_\mu = \hbar k_\mu. \quad (2.9)$$

The 4-velocity is proportional to the momentum, hence we must have

$$k^\mu = \frac{dx^\mu}{d\lambda}, \quad (2.10)$$

for some affine parameter λ . Being massless particles, photons' wave vector must satisfy the null measure condition

$$k^\mu k_\mu = 0. \quad (2.11)$$

The angular frequency of a signal is an observable. An observer moving in spacetime with a clock t_o will measure it as:

$$\omega_o = \frac{d\alpha}{dt_o} = \frac{\partial \alpha}{\partial x^\mu} \frac{dx^\mu}{dt_o} = k_\mu u^\mu, \quad (2.12)$$

where $u^\mu = \frac{dx^\mu}{dt_o}$ is the 4-velocity of the observer, not necessarily a comoving observer.

Since k^μ plays the role of light's 4-velocity it follows that k_μ is parallelly transported along the photon's null geodesic:

$$k_{\mu;\nu} k^\nu = 0. \quad (2.13)$$

We can write an ansatz for the components of k_μ :

$$\begin{aligned} k_0 &= \nu \\ k_i &= \frac{\nu}{c} e_i \end{aligned} \quad (2.14)$$

The null condition (2.11) is satisfied as long as $\gamma_{ij}e^ie^j = 1$, i.e. the vector $e^i \equiv \gamma^{ij}e_j$ is unitary in the 3-dimensional space hypersurface metric. Furthermore, the geodesic equation (2.13), using the affine connection of the background (2.4), implies that ν is a constant of motion:

$$\frac{d}{d\lambda}\nu = k_{0,\mu}k^\mu = \Gamma^\nu_{0\mu}k_\nu k^\mu = 0. \quad (2.15)$$

Using the fact that $k_0 = \nu$ remains constant along the photon's geodesic, we find a relation for the redshift of a signal emitted (s for source) and received (o for observation) by comoving observers:

$$1 + z = \frac{\omega_s}{\omega_o} = \frac{(k_\mu u^\mu)|_s}{(k_\nu u^\nu)|_o} = \frac{a(\tau_o)}{a(\tau_s)}. \quad (2.16)$$

which is a noteworthy feature of the FLRW space-time which does not depend on the choice of coordinates but on its topological structure.

Distance

At any time τ , one can measure the length of paths and distances in space hypersurface as $a(\tau)\Delta\chi$ where χ is a variable that denotes *comoving distance*, i.e. $d\chi^2 = \gamma_{ij}dx^i dx^j$. Photons can again be used to measure distances, because the null condition on their 4-velocity (the constancy of the speed of light) implies that $d\tau = d\chi$. Therefore we can obtain a relation between the distance to the source of a signal and its redshift:

$$d|_s^o = a(\tau_o) \int_s^o d\chi = a(\tau_o) \int_0^z \frac{dz'}{\mathcal{H}(z')(1+z')}. \quad (2.17)$$

Since the absolute value of scale factor itself is not measurable, but its relative value, it is customary to set $a(\tau_o) = 1$ at the present time. Hubble's law follows from (2.17) when $z \ll 1$.

$$d \approx \frac{z}{\mathcal{H}_0}. \quad (2.18)$$

The age of the source can be computed in a similar fashion using the comoving observer proper time $dt = a d\tau$:

$$t = \int dt = \int_0^z \frac{dz'}{\mathcal{H}(z')(1+z')^2}. \quad (2.19)$$

Friedmann equations

The set of Einstein field equations can be written down for FLRW metric, and they provide constraints on the time evolution of the scale factor and its relation with the matter contents of the universe.

We would first need to compute the expressions for the Riemann and Ricci tensors computed from the affine connection (2.4). We use the following convention for the Riemann tensor:

$$R^\mu_{\alpha\nu\beta} = \Gamma^\mu_{\alpha\beta,\nu} - \Gamma^\mu_{\alpha\nu,\beta} + \Gamma^\mu_{\lambda\nu} \Gamma^\lambda_{\alpha\beta} - \Gamma^\mu_{\lambda\beta} \Gamma^\lambda_{\alpha\nu}, \quad (2.20)$$

and then the Ricci tensor is computed as:

$$R_{\alpha\beta} = R^\nu_{\alpha\nu\beta} = \Gamma^\nu_{\alpha\beta,\nu} - \Gamma^\nu_{\alpha\nu,\beta} + \Gamma^\nu_{\lambda\nu} \Gamma^\lambda_{\alpha\beta} - \Gamma^\nu_{\lambda\beta} \Gamma^\lambda_{\alpha\nu}. \quad (2.21)$$

and it takes the following form after substitution of the connection components (2.4):

$$\begin{aligned} R_{00} &= -3\mathcal{H}_{,0} \\ R_{0i} &= R_{i0} = 0 \\ R_{ij} &= R^{(3)}_{ij} + \frac{\gamma_{ij}}{c^2} (\mathcal{H}_{,0} + 2\mathcal{H}^2) , \end{aligned} \quad (2.22)$$

where $R^{(3)}_{ij}$ corresponds to the Ricci tensor of the spatial hypersurface connection:

$$R^{(3)}_{ij} = \Gamma^{(3)l}_{ij,l} - \Gamma^{(3)l}_{il,j} + \Gamma^{(3)l}_{kl} \Gamma^{(3)k}_{ij} - \Gamma^{(3)l}_{kj} \Gamma^{(3)k}_{il} . \quad (2.23)$$

After raising the first index we get the following expression for the Ricci tensor components:

$$\begin{aligned} R^0_0 &= \frac{3}{a^2 c^2} \mathcal{H}_{,0} \\ R^0_i &= R^i_0 = 0 \\ R^i_j &= \frac{R^{(3)i}_j}{a^2} + \frac{\delta^i_j}{a^2 c^2} (\mathcal{H}_{,0} + 2\mathcal{H}^2) . \end{aligned} \quad (2.24)$$

The Ricci scalar is then computed as:

$$R = R^\nu_\nu = \frac{R^{(3)}}{a^2} + \frac{6}{a^2 c^2} (\mathcal{H}_{,0} + \mathcal{H}^2) . \quad (2.25)$$

Next, in order to write Einstein's equations we need to propose a functional expression for the energy momentum tensor in the coordinate system we have chosen. Considering an isotropic and translational invariant universe we can write the energy momentum tensor of a perfect fluid as:

$$\begin{aligned} T^0_0 &= \rho c^2 \\ T^i_0 &= T^0_i = 0 \\ T^i_j &= -p \delta^i_j \end{aligned} \quad (2.26)$$

where ρ is the mass density and p the pressure, and they only depend on time.

By substituting the expression we have found for the Ricci and Energy-Momentum tensors eqs. (2.24) and (2.26) into Einstein's equations:

$$R^\mu_\nu - \frac{R}{2} g^\mu_\nu + \Lambda g^\mu_\nu = -\frac{8\pi G}{c^4} T^\mu_\nu , \quad (2.27)$$

we obtain as a first result that:

$$R^{(3)i}_j = \frac{K}{3} \delta^i_j , \quad (2.28)$$

where $K = R^{(3)}$ is a constant and by definition it represents the curvature of space. Also from the (0,0) component of (2.27) we obtain the so called *first Friedmann equation*:

$$\frac{1}{a^2} \mathcal{H}^2 + \frac{K c^2}{6a^2} = \frac{\Lambda c^2}{3} + \frac{8\pi G}{3} \rho , \quad (2.29)$$

while for the (i, j) components we find:

$$-\frac{1}{a^2} \mathcal{H}^2 - \frac{2}{a^2} \mathcal{H}_{,0} - \frac{K c^2}{6a^2} + \Lambda c^2 - \frac{8\pi G}{c^2} p = 0 .$$

Combining with (2.29) and (2.1) we obtain the *second Friedmann equation*:

$$\frac{1}{a^2} \mathcal{H}_{,0} - \frac{\Lambda c^2}{3} + 4\pi G(\rho/3 + p/c^2) = 0 . \quad (2.30)$$

2.2 Perturbation theory in differential geometry

Having presented the main characteristics of the background for the ideal FLRW universe, we can present very briefly the *passive* approach to relativistic perturbation theory. The goal of this mathematical formulation is to be able to solve Einstein equations to obtain the metric \mathbf{g} of space-time in the presence of an inhomogeneous universe.

Our presentation of the perturbation theory starts with the assumption that in some coordinate system the components of the metric \mathbf{g} of the space-time manifold M is approximately equal to some fixed function $\bar{g}_{\mu\nu}$ in \mathbb{R}^4 ; this function is called the *background metric*. We will then try to deduce a set of equations to estimate the real metric components $g_{\mu\nu}$ in a Taylor expansion of the the differences $\Delta g_{\mu\nu} = g_{\mu\nu} - \bar{g}_{\mu\nu}$. As a general convention, the bar on top of geometric symbols will denote background quantities, the Δ in front will represent the perturbation and δ will be reserved for the infinitesimal variations under coordinate transformations.

In this context, it is important to remark that the symbol $\bar{g}_{\mu\nu}$ representing the components of the background metric is not a geometric object (Mukhanov, Feldman, and Brandenberger 1992), meaning it doesn't change under coordinate transformations. For that reason it might happen that the physical metric equals the background on some coordinate system and by changing coordinates their difference becomes non-zero. The perturbation $\Delta g_{\mu\nu}$ is then a function of the coordinate system and the position.

Example 1. Let $\bar{g}_{\mu\nu} = \text{diag}(-1, 1, 1, 1)$ and $g_{\mu\nu} = \text{diag}(-1, 1, 1, 1)$ in the coordinates $x = (t, x, y, z)$. By changing coordinates to $x' = (t, r, \theta, z)$ such that

$$\begin{aligned} x &= r \cos \theta; \\ y &= r \sin \theta; \end{aligned}$$

we obtain $g_{\mu\nu} = \text{diag}(-1, 1, r^2, 1)$, which differs from $\bar{g}_{\mu\nu}$, even though the metric \mathbf{g} itself didn't change. In the first case the perturbation of the metric is zero and in the second case it is not:

$$\begin{aligned} \Delta g(p, x) &= \text{diag}(0, 0, 0, 0), \\ \Delta g(p, x') &= \text{diag}(0, 0, r^2 - 1, 0), \end{aligned}$$

here $p \in M$ represents an event (a point in space-time) while x and x' denote two choices for coordinate systems.

In order to understand the differential structure of the perturbations we must investigate the effects of infinitesimal changes of coordinates on those perturbations. To this aim consider a vector field ξ , that induces a change of coordinates from chart¹ (U, x) to (U', x') via the diffeomorphism $\phi_t: M \rightarrow M$ where

$$\frac{d}{dt} f(\phi_t(p)) = \xi_p(f); \tag{2.31}$$

for any $p \in M$ and $f: M \rightarrow \mathbb{R}$ smooth function. $\phi_t(p)$, for a fixed p , is a curve with parameter p and tangent vector ξ . On the other hand, ϕ_t is a one-parameter family of

¹A chart (U, x) is rigorous way of defining coordinates on the manifold, $U \subset M$ is some open set on the manifold M and $x: M \rightarrow \mathbb{R}^4$ is a map from M into real numbers.

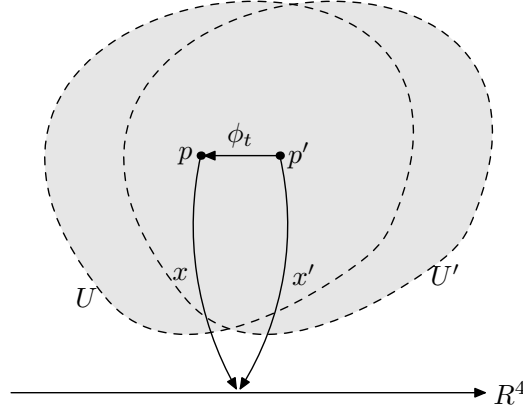


Figure 2.1: Graphical representation of the induced chart (U', x') from the one-parameter diffeomorphism ϕ_t and the original chart (U, x) .

diffeomorphisms. Starting from a chart (U, x) it is natural to define another chart (U', x') such that

$$\begin{aligned} U' &= \text{preimg}_{\phi_t} U, \\ x' &= x \circ \phi_t. \end{aligned} \quad (2.32)$$

Figure 2.1 shows a graphical representation of the change of coordinates induced by ϕ_t . Notice that if $p = \phi_t(p')$ it follows $x'(p') = x(\phi_t(p')) = x(p)$. According to (2.31) we have an infinitesimal law of coordinate transformation:

$$\frac{d}{dt} x^\mu(\phi_t(p)) = \xi_p(x^\mu) = \xi_p^\mu, \quad (2.33)$$

thus

$$x^\mu(\phi_t(p)) - x^\mu(p) \approx t\xi_p^\mu, \quad (2.34)$$

and it follows that

$$x'^\mu(p) \approx x^\mu(p) + t\xi_p^\mu. \quad (2.35)$$

The infinitesimal variation of the perturbation of tensors is proportional to the Lie derivative with respect to ξ of the same quantity in the background (Weinberg 2008). Without giving a general proof, we will show how this is true for some types of fields.

Scalar perturbations

Let us recall the definition of the Lie derivative² for a tensor T with respect to the vector field ξ :

$$(\mathcal{L}_\xi T)_p = \lim_{t \rightarrow 0} \frac{1}{t} (T - \phi_{t*} T)_{\phi_t(p)}. \quad (2.36)$$

Consider the scalar field $Q: M \rightarrow \mathbb{R}$ and a background $\bar{Q}: \mathbb{R}^4 \rightarrow \mathbb{R}$. The respective perturbations of Q at the position-coordinate pairs (p, x) and (p', x') are respectively:

$$\begin{aligned} \Delta Q(p, x) &= Q(p) - \bar{Q}(x(p)), \\ \Delta Q(p', x') &= Q(p') - \bar{Q}(x'(p')). \end{aligned} \quad (2.37)$$

²The theory of differential geometry can be found in Hawking and Ellis 1973.

By choosing $p = \phi_t(p')$ and $x' = x \circ \phi_t$ we get the variation of the perturbation induced by the vector field ξ :

$$\delta\Delta Q(p, x) = \Delta Q(p', x') - \Delta Q(p, x) = Q(p') - Q(\phi_t(p')), \quad (2.38)$$

which is, according to eq. (2.36) proportional to the Lie derivative of Q with respect to ξ . To show this consider that

$$(\phi_{t*}Q)_{\phi_t(p)} = (Q \circ \phi_t^{-1})_{\phi_t(p)} = Q(p), \quad (2.39)$$

and

$$(\mathcal{L}_\xi Q)_p = \lim_{t \rightarrow 0} \frac{1}{t} (Q(\phi_t(p)) - Q(p)) \approx \frac{1}{t} (Q(\phi_t(p')) - Q(p')). \quad (2.40)$$

Hence from (2.38) we find that

$$\delta\Delta Q(p, x) = Q(p') - Q(\phi_t(p')) \approx -t(\mathcal{L}_\xi Q)_p. \quad (2.41)$$

Notice that the Lie derivative is computed on geometric objects, in this case Q . It would have little sense to talk about the Lie derivative of \bar{Q} . It is interesting to notice that $\delta\Delta Q(p, x)$ is independent of the background and the coordinate system.

One-form perturbations

Similarly, for the perturbations of the 1-form ω given the background $\bar{\omega}$ we have

$$\begin{aligned} \Delta\omega_\mu(p, x) &= \omega_\mu(p) - \bar{\omega}_\mu(x(p)), \\ \Delta\omega_\mu(p', x') &= \omega'_\mu(p') - \bar{\omega}_\mu(x'(p')), \end{aligned} \quad (2.42)$$

and the variation of the perturbation becomes

$$\begin{aligned} \delta\Delta\omega_\mu(p, x) &= \omega'_\mu(p') - \omega_\mu(p) = \omega_{p'} \left(\frac{\partial}{\partial x'^\mu} \right) - \omega_p \left(\frac{\partial}{\partial x^\mu} \right) \\ &= (\omega_{p'} - (\phi_t^*\omega)_{p'}) \left(\frac{\partial}{\partial x'^\mu} \right) \approx -t(\mathcal{L}_\xi\omega)_p \left(\frac{\partial}{\partial x^\mu} \right). \end{aligned} \quad (2.43)$$

To derive (2.43) we have used the fact that³:

$$\frac{\partial}{\partial x^\mu} = \phi_{t*} \frac{\partial}{\partial x'^\mu}. \quad (2.44)$$

Vector perturbations

Similarly, for the perturbations of the vector field V given the background \bar{V} we have:

$$\begin{aligned} \Delta V^\mu(p, x) &= V^\mu(p) - \bar{V}^\mu(x(p)), \\ \Delta V^\mu(p', x') &= V'^\mu(p') - \bar{V}^\mu(x'(p')), \end{aligned} \quad (2.45)$$

and the variation of the perturbation becomes:

$$\begin{aligned} \delta\Delta V^\mu(p, x) &= V'^\mu(p') - V^\mu(p) = V_{p'}(x'^\mu) - V_p(x^\mu) = V_{p'}(x^\mu \circ \phi_t) - V_p(x^\mu) \\ &= (\phi_{t*}V)_p(x^\mu) - V_p(x^\mu) \approx -t(\mathcal{L}_\xi V)_p(x^\mu). \end{aligned} \quad (2.46)$$

³This can be shown by choosing any smooth function f and observing that $\frac{\partial}{\partial x^\mu} f = \partial_\mu(f \circ x^{-1}) = \partial_\mu(f \circ \phi_t \circ x'^{-1}) = \frac{\partial}{\partial x'^\mu} (f \circ \phi_t) = \left(\phi_{t*} \frac{\partial}{\partial x'^\mu} \right) f$.

(0, 2)-tensor perturbations

For the perturbations of the (0, 2)-tensor field T given the background \bar{T} we have:

$$\begin{aligned}\Delta T_{\mu\nu}(p, x) &= T_{\mu\nu}(p) - \bar{T}_{\mu\nu}(x(p)), \\ \Delta T_{\mu\nu}(p', x') &= T'_{\mu\nu}(p') - \bar{T}_{\mu\nu}(x'(p')), \end{aligned} \quad (2.47)$$

and the variation of the perturbation becomes:

$$\begin{aligned}\delta\Delta T_{\mu\nu}(p, x) &= T'_{\mu\nu}(p') - T_{\mu\nu}(p) = T_{p'} \left(\frac{\partial}{\partial x'^{\mu}}, \frac{\partial}{\partial x'^{\nu}} \right) - T_p \left(\frac{\partial}{\partial x^{\mu}}, \frac{\partial}{\partial x^{\nu}} \right) \\ &\approx -t(\mathcal{L}_{\xi}T)_p \left(\frac{\partial}{\partial x^{\mu}}, \frac{\partial}{\partial x^{\nu}} \right). \end{aligned} \quad (2.48)$$

Perturbation of the metric

Metrics are symmetric (0, 2)-tensors, hence equation (2.48) is valid for the perturbations of \mathbf{g} . In particular if we denote $\bar{g}_{\mu\nu}$ the background metric we can write:

$$\begin{aligned}\delta\Delta g_{\mu\nu}(p, x) &= g'_{\mu\nu}(p') - g_{\mu\nu}(p) \approx -t(\mathcal{L}_{\xi}g)_p \left(\frac{\partial}{\partial x^{\mu}}, \frac{\partial}{\partial x^{\nu}} \right) \\ &= -t(\xi^{\sigma} g_{\mu\nu,\sigma} + \xi^{\sigma}{}_{,\mu} g_{\sigma\nu} + \xi^{\sigma}{}_{,\nu} g_{\mu\sigma}) \\ &\approx -t(\xi^{\sigma} \bar{g}_{\mu\nu,\sigma} + \xi^{\sigma}{}_{,\mu} \bar{g}_{\sigma\nu} + \xi^{\sigma}{}_{,\nu} \bar{g}_{\mu\sigma}). \end{aligned} \quad (2.49)$$

The background, not being a geometric object, cannot be Lie-derived and hence there is no such thing as the Lie derivative of the background metric. Equation (2.49) is just a numeric result valid only for the coordinates in which the individual components of \mathbf{g} become very close to $\bar{g}_{\mu\nu}$.

The Lie derivative of the metric can be written in terms of the covariant derivatives of the field ξ as⁴:

$$(\mathcal{L}_{\xi}g)_{\mu\nu} = \xi_{\nu;\mu} + \xi_{\mu;\nu} = \xi^{\sigma}{}_{,\mu} g_{\sigma\nu} + \xi^{\sigma}{}_{,\nu} g_{\mu\sigma} + \xi^{\alpha} g_{\mu\nu,\alpha}, \quad (2.50)$$

and the variation of the perturbation (2.49) then becomes:

$$\delta\Delta g_{\mu\nu}(p, x) \approx -t(\xi_{\mu;\nu} + \xi_{\nu;\mu}). \quad (2.51)$$

In our notation an index after a comma represents a partial derivative while an index after a semi-colon represent a covariant derivative.

We can use the affine connection components of the background (2.4) to compute the variation of the metric perturbations (2.51):

$$\begin{aligned}-\frac{\delta\Delta g_{00}}{a^2 t} &= -2c^2(\xi^0{}_{,0} + \mathcal{H}\xi^0) \\ -\frac{\delta\Delta g_{0i}}{a^2 t} &= -c^2\xi^0{}_{,i} + \xi_{i,0} \\ -\frac{\delta\Delta g_{ij}}{a^2 t} &= 2\mathcal{H}\xi^0 \gamma_{ij} + \xi_{i|j} + \xi_{j|i}, \end{aligned} \quad (2.52)$$

where $\xi_i = \gamma_{ij} \xi^j$ and $\xi_{i|j}$ represents the covariant derivative using the affine connection of the spatial metric γ , i.e. $\xi_{i|j} = \xi_{i,j} - \bar{\Gamma}^{(3)k}{}_{ij} \xi_k$. The three degrees of freedom of the spatial

⁴This is a well know identity which can be found in textbooks like Hawking and Ellis 1973.

field	dof
Ψ	1
Φ	1
A	1
C	1
B_i	2
D_i	2
h_i	2

Table 2.1: Perturbation fields and their number of degrees of freedom.

part of the field ξ carry two components, one divergenless three-field ξ_{\perp} (with $\xi_{\perp}{}^i{}_{|i} = 0$) and the other is the divergence of a scalar ξ_{\parallel} :

$$\xi_i = \xi_{\perp i} + \xi_{\parallel,i}. \quad (2.53)$$

Then the perturbations of the metric can be written as:

$$\begin{aligned} -\frac{\delta\Delta g_{00}}{a^2 t} &= -2c^2(\xi^0{}_{,0} + \mathcal{H}\xi^0) \\ -\frac{\delta\Delta g_{0i}}{a^2 t} &= (-c^2\xi^0 + \xi_{\parallel,0})_{,i} + \xi_{\perp i,0} \\ -\frac{\delta\Delta g_{ij}}{a^2 t} &= 2\mathcal{H}\xi^0\gamma_{ij} + 2\xi_{\parallel||ij} + \xi_{\perp i|j} + \xi_{\perp j|i}. \end{aligned} \quad (2.54)$$

Equations (2.54) suggest that the physical metric projected in the coordinate system (τ, x^1, x^2, x^3) has components such that the line element takes the form:

$$ds^2 = a^2 \left(-c^2 d\tau^2 (1+2\Psi) - 2c d\tau dx^i (B_i + A_{,i}) + dx^i dx^j (\gamma_{ij} (1-2\Phi) + C_{|ij} + D_{i|j} + D_{j|i} + h_{ij}) \right), \quad (2.55)$$

such that

$$\begin{aligned} B^i{}_{|i} &= 0 \\ D^i{}_{|i} &= 0 \\ h^i{}_{,i} &= h^i{}_{j|i} = 0 \\ h_{ij} &= h_{ji}. \end{aligned} \quad (2.56)$$

The fields $\Psi, \Phi, A, C, B_i, D_i, h_{ij}$ don't have to be necessarily small, they just represent the difference between the physical metric and the background, in a few words they are the *perturbations* of the metric. Notice that once we consider the constraints (2.56) we obtain that each one of the perturbations carry a certain amount of degrees of freedom that add up to 10, which is the total functional degrees of freedom of the metric, due to the its symmetric constraint (see table 2.1).

From the line element (2.55) we can write explicitly the perturbations of the metric in terms of those fields:

$$\begin{aligned} \Delta g_{00} &= -2c^2 a^2 \Psi \\ \Delta g_{0i} &= -c a^2 (B_i + A_{,i}) \\ \Delta g_{ij} &= a^2 (-2\Phi \gamma_{ij} + C_{|ij} + D_{i|j} + D_{j|i} + h_{ij}). \end{aligned} \quad (2.57)$$

The variation of the perturbations (2.54) applied to the newly defined fields (2.57), taking into account their constraints, produces the following variations of the perturbations under an infinitesimal coordinate transformation induced by ξ (making the parameter $t = 1$):

$$\begin{aligned}
\delta\Psi &= -\xi^0{}_{,0} - \mathcal{H}\xi^0, \\
\delta B_i &= \frac{\xi_{\perp i,0}}{c}, \\
\delta A &= \frac{\xi_{\parallel,0} - c^2\xi^0}{c}, \\
\delta\Phi &= \mathcal{H}\xi^0, \\
\delta C &= -2\xi_{\parallel}, \\
\delta D_i &= -\xi_{\perp i}, \\
\delta h_{ij} &= 0.
\end{aligned} \tag{2.58}$$

2.3 Gauges

In the context of the perturbations of the metric, choosing a *gauge* means to exploit the 4 degrees of freedom of ξ to establish 4 more constraints of the perturbations fields, usually performed to achieve a simplified version of the equations of motion. Fixing the gauge is equivalent to a choice of a particular coordinate system induced by ξ as it was explained in section 2.2. The set of transformations (2.58) are hence called *gauge transformations*. By cleverly combining the perturbation fields, we are able to find other fields which are invariant under such transformations. For example:

$$\begin{aligned}
\delta\left(B_i + D_{i,0}/c\right) &= 0, \\
\delta\left(\Phi + \frac{1}{2c^2}\mathcal{H}(2cA + C_{,0})\right) &= 0, \\
\delta\left(\Psi - \frac{1}{2c^2 a}(2acA + aC_{,0})\right) &= 0.
\end{aligned} \tag{2.59}$$

The last two fields are called Bardeen potentials (Bardeen 1980; Mukhanov, Feldman, and Brandenberger 1992).

Gauges can be fixed by putting constraints on the perturbation fields. For example, consider the *synchronous gauge*, defined by fixing the conditions:

- $\Psi = 0$, which fixes ξ^0 ,
- $B_i = 0$, which fixes $\xi_{\perp i}$,
- $A = 0$, which fixes ξ_{\parallel} .

In the synchronous gauge the line element simplifies to:

$$ds^2 = a^2\left(-c^2 d\tau^2 + dx^i dx^j (\gamma_{ij}(1 - 2\Phi) + C_{|ij} + D_{i|j} + D_{j|i} + h_{ij})\right). \tag{2.60}$$

Another well known gauge is the *conformal Newtonian* or *conformal Poisson gauge*. Defined by fixing the conditions:

- $C = 0$, which fixes ξ_{\parallel} ,
- $A = 0$, which fixes ξ^0 ,

- $D_i = 0$, which fixes $\xi_{\perp i}$.

In the conformal Poisson gauge the line element becomes:

$$ds^2 = a^2 \left(-c^2 d\tau^2 (1 + 2\Psi) - 2c d\tau dx^i B_i + dx^i dx^j (\gamma_{ij} (1 - 2\Phi) + h_{ij}) \right). \quad (2.61)$$

In this gauge, the fields Ψ and Φ become numerically equal to the Bardeen potentials, that can then be found solving a Poisson equation. Also B_i is numerically equal to the gauge invariant field $B_i + D_{i,0}/c$. From this point on we will be always working in the conformal Poisson gauge.

2.4 Field Equations

The field equations can be found by either writing the Einstein-Hilbert action and then find the variational extremum solution with respect to the field variables—in the case of the perturbations the variational approach would demand the functional derivatives of the lagrangian density with respect to Ψ, Φ, B_i, h_{ij} from eq. (2.61)—or we could write directly Einstein's equations (2.27) and substitute the metric as the background plus the perturbation discarding high-order terms of the perturbations. We will follow the latter approach.

Physical metric

As it was mentioned before, we will be working in the Poisson gauge (2.61) hence the metric components can be written as:

$$\begin{aligned} g_{00} &= -c^2 a^2 (1 + 2\Psi), \\ g_{0i} &= -c a^2 B_i, \\ g_{ij} &= a^2 (\gamma_{ij} (1 - 2\Phi) + h_{ij}). \end{aligned} \quad (2.62)$$

The corresponding perturbations are:

$$\begin{aligned} \Delta g_{00} &= -2c^2 a^2 \Psi, \\ \Delta g_{0i} &= -c a^2 B_i, \\ \Delta g_{ij} &= a^2 (-2\Phi \gamma_{ij} + h_{ij}). \end{aligned} \quad (2.63)$$

It would be cumbersome to compute an exact expression for the inverse of the metric from (2.62). However, we can find g^{-1} up to first order in the perturbations using the fact that:

$$1 = g g^{-1} = (\bar{g} + \Delta g)(\bar{g}^{-1} + \Delta g^{-1}) = 1 + \bar{g} \Delta g^{-1} + \Delta g \bar{g}^{-1} + \mathcal{O}(\epsilon^2). \quad (2.64)$$

Here $\epsilon > 0$ has the same order of magnitude as Ψ, Φ, B_i or h_{ij} . Therefore

$$\Delta g^{-1} = -\bar{g}^{-1} \Delta g \bar{g}^{-1} + \mathcal{O}(\epsilon^2), \quad (2.65)$$

and we can compute each component as:

$$\begin{aligned} \Delta g^{00} &= \frac{2}{c^2 a^2} \Psi + \mathcal{O}(\epsilon^2), \\ \Delta g^{0i} &= -\frac{1}{c a^2} B^i + \mathcal{O}(\epsilon^2), \\ \Delta g^{ij} &= \frac{1}{a^2} (2\Phi \gamma^{ij} - h^{ij}) + \mathcal{O}(\epsilon^2). \end{aligned} \quad (2.66)$$

where $B^i = \gamma^{il} B_l$ and $h^{ij} = \gamma^{il} h_{ls} \gamma^{sj}$.

Next we compute the components of the affine connection of the physical metric. We will do so up to the first order in the perturbation fields plus quadratic terms that contain spatial derivatives of Φ and Ψ . These last terms are considered because they produce in the Einstein equations some quadratic terms of the form $\Phi\Phi_{|ij}$, $\Psi\Psi_{|ij}$, $\Phi\Psi_{|ij}$, $\Psi\Phi_{|ij}$, $\Phi_{|i}\Psi_{|j}$ that according to Adamek et al. 2016, have the same amplitude as the linear terms Ψ , Φ , B_i and h_{ij} . As a general rule spatial derivatives on Ψ and Φ reduce the order of the perturbation by $\epsilon^{1/2}$. Notice that following this rule we could in principle have second order terms in the Einstein equations of the form $B_i\Psi_{|nm}$, $B_i\Phi_{|nm}$, $h_{ij}\Phi_{|nm}$ and $h_{ij}\Psi_{|nm}$ which are not shown in Adamek et al. 2016. Also quadratic terms containing $\chi \equiv \Phi - \Psi$ are neglected in Adamek et al. 2016, This is because, as we will see latter, the field equations for B_i , h_{ij} and χ indicate that they are sourced by perturbations in the energy-momentum tensor which are sums of either linear or quadratic velocity terms v/c of massive particles, and in a matter dominated universe these are usually non-relativistic; this means that v/c terms and consequently the B_i potential, responsible for *frame dragging* or *gravitomagnetic* effects, as well as h_{ij} , the gravitational waves tensor, and χ are much smaller than the scalars Φ and Ψ .

Einstein equations

We proceed to find the equations of motion of the perturbations starting from Einstein's equations. Following to the assumption that the perturbations of the metric are small, we neglect terms with order higher than ϵ in the expansion of the Ricci tensor and scalar. The computations are lengthy and not shown here, some intermediate results are shown in Appendix A.

From Einstein's field equations (2.27) one can subtract the background equations:

$$\bar{R}^\mu{}_\nu - \frac{\bar{R}}{2}g^\mu{}_\nu + \Lambda g^\mu{}_\nu = -\frac{8\pi G}{c^4}\bar{T}^\mu{}_\nu, \quad (2.67)$$

and we obtain the Einstein equations for the perturbations:

$$\Delta R^\mu{}_\nu - \frac{\Delta R}{2}g^\mu{}_\nu = -\frac{8\pi G}{c^4}\Delta T^\mu{}_\nu. \quad (2.68)$$

The time-time component of eq. (2.68) produces:

$$\Phi_{|n}{}^n(1 + 4\Phi) - 3\frac{\mathcal{H}}{c^2}\Phi_{,0} - 3\frac{\mathcal{H}^2}{c^2}\Psi + \frac{3}{2}\Phi_{|n}\Phi_{|n}{}^n + \frac{\Phi}{2}K = \frac{4\pi Ga^2}{c^4}\Delta T^0{}_0. \quad (2.69)$$

If the spatial curvature is set to zero (for a flat universe) and we define $\chi = \Phi - \Psi$ we obtain equation (2.9) shown in GEVOLUTION's paper (Adamek et al. 2016):

$$\Phi_{|n}{}^n(1 + 4\Phi) - 3\frac{\mathcal{H}}{c^2}\Phi_{,0} + 3\frac{\mathcal{H}^2}{c^2}(\chi - \Phi) + \frac{3}{2}\Phi_{|n}\Phi_{|n}{}^n = \frac{4\pi Ga^2}{c^4}\Delta T^0{}_0. \quad (2.70)$$

The time-space component of eq. (2.68) produces:

$$-\frac{B_{|n}{}^n}{4c} - \frac{\Phi_{,i0}}{c^2} - \frac{\mathcal{H}}{c^2}\Psi_{,i} - \frac{B_i}{6c}K = -\frac{4\pi Ga^2}{c^4}\Delta T^0{}_i. \quad (2.71)$$

Again, by setting the spatial curvature to zero we obtain equation (2.11) of Adamek et al. 2016:

$$-\frac{B_{|n}{}^n}{4c} - \frac{\Phi_{,i0}}{c^2} - \frac{\mathcal{H}}{c^2}(\Phi_{,i} - \chi_{,i}) = -\frac{4\pi Ga^2}{c^4}\Delta T^0{}_i, \quad (2.72)$$

that can be simplified to:

$$-\frac{B_{i|n}{}^n}{4c} = -\frac{4\pi Ga^2}{c^4} P_{\perp} \Delta T^0{}_i, \quad (2.73)$$

taking advantage of the gauge condition $B_{n|}{}^n = 0$, where P_{\perp} is a linear operator that selects from a vector field its divergenceless component.

The trace of the spatial section of eq. (2.68) leads to:

$$-\chi_{|n}{}^n + 9\frac{\mathcal{H}}{c^2}\Phi_{,0} + 3\frac{1}{c^2}\Phi_{,00} + 6\frac{\mathcal{H}_{,0}}{c^2}\Phi + 3\frac{\mathcal{H}^2}{c^2}\Phi = -\frac{4\pi Ga^2}{c^4}\Delta T^n{}_n. \quad (2.74)$$

that determines χ . Since the source of χ is the perturbation of the trace of the energy-momentum tensor $\Delta T^n{}_n$, its amplitude in a matter dominated universe is suppressed by a factor $(v/c)^2$. That is equivalent to say: since dark matter is non-relativistic χ must be very small with respect to Φ or even B_i . Hence quadratic terms involving χ are neglected.

On the other hand, the traceless components of eq. (2.68) yields an set of equations for h_{ij} :

$$\frac{h^i{}_{j,00}}{2c^2} - \frac{h^i{}_{j|n}{}^n}{2} + \frac{\mathcal{H}}{c^2}h^i{}_{j,0} = -\frac{8\pi Ga^2}{c^4} P_{\perp} \left(\Delta T^i{}_j - \frac{\delta^i{}_j}{3} \Delta T^n{}_n \right), \quad (2.75)$$

which can be recognized as a wave equation since h_{ij} carries the gravitational waves.

2.5 Geodesics

To study the motion of free-falling particles in the presence of perturbations we can start by writing the physical action associated to their motion, which is proportional to the integral of the particle's proper time along it's trajectory.

$$S = -mc^2 \int dt_o = -mc \int d\tau \sqrt{-g_{\mu\nu} \frac{dx^\mu}{d\tau} \frac{dx^\nu}{d\tau}} \quad (2.76)$$

The Lagrangian of this action can be approximated by expanding it in powers of the perturbations up to the linear order

$$\begin{aligned} \mathcal{L} &= -mc \sqrt{-g_{\mu\nu} \frac{dx^\mu}{d\tau} \frac{dx^\nu}{d\tau}} \\ &= -mac^2 \sqrt{\left(1 - \frac{|v|^2}{c^2}\right) + 2\Psi + 2\Phi \frac{|v|^2}{c^2} + 2\frac{v^i}{c} B_i - \frac{v^i v^j}{c^2} h_{ij}} \\ &= -mac^2 \left(\sqrt{1 - (|v|/c)^2} + \frac{\Psi + \Phi \frac{|v|^2}{c^2} + \frac{v^i}{c} B_i - \frac{v^i v^j}{2c^2} h_{ij}}{\sqrt{1 - (|v|/c)^2}} \right) + \mathcal{O}(\epsilon^2) \end{aligned} \quad (2.77)$$

We have defined $v^i = \frac{dx^i}{d\tau}$, $v_i = \gamma_{ij} v^j$ and $|v|^2 = \gamma_{ij} v^i v^j$.

The canonical momentum is

$$\begin{aligned} p_i &= \frac{\partial \mathcal{L}}{\partial v^i} \\ &= \frac{mav_i}{\sqrt{1 - |v|^2/c^2}} - \frac{mac}{\sqrt{1 - |v|^2/c^2}} B_i + \frac{ma}{\sqrt{1 - |v|^2/c^2}} v^n h_{ni} \\ &\quad - \frac{mav_i}{\sqrt{1 - |v|^2/c^2}} \left(2\Phi + \frac{\Psi + \Phi |v|^2/c^2 + B_n v^n/c - \frac{1}{2} h_{nm} v^n v^m/c^2}{1 - |v|^2/c^2} \right) + \mathcal{O}(\epsilon^2) \end{aligned} \quad (2.78)$$

Similar to v_i , we can define $p^i = \gamma^{ij} p_j$.

The Euler-Lagrange equations of the motion of the particles is a second-order equation in time. It is more convenient to have first-order coupled differential equations. Hence we start building the Hamiltonian of the particle. We first need to write v^i as a function of p_i . To make equations simpler to write we define the dimensionless quantities $u^i = v^i/c$ and $q_i = p_i/(mca)$. The equation of the canonical momentum (2.78) becomes:

$$q_i = \frac{u_i}{\sqrt{1-|u|^2}} - \frac{B_i}{\sqrt{1-|u|^2}} + \frac{u^n h_{ni}}{\sqrt{1-|u|^2}} - \frac{u_i}{\sqrt{1-|u|^2}} \left(2\Phi + \frac{\Psi + \Phi|u|^2 + B_n u^n - \frac{1}{2} h_{nm} u^n u^m}{1-|u|^2} \right) + \mathcal{O}(\epsilon^2). \quad (2.79)$$

Eq. (2.79) can be inverted to obtain an expression for u_i as a function of q_i :

$$u_i = \frac{q_i}{\sqrt{1+|q|^2}} + B_i - \frac{q^j h_{ij}}{\sqrt{1+|q|^2}} + \frac{q_i}{\sqrt{1+|q|^2}} \left(\Phi \frac{2+|q|^2}{1+|q|^2} + \Psi + \frac{h_{jk} q^j q^k}{2(1+|q|^2)} \right) + \mathcal{O}(\epsilon^2), \quad (2.80)$$

where $|q|^2 = \gamma_{ij} q^i q^j$. Then we can use (2.80) and (2.77) to build the Hamiltonian for the motion of the particle:

$$\begin{aligned} \mathfrak{H} &= v^i p_i - \mathcal{L} \\ &= c \left(\sqrt{(mca)^2 + p^2} + \Psi \sqrt{(mca)^2 + p^2} + p_n B^n + \frac{\Phi p^2}{\sqrt{(mca)^2 + p^2}} - \frac{h^{ij} p_i p_j}{2\sqrt{(mca)^2 + p^2}} \right). \end{aligned} \quad (2.81)$$

The Hamilton equations, for the case $\gamma_{ij} = \delta_{ij}$, that is in flat space using cartesian coordinates, result:

$$\begin{aligned} \frac{dx^i}{d\tau} &= \frac{\partial \mathfrak{H}}{\partial p_i} \\ &= \frac{cp^i}{\sqrt{(mca)^2 + p^2}} + cB^i - \frac{ch^{ni} p_n}{\sqrt{(mca)^2 + p^2}} \\ &\quad + \frac{cp^i}{\sqrt{(mca)^2 + p^2}} \left(\Psi + \Phi \frac{2(mca)^2 + p^2}{(mca)^2 + p^2} + \frac{h_{nm} p^n p^m}{2((mca)^2 + p^2)} \right), \end{aligned} \quad (2.82)$$

$$\begin{aligned} \frac{dp_i}{d\tau} &= -\frac{\partial \mathfrak{H}}{\partial x_i} \\ &= -c \left(p^n B_{n,i} + \Psi_{,i} \sqrt{(mca)^2 + p^2} + \frac{p^2 \Phi_{,i}}{\sqrt{(mca)^2 + p^2}} - \frac{p^n p^m h_{nm,i}}{2\sqrt{(mca)^2 + p^2}} \right). \end{aligned} \quad (2.83)$$

The right-hand side in the last equation is the generalized force acting on the particles. The term proportional to $\Psi_{,i}$ becomes the Newtonian force in the limit of small velocities, $p^n B_{n|i}$ represent the corrections due to frame dragging, the third term in parenthesis is a further relativistic correction and the last term represents the interaction of the particle with the gravitational waves.

Newtonian limit

The Newtonian limit is recovered when we consider Fourier modes larger than \mathcal{H}/c and we further neglect B_i and consider $\Phi \ll 1$; then equation (2.70) becomes

$$\frac{c^2}{a^2} \nabla^2 \Phi = 4\pi G \Delta \rho, \quad (2.84)$$

where $\Delta \rho = \Delta T^0_0/c^2$. Here $c^2 \Phi$ plays the role of the Newtonian gravitational potential. Also, if physical coordinates $\vec{r} = a\vec{x}$ are used instead of comoving coordinates, the operator $\frac{1}{a^2} \nabla^2$ needs to be replaced by $\nabla_{\vec{r}}^2$.

Similarly, in the small velocity limit, i.e. $q = p/(mca) \ll 1$, we can neglect second-order terms of any combination of q , Φ , Ψ , B_i and h_{nm} . Then the hamiltonian equations (2.82) and (2.83) become

$$\frac{dx^i}{d\tau} = c \frac{p^i}{mac} + cB^i, \quad (2.85)$$

$$\frac{dp_i}{d\tau} = -c(p^n B_{n,i} + \Psi_{,i} mac), \quad (2.86)$$

and by combining these two equations we obtain

$$\frac{d^2 x^i}{d\tau^2} = -\mathcal{H} \frac{dx^i}{d\tau} + \mathcal{H} c B_i - c^2 \Phi_{,i} + c B_{i,0} + c \left(\frac{dx^n}{d\tau} B_{i,n} - \frac{dx^n}{d\tau} B_{n,i} \right). \quad (2.87)$$

If we differentiate with respect to the physical proper time of the comoving observers instead of the conformal time $dt = a d\tau$ we obtain

$$\frac{d^2 x^i}{dt^2} = -2H \frac{dx^i}{dt} + \frac{c}{a} (H + \partial_t) B_i - \frac{c^2}{a^2} \Phi_{,i} + \frac{c}{a} \left(\frac{dx^n}{dt} B_{i,n} - \frac{dx^n}{dt} B_{n,i} \right), \quad (2.88)$$

where $H = a^{-1} \frac{da}{dt}$. In (2.88) we can interpret the role of $c^2 \Phi/a^2$ as the Newtonian Gravitational potential, while B_i is some kind of vector potential similar to the vector potential of the electrodynamics that generates the magnetic force field. In cartesian vector notation, eq. (2.88) results:

$$\frac{d^2 \vec{x}}{dt^2} = -2H \frac{d\vec{x}}{dt} + \frac{c}{a} (H + \partial_t) \vec{B} - \frac{c^2}{a^2} \vec{\nabla} \Phi - \frac{c}{a} \frac{d\vec{x}}{dt} \times (\vec{\nabla} \times \vec{B}), \quad (2.89)$$

that makes the presence of a term similar to the Lorentz force, due to the gravitomagnetic effect of the B_i potential, more evident. It is also worth noticing that the presence of the Hubble flow term $-2H \frac{d\vec{x}}{dt}$ acts as a viscous force progressively slowing down the relative motion of free falling particles with respect to the comoving coordinates.

2.6 Energy Momentum Tensor

The energy-momentum tensor is an additive tensor, that means that it is computed as the sum of the individual energy-momentum tensors for each particle species, radiation or any other source of energy and momentum.

For an ensemble of massive particles the energy-momentum tensor can be constructed from the knowledge of positions and momenta, but it also depends on the perturbed metric components. The general equation for the energy-momentum tensor is (Weinberg 1972):

$$T^\mu{}_\nu = \frac{c}{\sqrt{-g}} \sum_i v^\mu P_\nu \delta^{(3)}(x - x_i(\tau)), \quad (2.90)$$

where P_μ is the four-momentum of particle i .

The 4-momentum of a particle is computed in our perturbed metric as

$$P_0 = \mathfrak{H} = mc^2 a \sqrt{1+q^2} \left(1 + \Psi + \Phi \frac{q^2}{1+q^2} + B^n \frac{q_n}{\sqrt{1+q^2}} - h_{ij} \frac{q^i q^j}{2(1+q^2)} \right), \quad (2.91)$$

for the time component, while the spatial components are computed as

$$P_i = -p_i = -mcaq_i. \quad (2.92)$$

Taking equations (2.91) and (2.92) into (2.90) and converting the velocity to momentum using (2.80) we obtain the following expressions for the components of the Energy-Momentum Tensor

$$T^0_0 = \sum_k \delta^{(3)}(x - x_k(t)) \frac{mc^2}{a^3 |\gamma|^{1/2}} \sqrt{1+q^2} \left(1 + 3\Phi + \Phi \frac{q^2}{1+q^2} + \frac{q_n B^n}{\sqrt{1+q^2}} - \frac{h_{ij} q^i q^j}{2(1+q^2)} \right), \quad (2.93)$$

$$T^0_i = \sum_k \delta^{(3)}(x - x_k(t)) \frac{-mcq_i}{a^3 |\gamma|^{1/2}} (1 - \Psi + 3\Phi), \quad (2.94)$$

$$T^i_0 = \sum_k \delta^{(3)}(x - x_k(t)) \frac{mc^2}{a^3 |\gamma|^{1/2}} c \left(q^i (1 + \Psi + 5\Phi + \frac{q_n B^n}{\sqrt{1+q^2}}) + \sqrt{1+q^2} B^i - q_n h^{ni} \right), \quad (2.95)$$

$$T^i_j = \sum_k \delta^{(3)}(x - x_k(t)) \frac{-mcq_j}{a^3 |\gamma|^{1/2}} c \left(\frac{q^i}{\sqrt{1+q^2}} \left(1 + 4\Phi + \frac{\Phi}{1+q^2} + \frac{h_{nm} q^n q^m}{2(1+q^2)} \right) + B^i - \frac{q_n h^{ni}}{\sqrt{1+q^2}} \right). \quad (2.96)$$

These particular set of expressions to compute the components of the energy-momentum tensor from the state of an ensemble of particles become useful for the kind of computer simulation codes described in this thesis. On the left hand side of equations (2.93), (2.94), (2.95) and (2.96) we have an average operation acting on the microscopic state of the system represented by discrete particle elements and on the right hand side we have an effective field quantity that represents a macroscopic property of the matter field. This basically constitutes the connection between a Lagrangian description of the matter contents of the universe (this concept will be clarified in the next chapter) and the source terms for an Eulerian representation that will be used to track the GR equations of motion on large scales.

In this chapter we have explored the a perturbative weak field limit approach to general relativity in the context of cosmology. This is the theoretical framework implemented in GEVOLUTION code and used throughout the development of GRGADGET.

Numerical methods in cosmology

Computer simulations of structure formation play an important role in cosmology, because only by direct numerical computation one can produce accurate predictions of the non-linear regime arising with the gravitational collapse of matter at the late times of the evolution of the universe. Analytic methods are restricted to highly symmetric idealized systems or to approximate treatments of inherently non-linear ones.

Through simulations we can establish relationships between the simple high redshift primordial Universe and the complex structures we observe today. They also allow us to study in detail the formation of clusters of galaxies, the interactions between isolated galaxies and the evolution of the intergalactic gas.

3.1 N -body cosmological simulations

Dark matter and baryonic gas can be modeled as self-gravitating fluids. Therefore, in theory, they can be described by the Boltzmann equation in comoving coordinates and with proper pressure and force terms to account for the expanding universe and the gravitational interaction. In practice, current cosmological codes employ two different hydrodynamical method categories to solve the state evolution of the matter fluid:

- *Eulerian*, which are methods based on the discretization of space as a mesh and representing macroscopic effective fluid properties on the mesh;
- and *Lagrangian* methods, in which the fluid is discretized as a set of mass elements (particles), so that macroscopic effective properties are obtained by averaging over these particles.

Lagrangian methods are intrinsically adaptive, because, by definition, high density regions are those in which there is also a high concentration of tracer particles, hence there is an automatic increase in the resolution and allocation of computational resources towards those regions. Coincidentally, these are of primary interest to cosmology, because it is there where, depending on the scale, the formation of large scale structures or galaxies or star formation happens.

This thesis will concentrate uniquely on Lagrangian methods.

In computational cosmology, N -body simulations refer to a class of computer modeling of an idealized dynamical system composed of many point-like particles. These can be used to study the complex motion of a few gravitationally interacting objects, like the

planets and the Sun in the Solar System or galaxies within a cluster. For example Peebles 1970 performed an N -body simulation of the Coma Cluster, in which simulation particles are used to represent galaxies or proto-galaxies. But N -body simulations are also used to study the time evolution of fluids, in particular the hydrodynamical representation of the Universe, using the Lagrangian approximation. In this case the results obtained from the simulations should not depend on the number N_p of fluid elements (or particles) as they grow very large, i.e. the observables must converge in the limit of $N_p \rightarrow \infty$.

Time integration

In a cosmological N -body simulation, the particle system evolves according to the mechanical and thermodynamical laws that we attribute to the physical model we want to represent. In a dark matter only representation of the universe, for example, particles would interact among each other according to the gravitational interaction and their motion will be affected by the expansion of the universe. A more realistic representation of the universe, needed to explain the formation of galaxies, would require to take into account the presence of baryons and radiation, and in such simulation one could have different classes of particles to represent those entities and their interactions will be gravity plus effective forces that take into account their thermodynamical state (Springel 2005). However, this thesis focus on the mechanical state and evolution of particles in N -body systems, therefore from our stand point all particles are effectively non-collisional dark matter particles.

The most common time evolution approach in numerical N -body consists in integrating the Hamiltonian equations of motion of the mechanical state of the system, which in this case is defined by the $6N_p$ -tuple $\mathbf{z} = (\mathbf{x}, \mathbf{p})$ of coordinates and momenta of N_p particles. *Symplectic integrators* are specific to these systems. At each time step they are constructed as a combination of canonical transformations, hence preserving the symplectic two-form of the system (Saha and Tremaine 1992).

Symplectic integrators assume that the Hamiltonian \mathfrak{H} of the system is of the form

$$\mathfrak{H} = \mathfrak{H}_A + \mathfrak{H}_B, \quad (3.1)$$

where \mathfrak{H}_A and \mathfrak{H}_B , considered separately, are both integrable exactly. Hamilton's equations of motion can then be written as

$$\frac{d}{d\tau} \mathbf{z} = \{ \mathbf{z}, \mathfrak{H}_A + \mathfrak{H}_B \}, \quad (3.2)$$

where $\{ , \}$ are Poisson brackets. By defining the operators

$$A \equiv \{ , \mathfrak{H}_A \} \quad \text{and} \quad B \equiv \{ , \mathfrak{H}_B \}, \quad (3.3)$$

the formal solution to (3.2) becomes

$$\mathbf{z}(\tau) = \exp(\tau(A + B))\mathbf{z}(0). \quad (3.4)$$

Since we have assumed that \mathfrak{H}_A and \mathfrak{H}_B are integrable, we should have the knowledge of an exact expression for $\exp(\tau A)$ and $\exp(\tau B)$.

Example 2. *Consider the time evolution of a system of dark matter particles evolving in a perturbed FLRW spacetime. In section 2.5 it was found an expression for the Hamiltonian (2.81) of an individual particle that follows a geodesic trajectory in a perturbed FLRW*

universe. In the limit of small velocities and neglecting all perturbations except Φ , equation (2.81) becomes

$$\mathfrak{H}_i = \frac{p_i^2}{2m_i a} + m_i c^2 a \Phi(x_i), \quad (3.5)$$

the index i denotes the i -th particle and the field equation for Φ (2.84) combined with the expression for the $(0,0)$ -component of the energy-momentum tensor (2.93) becomes

$$\nabla^2 \Phi(x) = \frac{4\pi G}{ac^2} \sum_k \delta^{(3)}(x - x_k) m_k, \quad (3.6)$$

which can be solved as

$$\Phi(x) = -\frac{G}{ac^2} \sum_k \frac{m_k}{|x - x_k|}. \quad (3.7)$$

Then the Hamiltonian of the entire system, written as a sum of the Hamiltonian of the individual particles is

$$\mathfrak{H} = \sum_i \frac{p_i^2}{2m_i a} - \frac{1}{2} \sum_{i \neq j} \frac{G m_k m_i}{|x_i - x_k|}, \quad (3.8)$$

which can be split into $\mathfrak{H} = \mathfrak{H}_A + \mathfrak{H}_B$ with

$$\mathfrak{H}_A = \sum_i \frac{p_i^2}{2m_i a}, \quad (3.9)$$

and

$$\mathfrak{H}_B = -\frac{1}{2} \sum_{i \neq j} \frac{G m_k m_i}{|x_i - x_k|}. \quad (3.10)$$

Following from the fact that \mathfrak{H}_A has no dependence on the particles' positions, $\frac{\partial \mathfrak{H}_A}{\partial x_i} = 0$, it follows that

$$\exp(\Delta\tau A)\mathbf{z} = \begin{cases} x_i \mapsto x_i + \frac{p_i}{m_i} \int_{\tau}^{\tau+\Delta\tau} \frac{d\tau'}{a} \\ p_i \mapsto p_i \end{cases} \quad (3.11)$$

This evolution operator is commonly known as drift, because it changes particles' positions while leaving their momenta constant.

Since \mathfrak{H}_B does not depend on the particles' momenta, $\frac{\partial \mathfrak{H}_B}{\partial p_i} = 0$, it follows that

$$\exp(\Delta\tau B)\mathbf{z} = \begin{cases} x_i \mapsto x_i \\ p_i \mapsto p_i - \Delta\tau \sum_j G m_i m_j \frac{(x_i - x_j)}{|x_i - x_j|^3} \end{cases} \quad (3.12)$$

This evolution operator is commonly known as kick, because it changes particles' momenta while leaving their position constant.

A simple symplectic time evolution operator can be built as follows

$$U_1(\tau) \equiv \exp(\tau A) \exp(\tau B), \quad (3.13)$$

which is approximately $\exp(\tau(A+B))$. The approximation error can be assessed using Baker-Campbell-Hausdorff (BCH) formula

$$\exp(\tau A) \exp(\tau B) = \exp\left(\tau(A+B) + \frac{\tau^2}{2}[A, B] + \frac{\tau^3}{12}[A-B, [A, B]] + \dots\right). \quad (3.14)$$

The error comes from the highest order term after the $\tau(A+B)$ operator. In this case the next to leading order is the term $\frac{\tau^2}{2}[A, B]$. Therefore $U_1(\tau)$ is an order $\mathcal{O}(\tau)$ integrator.

Another symplectic time evolution operator, called *leap-frog* is defined as

$$U_2(\tau) \equiv \exp\left(\frac{\tau}{2}A\right) \exp(\tau B) \exp\left(\frac{\tau}{2}A\right), \quad (3.15)$$

The approximation error can be assessed using BCH formula

$$\exp\left(\frac{\tau}{2}A\right) \exp(\tau B) \exp\left(\frac{\tau}{2}A\right) = \exp\left(\tau(A+B) + \frac{\tau^3}{12}[[A, B], \frac{A}{2} + B] \dots\right). \quad (3.16)$$

In this case the next to leading order is the term $\frac{\tau^3}{12}[[A, B], \frac{A}{2} + B]$. Therefore, leap-frog is an order $\mathcal{O}(\tau^2)$ integrator. From the symmetry of expression (3.15) it follows that leap-frog is a time reversible operator.

Depending on whether \mathfrak{H}_A is the kinetic energy term in the Hamiltonian while \mathfrak{H}_B is the potential (or vice versa) the expression (3.15) represents a *Drift-Kick-Drift* (DKD) or a *Kick-Drift-Kick* (KDK) operator.

The mainstream cosmological simulation code GADGET-2 (Springel 2005) has brought leap-frog integrators to the spotlight as a tool for N -body simulations, for their simplicity and accuracy.

Gravity solvers

Gravity is the predominant force driving the motion of ordinary matter and the only interaction in the case of dark matter which constitutes around the 80% of the matter contents of the universe (Planck Collaboration et al. 2020). In a cosmological simulation getting an accurate and efficient implementation of gravity is particularly challenging due to its long-range nature and the high dynamic range of the information produced by the formation of structures. Whenever we want to apply a kick operator to the state of the N -body system one must compute for each particle the force acting on it, which in the case of a Newtonian approximation (like in the Example 2) is expressed as a sum of terms arising from the interaction of the particle with the rest of the system (3.12). One naïve approach would be to perform this summation directly in the computer but that means that for every time step one must perform $\mathcal{O}(N_p^2)$ computational operations, for N_p particles. This algorithm does not scale well with the number of particles; still this *direct summation* or *Particle-Particle* (PP) method, as it is often called, it is used as a benchmark to test other more efficient but complex methods. Also for values of N_p up to 10^5 is still feasible to use PP to resolve particle interactions. For example, in White 1976 $N = 700$ particles were used to simulate the formation of galaxy clusters and with $N_p = 500$ particles White 1978 has performed simulations of galaxy merging, using PP method in both papers.

Yet it desirable to be able scale the number of particles in simulations to increase the mass resolution. This need has driven the research for more computationally efficient methods and as a result we can count today with a class of methods called *Particle-Mesh* (PM) in which the interaction potentials, for example the Newtonian potential Φ in Eq. (3.7) in a gravitationally interacting N -body problem, is computed on discrete points of space and then used to compute the forces acting on each particle. A popular PM method uses the Discrete Fourier representation of the potentials to solve their field equations in Fourier space (Hockney and Eastwood 1988). This reduces the complexity of the Kick to $\mathcal{O}(N_p + N \log N)$, where N is the mesh size, now bounded by the cost of performing Fourier transforms. In the next section we will discuss in details this PM method.

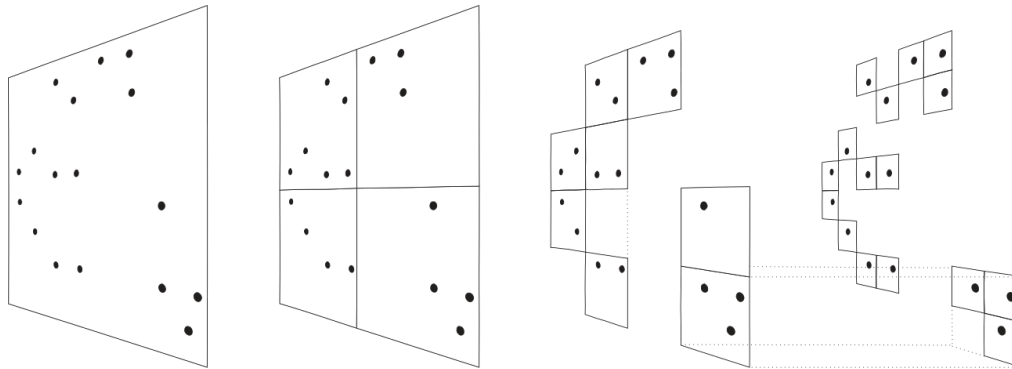


Figure 3.1: Schematic illustration of the BH-TREE in two dimensions. The particles are first enclosed in a square root node. This square is then iteratively subdivided in four squares of half the size, until exactly one particle is left in each final square (leaves of the tree). Image reproduced from Springel, Yoshida, and White 2001.

When we use a PM, we sacrifice a very fine grained spatial resolution N -body for a better time-to-solution, because the spatial resolution of the PM is constrained by the amount of memory required to store the mesh and this limitation is very often evident in the practice. There is a combined PM and PP method to overcome this limitation and try to obtain the best of both worlds. It is called P³M (Hockney and Eastwood 1988) and it supplements the force computation on scales below the mesh size with a direct summation in highly clustered regions.

Tree

Another technique to reduce computational complexity of the force computation is the BH-TREE, described in Barnes and Hut 1986, in which forces are computed with the help of a tree data structure that divides space into cubic cells in a hierarchical fashion. Each node in the BH-TREE is associated to a cubic region of space, and it can have up to 8 children nodes obtained by dividing the current cube into 8 identical non-overlapping smaller cubes of half the side length. The root node encompasses the entire simulation box. Each node stores the information about the particles inside its cube and no nodes are allowed to contain zero particles. Figure 3.1 illustrates the concept of the BH-TREE in two dimensions; each node represents a square region of space that can be split into 4 children squares with half the side length.

To compute forces using the BH-TREE one considers that the multipole expansion of the $1/r$ potential from a compact mass distribution inside a box of size l , centered at \vec{x}_c , computed at a certain location \vec{x} can be approximated as

$$\phi(x) \equiv - \sum_i \frac{m_i}{|\vec{x} - \vec{x}_i|} \approx - \frac{M}{r}, \quad (3.17)$$

where $M = \sum_i m_i$ and $\vec{r} = \vec{x} - \vec{x}_c$, when $l \ll d$. A tolerance to trigger the approximation is fixed with the parameter θ_c called *critical opening angle*, and the approximation (3.17) will be used if and only if

$$\frac{l}{r} < \theta_c. \quad (3.18)$$

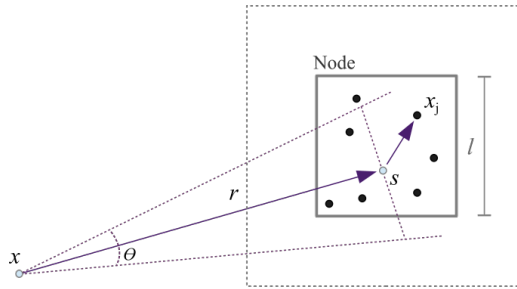


Figure 3.2: The combined gravitational force exerted by the particles contained in a node can be approximated to the monopole as long as the distance of the test particle to the center of the node's box r is small compared to the box size l (Springel et al. 2021).

The force derived from (3.17) is computed as

$$\vec{f}(x) \approx -\frac{M}{r^3} \vec{r}, \quad (3.19)$$

Keeping these aspects in mind, the BH-TREE has to be traversed starting from the root and at every node one checks if the *opening condition* (3.18) is met. If the answer is yes, then one uses the multipole approximation (3.19) and the tree traversal is pruned at that node. Otherwise, the node is *opened*, i.e. the traversal continues recursively on the children of the current node. If the current node has no children then we use direct summation of forces for all the particles contained in that node. Figure 3.2 depicts graphically the quantities involved in the opening condition for nodes.

The smaller the value of the critical opening angle the more accurate are the forces, but also more expensive to compute because the opening condition will be harder to meet and it will lead to less search pruning and more nodes will be visited. Barnes and Hut 1986 estimates that the cost of computing all the forces acting on all the particles is accomplished with the tree in $\mathcal{O}(N_p \log N_p)$ time.

The volume of the nodes is reduced exponentially with the depth of the BH-TREE therefore this data structure naturally cover a high dynamic range of sizes of structures in the simulation. Also in terms of efficiency, BH-TREE adapts to the clustering of particles without significant losses in the runtime.

3.2 Particle-Mesh

The term Particle-Mesh (PM) refers to a class of methods to compute interaction between particles using the potential fields simulated with a discretization of space as intermediary agents. The most simple PM method consist in establishing a cartesian regular mesh that covers the finite simulation box, and there one can use the discrete Fourier transform of the field quantities to solve the partial differential equations that define the value of the interaction potentials. This section is dedicated to the exploration of this type of PM, and throughout the thesis we will not deal with other PM types.

In practice, PM methods perform orders of magnitude faster than the BH-TREE. The tradeoff to take into account is that a PM with a fixed size lacks the ability to zoom-in details in higher dense regions, in exchange for a robust representation of fields in space and faster execution times.

Force solver

The core of a PM code is constituted by the routines that compute the forces acting on the particles once their positions are introduced as input. The *Mesh* is a 3-dimensional grid of length N , that is used to represent the continuous field; that in the case of a Newtonian gravity simulation they will be: the density ρ , potential ϕ and force \vec{F} fields. The steps for computation can be summarized as follows (Hockney and Eastwood 1988):

1. assign the masses to the mesh, i.e. compute ρ from the knowledge of $\{\vec{x}_i\}_{i=1\dots N_p}$ and $\{m_i\}_{i=1\dots N_p}$,
2. solve the Poisson equation to compute ϕ from ρ ,
3. compute the force field \vec{F} from the potential ϕ ,
4. interpolate \vec{F} from the mesh to the particles' positions.

The step in which ϕ is computed through the knowledge of ρ relies on Fourier methods. Let's for simplicity study the method in one dimension. The grid then helps to represent fields in the cubic box $[0, L]$, and we will assume they are two times differentiable and square integrable. Those fields form a Hilbert space called Sobolev, here we will denote that space as $\mathcal{H}^2([0, L])$, with inner product \langle, \rangle such that:

$$\langle A, B \rangle = \frac{1}{L} \int_0^L dx A^\dagger(x) B(x). \quad (3.20)$$

Then according to Fourier's theorem the set $\{u_n\}_{n \in \mathbb{Z}}$, where $u_n(x) = \exp(i\frac{2\pi}{L}xn)$, is an orthonormal basis of $\mathcal{H}^2([0, L])$. That is, for any $f \in \mathcal{H}^2([0, L])$ we can write:

$$f(x) = \sum_{k=-\infty}^{\infty} \langle u_k, f \rangle u_k(x). \quad (3.21)$$

Notice that f is a real-valued function if and only if $\langle u_k, f \rangle^* = \langle u_{-k}, f \rangle$.

Let us emphasize that the numbers $\langle u_k, f \rangle$ are, by definition, computed as integrals in real space

$$\langle u_k, f \rangle \equiv \frac{1}{L} \int_0^L dx u_k^\dagger(x) f(x), \quad (3.22)$$

and we will refer them as the *Fourier coefficients* or the coefficients of the Fourier Transform (FT). Another set of numbers \tilde{f}_k can be defined as

$$\tilde{f}_k \equiv \frac{1}{N} \sum_{n=0}^{N-1} \omega_N^{-kn} f(nL/N), \quad (3.23)$$

where $\omega_N = \exp(i\frac{2\pi}{N})$, and they are called *discrete Fourier coefficients* or the coefficients of the Discrete Fourier Transform (DFT).

In Fourier space the differential equations of the field take a simpler form, because the differential operator are replaced with multiplications of the mode number k . However we need to take into account that our computer simulation does not count with infinite resources to model neither continuous space nor infinite Fourier modes. Our discretized space representation of fields and their relation with the Fourier transform is understood through Nyquist-Shannon theorem¹.

¹In the Appendix B we present a proof of Nyquist-Shannon theorem as presented here (Theorem 1)

Theorem 1 (Nyquist-Shannon). *For any function $f \in \mathcal{H}^2([0, L])$ and for a given N , if there are no modes above $k_N = \lfloor \frac{N-1}{2} \rfloor$, i.e. $\langle u_k, f \rangle = 0$ for every $|k| > k_N$, then the FT coefficients $\langle u_k, f \rangle$ become equal to the DFT coefficients \tilde{f}_k .*

$$\forall |k| \leq k_N: \langle u_k, f \rangle = \tilde{f}_k \quad (3.24)$$

The number k_N is called Nyquist mode.

Nyquist-Shannon theorem assures us that as long as the fields do not have modes above k_N , we can compute Fourier transforms, not using an integral in $[0, L]$ but using the DFT (3.23). An immediate corollary is that we can move from the real space representation $\{f(x_n)\}_{n=0, \dots, N-1}$ to the Fourier space representation $\{\tilde{f}_k\}_{|k| \leq k_N}$ using (3.24) and back to real space using the inverse discrete Fourier transformation

$$f(x_n) = \sum_{|k| \leq k_N} \tilde{f}_k \omega_N^{kn}. \quad (3.25)$$

without losing information. Another corollary is that f is completely determined by the value it takes at the points $\{x_n\}_{n \in [0, N]}$.

If the mass density ρ field had no modes greater than k_N for a fixed grid size N , then we would have an exact representation of ρ once we sample it at the points of the grid $\{x_i\}_{i \in [0, N]}$, according to the Theorem 1. Then the Poisson equation for the Newtonian gravitational potential field

$$\nabla^2 \phi = 4\pi G \rho, \quad (3.26)$$

can be expressed in Fourier space, generalizing to 3-dimensions, as

$$\left(i \frac{2\pi}{L} |\vec{k}|\right)^2 \tilde{\phi}_k = 4\pi G \tilde{\rho}_k. \quad (3.27)$$

Sampling filters

In general, the density field will not satisfy the condition necessary for Theorem 1 to hold, that's because ideal density ρ_o is defined for point particles as

$$\rho_o(\vec{x}) = \sum_i m_i \delta^{(3)}(\vec{x} - \vec{x}_i), \quad (3.28)$$

which does have modes above k_N because of the Dirac delta functions. For that reason we cannot have an exact representation of the ideal density ρ_o on the mesh. We can try instead to approximate ρ_o to some field ρ that doesn't have modes above k_N , so that we can store it in the mesh without losing information and be able to use Theorem 1 to obtain the field's Fourier modes. The passage from ρ_o to ρ is called *sampling*.

One possible solution to this sampling problem would be to apply a low-pass filter to ρ_o thus discarding all modes above k_N :

$$\rho(x) = \int dx' W_N^{(\text{low})}(x - x') \rho_o(x'). \quad (3.29)$$

The low-pass filter can actually be computed analytically and its convolution with Dirac deltas would be again itself. Considering the Fourier series for $W_N^{(\text{low})}$ we get

$$\begin{aligned} \langle u_k, \rho \rangle &= \int dx' \sum_{k'} \langle u_{k'}, W_N^{(\text{low})} \rangle \delta_{kk'} u_{k'}^\dagger(x') \rho_o(x') \\ &= L \langle u_k, W_N^{(\text{low})} \rangle \langle u_k, \rho_o \rangle, \end{aligned} \quad (3.30)$$

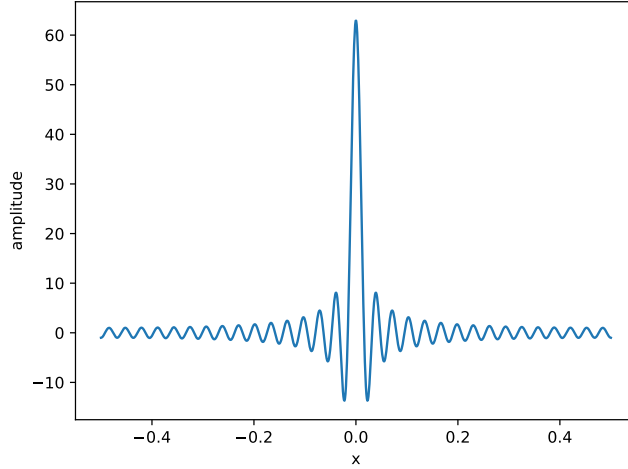


Figure 3.3: The low-pass filter $W_N^{(\text{low})}$ for $N = 64$ and $L = 1$.

therefore

$$\langle u_k, W_N^{(\text{low})} \rangle = \begin{cases} 0, & |k| > k_N \\ 1/L, & |k| \leq k_N \end{cases} \quad (3.31)$$

and $W_N^{(\text{low})}$ can be reconstructed analytically from its Fourier expansion:

$$W_N^{(\text{low})}(x) = \frac{1}{L} \sum_{k=-k_N}^{k_N} u_k(x) = \frac{1}{L} \frac{\sin(\frac{\pi}{L}x(2k_N + 1))}{\sin(\frac{\pi}{L}x)}. \quad (3.32)$$

From equation (3.32) we see that $W_N^{(\text{low})}$ is a distribution which wiggles and decays slowly as we go further from the origin, see figure 3.3, having as consequence that in order to sample the density field ρ_o we would need to add the contribution of every particle at every grid point, leading to an algorithmic complexity, for a 1-dimensional grid, of $\mathcal{O}(N_p \cdot N)$, which is not efficient in practice.

In order to lower the complexity to $\mathcal{O}(N_p)$ one needs to approximate the density field produced by a point particle to a distribution with a compact support such as: nearest-grid-point (NGP), cloud-in-cell (CIC), triangular-shaped-cloud (TSC) or a piecewise-cubic-spline (PCS) (Sefusatti et al. 2016; Hockney and Eastwood 1988).

$$W^{\text{NGP}}(sL/N) = \frac{N}{L} \times \begin{cases} 1, & |s| < 1/2 \\ 0, & \text{otherwise} \end{cases} \quad (3.33)$$

$$W^{\text{CIC}}(sL/N) = \frac{N}{L} \times \begin{cases} 1 - |s|, & |s| < 1 \\ 0, & \text{otherwise} \end{cases} \quad (3.34)$$

$$W^{\text{TSC}}(sL/N) = \frac{N}{L} \times \begin{cases} 3/4 - s^2, & |s| < 1/2 \\ \frac{1}{2}(3/2 - |s|)^2, & 1/2 \leq |s| < 3/2 \\ 0, & \text{otherwise} \end{cases} \quad (3.35)$$

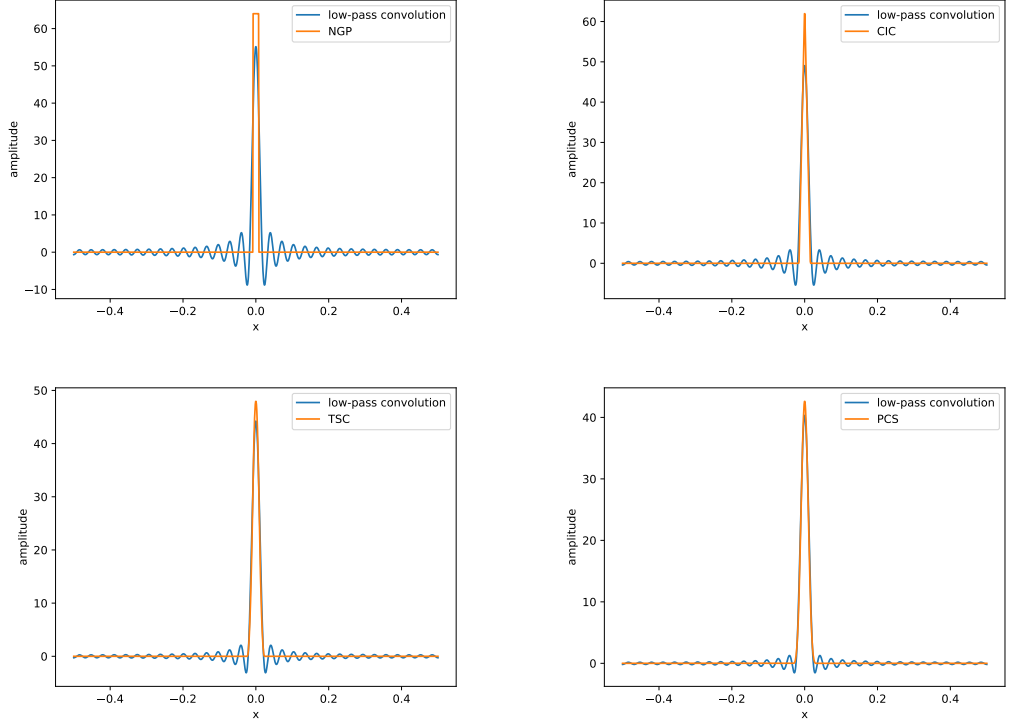


Figure 3.4: Different sampling kernels at $N = 64$, with a boxsize $L = 1$. We plot the shape of the filters and the result of its convolution with $W_N^{(\text{low})}$.

$$W^{\text{PCS}}(sL/N) = \frac{N}{L} \times \begin{cases} \frac{1}{6}(4 - 6s^2 + 3|s|^3), & |s| < 1 \\ \frac{1}{6}(2 - |s|)^3, & 1 \leq |s| < 2 \\ 0, & \text{otherwise} \end{cases} \quad (3.36)$$

The prefactor ensures the normalization of the filters, i.e. $\int_{-L/2}^{L/2} dx W(x) = 1$.

Those filters have, however, non-negligible modes higher than k_N . We can check this either by computing their Fourier coefficients or checking if they remain invariant under convolution with $W_N^{(\text{low})}$. See for instance in the Fig. 3.4 we show the result of applying $W_N^{(\text{low})}$ to NGP, CIC, TSC and PCS filters. The resulting shape is different from the original. Also the power spectrum of all proposed filters in Fig. 3.5 show that even though the amplitude of the Fourier coefficients decreases after k_N , they do not reduce instantaneously to zero. This means that the filters NGP, CIC, TSC and PCS though they might be good to reduce the non-locality of a density function for every particle, they will introduce aliasing effects when used for sampling because they present non-negligible mode amplitudes for $|k| > k_N$,

Interpolation

Fields encoded on the mesh can be evaluated at any point inside the simulation box. This procedure is called *interpolation*, and it is useful in order to recover information from the

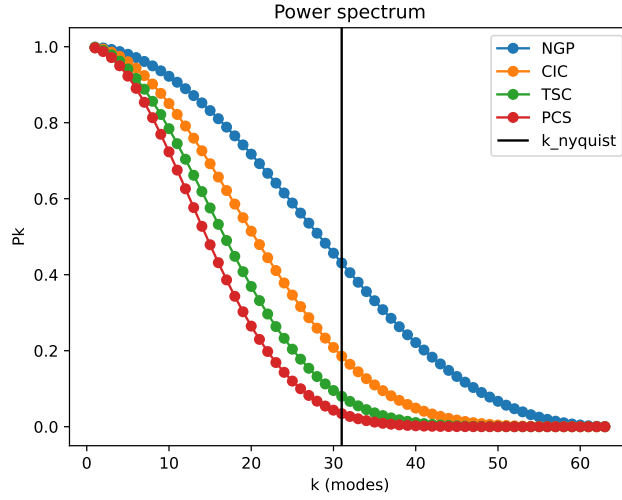


Figure 3.5: Power spectrum of different sampling kernels at $N = 64$.

mesh into the position of the particles; e.g. evaluation of the potentials or density at the position of the particle, or to compute the PM forces acting on a particle.

According to Theorem 1 the knowledge of $\tilde{\phi}$ is sufficient to recover $\phi(x)$ exactly at every point $0 \leq x < L$ using the Fourier expansion:

$$\phi(x) = \sum_{k=-k_N}^{k_N} \tilde{\phi}_k u_k(x). \quad (3.37)$$

However, the direct evaluation of that equation would imply a computational cost of $\mathcal{O}(N)$, which is too expensive for an N -body simulation.

To reduce that computational cost we can alternatively try to construct $\phi(x)$ from the knowledge of the field evaluated at the mesh points around x . That would be effectively a convolution operation with some interpolation filter W . Using Fourier Theorem one obtains that

$$\begin{aligned} (\phi * W)(x) &= \int dx' \phi(x - x') W(x') \\ &= \sum_{k=-\infty}^{\infty} \langle u_k, \phi \rangle u_k(x) L\langle u_k, W \rangle. \end{aligned} \quad (3.38)$$

Since ϕ satisfies the necessary conditions for Nyquist-Shannon Theorem, then we can write $\langle u_k, \phi \rangle$ as a sum over the sampled values at grid points

$$\begin{aligned} (\phi * W)(x) &= \frac{1}{N} \sum_{n=0}^{N-1} \sum_{k=-\infty}^{\infty} \phi_n \omega^{-kn} u_k(x) L\langle u_k, W \rangle \\ &= \frac{L}{N} \sum_{n=0}^{N-1} \phi_n W\left(x - \frac{n}{N}L\right), \end{aligned} \quad (3.39)$$

where $\phi_n \equiv \phi(nL/N)$ for $n = 1, \dots, N-1$ are the real space values of ϕ encoded in the mesh points.

In deducing Eq. (3.39) we have assumed that ϕ satisfies the necessary condition for the Nyquist-Shannon theorem, that is $\phi = \phi * W_N^{(\text{low})}$, therefore one obtains that

$$\phi(x) = \frac{L}{N} \sum_{n=0}^{N-1} \phi_n W_N^{(\text{low})} \left(x - \frac{n}{N}L \right), \quad (3.40)$$

which goes by the name of *Whittaker-Shannon interpolation formula* (WS). This formula provides another exact interpolation algorithm for the PM. But just like Eq. (3.37), it cannot be used in practice for interpolation.

If we use another filter instead of $W_N^{(\text{low})}$ in Eq. (3.39), one with compact support, for example NGP, CIC, TSC and PCS; we will reduce the computational cost of interpolation to $\mathcal{O}(1)$, but we would be interpolating the values of $(\phi * W)(x)$ instead of the actual $\phi(x)$. That is in essence how interpolation is performed in the implementation of a PM and the tradeoffs of accuracy for performance at the core of the method.

Filter corrections

We have discussed the fact that the $W_N^{(\text{low})}$ filter is exact for sampling, but it is computationally expensive to apply. Hence we use other filters with a compact support to reduce the computational complexity of sampling to $\mathcal{O}(1)$. Since these alternative filters W have non uniform modes below k_N their convolution with ρ_o does not preserve $\langle u_k, \rho_o \rangle$. Yet, we can estimate $\langle u_k, \rho_o \rangle$ from $\tilde{\rho}_k$.

Let's start by writing the definition of $\tilde{\rho}_k$ as the DFT of $\{\rho_n\}_{n=1, \dots, N-1}$

$$\tilde{\rho}_k = \frac{1}{N} \sum_{n=0}^{N-1} \omega_N^{-kn} \rho(x_n) = \sum_{z \in \mathbb{Z}} \langle u_{k+zN}, \rho \rangle, \quad (3.41)$$

where we have used the Fourier expansion of $\rho(x_n)$. Eq. (3.41) shows the distinction between DFT and FT coefficients. The presence of modes of ρ above k_N lead to the effect of *aliasing*; i.e. the fact that $\tilde{\rho}_k \neq \langle u_k, \rho \rangle$. Then, by writing ρ as a convolution $W * \rho_o$

$$\tilde{\rho}_k = \sum_{z \in \mathbb{Z}} L \langle u_{k+zN}, W \rangle \langle u_{k+zN}, \rho_o \rangle. \quad (3.42)$$

If $\langle u_k, W \rangle$ is assumed to be leading in order with respect to all the rest of the aliases $k+z, N$, then we can approximate the previous equation and we obtain

$$\langle u_k, \rho_o \rangle \approx \frac{\tilde{\rho}_k}{L \langle u_k, W \rangle}, \quad (3.43)$$

which should be a better estimator for $\langle u_k, \rho_o \rangle$ than $\tilde{\rho}_k$ alone. Equation (3.43) is what we call the *sampling filter correction*. To apply the filter corrections we need to know the FT coefficients of the sampling filter. From Sefusatti et al. 2016 we learn that the four interpolating filters NGP, CIC, TSC and PCS have FT coefficients according to the formula

$$L \langle u_k, W^p \rangle = \left(\frac{\sin(\pi k/N)}{(\pi k/N)} \right)^p, \quad (3.44)$$

where p is either 1, 2,3 or 4 for NGP, CIC, TSC or PCS respectively.

3.3 TreePM

The PM and BH-TREE algorithms can be combined to create a hybrid TREEPM method, first described by Xu 1995. In the TREEPM method the BH-TREE is applied to compute the short-range forces—similar to the role of particle-particle direct summation in P³M codes—while the PM covers the long-range gravitational interaction. The advantage of the combined method over the pure BH-TREE algorithm is that fields quantities are better represented by a PM structure, this is particularly useful when fields not carried by the particles are needed from the simulations (e.g. the gravitational potential to produce weak lensing); in addition, there is a decrease in overall runtime because the PM tends in practice to be faster than the BH-TREE. Moreover, the TREEPM method achieves a high-dynamic range of spatial resolution, allowing us to resolve the matter clustering on small scales even in large simulation volumes—high resolution with a pure PM is achieved by increasing the number of mesh points N per dimension, but the computational and memory cost becomes quickly unfeasible because of its $\mathcal{O}(N^3)$ scaling dependence.

Both forces, PM and BH-TREE, can be coherently combined with the appropriate numerical algorithm. To that end the gravitational potential can be split into the sum of two fields:

$$\phi = \phi^{(L)} + \phi^{(S)}, \quad (3.45)$$

where $\phi^{(L)}$ is the solution to the Poisson equation for long modes and $\phi^{(S)}$ corresponds to the solution for short modes. Written in Fourier space (tilde on top of symbols denote the Fourier transform), the Newtonian gravity Poisson equation reads

$$\tilde{\phi}_k = -\frac{4\pi G}{k^2} \tilde{\rho}_k \quad (3.46)$$

where ρ denotes the mass density. Then we define a quantity r_a called *smoothing scale* and the source term in Eq. (3.46) can be split as the sum of two terms

$$\tilde{\phi}_k = -\frac{4\pi G}{k^2} \tilde{\rho}_k (1 - \exp(-k^2 r_a^2)) - \frac{4\pi G}{k^2} \tilde{\rho}_k \exp(-k^2 r_a^2). \quad (3.47)$$

This means that we can obtain $\phi^{(S)}$ by solving the modified Poisson equation for short modes

$$\tilde{\phi}_k^{(S)} = -\frac{4\pi G}{k^2} \tilde{\rho}_k (1 - \exp(-k^2 r_a^2)), \quad (3.48)$$

and $\phi^{(L)}$ by solving the modified Poisson equation for long modes

$$\tilde{\phi}_k^{(L)} = -\frac{4\pi G}{k^2} \tilde{\rho}_k \exp(-k^2 r_a^2). \quad (3.49)$$

The factor r_a has units of distance and indicates the scale at which we split long and short range modes.

The long-mode Poisson equation (3.49) can be solved by the PM, since at its core, this data structure uses Fourier methods and hence that equation is written as it is in the code as a multiplication in Fourier space. The force tree on the other hand works exclusively in real space hence, equation (3.48) has to be written in real space before one can implement it in the BH-TREE. Fortunately, the Green function to that modified Poisson equation has an analytic form in terms of the error function erfc . So that for a point-like distribution of particles one obtains

$$\phi^{(S)}(\vec{x}) = -G \sum_i \frac{m_i}{|\vec{x} - \vec{r}_i|} \text{erfc} \left(\frac{|\vec{x} - \vec{r}_i|}{2r_a} \right). \quad (3.50)$$

3.4 Gadget-4

GADGET-4 is the latest release of a hydrodynamical cosmological simulation N -body code written in C++ (Springel et al. 2021). The name stands for *Galaxies with Dark Matter and Gas Interact*. The code is optimized to run on massively parallel hardware and it is capable of following a collisionless fluid with the N -body method, and an ideal gas by means of *smoothed particle hydrodynamics* (SPH). From its first version (Springel, Yoshida, and White 2001) to the latest one, the development of GADGET has been hand in hand with the proposal and improvement of the numerical methods for N -body simulations (such like BH-TREE, PM and TREEPM), some of which have been presented by the authors of the code themselves.

Gravity in GADGET is Newtonian added on top of an expanding universe. This approach is consistent with General Relativity in the Poisson Gauge if we remain in the leading order weak field limit by neglecting the metric degrees of freedom B_i , χ and h_{ij} , and when the modes of the gravitational field are larger than \mathcal{H}/c ; i.e. wavelengths much smaller than the Hubble radius. In our notation, this means that geodesic equations (2.82) and (2.83) become

$$\frac{dx^i}{d\tau} = \frac{p^i}{ma}, \quad (3.51)$$

and

$$\frac{dp_i}{d\tau} = -m\phi_{,i}. \quad (3.52)$$

The field equation for the Newtonian potential $\phi = ac^2\Phi$ is

$$\nabla^2\phi(x) = 4\pi G \sum_k \delta^{(3)}(x - x_k)m_k. \quad (3.53)$$

In GADGET, gravitational forces are computed with a TREEPM algorithm, described in section 3.3. Time integration is based on a quasi-symplectic KDK (see section 3.1) scheme where long-range and short-range forces can be integrated with different time-steps. The PM in GADGET takes into account the sampling and interpolation losses of power at short wavelengths, by convolving the gravitational potential with a CIC correction.

Gadget is parallelized with the Message Passing Interface (MPI) paradigm, in which the copy of the same code is executed in multiple processes that could be running distributed in the same or even different physical machines. Each process has ownership of his own memory space, this is why this approach is called *distributed memory parallelism*. This feature allows to scale the memory and computational resources available to the application. The coordination between processes is obtained with a standard programming interface defined in the MPI standard², which allows the code to be compiled for different machine architectures. MPI libraries take care of optimizing the inter-processes communication so that processes running on the same machine will send and receive messages through shared memory, while processes running on different machines will communicate through TCP/IP; the programmer does not need to write any logic to make those optimization decision.

The code parallelization is a technological necessity for cosmological simulations. Like in other branches of computational physics there is an ever increasing demand for realistic realizations that often drives the computational requirements towards memory allocation. For example, in order to simulate the photometric catalog of the Euclid mission reaching

²The latest version of the MPI standard is version 4, see Message Passing Interface Forum 2021.

up to redshift 2 (Amendola et al. 2018), one would need a box of at least³ $L = 3 \text{ Gpc}/h$ and $N_{\text{part}} = 2 \times 10^{11}$ particles to achieve a mass resolution⁴ of $m_1 = 10^{10} M_{\odot}/h$; then if one considers that for each particle at least 48 bytes of memory are needed (8 bytes for each spatial coordinate and velocity component) then at least $N_{\text{part}} \times 48 \text{ bytes} \approx 10 \text{ Tbytes}$ are required for hosting such simulation. Even though this estimation is very conservative, it is clear that cosmological simulations are memory-bound and cannot be run on single machine architectures. Instead the simulation data must be distributed among physical computers by a software layer that is able to decompose it into several parts, called *domains* that are assigned to individual MPI tasks.

To perform the domain decomposition the code uses a Peano-Hilbert space-filling curve⁵ that maps a discrete representation of the simulation volume into a one-dimensional parameter index preserving locality; i.e. two neighboring PH indexes correspond to neighbor regions of 3D space. The range of values of the PH index can be divided into pieces and they will correspond to 3D domains of the simulation box. PH curves at any order n divide the 3D box into 2^{3n} small and identical cubes (see Fig. 3.6). Therefore there is a correspondence between the domains defined by the PH curve and the BH-TREE. That is, every domain can be expressed as the union of non-overlapping tree nodes. This property allows the code to perform a parallel distribution of the BH-TREE. While abstractly the combined memory of the processors contain the entire BH-TREE, individual processes only store in their local memory the top nodes of the tree and full expansion in depth until the leaves of those nodes that lie within the local spatial domain; see Fig. 3.7.

Because of the PH curve properties, the domain regions have small surface to volume ratios which is an optimality condition for running time, because the higher the boundary surface between the domains, the higher the size of the communication packets they need to exchange to synchronize. Processors are very powerful these days, usually the bottleneck of this kind of simulations is the communication instead of actual computation in the CPU, this is the reason why much of the design effort is aimed at reducing the inter-process communication.

Compared to previous version of GADGET, the latest GADGET-4 offers a Fast Multipole Method (FMM) as an alternative to a plain BH-TREE. It also provides an improved domain decomposition logic for better distribution of memory and computation work among processes. Other features of the new code include: the support for built-in merger trees, the construction of lightcone outputs, the identification of halos directly on the lightcone, the measurement of power spectrum during runtime, and the support for HDF5 output.

The PM domain decomposition has also been improved, from the traditional slab decomposition to column based, and the corresponding parallel FFT algorithm has been adapted to this new decomposition. See Fig. 3.8 for a graphical representation of the column-based domain decomposition. This change in the domain decomposition of the PM solves the scalability problem of the parallel PM and FFT. In the section about LATFIELD2 library we will explore in more detail the nature of this scalability problem and how it is solved in LATFIELD2 with a more robust method using a 2-dimensional

³With a simulation box of length $L = 3 \text{ Gpc}/h$ one can extract a deep field lightcone up to $z = 2$ that covers a 3% fraction of the sky.

⁴The total matter mass in the simulation is obtained from the critical density at the present time multiplied by the fraction of matter mass and the box volume $M = \rho_c \Omega_m L^3$. The value $m_1 = 10^{10} M_{\odot}/h$ is a typical choice of mass resolution for cosmological simulations in order to reproduce the formation of the dark matter haloes that host galaxies.

⁵For details about Peano-Hilbert curves are defined and constructed see Haverkort 2011.

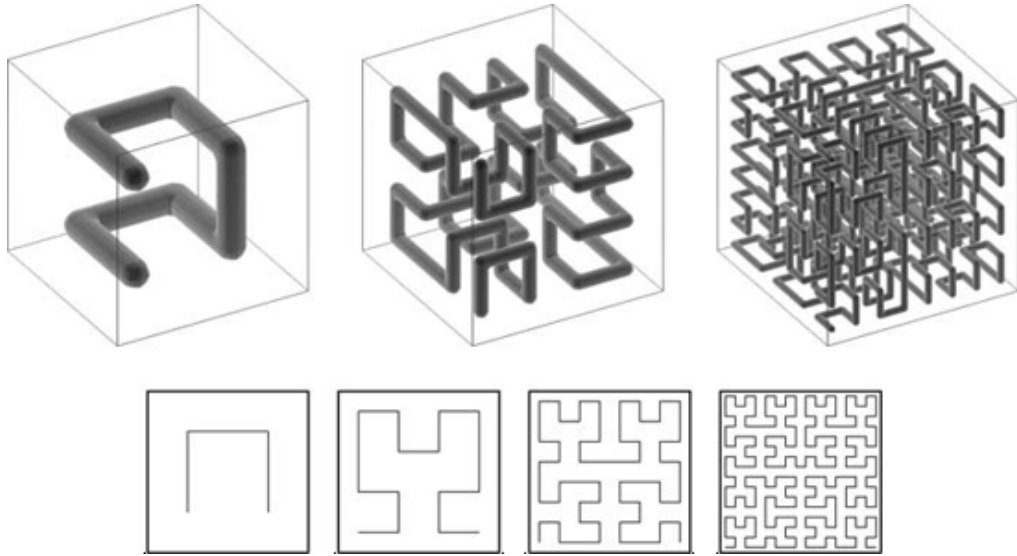


Figure 3.6: Space-filling Peano-Hilbert curve in two (bottom) and three (top) dimensions (Springel 2005).

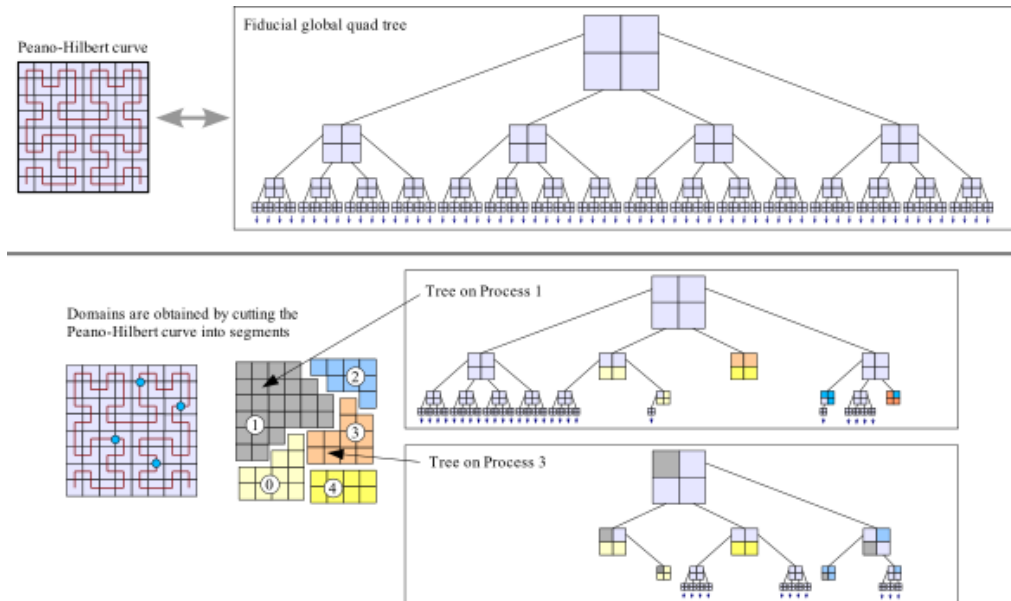


Figure 3.7: Relation between the Barnes-Hut tree and a domain decomposition based on a Peano-Hilbert curve. For clarity, the sketch is drawn in two dimensions (Springel 2005).

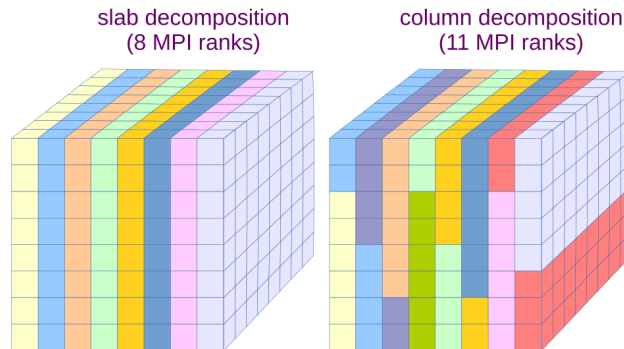


Figure 3.8: Domain decomposition schemes for the PM data. Slab-based to the left and column based to the right (Springel et al. 2021).

domain decomposition of the mesh. At a lower level, GADGET-4 still uses FFTW3 library⁶ to perform 1-dimensional FFTs on the fastest running index of the PM memory array.

The algorithms produced to improve the computational performance and scalability of GADGET have allowed to increase the size of N -body simulations from $N \sim 1000$ using PP up to $N \sim 10^{12}$ (Pakmor et al. 2022) with the latest version GADGET-4 running on a state of the art high performant machine.

3.5 Gevolution

GEVOLUTION is another N -body cosmological simulation code also written in C++ and parallelized according the MPI paradigm (Adamek et al. 2016).

Numerically, this code works as a pure PM to resolve the dynamics of the particles. The mesh in GEVOLUTION, a cartesian grid, has a fixed resolution throughout the simulation. This equal spacing of grid points per dimension is a natural choice for the discretization of the cubic volume that represents the simulation space, that allows the use of Fourier methods to solve field equations. However, it has the drawback that if one wishes to increase the resolution of the simulation, one must do so for the entire volume hence with a cost in memory of $\mathcal{O}(N^3)$, where N is the number of grid points per dimension.

GEVOLUTION, can run in either *Newton* or *General Relativity* modes. When running the General Relativity mode, it solves Einstein equations in the weak field limit, using a combination of the field equations derived in Section 2.4, Eqs. (2.70), (2.73), (2.74), and (2.75) that suits the best for the numerical stability of the code. All the field quantities, metric perturbations and components of the energy-momentum tensor are represented on the mesh. The hamiltonian forces and velocities to which particles are subjected are computed from the gravitational potentials arising from the perturbative treatment of gravity in a FLRW background in the Poisson gauge as described in Chapter 2.

GEVOLUTION code relies on Fourier methods to compute the perturbation potentials from Einstein equations. That numerical approach is facilitated with the use of a C++ library called LATFIELD2, that defines classes and methods to represent classical fields in distributed memory applications. LATFIELD2 exposes a programming interface to perform operations on the fields, either in their real and Fourier space representations. This library implements Fast Fourier Transforms (FFT) of 3-dimensional fields whose memory

⁶<http://fftw.org/>

is distributed among parallel processes following a 2-dimensional uniform decomposition of space, in which each process owns in memory a portion of the grid with a *rod* shape (Daverio, Hindmarsh, and Bevis 2015). In this way LATFIELD2 overcomes the scaling limitations of a more simpler 1-dimensional domain decomposition (or *slabs*) provided by the mainstream FFTW3 Fast Fourier Transform library. We will come back to this argument in more detail in the next chapter.

GEVOLUTION has been used to study the effects of early radiation and massive neutrinos in the clustering at large scales and the halo mass function (Adamek et al. 2017; Adamek, Durrer, and Kunz 2017), investigate the generation of vorticity and its relation with the frame dragging potential (Jelic-Cizmek et al. 2018), to obtain the halo counts and its angular power spectrum with relativistic effects and its cross-correlation with weak gravitational lensing convergence (Lepori et al. 2022), to study k-essence model of clustering dark energy (Hassani et al. 2019) and to quantify the systematic error from mapping Newtonian simulations to light-cone observables (Adamek and Fidler 2019).

GrGadget

Given the fact that General Relativity has been so far proven to be the correct theory to explain gravitational phenomena, we support the idea that cosmological N -body simulations should be based on GR from first principles. The complexity of the implementation of a code like that is clearly greater and more time consuming, but we obtain the following benefits:

- There is a direct interpretation of quantities expressed in computer code from quantities in the theory, e.g. field values, coordinates and momentum of particles, that are gauge dependent, and we can correctly appreciate how their relationship with observables quantities is.
- Simulations produce more realistic outcomes.
- It should be easier to introduce modifications to GR, either by adding extra degrees of freedom and/or changing the field and geodesic equations.

Given the fact that GR is a field theory, a discretization of space is needed to represent the state of the fields. The most viable technique to represent a field theory like GR in a simulation is through a mesh, and since we will pay close attention to phenomena that arise from the non-ideal fluid dynamics of the matter content of the universe, a Lagrangian representation of matter aided by a relativistic PM to solve interactions seems the most reasonable choice. On the other hand we have discussed in Chapter 3 the limitations of PM codes to reproduce high dynamic clustering as it occurs naturally in matter evolution and how TREEPM strategy has been developed to overcome these limitations and improve the parallel scalability of N -body codes. Therefore, we propose to study relativistic effects in N -body simulations with the use of a TREEPM method.

Our approach is such that matter and energy is transported by particles elements and the state of the gravitational field is encoded in a regular grid discretization of space. A PM structure and methods are needed to perform the time evolution of the gravitational fields due to the changing underlying matter distribution through Einstein equations. The PM is also needed to compute the geodesic trajectory of particles from the knowledge of the space-time metric. Short range gravitational interactions between particles are resolved in the Newtonian limit, specially in very dense but small regions of space—the size of the grid resolution and below—and computed with a BH-TREE.

We have used the TREEPM code GADGET-4 and added to it a relativistic PM to substitute the Newtonian PM. Instead of writing this PM from scratch, we have taken

the already existing open source code of `GEVOLUTION` and we have built from it a C++ library named `LIBGEVOLUTION` that exposes its resources and methods to perform general relativistic computations in the mesh. We are exploiting the similarities in both `GEVOLUTION` and `GADGET-4` designs: firstly, both codes are written in C++ so programming interfaces are portable between them; secondly, in both cases the PM is a Fourier uniform mesh; and thirdly, the equations of motion for particles in `GEVOLUTION` converge to the Newtonian equations in the non-relativistic limit and small scales. The code described in this thesis is publicly available at <https://github.com/GrGadget>.

4.1 LATfield2

Before we make an in-depth exploration of the technical details of `LIBGEVOLUTION` it is necessary to dedicate some attention to a library called `LATFIELD2` (Daverio, Hindmarsh, and Bevis 2015), which is a building block of `GEVOLUTION` code. As a matter of fact all the programming technicalities of the PM are provided by `LATFIELD2`, while the implementation of the physics—Einstein field equations, geodesic motion, etc.—are written in the second layer of `GEVOLUTION`. The design choices made by `LATFIELD2` developers have a footprint in the workflow of operations in `GEVOLUTION` and they shape as well the syntax of `LIBGEVOLUTION` to a great extent.

`LATFIELD2` is C++ library with the purpose of providing a programming framework for classical field theories. The library is restricted to applications running on parallel processes using the MPI model. `LATFIELD2` provides to the user 5 different classes: `Parallel`, `Lattice`, `Site`, `Field` and `PlanFFT`. I will briefly explain their purpose and the interrelation between them.

First of all there is `Parallel`, which instantiated only once as a global variable. This class serves two purposes: it represents the global state of the library and it predefines a topology of participating MPI processes into a cartesian two-dimensional array. `Parallel` owns an MPI communicator of all the application processes. The organization of all processes into a two-dimensional grid is performed upon initialization and it is used as a template for the domain decomposition. In `LIBGEVOLUTION` we have modified this behavior, so that `Parallel` can be initialized with a user provided communicator. From a code design philosophy perspective, the new version of `Parallel` is only managing the MPI communicator of the PM as an abstract resource and not grabbing that resource from the user. This allows, for example, `LATFIELD2` to run on a subset of all application processes. This is essential to port `LIBGEVOLUTION` into `GADGET-4`, because the previous version of `LATFIELD2` would run only for an specific number of MPI processes that depended on the number of mesh points per dimension, while `GADGET-4` does not impose any bound in this respect.

The `Lattice` class describes a discretization of three-dimensional space into a regular grid—in the context of this thesis lattice, grid and mesh are synonymous. A `Lattice` can be instantiated by specifying the number of points for each one of the three spatial dimensions, the internal book-keeping routines will take care of defining how the mesh will be divided into domains assigned to the MPI processes according to their location in the process grid within `Parallel`. `Lattice` also contains information about the extension of *ghost cells*—also called *halo* in `LATFIELD2`'s own terminology. In distributed memory parallelism the ghost cells relative to an MPI process are those regions of space right outside the boundary of the domain belonging to that process and their information is cached in memory to prevent wasteful intraprocess communication. In `LIBGEVOLUTION` the extent of the ghost cells is determined by the order of the interpolation and sampling

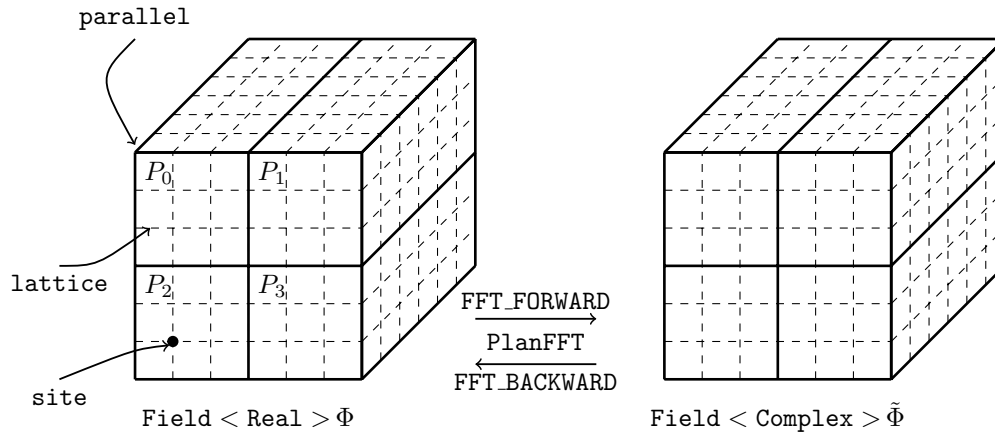


Figure 4.1: Illustration of LATFIELD2 classes to represent classical fields on a discrete mesh, and MPI parallelism. `Parallel` handles the MPI processes coordination and domain decomposition, `Lattice` regards the discretization of space and manages ghost cells along domain boundaries, `Field` allocates the memory for complex or real classical fields embedded in the geometry defined by the `Lattice`, `Site` acts like a pointer to specific locations on the mesh and `PlanFFT` is used to move from the real to the complex Fourier space and viceversa using FFTs.

kernels, as well as the order of the finite differences for derivatives. As we will describe later, LIBGEVOLUTION uses a CIC (see section 3.2) interpolation/sampling kernel which requires only one ghost cell, and finite differences are computed at 4th order and required up to 2 ghost cells.

A `Field` implements the concept of a classical field represented on discrete points of space with the help of a `Lattice`. The field could be a scalar, vector or tensor, and it can take either real or complex values. Each field can be imagined as a three-dimensional matrix which is split along the x and y axis into rod pieces, following the `Parallel` process grid template, so that each MPI process stores in memory only a portion of that matrix. Figure 4.1 illustrates the role of LATFIELD2 classes put together to represent fields in space and across MPI processes.

The class `Site` has close ties with `Lattice` and `Field`, conceptually it constitutes the *iterator* or the indexing of lattice points. It can be used to access the individual elements of the `Field`.

Finally, the class `PlanFFT` links together a real and a complex field and implements the concept of the DFT (Discrete Fourier Transformation) applied to `Field`. Under the hood, this class performs a series of one-dimensional FFT transformations along the z -axis using FFTW3 library¹ (Frigo and Johnson 2005), that combined with matrix transpositions involving all MPI processes results in the three-dimensional DFT of the `Field`.

4.2 Domain decomposition

The spatial resolution of the fields represented in LATFIELD2 depends on how many grid points N there are per space dimension. One can immediately see that the mem-

¹FFTW3 official website is <http://fftw.org>.

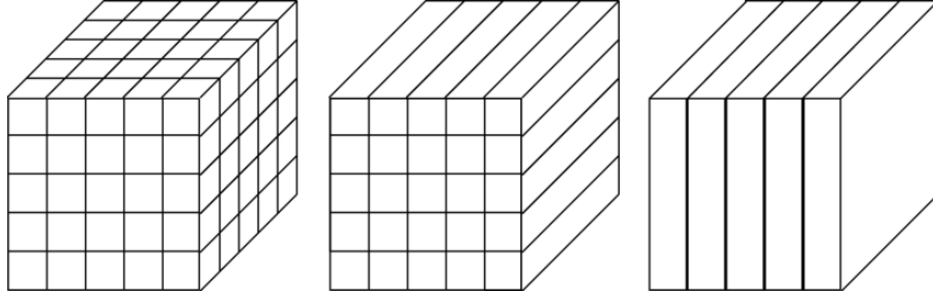


Figure 4.2: Block, pencil, and slab domain decomposition of a cubic region of space (from left to right). FFTW3 natively supports FFT for slab domains in its API (right), while LATFIELD2 is based on pencil domains (center).

ory requirements grow as $\mathcal{O}(N^3)$, which corresponds to the size of the three-dimensional field data matrix, and the computational complexity of performing Fourier transforms is $\mathcal{O}(N^3 \log N)$. When the memory and computation is evenly distributed across P MPI processes, the memory per process becomes $\mathcal{O}(N^3/P)$ and the computational complexity reduced to $\mathcal{O}((N^3/P) \log N)$. If the lattice domain was decomposed along one dimension—this goes by the name of *slab* domain decomposition— P would be limited to have a value at most equal to N , and by increasing P we can get at most that a memory requirement of $\mathcal{O}(N^2)$ and $\mathcal{O}(N^2 \log N)$ execution time; this is quite inefficient and we cannot increase P beyond N to overcome that burden on the computational resources. FFTW3 library follows this one-dimensional domain decomposition scheme. This problem and possible solutions are documented by Pippig 2013.

LATFIELD2 overcomes this scaling limitation by producing a two-dimensional domain decomposition—namely a *pencil* domain decomposition—so that theoretically the number of MPI processes can be increased up to the number of lattice points in the xy plane, which for cubic grid becomes N^2 . See Fig. 4.2 for an illustration of how different mesh domain decompositions are performed. For example LATFIELD2 has been tested with lattices of 4096^3 grid points distributed over 32×10^3 MPI processes (Daverio, Hindmarsh, and Bevis 2015), this amounts to roughly 2×10^6 grid points stored on each process; In this case a slab domain decomposition would be limited to 4096 MPI processes and 16×10^6 grid points per process which carries 8 times higher work and memory loads for the CPUs. If N was multiplied by 2, a pencil domain decomposition could still accommodate the problem in a machine 4 times bigger and the burden per process is at most doubled, while a slab-based code can only increase by 2 the size of the machine and as a result the memory and computation per process will be 4 times higher.

4.3 Concept and design rules

The rationale of LIBGEVOLUTION is that the encapsulation of GEVOLUTION’s resources and methods into C++ classes will result in the following benefits: first, maintenance is eased with the logical modularization of the code; second, we are able to port GEVOLUTION methods to other applications, such as GRGADGET in the present thesis.

Conceptually, LIBGEVOLUTION provides a high level interface that implements a relativistic PM, which is able to perform the following tasks: take as an input the state of all the simulation particles, compute the gravitational fields (perturbations of the metric

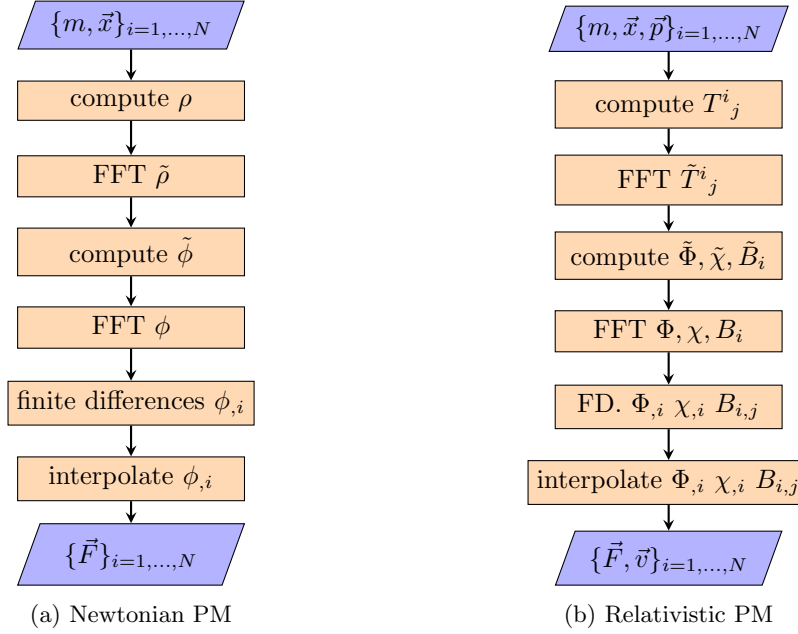


Figure 4.3: General concept of a PM and its workflow for the case of a Newtonian PM (to the left) and a relativistic PM (to the right).

or relativistic potentials) and compute the kick and drift operators for time evolution of the particles' state; see figure 4.3.

A Newtonian PM considers the masses and position of the particles as input to compute the density field ρ by sampling the Dirac delta mass distribution of each particle into the mesh. Then in Fourier space it solves Newtonian gravity Poisson equation to find the Newtonian potential ϕ . Using finite differences and an interpolation kernel, the first partial derivatives of ϕ can be computed at the position of each particle and thus the gravitational force is obtained.

A relativistic PM, on the other hand (see Fig. 4.3), considers an extended state of the particles comprising mass, position and momenta, to compute all the components of the energy-momentum tensor. Then in Fourier space it solves Einstein equations to find the components of the metric Φ , χ and B_i . Then, similarly to the Newtonian case, finite differences and an interpolation kernel are used to compute the derivatives of those potentials at the position of each particle and using equations (2.82) and (2.83) the force and velocity are obtained. It should be noted that in the aforementioned equations we write h_{ij} which we do not treat in GRGADGET at the moment, also they are written in terms of Φ , Ψ instead of $\chi \equiv \Phi - \Psi$ but internally, GRGADGET as well as GEVOLUTION work in terms of the scalars (Φ, χ) and not (Φ, Ψ) .

The design of LIBGEVOLUTION is based on 4 rules or guarantees we want to maintain:

1. Simplicity and backwards compatibility: we hide implementation details from the user, the classes represent concepts (like particles, fields and grids) and their methods are coherent with the concept they represent and do not reflect the underlying algorithms. In the future we might, for example, change the way finite differences are computed or substitute FFT-based solutions to Einstein equations but the user interface should not be affected for backwards compatibility.

2. Generality: use C++ templates as much as possible, make the code portable to other applications. For example, particle data could be specified by the user according to his needs.
3. No resource grabbing: `GEVOLUTION` and `LATFIELD2` make intensive use of standard IO, memory allocation and MPI communicators. Those resources should be handled through the user interface and released when not needed; in this we follow the RAII principle (resource acquisition is initialization) strongly suggested by Stroustrup 2013. For example, messages coming from `LIBGEVOLUTION` should be passed to the user through a logging mechanism instead of printing to standard output/error. The user should have control over the memory allocation performed by `LIBGEVOLUTION`, for example, by passing a custom allocator like the one used by `GADGET-4`. The MPI space of `LIBGEVOLUTION` should also be decided by the user, one should be able to choose how many processes are allocated to run the PM if one has in mind an heterogeneous parallel execution of code.
4. Error handling with C++ exceptions: exceptions constitute the native error handling mechanism in C++ (Stroustrup 2013), we started working on integrating that kind of logic into `LIBGEVOLUTION` which has a further complex structure due to MPI parallelization. In general, error events can be caused by either unexpected user input, failure of acquisition of a resource (memory allocation, opening a file, etc.) or a bug that breaks the correct execution of the program; in any case `LIBGEVOLUTION` should not cause the simulation to crash without letting the main application perform book-keeping measures, like saving data of the current state of the particles, logs of what event may have triggered the error, and properly release any locked resource.

Not all of these guarantees are fully implemented at the moment, but we are heading into that direction.

4.4 Classes and memory model

`LIBGEVOLUTION` is a PM-library whose cornerstone consists of three main components: a particle container implemented through the class `Particles_gevolution`, a PM data structure named `particle_mesh` that can be used either as a `relativistic_pm` or a `newtonian_pm` and an executable application that uses the previous components to produce N -body simulations as the original `GEVOLUTION` code does.

The `particle_mesh` has to be understood as a container that is aware of the MPI parallelization and distributed memory, that holds the state of the gravitational field (the metric perturbation components in the case of relativistic PM or the single gravitational potential in the case of Newtonian gravity) and allows the user to compute the forces acting on the simulation particles. The relevant user interface declared in `particle_mesh` consists of the following functions

- `sample(...)`, that builds the sources (desity field or energy momentum tensor) by projecting particles into the mesh.
- `compute_potential(...)`, that solve Poisson equations to compute the potential fields.
- `compute_forces(...)`, that computes the forces acting on the simulation particles.

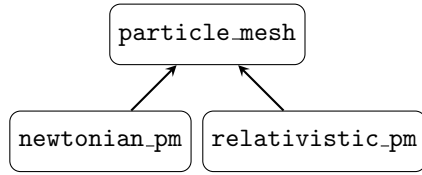


Figure 4.4: PM class hierarchy in LIBGEVOLUTION. `particle_mesh` is a virtual super-class containing prototypes and common methods. The `newtonian_pm` and `relativistic_pm` classes inherit from `particle_mesh` and correspond to a Newtonian and GR specialization respectively.

The `particle_mesh` is specialized to solve a Newtonian problem or a General Relativistic problem through class inheritance. Figure 4.4 illustrates the class hierarchy of LIBGEVOLUTION’s `particle_mesh`.

The `newtonian_pm` is the specialization of `particle_mesh` that contains a single scalar field represented as a real `LATfield2::Field`, the Newtonian gravitational potential Φ_{Newton} , plus a complex `LATfield2::Field`, the Fourier transform of the latter $\tilde{\Phi}_{\text{Newton}}$, and a `LATfield2::PlanFFT` that connects Φ_{Newton} with $\tilde{\Phi}_{\text{Newton}}$ through discrete Fourier transformation.

The `relativistic_pm` is the specialization of `particle_mesh` that contains all degrees of freedom of the perturbed FLRW metric. That is, Φ , B_i and χ as fields are represented as real `LATfield2::Field` variables with a complex `LATfield2::Field` counterpart and a respective `LATfield2::PlanFFT` for each one that connects them through discrete Fourier transformation.

4.5 Boosting Gevolution

Part of the development of LIBGEVOLUTION has been dedicated to the improvement of the numerical methods. This action included the upgrade the computation of spatial derivatives from the first order finite differences (FD), where derivatives are obtained by computing the difference between the current and the next mesh point:

$$\frac{\partial \Phi_i}{\partial x} = \frac{\Phi_{i+1} - \Phi_i}{h} + \mathcal{O}(h), \quad (4.1)$$

to the fourth-order approach, in which up to 2 mesh points in each direction are considered

$$\frac{\partial \Phi_i}{\partial x} = 8 \frac{\Phi_{i+1} - \Phi_{i-1}}{12h} - \frac{\Phi_{i+2} - \Phi_{i-2}}{12h} + \mathcal{O}(h^4), \quad (4.2)$$

where $h = L/N$ is the size of the mesh cells. These equations can be obtained by Taylor expansion. The fourth order FD equation is used in GADGET-4, thus we have decided to implement it as well into LIBGEVOLUTION to reduce the numerical discrepancies between both codes as much as possible.

Another numerical tuning was the application of correcting filters in Fourier space to compensate for the loss of power of the fields due to the sampling of particle density by means of the cloud-in-cell (CIC) kernel and the interpolation of force fields, again with CIC, at the particle’s positions. Thus, for the corrected mass density ρ' in a 1-dimensional mesh, its Fourier modes will be estimated starting from the density value obtained by

particle sampling ρ according to

$$\tilde{\rho}'_k = \tilde{\rho}_k \left(\frac{\pi k/N}{\sin(\pi k/N)} \right)^4. \quad (4.3)$$

This follows from eqs. (3.43) and (3.44), for CIC the power of the correction is $p = 2$, but since we apply the correction twice (one for sampling and one for interpolation effects) we get in total a power 4. The equation generalizes to 3 dimensions by multiplying for more $\left(\frac{\pi k/N}{\sin(\pi k/N)} \right)^4$ factors for each dimension.

To make the Newtonian PM scheme equivalent to that of GADGET-4 we changed the form of the discrete Laplacian operator in the Poisson equation solver from its original form

$$\nabla^2 \rightarrow -\frac{4N^2}{L^2} \left(\sin^2 \frac{\pi k_x}{N} + \sin^2 \frac{\pi k_y}{N} + \sin^2 \frac{\pi k_z}{N} \right), \quad (4.4)$$

described in Adamek et al. 2016, equation (C.5), to the form used in GADGET-4:

$$\nabla^2 \rightarrow -\frac{4\pi^2}{L^2} \left(k_x^2 + k_y^2 + k_z^2 \right). \quad (4.5)$$

This effort was aimed at improving the precision of the numerical methods in LIBGEVOLUTION as well as to make them compatible with GADGET-4's own recipes so that we could have a stable baseline simulation to check against.

From the code infrastructure point of view, some of the features we wanted to make available in LIBGEVOLUTION so that it could be integrated naturally into higher level applications, required some flexibility at the level of LATFIELD2. We cared to introduce that flexibility by using available infrastructure libraries on the market. The collection of Boost libraries² contains all the ingredients we needed to accomplish our purposes.

Boost is an open-source collaboration that created and maintains a set of C++ libraries to complement the lack of features of the Standard Library. The code in Boost is peer reviewed by noteworthy contributors and users alike, some of which are part of the standard comitee of C++. As a result, Boost libraries are considered one of the best examples of implementation best practices and reliable libraries for C++. Very often Boost libraries and concepts are evaluated by the rigorous C++ standard comitee and accepted as part of the standard library, for example `filesystem` library, *Polymorphic memory resources*, *smart pointers*, `chronos` date and time library and `string_view` class, to mention a few (Josuttis 2012; Josuttis 2017).

We *Boosted* our implementation of LIBGEVOLUTION and customized fork of LATFIELD2 with the aid of Boost.MPI and Boost.Serialization libraries. Boost.MPI, which is a C++ wrapper for MPI C-API, supports modern C++ development styles, including support for user-defined data types, Standard Library types and general function objects for collective algorithms. While Boost.Serialization, a library for the reversible deconstruction of an arbitrary C++ data structure into bytes, was used to write particle data into disk for testing and debugging, and to use the collective communication functions of Boost.MPI for our custom particle data.

With the help of these tools we were able to implement an initialization mechanism for LATFIELD2 with a user provided MPI communicator. This is a pattern of resource sharing based on rules and it is encouraged by the MPI standard best practices. In particular this will allow us to use LIBGEVOLUTION in applications in which some MPI

²<http://boost.org>

processes are selected to perform LIBGEVOLUTION tasks while others are in charge of other tasks; for an heterogeneous task distribution pattern. Without this feature we wouldn't be able to use LIBGEVOLUTION into GRGADGET.

We have also introduced a generic `for_each` function into LIBGEVOLUTION particle containers that allow us to perform any operation looping over the simulation particles. This function is a template so it is declared as

```
template<typename Functor>
void for_each(Functor updateF);
```

and it takes a function as its only argument. The syntax for the most common use case would involve calling it passing a lambda function

```
part_container.for_each(
    [&](particle& part, const Site& xpart)
    {
        // do whatever with part at xpart
    }
);
```

This is a pattern of functional and generic programming, inspired by the `std::for_each` function of the Standard Library. With this function we reduce redundant code and complicated function signatures; e.g. before this feature, there was a function `updateVel`, `updatePos`, with a long list of pointer inputs that tried to generalize the user case for performing operations of particle position and velocity updates—drifts and kicks basically. With our improvement we provide a more general and flexible interface and we gain performance because the compiler is able to inline at compile time the lambda function inside the `for_each` loop. This would not be the case for functions passed by pointers.

4.6 GrGadget code

We have used the code GADGET-4 from the public domain to construct a modified version that uses LIBGEVOLUTION as its PM solver so that we can, at least at the fixed resolution of the mesh, obtain relativistic effects on top of the realistic and highly non-linear simulation that the TREEPM method of GADGET-4 is able to deliver. Our new code was named GRGADGET.

The implementation of LIBGEVOLUTION in GADGET-4 was performed as follows. We created a new PM class with a similar interface as the original one in GADGET-4, so that it is initialized and executed with the same functions as GADGET-4, i.e. `init_periodic()` and `pmforce_periodic()`. A new class `relativistic_pm` was implemented within an `gadget::gevolution_api` namespace, avoiding to use the wider `gadget` namespace to make a clear distinction of purpose between the original GADGET-4 code and our additional features. This `relativistic_pm` class acts like a mediator taking information in and out of `gadget` simulation particles, processing the correct units conversion and calling the methods on `gevolution` namespace. Figure 4.5 shows a diagram that summarizes the contents of this PM class, its relation with GADGET-4's resources and the entry points for `gevolution`'s api.

Inside of GRGADGET, the `relativistic_pm` consists of:

- A variable of type `smparticle_handler` that acts as a wrapper for providing particle information from GRGADGET's `smparticles` global variable into a `Particles` `gevolution` data structure that integrates natively with LIBGEVOLUTION's methods

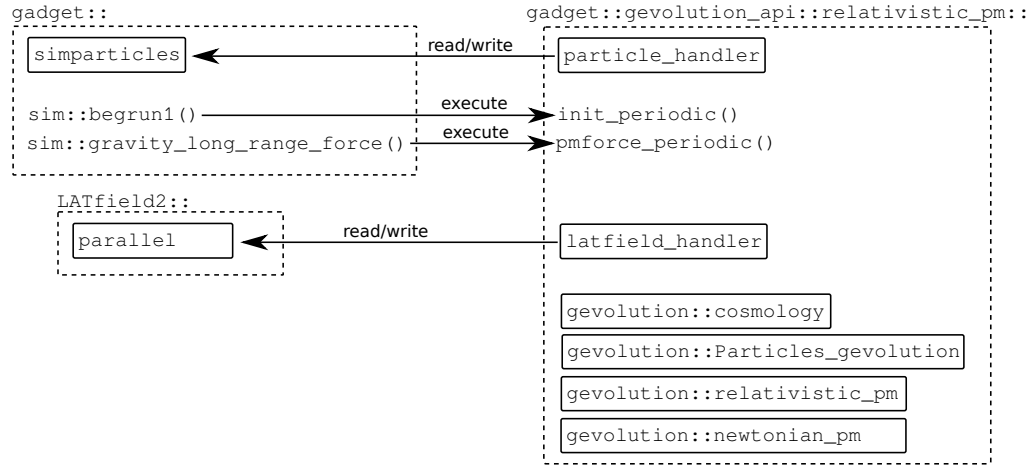


Figure 4.5: Diagram of resource ownership and relations for LIBGEVOLUTION integrated into GADGET-4’s workflow. Each solid box represents a memory resource (an instantiation of a variable type) while the dashed boxes indicate ownership. The newly developed code, represented in the right part of the diagram denoted with the namespace `gadget::gevolution_api`, consists in a class named `relativistic_pm` that owns a `particle_handler` object that reads and writes directly into `gadget::simparticles`, a `latfield_handler` that takes care of setting up and inspect the state of `LATfield2::parallel`, and some types defined in LIBGEVOLUTION, that are defined in `gevolution` namespace, like `cosmology`, `Particles_gevolution` and `relativistic_pm`. The methods `sim::begrun1()` and `sim::gravity_long_range_force()` in `gadget` interact with the `relativistic_pm` through its interface `init_periodic()` and `pmforce_periodic()`.

to construct the energy-momentum tensor on the mesh. `simparticle_handler` is also used to write back the data produced by LIBGEVOLUTION’s PM into GRGADGET particles. These procedures take into account the fact that the domain decomposition of the parallel BH-TREE and PM follow different patterns (Peano-Hilbert partition of space in the case of the tree and two-dimensional pencil decomposition of the mesh) with the result that each particle could belong to the local domain of one process’ tree and at the same time its position in space lies inside the mesh domain of another process. This problem is solved in a very robust but inefficient way: particles are copied into buffer memory and sent from their BH-TREE domains into their corresponding PM domains everytime the PM is called.

- A variable of type `latfield_handler` that takes care of correctly initializing LATFIELD2 global state. Indeed, while GRGADGET can run with any number of MPI processes, LATFIELD2 imposes limitations that depend on the number of grid points in the PM; such that the number of MPI process must be a composite number $P = P_x P_y$, where P_x and P_y are the sizes of the process grid in the x and y spatial directions, furthermore, P_x and P_y must divide the number of grid points per dimension N . In GRGADGET we avoid these constraints by automatically selecting a subset from the global MPI communicator; the maximum number of processes that satisfy LATFIELD2 conditions and initialize `Parallel` state with that sub-communicator.

- A variable of type `gevolution::cosmology` that contains the parameters for the background evolution.
- A container of type `gevolution::Particles_gevolution` that holds particle information, stored according to their location on the PM grid. This container integrates natively with `LIBGEVOLUTION` methods that construct the energy-momentum tensor by sampling particle's dynamical state and the interpolation of field quantities into particles positions.
- Variables of type `gevolution::relativistic_pm` and `gevolution::newtonian_pm` that perform the actual PM computations, i.e. construct the sources, either density or the components of the energy-momentum tensor, compute the gravitational potential or the metric perturbation fields and the forces that act upon the particles.
- The methods `pm_init_periodic` and `pmforce_periodic`, for initialization and execution of the PM, respectively.

In order to keep the Hamiltonian character of the equations of motion in GRGADGET, we have to describe the state of each particle through its position and momentum, not velocity. Following a leap-frog scheme, the momentum should be updated with a kick operation using the full relativistic Eqs. (2.82) and (2.83). However, velocities in GRGADGET are to be interpreted as momenta (per unit mass) of non-relativistic particles in the Newtonian limit. Then we redefine GADGET-4's kick and drift operators assuming non-relativistic matter, $p \ll mca$, and further neglecting the very small contribution coming from χ :

$$\frac{dx^i}{d\tau} = \frac{p^i}{ma} (1 + 3\Phi) + cB^i, \quad (4.6)$$

$$\frac{dp_i}{d\tau} = -cp^n B_{n|i} - \Phi_{,i} mc^2 a. \quad (4.7)$$

The right hand side of (4.7) is what we call *force*.

To combine the forces computed with the relativistic PM and GADGET's Newtonian Tree we have extended the idea of the TREEPM coupling. From equation (3.45) one obtains that the force acting on a particle in a TREEPM scheme consists of two terms:

$$\vec{F} = L_{r_a}[\vec{F}_{\text{Newton}}^{\text{PM}}] + S_{r_a}[\vec{F}_{\text{Newton}}^{\text{Tree}}]. \quad (4.8)$$

The first term is the force computed using the PM on which an exponential cut-off filter L_{r_a} is applied, that removes short-wavelength and leaves long-wavelength modes; the L stands for *long*. The second term corresponds to the BH-TREE force on which the complementary filter S_{r_a} is applied to leave short-wavelength modes and remove the long-modes; the S stands for *short*. The symbols S_a and L_a formally denote the operators:

$$L_{r_a}[f](\vec{r}) = \frac{1}{N} \sum_{\vec{k}} \tilde{f}_{\vec{k}} \exp(-k^2 r_a^2) \exp(i\vec{k} \cdot \vec{r}), \quad (4.9)$$

and

$$S_{r_a}[f](\vec{r}) = \frac{1}{N} \sum_{\vec{k}} \tilde{f}_{\vec{k}} (1 - \exp(-k^2 r_a^2)) \exp(i\vec{k} \cdot \vec{r}). \quad (4.10)$$

The *grid smoothing scale* r_a is a simulation parameter expressed in units of the PM mesh size L/N , its value is optimized in GADGET-4 to minimize the impact of the two different treatments of the gravitational force.

In order to account for the relativistic dynamics while preserving the match between tree and PM contributions that is valid in the Newtonian case, we choose the following strategy: GRGADGET calls both `newtonian_pm` and `relativistic_pm` from LIBEVOLUTION, the Newtonian value of the force is added to the Tree force as in a standard Newtonian simulation, while the difference between the Newtonian and the relativistic forces is added on top as a correction, but filtered on a different scale r_b , that we call *gr-smoothing scale*. Eq. (4.8) then becomes:

$$\vec{F} = S_{r_a}[\vec{F}_{\text{Newton}}^{\text{Tree}}] + L_{r_a}[\vec{F}_{\text{Newton}}^{\text{PM}}] + L_{r_b}[\vec{F}_{\text{GR}}^{\text{PM}} - \vec{F}_{\text{Newton}}^{\text{PM}}]. \quad (4.11)$$

The case $r_a = r_b$ would correspond to simply adding the relativistic force to the Tree:

$$\vec{F} = S_{r_a}[\vec{F}_{\text{Newton}}^{\text{Tree}}] + L_{r_b}[\vec{F}_{\text{GR}}^{\text{PM}}]. \quad (4.12)$$

However, while the size of r_a , that regulates the match between Newtonian Tree and PM forces, is very well tested within GADGET-4, the optimal value of r_b is to be found; we will show in the next Chapter that using r_b larger than r_a allows us to achieve percent accuracy at small scales.

Code validation

In Chapter 4 we have discussed some changes we have introduced in the back-end of LIBGEVOLUTION’s numerical algorithms that we consider as improvements with respect to the original GEVOLUTION code for our purposes. The choice for a specific back-end in scientific computing is generally justified if it favours one desired feature over another. For example we may be more interested in obtaining a faster code at the expenses of accuracy, or maybe the typical problem we are dealing with maybe better solved with a code that does not perform well in other domains. While designing LIBGEVOLUTION we had in mind its incorporation in GADGET-4, hence we had chosen a back-end that could reproduce with accuracy the results that we could obtain with GADGET-4’s original PM. The main reason for this choice is to be able to have a common baseline in Newtonian simulations either by running GADGET-4 original code or by running GRGADGET in Newtonian mode. In this chapter we will present some tests we have performed to justify and validate the changes in LIBGEVOLUTION’s back-end.

The test were based on running the different codes on a few realizations of initial conditions, listed in table 5.1. These were generated with GADGET-4’s `ngenic` code at $z = 19$, starting from a linear power spectrum generated with CAMB¹ and with cosmological parameters consistent with Planck 2018 result (Planck Collaboration et al. 2020): $\Omega_b h^2 = 0.0223$, $\Omega_c h^2 = 0.120$, $H_0 = 67.3 \text{ km s}^{-1} \text{ Mpc}^{-1}$, $A_s = 2.097 \times 10^{-9}$ and $n_s = 0.965$. All simulations are dark matter only.

Simulations were performed with the HOTCAT system of INAF with Intel® Xeon® Gold 5118 CPU @2.30 GHz processors with 12 cores each and 128GB of RAM, every compute node on this machine has 4 sockets for a total of 48 cores and 512GB of RAM per node (Taffoni et al. 2020; Bertocco et al. 2020).

¹<https://camb.info/>

name	N_p (particles)	N (PM grid points)	L (box size)
N64	64^3	64	1 Gpc/h
N256	256^3	256	1 Gpc/h
high_res	512^3	512	500 Mpc/h

Table 5.1: Cosmological simulation configurations used to validate GRGADGET.

5.1 Gevolution and Gadget-4 original codes

GEVOLUTION computes Newtonian forces that are at some extent different from those obtained with GADGET-4's PM—in Chapter 4 we have listed the main differences in the numerical methods used by the PM routines in these two N -body codes. These differences can be noticed by running dark matter only simulations with both codes starting from the same initial conditions and printing out the 3-dimensional PM forces acting on individual particles.

Figure 5.1, for example shows the dispersion of the force terms with respect to the theoretical real force in a Newtonian simulation. This was obtained by running a simulation with a small number of particles, corresponding to the configuration N64 (see table 5.1). At later times, $z = 8$ and $z = 0$, we have taken snapshots of the gravitational forces that the particles experience and we have compared those to the result of direct summation PP. We have classified those particles into bins according to the value of the force obtained with PP. The curves represent, for each bin, the mean difference of a single component of the real force, calculated by direct summation and the estimated force from a PM or TREEPM method. The shaded regions indicate one standard deviation. Forces are given in GADGET-4's default units, which are actually accelerations, force per unit of mass, equivalent to $10H_0\text{km/s} = h\text{km}^2\text{s}^{-2}\text{kpc}^{-1}$. The green line represents the pure PM method using the original GEVOLUTION code, the red line was obtained from pure PM using GADGET-4's original code and for completeness we also show a black line obtained with the TREEPM method of GADGET-4. The new PM implementation in LIBGEVOLUTION produces the same results that we have obtained with GADGET-4's original PM.

Out of those plots we can deduce that the TREEPM method produces far less dispersion and bias when estimating forces, see for instance in figure 5.1 at $z = 8$: the error is approximately equal to $0.1h\text{km}^2\text{s}^{-2}\text{kpc}^{-1}$ while for $z = 0$ it is barely visible. We also notice that at $z = 0$ there is a trend of the PM method to underestimate the Newtonian force as its value increases. Higher magnitudes of the forces are obtained either with higher mass densities or closer distances, by failing to resolve interaction at scales smaller than the grid resolution the PM is systematically underestimating the gravitational forces. Another interesting feature out of this test is the presence of an S shape in GEVOLUTION's PM force dispersion much more visible at $z = 8$, which is mostly due to the first-order interpolation used to find the gradient of the potential.

In Fig. 5.2 we show the matter power spectra² obtained from a set of larger simulations with the configuration N256—see table 5.1. The red solid line shows the result obtained with the original GADGET-4 code with its TREEPM method, while the red dotted line shows the results obtained by switching off the Tree so that the forces are computed using the PM alone. The green lines show results obtained with the latest version of GEVOLUTION that implements higher order schemes for finite differences; the dotted line gives results obtained with `GRADIENT_ORDER=1` and is identical to the result obtained with V1.2 of GEVOLUTION, the green solid line uses `GRADIENT_ORDER=2`, that corresponds to a second-order scheme. These power spectra show that the matter distribution in GEVOLUTION using first-order gradients loses power in what seems to be a uniform trend for large-scale modes. This is a behaviour that is not inherent to the PM nature of the code, since that type of numerical approximation should predict very well the linear evolution at large scales; indeed, the higher-order scheme recovers power on large scales

²All particle power spectra were computed using `PowerI4` code presented in Sefusatti et al. 2016. Unless otherwise state, all power spectra are computed up to the Nyquist frequency of the PM.

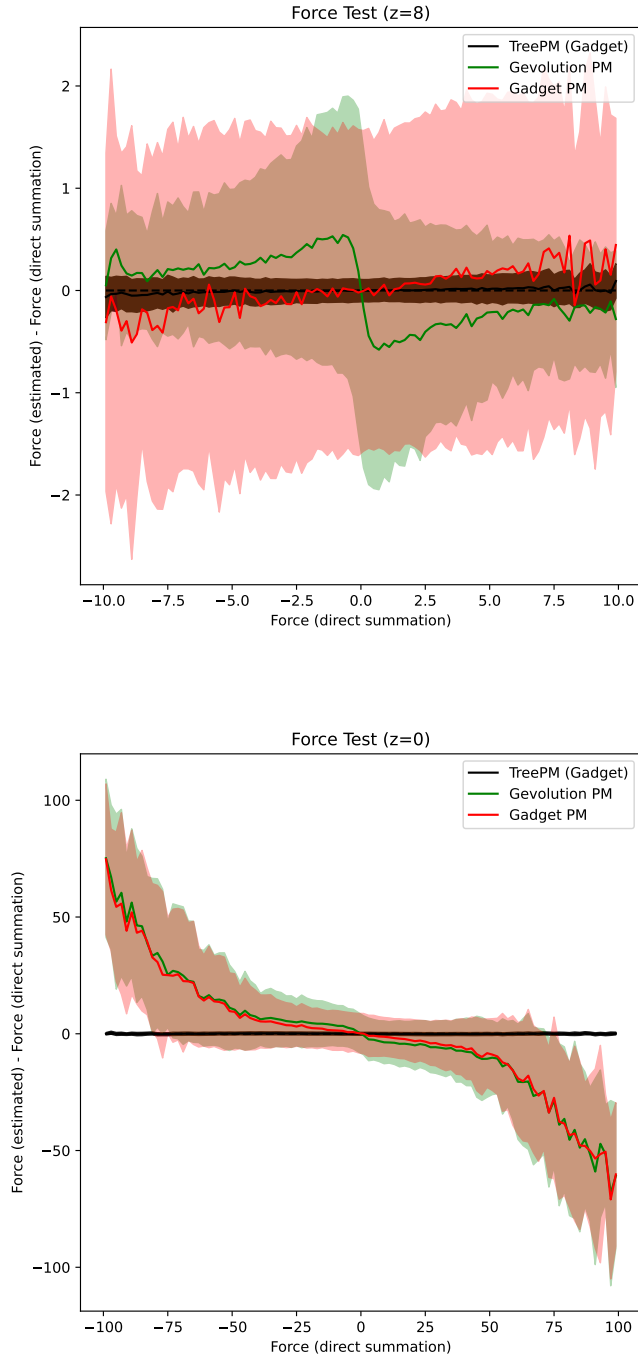


Figure 5.1: Difference of gravity force with respect to the true PP value, binned according to the true force, for $N64$ initial conditions. The top and bottom panels correspond to snapshots at $z = 8$ and $z = 0$ respectively. The lines represent the mean value of the force difference in the bins and the shaded regions indicate its standard deviation. The forces are measured in units of $h \text{ km}^2 \text{ s}^{-2} \text{ kpc}^{-1}$.

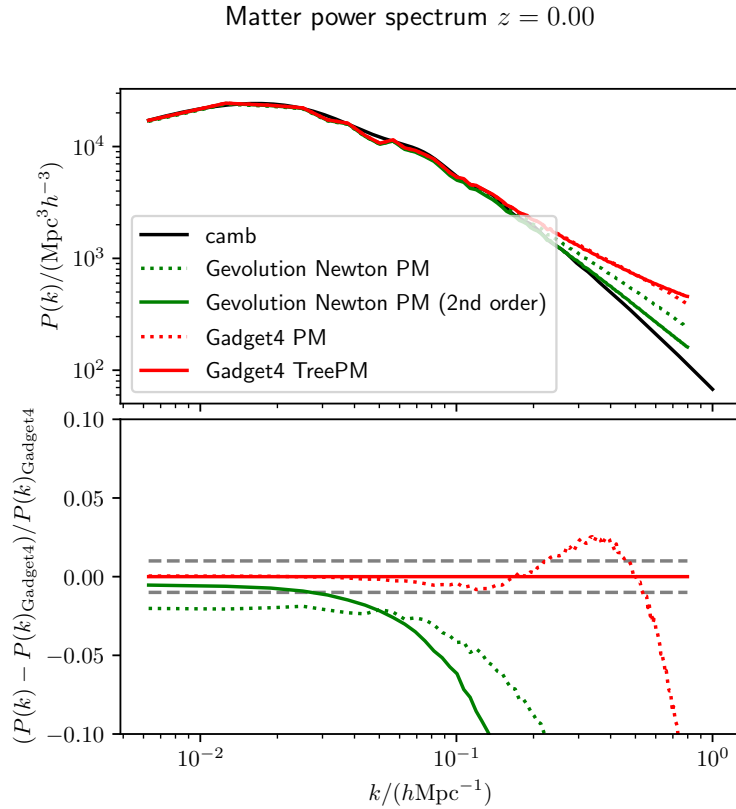


Figure 5.2: Matter power spectrum of N256 cosmological simulations. The lower panel shows residuals with respect to GADGET-4’s original code (in red), used as baseline. The black line shows the linear power spectrum obtained with CAMB. Red lines show results obtained with GADGET, with the Tree part on (solid line) or switched off (dotted line). Green lines show results obtained with GEVOLUTION in Newtonian configuration, with finite differences at first order (dotted line) or second order (solid line).

to sub-percent accuracy. Conversely, GADGET-4’s PM and TREEPM agree very well at wavenumbers below $k \sim 0.1 \text{ Mpc}^{-1} h$.

The higher-order differentiation worsens the loss of power of GEVOLUTION for high values of k , that is not present in GADGET-4. This can be explained as a consequence of the particle-to-mesh sampling and mesh-to-particle interpolation described in section 3.2. GADGET-4’s PM corrects for these effects, resulting in a power spectrum that degrades only at very high values of k as we approach the Nyquist frequency, while producing a $\sim 2\%$ overcorrection at $k \sim 0.4 \text{ Mpc} h$.

5.2 Newtonian forces

We have tested our implementation of the GRGADGET code by running a standard test in GADGET-4: we create an N -body configuration in which there is a single massive particle in the entire simulation box, while other massless test particles are placed at different distances from the first. In this setting the exact value of the force on each

particle is known, hence one can compare the numerical results coming from the TREEPM algorithm to the analytical solution. This test explores the precision of the two methods for computing forces at long and short distances and the way they are combined at the intermediate scales.

The results are shown in figure 5.3, where each dot represents a test particle. The x-axis gives the distance to the massive particle that sources the gravitational field, in units of the PM resolution (L/N), while the y-axis gives the corresponding absolute value of the relative difference of the true and estimated forces acting on the test particle. The red and blue lines correspond to the mean value of force residuals, for particles binned into distance bins; the red line denotes the statistics obtained from a simulation using GADGET-4's original TREEPM implementation and the blue line was produced using GRGADGET, in this case with the Newtonian gravity engine. This figure shows that the accuracy with which the TREEPM code reproduces the gravitational force is at worst at percent level on scales of a few mesh cells, corresponding to the scale where the PM and Tree contributions are matched, and gets very accurate in the limits where either the Tree (small scales) or the PM (large scales) dominates. These numbers are consistent with the TREEPM accuracy reported in GADGET-4's paper (Springel et al. 2021). GADGET-4's and GRGADGET's Newtonian PMs show basically the same accuracy, even though their PM implementations are different. However both codes use the same set of numerical recipes to perform PM operations.

In Fig. 5.4 we show the matter power spectra of a set of N256 simulations (see table 5.1). In this case we are comparing the matter clustering of GRGADGET, in blue (with Newtonian forces for testing purposes), against GADGET-4, in red. In agreement with the previous test of force differences, we find that both codes produce the same matter power spectrum up to floating point errors. This is verified both in the case of simulations computing forces using a pure PM and in the case of TREEPM.

5.3 Relativistic Cosmology with GrGadget.

We present here results obtained by running GRGADGET with `relativistic.pm`, comparing them with the corresponding relativistic version of GEVOLUTION. We expect that the power spectrum of the matter density displays some relativistic features at the large scales due to terms preceded by \mathcal{H} in the field equation (2.70), while at small scales the results should be compatible with GADGET-4's Newtonian simulations. However, the matter power spectrum shown here is not an observable quantity, so this comparison is just meant to give a first validation of the results.

Figure 5.5 shows the matter power spectra for a series of N256 simulations (see table 5.1). In this case GEVOLUTION and GRGADGET are run in GR mode. The parameter that regulates the scale of the relativistic correction (Eq. 4.11) is set to $r_b = 6L/N \approx 23 \text{ Mpc}/h$, i.e. the relativistic corrections of the PM method are smoothed at a distances below 6 grid cells. The plot shows that relativistic PM-only simulations, GRGADGET (blue dotted line) and GEVOLUTION (green lines) are compatible in the large scales ($k < 0.03 \text{ Mpc}^{-1}h$) up to a small percent-level difference that it is likely caused by the use of different gradient approximation order. The plot also confirms that our combination of Tree and PM forces in the relativistic weak field limit with GRGADGET (blue solid line) reproduces the Newtonian non-linear features to sub-percent level at small scales, that is for $k > 0.1 \text{ Mpc}^{-1}h$; here GADGET-4 (red solid line) is again our reference for the non-linear clustering.

Designed for the use of Fourier methods from the start, LIBGEVOLUTION provides an

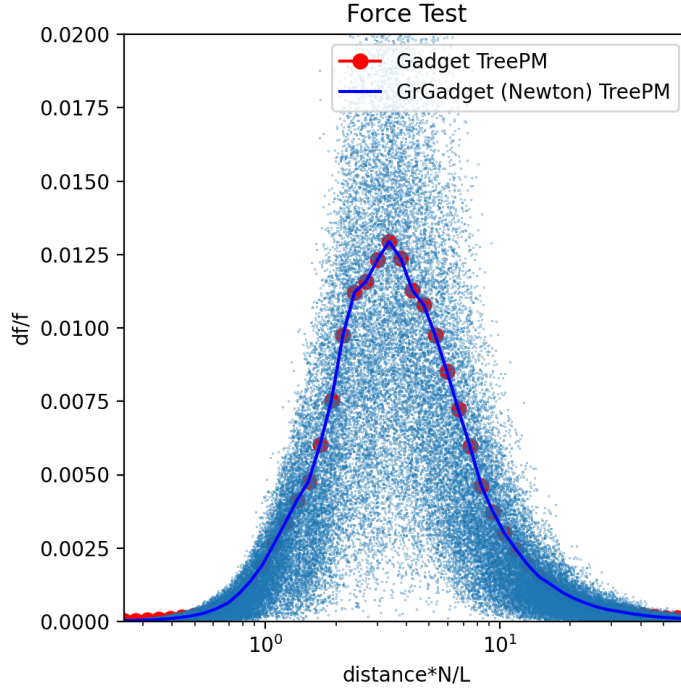


Figure 5.3: Forces due to a point source: the points are test particles located at different distances (in units of the mesh resolution L/N) from the source and the lines represent the RMS of the difference between real and TREEPM forces in different distance bins. The red line corresponds to GADGET-4 original TREEPM while the blue line was obtained with GRGADGET in Newtonian mode. As for the the grid smoothing scale, the default value was used: $r_a = 1.25L/N$. For this test we have used $N = 256$ and $L = 1 \text{ Gpc}/h$.

interface for the computation of the power spectrum of the fields defined within. Thus we can also extract and analyse the power spectra of the individual components of the metric perturbations from the relativistic simulations. Figures 5.6 and 5.7 show the power spectra of the relativistic potentials, Φ , B_i and χ , for a high resolution configuration `high_res` (see table 5.1). These plots show a comparison of PM (blue lines) and TREEPM (red lines) simulations. The power spectrum of the gravitational potentials converge for both methods in the large scales. However, below $1 \text{ Mpc}/h$ the PM-only simulation loses power with respect to the TREEPM, the differences can reach up to 40% as we approach the Nyquist frequency. This pattern is equally found for the scalar fields Φ and χ , as well as for the individual components of B_i .

The bottom plot in Fig. 5.6 helps to understand the reason behind this result. Generally speaking, energy density, momentum density and their respective density currents (the components of the Energy-Momentum tensor) are sources of the metric perturbations. Even though those quantities, as fields, are found at discrete positions of space defined by the mesh, their values are computed by sampling the energy and momentum carried by the particle distribution, which contain information on the clustering due to the short range interactions (through the Tree) that goes well below the mesh resolution L/N . Therefore, TREEPM simulations, having power on scales well smaller than the PM

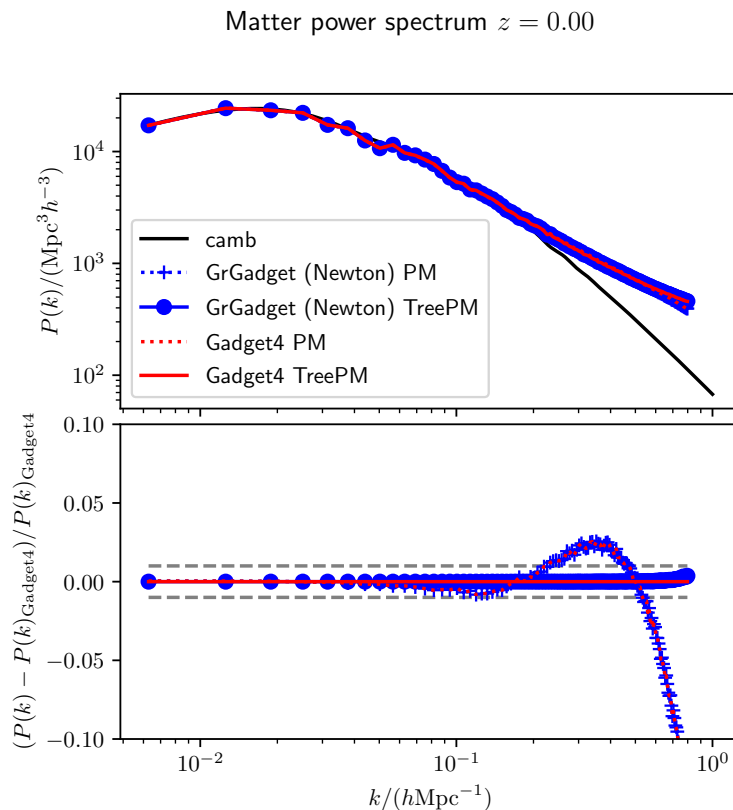


Figure 5.4: Matter power spectrum of four simulations starting from the same initial conditions `high_res`: blue lines give results for GADGET-4 original code, red lines give results for GRGADGET. In both cases dotted lines refer to runs with PM-only, solid lines refer to runs with full TREEPM.

resolution, give a better representation of the source of metric perturbation, and thus allow to recover power at the highest frequency modes right below Nyquist. Fig. 5.6 to the bottom highlights the particular case of T_0^0 (the matter density) as a source for Φ ; by comparing T_0^0 with $k^2\Phi$, we are verifying the Poisson equation $k^2\Phi \approx \tilde{T}_0^0$ that is valid for wavelengths below the Hubble horizon. This confirms that the presence of small-scale clustering in the particle distribution propagates to the gravitational fields up to the maximum resolution that the PM allows. The same thing is visible in the vector modes B_i and in χ (Figure 5.7), where we also notice a small, few-percent mismatch on large scales on these fields that are known to give sub-percent effects on observables, so this difference, that is likely due to some degree of mode coupling, is non considered as a problem.

In figure 5.8 we show how the matter power spectrum obtained using GRGADGET is affected by the choice of the gr-smoothing scale parameter r_b . We have used an N256 box configuration to perform this test. For the values tried $r_b = 1.5, 3, 6$ in units of $L/N \approx 4\text{Mpc}/h$, we find that the large scales correlations do not depend on the value of the parameter. The structures at scales below the PM resolution are resolved by the Tree algorithm, hence for $k > k_{\text{Nyquist}}$ there is a convergence of all simulations to a common non-linear power spectrum tail. It is in the medium to small scales $k_{\text{Nyquist}} > k > 0.2\text{Mpc}^{-1}h$

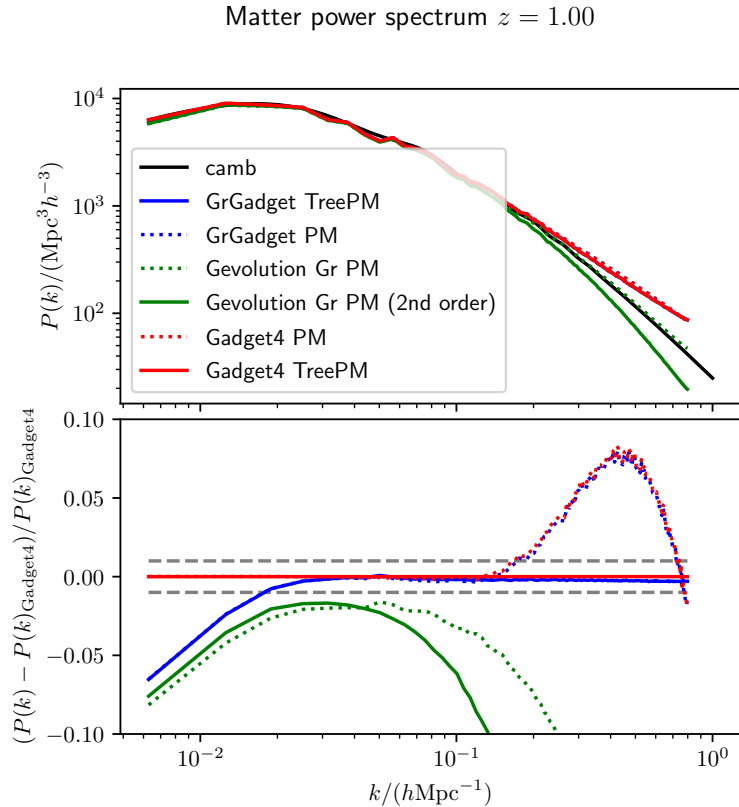


Figure 5.5: Matter power spectrum of simulations run with GADGET-4, GEVOLUTION and GRGADGET (in relativistic mode) with the N256 configuration. The upper panel shows the absolute value and the lower panel the relative difference with respect to GADGET-4’s TREEPM. The black line gives the linear matter power spectrum; red and blue lines give GADGET-4 and GRGADGET results, with full TREEPM forces (solid lines) or with the Tree switched off (dotted lines). Green lines give GEVOLUTION results, dotted line referring to first-order finite differences (GRADIENT_ORDER=1) and solid line referring to second-order calculation (GRADIENT_ORDER=2).

that we notice differences in the power spectrum above the $\sim 1\%$ (dashed grey line). For small values of r_b ($\sim 1.5 L/N$), we obtain discrepancies in the power spectrum at $k \sim 0.5 \text{ Mpc}^{-1} h$ that can be as large as 5% and indicate the limitations of our force summation scheme, Eq. (4.11). A value of $r_b = 3 L/N$ or possibly higher is needed to obtain a good compatibility of GRGADGET and GADGET-4 for all modes greater than $0.1 \text{ Mpc}^{-1} h$, where relativistic features in the matter clustering is negligible.

The last test we present here regards the convergence of the numerical results for increasing resolution. Figure 5.9 shows the matter power spectrum obtained from running GADGET-4’s TREEPM (red lines), GRGADGET with PM-only (blue dotted lines) and GRGADGET with TREEPM (blue continuous line). These various code configurations were run with different combinations of the number of grid points per dimension $N = 256$, $N = 512$ and box length $L = 250, 500, 1000, 2000 \text{ Mpc}/h$; the number of particles was set $N_p = N^3$. In all cases we have set the PM smoothing scale to $r_a = 1.5 L/N$ and the

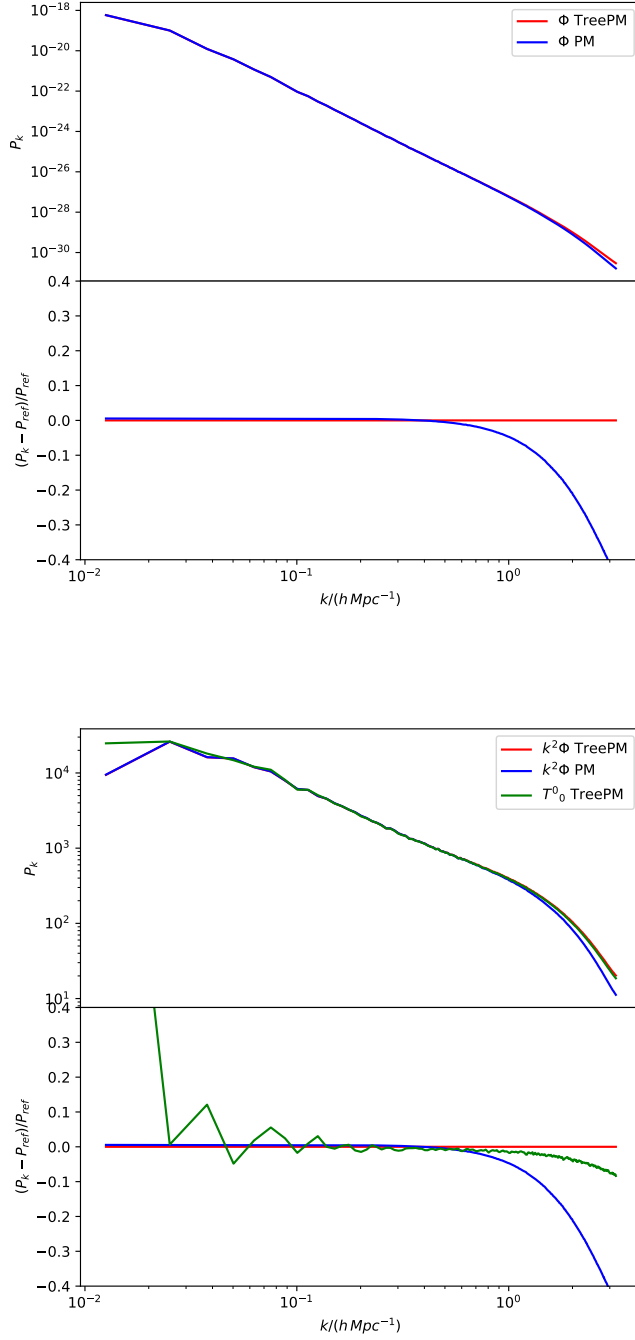


Figure 5.6: In the upper plot: power spectrum of the metric perturbation Φ in a `high_res` simulation obtained with GRGADGET. In the bottom plot: power spectrum of $k^2\Phi$ and T^0_0 . For modes well below the Hubble horizon and small perturbations it should be verified that $k^2\Phi \approx \hat{T}^0_0$.

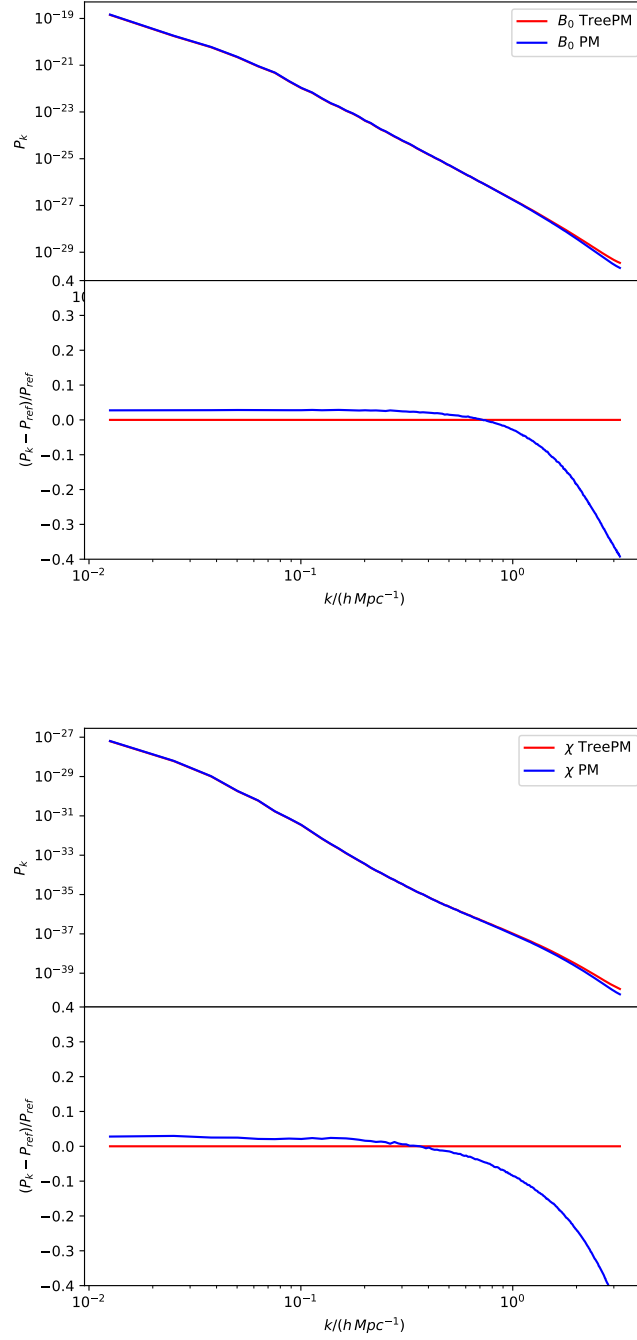


Figure 5.7: In the upper plot: power spectrum of the metric perturbation B_i in a `high_res` simulation obtained with GRGADGET, for one component. In the bottom plot: power spectrum of χ .

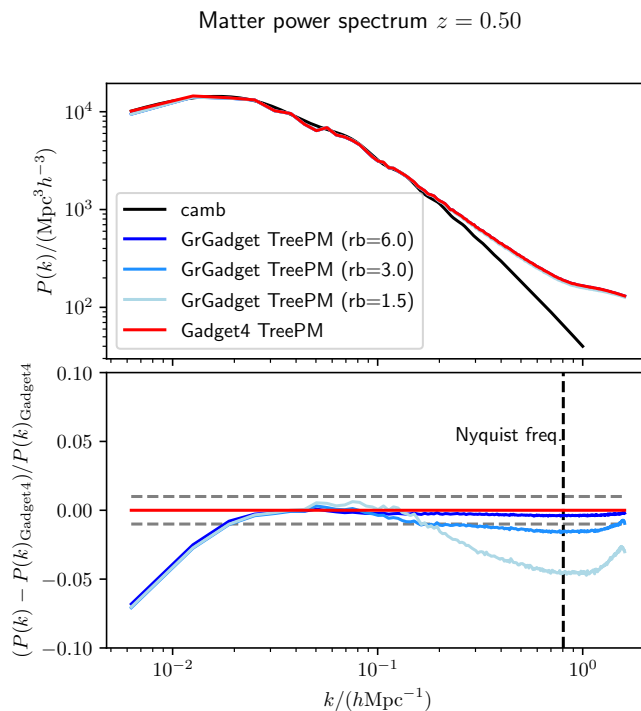


Figure 5.8: Power spectrum of matter density for GADGET-4 and GRGADGET, on a N256 configuration. The upper panel shows the absolute value and the lower panel the relative difference with respect to GADGET-4’s TREEPM. Different shades of blue indicate different values of the gr-smoothing scale parameter $r_b = 1.5, 3, 6$ in units of L/N . The PM smoothing scale is fixed to $r_a = 1.5 L/N$. The power spectra in this plot are computed beyond the Nyquist frequency to show the convergence of the matter distribution correlations for distances below the grid resolution, the Tree regime.

gr-smoothing scale to $r_b = 3 L/N$. It can be observed with the finest resolution, in the top plots, that there is a matching between General Relativity and Newtonian dynamics in the small scales. Then as the mesh size becomes coarser, in the middle plots, some discrepancies in the power spectrum start to appear which become more evident for even coarser meshes, in the bottom plots.

We argue that this mismatch is not caused by $r_b = 3 L/N$ moving towards larger scales as L/N increases, forcing the Newtonian limit on the motion equations (2.82) and (2.83) at scales where the Hubble horizon cannot be neglected. Because we saw already in Fig. 5.8 that the contrary effect occurs; i.e. the match between Newtonian simulation and the Relativistic one becomes better in the small scale regime when r_b is increased—at least up to the values that we tested that did not exceed $23 \text{ Mpc}/h$. On one hand in Fig. 5.8 we have shown that a value of $r_b = 6 L/N \approx 23 \text{ Mpc}/h$ (for $N = 256$ and $L = 1 \text{ Gpc}/h$) was a good choice to obtain an under percent matching of the TREEPM at all scales, on the other hand we see in Fig. 5.9 that $r_b = 3 L/N \approx 12 \text{ Mpc}/h$ for $L/N \approx 4 \text{ Mpc}/h$ produces an almost 2% deviation in the TREEPM power spectra approaching k_{Nyquist} and for $L/N \leq 2 \text{ Mpc}/h$ leading to $r_b \leq 6 \text{ Mpc}/h$ the deviations are again below the 1%. Therefore it seems that for a fixed resolution L/N higher values of r_b relative to

L/N improve the TREEPM agreement in our summation scheme (4.8). Also the finer the resolution of the mesh the better this agreement and bigger values of $r_b/(L/N)$ are not needed.

5.4 Performance

GRGADGET inherits from GADGET-4 and GEVOLUTION its parallel execution patterns. However, the work we have done integrating GEVOLUTION's PM into GADGET-4 could have effects degrading the overall performance of GRGADGET if it is not done properly. This could result in extremely large execution times even as the number working processors is increased. In order to run the code in production we first need to understand if the code *scales*, and it would be also interesting to know how much overhead we pay for using a relativistic PM instead of the Newtonian. This section shows the results of some timing and scalability tests we have subjected GRGADGET to.

In figure 5.10 we show the result of a comprehensive measurement of execution times for GRGADGET's relativistic PM with respect to the total execution of the simulation, in different problem configurations and by changing the number of MPI processes. In general, we learn that the PM in GRGADGET takes around 15% fraction of the total execution time in the worst case. This is consistent with the knowledge that PM methods are in practice far less computationally expensive than its BH-TREE counterpart.

The worst performance of the PM is obtained for the denser grids corresponding to $L = 250 \text{ Mpc}/h$ with $N = 512$ and $L = 500 \text{ Mpc}/h$ with $N = 512$. This is expected as the execution time of the PM code on an individual MPI process depends on the number of particles stored in local memory—the number of CIC projections and interpolations is what actually depends on how many local particles there are, while the field computations, that consist in FFTs and solving the partial differential equations for every mesh point, are not affected. Total execution time of the parallel PM is then the time it takes for the slowest process to finish. Hence, if there is an uneven distribution of particles among MPI processes, some of them will finish the PM execution earlier and others will have a delay because they perform more computations. In denser simulations there will be a better resolution of the non-linear clustering which results in regions of the simulation box containing more particles than others, at the same time the domain decomposition of the PM does not take that clustering into consideration, instead it's fixed by the geometry of the box (see Fig. 4.1), and thus there will be an uneven distribution of computational work among the MPI processes that results in the loss of performance of the PM with respect to the rest of the code.

From Fig. 5.10 we observe there is not a strong correlation of the relative execution time of the PM with the number of MPI processes running the code. That is, the PM scales with the increase of computational resources approximately the same way the rest of the code does.

In order to evaluate the overhead of the GR PM with respect to the Newtonian one, we show in figure 5.11 a series of time measurements of the PM relative to the total execution time for different simulation configurations. Again, we can appreciate the increase of relative runtime of both Newtonian (in red) and GR (in blue) PMs when the resolution of the simulation is increased (lower values of L/N); as we discussed previously this behaviour is expected. The main observation, however, is that the computational cost of running a GR PM is 10 times higher than the Newtonian original PM of GADGET-4. This is due to the fact that: first, the code has not yet been thoroughly optimized and secondly, in the GR PM there are more fields to take into account that increase the number of FFTs

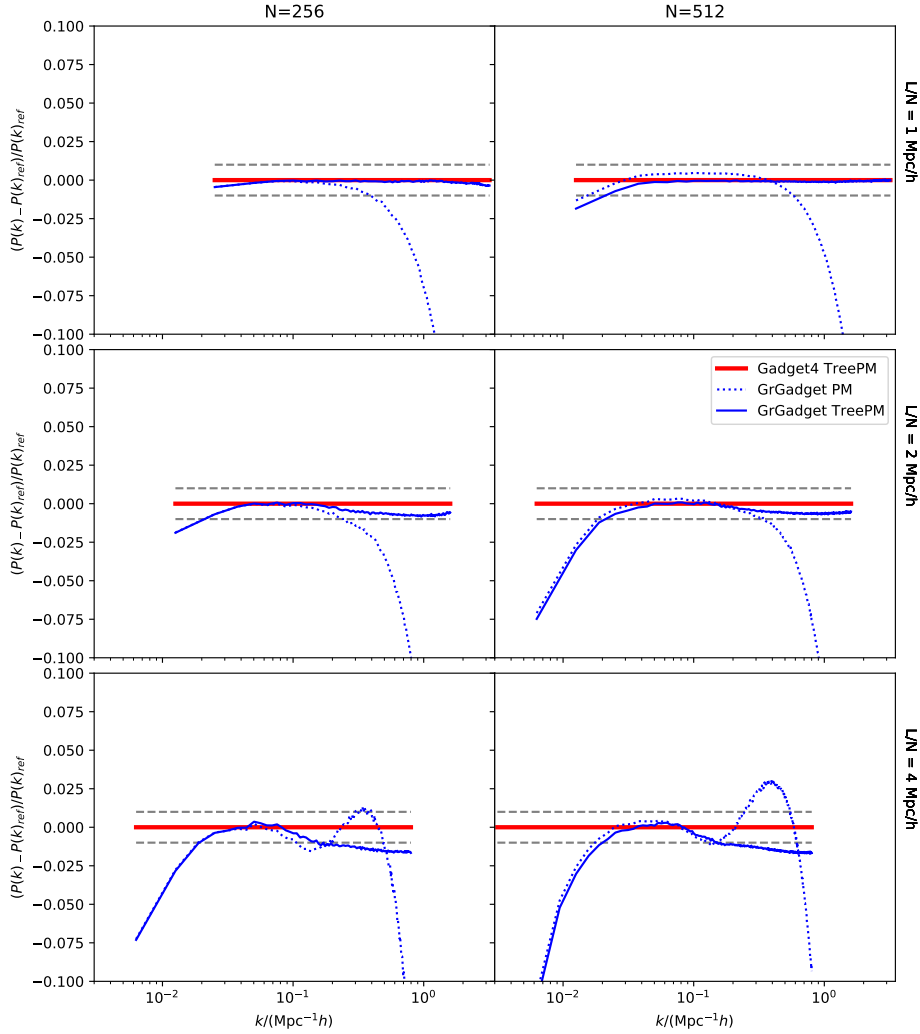


Figure 5.9: Matter power spectrum from cosmological simulations at $z = 0$ using GRGADGET (the blue lines) and compared to GADGET-4 (the red line) at $z = 0$. The dotted line is obtained with a simulation in which only the PM is used to compute forces. The plots show the relative difference with respect to the power spectrum obtained with GADGET-4. The left column corresponds to simulations with $N = 256$ grid points per dimension while for the right column $N = 512$. The boxsize changes along the ranks so that for the top plots the resolution is the highest $L/N \approx 1 \text{ Mpc}/h$, in the middle $L/N \approx 2 \text{ Mpc}/h$ and the bottom plots correspond to $L/N \approx 4 \text{ Mpc}/h$. In all cases $r_a = 1.5 L/N$ and $r_b = 3 L/N$. The grey dashed line indicate a 1% error.

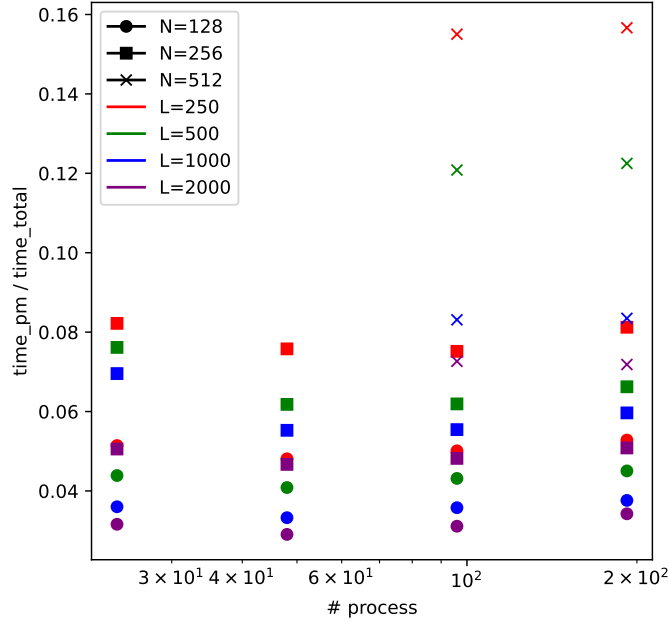


Figure 5.10: Fraction of total runtime spent in PM computations in GRGADGET for several combinations of the mesh size $N = 128, 256$ and 512 ; boxlength $L = 250, 500, 1000$ and $2000 \text{ Mpc}/h$; and number of MPI processes $N_{\text{MPI}} = 24, 48, 96$ and 192 . The number of particles is determined as $N_p = N^3$.

and linear operations to perform—in the case of the Newtonian PM there is only one gravitational potential field, while in the GR PM there are three scalars and one vector potentials adding up to six components in total.

Strong scalability

To take profit of parallel execution, codes need to scale as the number of computational resources increase—that is the number of processors. For example, it wouldn't make sense to run the code in a bigger machine if it takes the same amount of time or even less to run it in a smaller one. The scalability of a code depends on many factors: the nature of the problem (if it can be parallelized), the actual implementation (e.g. the domain decomposition) and the coordination overhead of the participating processes. Ideally, for a perfect parallel application the execution time $T_{P,M}$ depends on the number of processors P and the linear size of the problem M

$$T_{P,M} = \alpha \frac{M}{P}, \quad (5.1)$$

where α is a proportionality constant.

The *strong scalability* is a measure of how well the code follows the ideal scaling behaviour (5.1). It is obtained by fixing the problem size and increasing the number of

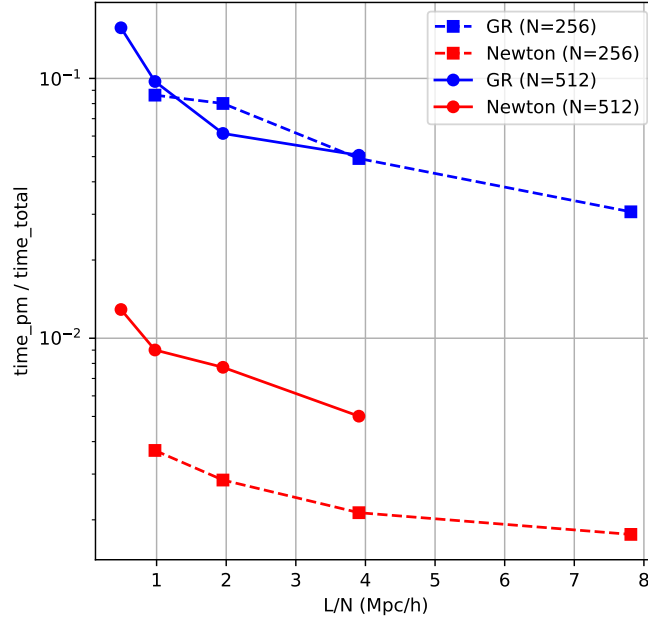


Figure 5.11: Fraction of total runtime spent in PM computations in GRGADGET (with the blue lines) and GADGET-4 (with the red lines) for two combinations of the mesh size $N = 256$ (dashed lines) and 512 (solid lines). The boxlength varies between $L = 250, 500, 1000$ and $2000 \text{ Mpc}/h$. The number of MPI processes for $N = 256$ and 512 are $N_{\text{MPI}} = 48$ and 192 respectively. The number of particles is determined as $N_p = N^3$.

MPI processes P , then the *speedup* s is computed as a function of P

$$s_P \equiv T_{P_0, M} / T_{P, M}, \quad (5.2)$$

where P_0 is a predefined number of processors chosen as a pivot, usually one has $P_0 = 1$. In the ideal case, if eq. 5.1 holds then

$$s_P = P / P_0. \quad (5.3)$$

Affecting the speedup there are at least three competing factors: (i) the reduced time of computation because the work load has been distributed among processes, (ii) if the work load is not distributed evenly the speedup deviates from the ideal trend, and (iii) there is also the overhead of inter-processes communication.

In figure 5.12 we show the result of strong scaling tests performed to the GRGADGET, for different simulation configurations: $N = 128, 256$ and 512, while $L = 250, 500, 1000$ and $2000 \text{ Mpc}/h$, the number of particles is always $N_p = N^3$. In this test, the problem size is fixed and the number of processors is changed from $P = 24, 48, 96$ until 192; our pivot is $P_0 = 24$ for the smaller problems with $N = 128$ and 256, while for $N = 512$ the pivot is $P_0 = 96$ because that simulation size, due to its memory requirements, does not fit into 1 or 2 computing nodes, we needed at least 3 nodes of HOTCAT cluster. The top plots correspond to the strong scalability of the GRGADGET total runtime and its GR

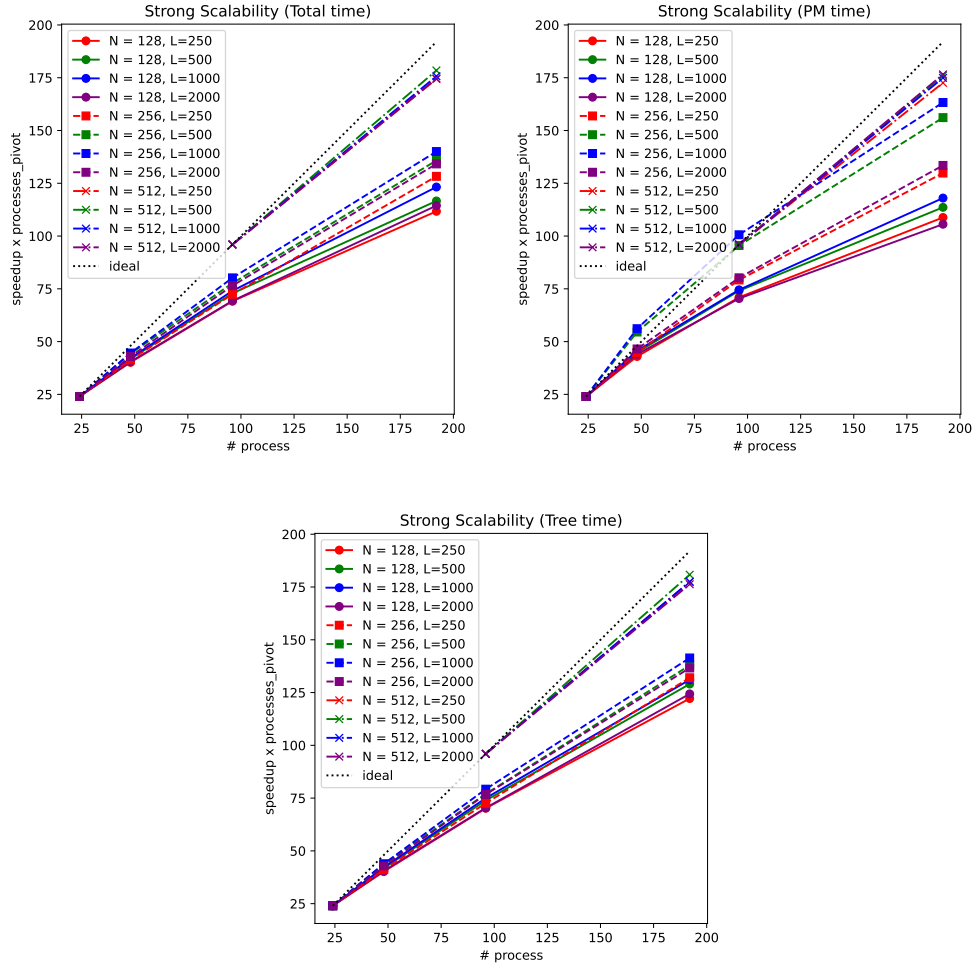


Figure 5.12: Strong scalability test for GRGADGET's total time (top left), PM (top right) and BH-TREE (bottom plot). The problem size is fixed for several combinations of $N = 128$ and 256 , with $L = 250, 500, 1000$ and 2000 Mpc/h and the number of processors is increased from $P = 24, 48, 96$ until 192 . For $N = 512$ the number of processors changes starting from $P = 96$ until 192 .

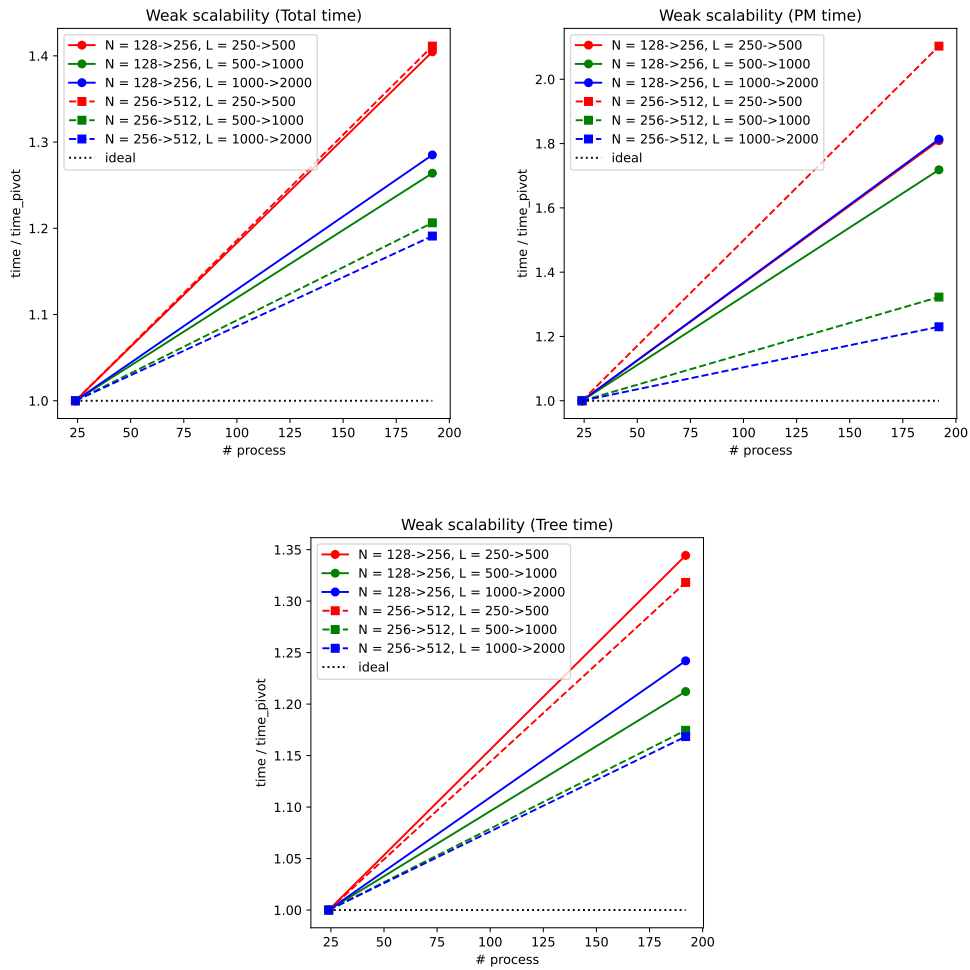


Figure 5.13: Weak scalability test for GRGADGET's total time (top left), PM (top right) and BH-TREE (bottom plot). The problem size (particles, mesh points and volume) and processors is multiplied by a factor of $\times 8$, seeking to keep the work load per processor constant. The number of particles is always $N_p = N^3$.

PM, from left to right respectively. In the bottom plot we show the strong scalability of the BH-TREE. In general we see that in all instances there is some deviation from the ideal trend, which is expected, but overall the code scales; i.e. it gains speedup with the increase of processors.

Weak scalability

Weak scalability is another way to measure the parallelization of the code, or how well eq. 5.1 holds. It is obtained by increasing the linear problem size M and the number of processors P proportionally. Ideally the work load M/P per process remains the same and one expects that the execution time $T_{P,M}$ remains constant.

In figure 5.13 we show the results of the weak scalability test on GRGADGET, for different problem configurations. Just like in the case of the strong scalability, we can notice that the BH-TREE scales better than the PM part of the code, but since the first takes much of the runtime it follows that the entire code scales reasonably well.

Conclusions

Cosmological N -body simulations where GR is implemented from first principles are needed as we enter the era of precision cosmology. In order to profit from the data that the upcoming galaxy surveys will provide with Euclid¹, Rubin-LSST², DESI³, Roman⁴, SKAO⁵ and Sphex⁶ instruments, we need to consider the relativistic effects arising from the geodesic trajectory of the light coming from the distant galaxies until it reaches our telescopes, like weak lensing effects that change the position of the sources in the line of sight, change its shape and magnification, redshift space distortions due to Doppler, Sachs-Wolfe and integrated Sachs-Wolfe effects (Yoo, Fitzpatrick, and Zaldarriaga 2009). These small effects could be the source of systematic errors if not taken into account, and if measured properly they could provide a testing tool for GR in the large scales. Native GR simulation codes are able to address those effects on the lightcone, naturally including correction terms from the contributions of all metric degrees of freedom like the vector potential that produces frame dragging. They can also be used to investigate not well known phenomena like the effect of radiation in matter clustering, and the role of the higher-order relativistic potentials like frame dragging in gravitational weak lensing and the formation of structures.

In this context we have constructed a relativistic N -body TREEPM code called GR-GADGET (Quintana-Miranda, Monaco, and Tornatore 2023), based on GADGET-4 in which short-range gravitational interactions are computed with the help of the BH-TREE implemented in GADGET-4, while the long-range contributions to the force are computed using a relativistic PM library LIBEVOLUTION, that we have developed based on the existing GEVOLUTION code. The code works on the assumption that for the purpose of cosmology, dark matter can be treated non-relativistically and then the equations of motion of tracer particles tend to the Newtonian limit at scales well below the Hubble horizon. In our setup we use a weak field approximation of GR, following the numerical approach already well established by GEVOLUTION, where the perturbations of the space-time metric with respect to FLRW background are encoded as fields and simulated by the PM. The simulation particles move according to their geodesic trajectories established by the metric of space-time.

¹sci.esa.int/web/euclid

²www.lsst.org

³www.desi.lbl.gov

⁴roman.gsfc.nasa.gov

⁵www.skao.int

⁶spherex.caltech.edu

The TREEPM in GRGADGET is able to resolve the clustering represented by the simulation particles down to scales orders of magnitude smaller than the mesh size. This means that the energy-momentum tensor, that sources the equations of the fields that represent the perturbations of the metric, is computed from a fully non-linear distribution of particles and as a consequence the predictive power of GRGADGET gives an improvement for mesh quantities with respect to the pure PM implementation of GEVOLUTION. We have checked this claim by testing the power spectra of the potential fields in the weak gravitational approximation, that show an increase in amplitude for short-wave modes in GRGADGET with respect to pure PM simulations; see figures 5.6 and 5.7. This may be very useful, e.g. when assessing the possibility of detecting the frame-dragging effect of collective rotatory motion of a matter distribution—characterized by the vorticity of the velocity field or more precisely by the amplitude of the divergenceless component of the ΔT^0_i part of the energy-momentum tensor containing the information of the local total momentum carried by matter, see eq. (2.73).

Through the testing phase of GRGADGET we have validated that the code performs simulations that on the Mpc scales and below, produce the same non-linear clustering that we obtain from a Newtonian code like GADGET-4, while in the Gpc scales as we approach the Hubble horizon the relativistic features in the matter power spectrum start to become evident and are in agreement with the results obtained from GEVOLUTION simulations.

By studying the convergence of the two-point correlation function of matter in GRGADGET for varying mesh resolutions, we find that the relativistic PM becomes numerically unreliable when it comes to resolving forces for modes one order of magnitude below the Nyquist frequency for a mesh size of $L/N \approx 4 \text{ Mpc}/h$ or above. In finer meshes, $L/N < 2 \text{ Mpc}/h$, we obtain a smoother transition from the PM and Tree force regimes. Another factor that plays in, is the gr-smoothing parameter r_b that we introduced to tune the scale in which we wish to transition from GR to predominantly Newtonian forces. We have tested that it can be reduced down to $r_b \approx 3 L/N$, enlarging the GR regime without compromising the integrity of the particle’s geodesics with the numerical noise coming from the relativistic PM at the distances where the discretization of space approximation starts to fall apart.

Outlook

GRGADGET’s and its supporting library LIBGEVOLUTION’s source codes are available in the public domain⁷. However, we state here as a disclaimer, that we are still working to implement all the API guarantees described in section 4.3. We aim to implement also a set of routines to obtain particle and fields information on the lightcone from which we can actually reproduce the observable quantities of astronomy.

We are planning an upcoming production simulation of GRGADGET, for which we have been granted 30’000 CPU-hours with ISCRA-C on the supercomputer Galileo100 at CINECA. With these resources we plan to run a simulation with a grid size $N = 2048$ in a comoving box of $L = 2 \text{ Gpc}/h$, with $N_p = 2048^3$ particles. For a simulation of this size we have a mass resolution of $m_0 \approx 8 \times 10^{10} M_\odot/h$, and we can obtain lightcone observables up to $z \approx 1.8$. The optimal capabilities of GRGADGET to simulate the formation of non-linear structures can be exploited, so that we can identify DM halos from the particle snapshots to study their mass functions as a function of redshift and study the effects that higher-order relativistic potentials can have on them. The same DM halos can be used either as biased tracers of galaxies or we can populate them with simulated galaxies using

⁷<https://github.com/rgadget>

Halo Occupation Distribution model (Berlind et al. 2003), to obtain a mock catalog from which we can extract the two-point correlation function and cross-correlation with the weak lensing convergence; both of which are observables in galaxy surveys. The effects of GR will be naturally incorporated into this analysis as we will perform ray tracing back in time from the observer until the source correcting for the photon’s deviation angle and for the variation in redshift due to the perturbations of the metric along its path. It will be interesting to compare the result from a simulation like this in GRGADGET with similar studies done with GEVOLUTION; for example, the two point statistics of halo number counts from relativistic simulations (Lepori et al. 2022).

With our code we will be able to study the vector modes of the metric perturbations. It has been shown that the acceleration produced by vector modes is at its peak 10^{-3} with respect to the standard gravitational acceleration (Barrera-Hinojosa et al. 2020) and the effects on weak lensing have been so far considered negligible (Thomas, Bruni, and Wands 2015). However, recent studies have concluded that their signal present in Redshift-Space Distortions should not be negligible in the upcoming surveys, so that not accounting for them would lead to systematic errors in theoretical predictions (Bonvin et al. 2018). The theoretical models for vector modes are challenged by the fact that vorticity becomes stronger in the non-linear scales where the dark matter fluid departs significantly from the perfect fluid approximation because of orbit crossing. The extraction of the signal of the vector modes would require the use of N -body simulations with a high resolution on those non-linear scales.

Furthermore, motivated by the fact that GRGADGET is a development branch of the widely used GADGET-4 code, and because the PM sector is called only by the computation of the gravity force, our code can be easily extended to simulations of galaxies or galaxy clusters by switching on the hydrodynamics, star formation and feedback sectors.

Another possible application of GRGADGET can be found in the sector of modified gravity, for theories whose formulation are worked out in a general relativistic context. The alternative equations can be naturally integrated in the PM code and short-range/small-velocity limits can be theoretically derived and introduced in the BH-TREE for a fully consistent treatment. This line of development would be precious in the formulation of tests of gravity, because relativistic effects may hide smoking-gun features of specific classes of modified gravity theories. An interesting possibility is to find the specific signal enhancement of vector modes in alternative models of gravity such as Hu-Sawicki’s $f(R)$; for example it is known that the amplitude of the vector potential could be 50% larger in $f(R)$ simulations with respect to Λ CDM (Thomas et al. 2015).

Appendix A

Perturbations of the Ricci tensor

In this appendix we show the symbolic computation of the connection and Ricci tensor for a perturbed FLRW metric. The results obtained here were used in Chapter 2 to obtain the field equations (2.69), (2.71), (2.74) and (2.75).

Connection

We compute the Christoffel symbols (affine connection components) for the physical metric g up to order $\epsilon^{3/2}$ in the perturbations. We write the Christoffel symbols as

$$\begin{aligned}\Gamma^\mu{}_{\nu\alpha} &= \frac{g^{\mu\sigma}}{2}(-g_{\nu\alpha,\sigma} + g_{\nu\sigma,\alpha} + g_{\alpha\sigma,\nu}) \\ &= \frac{\bar{g}^{\mu\sigma} + \Delta g^{\mu\sigma}}{2}(-\bar{g}_{\nu\alpha,\sigma} + \bar{g}_{\nu\sigma,\alpha} + \bar{g}_{\alpha\sigma,\nu} - \Delta g_{\nu\alpha,\sigma} + \Delta g_{\nu\sigma,\alpha} + \Delta g_{\alpha\sigma,\nu}) \\ &= \bar{\Gamma}^\mu{}_{\nu\alpha} + \Delta g^{\mu\sigma} \bar{g}_{\sigma\lambda} \bar{\Gamma}^\lambda{}_{\nu\alpha} + \frac{\bar{g}^{\mu\sigma} + \Delta g^{\mu\sigma}}{2}(-\Delta g_{\nu\alpha,\sigma} + \Delta g_{\nu\sigma,\alpha} + \Delta g_{\alpha\sigma,\nu})\end{aligned}$$

hence

$$\Delta\Gamma^\mu{}_{\nu\alpha} = \Delta g^{\mu\sigma} \bar{g}_{\sigma\lambda} \bar{\Gamma}^\lambda{}_{\nu\alpha} + \frac{\bar{g}^{\mu\sigma} + \Delta g^{\mu\sigma}}{2}(-\Delta g_{\nu\alpha,\sigma} + \Delta g_{\nu\sigma,\alpha} + \Delta g_{\alpha\sigma,\nu}) \quad (\text{A.1})$$

Then the perturbations of the Christoffel symbols are

$$\begin{aligned}\Delta\Gamma^0{}_{00} &= \Psi_{,0} - cB^s\Psi_{,s} + \mathcal{O}(\epsilon^2) \\ \Delta\Gamma^i{}_{00} &= c^2\Psi_{,i} - cB^i_{,0} - cB^i\mathcal{H} + c^2\Psi_{,l}(2\Phi\gamma^{li} - h^{li}) + \mathcal{O}(\epsilon^2) \\ \Delta\Gamma^0{}_{0i} &= \Psi_{,i} - \frac{B_i}{c}\mathcal{H} - 2\Psi\Psi_{,i} + \mathcal{O}(\epsilon^2) \\ \Delta\Gamma^i{}_{0j} &= -\Phi_{,0}\delta^i{}_j + \frac{c\gamma^{il}}{2}(B_{j|l} - B_{l|j}) + \frac{h^i{}_{j,0}}{2} + cB^i\Psi_{,j} + \mathcal{O}(\epsilon^2) \\ \Delta\Gamma^0{}_{ij} &= -2\Psi\mathcal{H}\frac{\gamma_{ij}}{c^2} + \frac{1}{2c}(B_{i|j} + B_{j|i}) + \frac{1}{2c^2}(h_{ij} - 2\Phi\gamma_{ij})_{,0} + \frac{\mathcal{H}}{c^2}(h_{ij} - 2\Phi\gamma_{ij}) \\ &\quad - \frac{B^l}{c}\Phi_{,l}\gamma_{ij} + \frac{1}{c}(B_i\Phi_{,j} + B_j\Phi_{,i}) + \mathcal{O}(\epsilon^2) \\ \Delta\Gamma^k{}_{ij} &= \Phi_{,i}\gamma_{kj} - \Phi_{,j}\gamma_{ki} - \Phi_{,i}\delta^k{}_j + \mathcal{H}\frac{B^k}{c}\gamma_{ij} + \frac{\gamma^{sk}}{2}(-h_{ij|s} + h_{is|j} + h_{js|i}) \\ &\quad + (h^{ks} - 2\Phi\gamma^{ks})(-\Phi_{,s}\gamma_{ij} + \Phi_{,i}\gamma_{sj} + \Phi_{,j}\gamma_{is}) + \mathcal{O}(\epsilon^2)\end{aligned} \quad (\text{A.2})$$

Ricci tensor

This section is devoted to the computation of the components of the Ricci tensor starting from the Christoffel symbols computed previously. The computations are quite long hence we are going to follow this strategy: as a general rule do not expand unless no further simplifications can be performed, write down partial results, and thoroughly use the spatial covariant derivative instead of partial derivatives. By writing $\Gamma^\mu{}_{\nu\alpha} = \bar{\Gamma}^\mu{}_{\nu\alpha} + \Delta\Gamma^\mu{}_{\nu\alpha}$ we can expand (2.21).

$$R_{\alpha\beta} = \bar{R}_{\alpha\beta} + \Delta\Gamma^\nu{}_{\alpha\beta;\nu} - \Delta\Gamma^\nu{}_{\alpha\nu;\beta} + \Delta\Gamma^\nu{}_{\lambda\nu}\Delta\Gamma^\lambda{}_{\alpha\beta} - \Delta\Gamma^\nu{}_{\lambda\beta}\Delta\Gamma^\lambda{}_{\alpha\nu} \quad (\text{A.3})$$

where $\Delta\Gamma^\mu{}_{\alpha\beta;\nu}$ denotes covariant derivative of $\Delta\Gamma^\mu{}_{\alpha\beta}$ with the background connection $\bar{\Gamma}$. Notice that $\Delta\Gamma$, being the difference of two connections it is in fact a tensor. The first index of $R_{\alpha\beta}$ is raised using the physical metric $g^{\mu\nu} = \bar{g}^{\mu\nu} + \Delta g^{\mu\nu}$:

$$R^\alpha{}_\beta = \bar{R}^\alpha{}_\beta + \bar{g}^{\alpha\sigma}\Delta R_{\sigma\beta} + \Delta g^{\alpha\sigma}\bar{R}_{\sigma\beta} + \Delta g^{\alpha\sigma}\Delta R_{\sigma\beta} \quad (\text{A.4})$$

Component R_{00}

Using equation (A.3)

$$\begin{aligned} \Delta R_{00} &= \Delta\Gamma^\nu{}_{00;\nu} - \Delta\Gamma^\nu{}_{0\nu;0} + \Delta\Gamma^\nu{}_{\lambda\nu}\Delta\Gamma^\lambda{}_{00} - \Delta\Gamma^\nu{}_{\lambda 0}\Delta\Gamma^\lambda{}_{0\nu} \\ &= \Delta\Gamma^n{}_{00|n} - \Delta\Gamma^n{}_{0n,0} + 3\mathcal{H}\Delta\Gamma^0{}_{00} - \mathcal{H}\Delta\Gamma^n{}_{0n} \\ &\quad + \Delta\Gamma^n{}_{0n}\Delta\Gamma^0{}_{00} + \Delta\Gamma^n{}_{ln}\Delta\Gamma^l{}_{00} - \Delta\Gamma^n{}_{00}\Delta\Gamma^0{}_{0n} - \Delta\Gamma^n{}_{0l}\Delta\Gamma^l{}_{0n} \\ &= \Delta\Gamma^n{}_{00|n} - \Delta\Gamma^n{}_{0n,0} + 3\mathcal{H}\Delta\Gamma^0{}_{00} - \mathcal{H}\Delta\Gamma^n{}_{0n} + \Delta\Gamma^n{}_{ln}\Delta\Gamma^l{}_{00} - \Delta\Gamma^n{}_{00}\Delta\Gamma^0{}_{0n} + \mathcal{O}(\epsilon^2) \end{aligned} \quad (\text{A.5})$$

$$\Delta R_{00} = c^2\Psi_{|n}{}^n + 3\mathcal{H}\Psi_{,0} + 3\Phi_{,00} + 3\mathcal{H}\Phi_{,0} - c^2\Psi_{,l}(\Phi^{,l} + \Psi^{,l}) + 2c^2\Phi\Psi_{|l}{}^l - c^2\Psi_{|nl}h^{nl} + \mathcal{O}(\epsilon^2) \quad (\text{A.6})$$

Terms like $B^n{}_{|n}$ and $h^n{}_n$ have been eliminated due to the constraints on B and h , see eq. (2.56).

Component R_{0i}

Using equation (A.3)

$$\begin{aligned} \Delta R_{0i} &= \Delta\Gamma^\nu{}_{0i;\nu} - \Delta\Gamma^\nu{}_{0\nu;i} + \Delta\Gamma^\nu{}_{\lambda\nu}\Delta\Gamma^\lambda{}_{0i} - \Delta\Gamma^\nu{}_{\lambda i}\Delta\Gamma^\lambda{}_{0\nu} \\ &= \Delta\Gamma^0{}_{0i,0} + \Delta\Gamma^n{}_{0i|n} - \Delta\Gamma^0{}_{00|i} - \Delta\Gamma^n{}_{0n|i} + 3\mathcal{H}\Delta\Gamma^0{}_{i0} - \mathcal{H}\frac{\gamma_{in}}{c^2}\Delta\Gamma^n{}_{00} \\ &\quad + \Delta\Gamma^n{}_{0n}\Delta\Gamma^0{}_{0i} + \Delta\Gamma^n{}_{ln}\Delta\Gamma^l{}_{0i} - \Delta\Gamma^0{}_{li}\Delta\Gamma^l{}_{00} - \Delta\Gamma^n{}_{li}\Delta\Gamma^l{}_{0n} \\ &= \Delta\Gamma^0{}_{0i,0} + \Delta\Gamma^n{}_{0i|n} - \Delta\Gamma^0{}_{00|i} - \Delta\Gamma^n{}_{0n|i} + 3\mathcal{H}\Delta\Gamma^0{}_{i0} - \mathcal{H}\frac{\gamma_{in}}{c^2}\Delta\Gamma^n{}_{00} + \mathcal{O}(\epsilon^2) \end{aligned} \quad (\text{A.7})$$

$$\Delta R_{0i} = 2\Phi_{,i0} + 2\mathcal{H}\Psi_{,i} - \frac{2}{c}\mathcal{H}^2 B_i - \frac{1}{c}\mathcal{H}_{,0}B_i + \frac{c}{2}(B_{i|n}{}^n - B_{n|i}{}^n) + cB^n\Psi_{|ni} + \mathcal{O}(\epsilon^2) \quad (\text{A.8})$$

Component R_{i0}

Using equation (A.3)

$$\begin{aligned}
\Delta R_{i0} &= \Delta\Gamma^\nu{}_{i0;\nu} - \Delta\Gamma^\nu{}_{i\nu;0} + \Delta\Gamma^\nu{}_{\lambda\nu}\Delta\Gamma^\lambda{}_{i0} - \Delta\Gamma^\nu{}_{\lambda 0}\Delta\Gamma^\lambda{}_{i\nu} \\
&= \Delta\Gamma^n{}_{i0|n} - \Delta\Gamma^n{}_{in,0} + 3\mathcal{H}\Delta\Gamma^0{}_{i0} - \mathcal{H}\frac{\gamma^{ni}}{c^2}\Delta\Gamma^n{}_{00} \\
&\quad + \Delta\Gamma^n{}_{0n}\Delta\Gamma^0{}_{i0} + \Delta\Gamma^n{}_{ln}\Delta\Gamma^l{}_{i0} - \Delta\Gamma^n{}_{00}\Delta\Gamma^0{}_{in} - \Delta\Gamma^n{}_{l0}\Delta\Gamma^l{}_{in} \\
&= \Delta\Gamma^n{}_{i0|n} - \Delta\Gamma^n{}_{in,0} + 3\mathcal{H}\Delta\Gamma^0{}_{i0} - \mathcal{H}\frac{\gamma^{ni}}{c^2}\Delta\Gamma^n{}_{00} + \mathcal{O}(\epsilon^2)
\end{aligned} \tag{A.9}$$

$$\Delta R_{i0} = \Delta R_{0i}$$

Component R_{ij}

Using equation (A.3)

$$\begin{aligned}
\Delta R_{ij} &= \Delta\Gamma^\nu{}_{ij;\nu} - \Delta\Gamma^\nu{}_{i\nu;j} + \Delta\Gamma^\nu{}_{\lambda\nu}\Delta\Gamma^\lambda{}_{ij} - \Delta\Gamma^\nu{}_{\lambda j}\Delta\Gamma^\lambda{}_{i\nu} \\
&= \Delta\Gamma^0{}_{ij,0} + \Delta\Gamma^n{}_{ij|n} - \Delta\Gamma^0{}_{i0|j} - \Delta\Gamma^n{}_{in|j} \\
&\quad + 2\mathcal{H}\Delta\Gamma^0{}_{ij} + \mathcal{H}\frac{\gamma^{ij}}{c^2}(\Delta\Gamma^0{}_{00} + \Delta\Gamma^n{}_{0n}) - \mathcal{H}(\frac{\gamma^{ni}}{c^2}\Delta\Gamma^n{}_{0j} + \frac{\gamma^{nj}}{c^2}\Delta\Gamma^n{}_{0i}) \\
&\quad + \Delta\Gamma^0{}_{00}\Delta\Gamma^0{}_{ij} + \Delta\Gamma^n{}_{0n}\Delta\Gamma^0{}_{ij} + \Delta\Gamma^0{}_{l0}\Delta\Gamma^l{}_{ij} + \Delta\Gamma^n{}_{ln}\Delta\Gamma^l{}_{ij} \\
&\quad - \Delta\Gamma^0{}_{0j}\Delta\Gamma^0{}_{i0} - \Delta\Gamma^n{}_{0j}\Delta\Gamma^0{}_{in} - \Delta\Gamma^0{}_{lj}\Delta\Gamma^l{}_{i0} - \Delta\Gamma^n{}_{lj}\Delta\Gamma^l{}_{in} \\
&= \Delta\Gamma^0{}_{ij,0} + \Delta\Gamma^n{}_{ij|n} - \Delta\Gamma^0{}_{i0|j} - \Delta\Gamma^n{}_{in|j} \\
&\quad + 2\mathcal{H}\Delta\Gamma^0{}_{ij} + \mathcal{H}\frac{\gamma^{ij}}{c^2}(\Delta\Gamma^0{}_{00} + \Delta\Gamma^n{}_{0n}) - \mathcal{H}(\frac{\gamma^{ni}}{c^2}\Delta\Gamma^n{}_{0j} + \frac{\gamma^{nj}}{c^2}\Delta\Gamma^n{}_{0i}) \\
&\quad + \Delta\Gamma^0{}_{l0}\Delta\Gamma^l{}_{ij} + \Delta\Gamma^n{}_{ln}\Delta\Gamma^l{}_{ij} - \Delta\Gamma^0{}_{0j}\Delta\Gamma^0{}_{i0} - \Delta\Gamma^n{}_{lj}\Delta\Gamma^l{}_{in} + \mathcal{O}(\epsilon^2)
\end{aligned} \tag{A.10}$$

$$\begin{aligned}
\Delta R_{ij} &= -\Psi_{|ij} + 2\Psi\Psi_{|ij} + \Psi_{|i}\Psi_{|j} - \frac{\mathcal{H}}{c^2}\Psi_{,0}\gamma_{ij} - 2\frac{\mathcal{H}_{,0}}{c^2}\Psi\gamma_{ij} - 4\frac{\mathcal{H}^2}{c^2}\Psi\gamma_{ij} \\
&\quad + \Phi_{|n}{}^n\gamma_{ij} + \Phi_{|ij} - \frac{\Phi_{,00}}{c^2}\gamma_{ij} - 2\frac{\mathcal{H}_{,0}}{c^2}\Phi\gamma_{ij} - 5\frac{\mathcal{H}}{c^2}\Phi_{,0}\gamma_{ij} - 4\frac{\mathcal{H}^2}{c^2}\Phi\gamma_{ij} \\
&\quad + 2\Phi\Phi_{|ij} + 2\Phi\Phi_{|n}{}^n\gamma_{ij} + \Phi_{|n}\Phi_{|n}{}^n\gamma_{ij} + 3\Phi_{|i}\Phi_{|j} + \Psi_{|n}\Phi_{|n}{}^n\gamma_{ij} - \Psi_{|i}\Phi_{|j} - \Psi_{|j}\Phi_{|i} \\
&\quad + \frac{\mathcal{H}}{c}(B_{i|j} + B_{j|i}) + \frac{1}{2c}(B_{i|j} + B_{j|i})_{,0} \\
&\quad + \frac{1}{2c^2}h_{ij,00} + \frac{\mathcal{H}_{,0}}{c^2}h_{ij} + \frac{\mathcal{H}}{c^2}h_{ij,0} + 2\frac{\mathcal{H}^2}{c^2}h_{ij} + \frac{1}{2}(h_{in|j}{}^n + h_{jn|i}{}^n - h_{ij|n}{}^n) \\
&\quad - h^{nm}\Phi_{|nm}\gamma_{ij} + h^n{}_j\Phi_{|in} + h^n{}_i\Phi_{|jn} + \mathcal{O}(\epsilon^2)
\end{aligned} \tag{A.11}$$

Component $R^0{}_0$

We use equation (A.4) to raise the first index of the Ricci tensor:

$$\Delta R^0{}_0 = \bar{g}^{0\sigma}\Delta R_{\sigma 0} + \Delta g^{0\sigma}\bar{R}_{\sigma 0} + \Delta g^{0\sigma}\Delta R_{\sigma 0}$$

But we don't substitute the results for the Ricci tensor with down indexes, it is less cumbersome first to arrive to an expression containing $\Delta\Gamma$ terms and then use (A.2)

$$(ac)^2\Delta R^0_0 = -6\Psi\mathcal{H}_{,0} + (2\Psi - 1)\Delta\Gamma^n_{00|n} + \Delta\Gamma^n_{0n,0} - 3\mathcal{H}\Delta\Gamma^0_{00} + \mathcal{H}\Delta\Gamma^n_{0n} \\ - \Delta\Gamma^n_{ln}\Delta\Gamma^l_{00} + \Delta\Gamma^n_{00}\Delta\Gamma^0_{0n} \quad (\text{A.12})$$

$$a^2\Delta R^0_0 = -\Psi_{|nm}h^{nm} - \Psi_{|n}{}^n(1 - 2\Psi + 2\Phi) - 3\frac{\mathcal{H}}{c^2}\Psi_{,0} - 6\frac{\mathcal{H}_{,0}}{c^2}\Psi + \Psi_{,n}(\Psi_{,n} + \Phi_{,n}) \\ - 3\frac{1}{c}\Phi_{,00} - 3\frac{\mathcal{H}}{c}\Phi_{,0} + \mathcal{O}(\epsilon^2) \quad (\text{A.13})$$

Component R^0_i

$$\Delta R^0_i = \bar{g}^{0\sigma}\Delta R_{\sigma i} + \Delta g^{0\sigma}\bar{R}_{\sigma i} + \Delta g^{0\sigma}\Delta R_{\sigma i}$$

$$(ac)^2\Delta R^0_i = -\Delta\Gamma^n_{i0|n} + \Delta\Gamma^n_{in,0} - 3\mathcal{H}\Delta\Gamma^0_{i0} + \mathcal{H}\frac{\gamma_{ni}}{c^2}\Delta\Gamma^n_{00} - cB^l\bar{R}_{li} \\ - cB^l\Delta\Gamma^n_{il|n} + cB^l\Delta\Gamma^0_{i0|l} + cB^l\Delta\Gamma^n_{in|l} + \mathcal{O}(\epsilon^2) \quad (\text{A.14})$$

$$a^2\Delta R^0_i = -2\frac{\Phi_{,i0}}{c^2} - 2\frac{\mathcal{H}}{c^2}\Psi_{,i} - \frac{B^l}{c}\bar{R}^{(3)}_{li} + \frac{1}{2c}(B_{n|i}{}^n - B_{i|n}{}^n) - \frac{B_i}{c}\Phi_{|n}{}^n - \frac{B^n}{c}\Phi_{|in} + \mathcal{O}(\epsilon^2) \quad (\text{A.15})$$

Component R^i_0

$$\Delta R^i_0 = \bar{g}^{i\sigma}\Delta R_{\sigma 0} + \Delta g^{i\sigma}\bar{R}_{\sigma 0} + \Delta g^{i\sigma}\Delta R_{\sigma 0}$$

$$a^2\Delta R^i_0 = \gamma^{is}(\Delta\Gamma^n_{s0|n} - \Delta\Gamma^n_{sn,0} + 3\mathcal{H}\Delta\Gamma^0_{s0}) - \frac{\mathcal{H}}{c^2}\Delta\Gamma^i_{00} + \frac{3}{c}B^i\mathcal{H}_{,0} - \frac{1}{c}B^i\Delta\Gamma^n_{00|n} + \mathcal{O}(\epsilon^2) \quad (\text{A.16})$$

$$a^2\Delta R^i_0 = 2(\Phi_{|i})_{,0} + 2\mathcal{H}\Psi_{|i} + \frac{c}{2}(B^i_{|n}{}^n - B_{n|i}{}^n) + 2\frac{\mathcal{H}_{,0}}{c}B^i - 2\frac{\mathcal{H}^2}{c}B^i + cB^n\Psi_{|i}{}^n - cB^i\Psi_{|n}{}^n + \mathcal{O}(\epsilon^2) \quad (\text{A.17})$$

Component R^i_j

$$\Delta R^i_j = \bar{g}^{i\sigma}\Delta R_{\sigma j} + \Delta g^{i\sigma}\bar{R}_{\sigma j} + \Delta g^{i\sigma}\Delta R_{\sigma j}$$

$$a^2\Delta R^i_j = (2\Phi\gamma^{il} - h^{il})\bar{R}_{lj} + \gamma^{is}\Delta\Gamma^0_{sj,0} + 2\mathcal{H}\gamma^{is}\Delta\Gamma^0_{sj} + \frac{\mathcal{H}}{c^2}\delta^i_j(\Delta\Gamma^0_{00} + \Delta\Gamma^n_{0n}) \\ - \frac{\mathcal{H}}{c^2}\Delta\Gamma^i_{0j} - \frac{\mathcal{H}}{c^2}\gamma^{il}\gamma_{nj}\Delta\Gamma^n_{0l} \\ + \gamma^{is}(\Delta\Gamma^0_{l0}\Delta\Gamma^l_{sj} + \Delta\Gamma^n_{ln}\Delta\Gamma^l_{sj} - \Delta\Gamma^0_{0j}\Delta\Gamma^0_{s0} - \Delta\Gamma^n_{lj}\Delta\Gamma^l_{sn}) \\ + (\gamma^{is}(1 + 2\Phi) - h^{is})(\Delta\Gamma^n_{sj|n} - \Delta\Gamma^n_{sn|j} - \Delta\Gamma^0_{s0|j}) + \mathcal{O}(\epsilon^2) \quad (\text{A.18})$$

$$\begin{aligned}
a^2 \Delta R^i_j &= -\Psi_{|j}^i - \Psi_{,0} \frac{\mathcal{H}}{c^2} \delta^i_j - \Psi \delta^i_j \left(2 \frac{\mathcal{H}_{,0}}{c^2} + 4 \frac{\mathcal{H}^2}{c^2} \right) \\
&\quad + \Phi_{|j}^i + \Phi_{|n}^n \delta^i_j - \frac{\Phi_{,00}}{c^2} \delta^i_j - 5 \Phi_{,0} \frac{\mathcal{H}}{c^2} \delta^i_j + 2 \Phi \bar{R}^{(3)i}_j \\
&\quad + \frac{1}{2c} (\partial_0 + 2\mathcal{H})(B^i_{|j} + B_{j|}^i) \\
&\quad + \frac{1}{2} (h^i_{n|j}{}^n + h^n_{j|}{}^i{}_n - h^i_{j|n}{}^n) \\
&\quad + \frac{1}{2c^2} h^i_{j,00} + \frac{\mathcal{H}}{c^2} h^i_{j,0} - h^i_n \bar{R}^{(3)n}_j \\
&\quad + 2\Psi \Psi_{|j}^i + \Psi_{|j}^i \Psi_{|j} - 2\Phi \Psi_{|j}^i + \Phi_{|n} \Psi_{|n}^n \delta^i_j - \Phi_{|j} \Psi_{|j}^i - \Psi_{|j} \Phi_{|j}^i \\
&\quad + 4\Phi \Phi_{|n}^n \delta^i_j + 4\Phi \Phi_{|j}^i + 3\Phi_{|j}^i \Phi_{|j} + \Phi_{|n}^n \Phi_{|n} \delta^i_j \\
&\quad - h^{nl} \Phi_{|nl} \delta^i_j - h^i_j \Phi_{|n}^n + h^n_j \Phi_{|n}^i + h^i_n \Psi_{|n}^n + \mathcal{O}(\epsilon^2)
\end{aligned} \tag{A.19}$$

Ricci scalar R

$$R = R^0_0 + R^i_i \tag{A.20}$$

hence

$$\begin{aligned}
a^2 \Delta R &= a^2 \Delta R^0_0 + a^2 \Delta R^i_i \\
&= -2\Psi_{|n}^n - 6 \frac{\mathcal{H}}{c^2} \Psi_{,0} - 12 \frac{\mathcal{H}_{,0}}{c^2} \Psi - 12 \frac{\mathcal{H}^2}{c^2} \Psi + 4\Phi_{|n}^n - 6 \frac{\Phi_{,00}}{c^2} - 18 \frac{\mathcal{H}}{c^2} \Phi_{,0} \\
&\quad + 4\Psi \Psi_{|n}^n + 2\Psi_{|n} \Psi_{|n}^n - 4\Phi \Psi_{|n}^n + 2\Psi_{|n} \Phi_{|n}^n + 16\Phi \Phi_{|n}^n + 6\Phi_{|n} \Phi_{|n}^n + 2\Phi \bar{R}^{(3)} \\
&\quad - h^l_n \bar{R}^{(3)n}_l - 2h^{nl} \Phi_{|nl} + \mathcal{O}(\epsilon^2)
\end{aligned} \tag{A.21}$$

Appendix B

Nyquist-Shannon Theorem

Here we present a proof to the Theorem 1.

Proof. Let $f \in \mathcal{H}^2([0, L])$ we can write:

$$f = \sum_{k \in \mathbb{Z}} \langle u_k, f \rangle u_k. \quad (\text{B.1})$$

Now let's define \tilde{f}_l , for $0 \leq l < N$, as the following sum:

$$\tilde{f}_l = \frac{1}{N} \sum_{n=0}^{N-1} \omega^{-ln} f(x_n), \quad (\text{B.2})$$

then using the expansion (B.1) we obtain:

$$\tilde{f}_l = \frac{1}{N} \sum_{k \in \mathbb{Z}} \sum_{n=0}^{n-1} \omega^{-ln} u_k(x_n) \langle u_k, f \rangle = \frac{1}{N} \sum_{k \in \mathbb{Z}} \sum_{n=0}^{n-1} (\omega^{k-l})^n \langle u_k, f \rangle, \quad (\text{B.3})$$

where we have used the fact that $u_k(x_n) = \omega^{kn}$. Since ω is an N -root of unity we have that:

$$\frac{1}{N} \sum_{n=0}^{N-1} (\omega^{k-l})^n = \begin{cases} 1, & k-l = zN, z \in \mathbb{Z} \\ 0, & \text{otherwise} \end{cases} \quad (\text{B.4})$$

and equation (B.3) becomes:

$$\tilde{f}_l = \sum_{z \in \mathbb{Z}} \langle u_{l+zN}, f \rangle. \quad (\text{B.5})$$

Each one of the terms $\langle u_{l+zN}, f \rangle$ is an *alias* of \tilde{f}_l , and the fact that there can be multiple non-zero aliases, leads to the *aliasing* phenomenon. If all non-zero modes of f lie in the interval $[-k_N, k_N]$ with $N - k_N > k_N$, ie. $N > 2k_N$, then for every $l \in [0, N)$ there is at most one non-zero alias.

$$\tilde{f}_l = \begin{cases} \langle u_l, f \rangle, & 0 \leq l \leq k_N \\ \langle u_{l-N}, f \rangle, & N - k_N \leq l < N \\ 0, & k_N < l < N - k_N \end{cases} \quad (\text{B.6})$$

Conversely for every $k \in [-k_N, k_N]$ we are able to find a finite sampling sum that computes it:

$$\langle u_k, f \rangle = \begin{cases} \tilde{f}_k, & 0 \leq k \leq k_N \\ \tilde{f}_{k+N}, & -k_N \leq k < 0 \end{cases} \quad (\text{B.7})$$

Due to the fact that $\omega^N = 1$, equation (3.24) and the statement of the Theorem follows from (B.7). \square

Exact interpolating kernel

In section 3.2 we have deduced an expression for $W_N^{(\text{low})}$ in real space, eq. (3.32), and we have also obtained the expression for Whittaker-Shannon interpolation formula that uses $W_N^{(\text{low})}$ as a perfectly interpolating filter following the conditions of the Nyquist-Shannon theorem.

Notice for N odd we can choose $W_N^{(\text{low})}$ with cut-off frequency that satisfies $2k_N + 1 = N$. Then $W_N^{(\text{low})}$ takes a form that we call *Sinc* filter because of its resemblance and similar role to the Sinc function:

$$W^{(\text{Sinc})}(x) = W_N^{(\text{low})} \Big|_{N \text{ odd}}(x) = \frac{1}{L} \frac{\sin(\frac{\pi N}{L}x)}{\sin(\frac{\pi}{L}x)}. \quad (\text{B.8})$$

Interesting is the fact that $W^{(\text{Sinc})}$ for any N will satisfy equation (3.40) for any $x = kL/N$, ie.

$$\phi_k = \frac{L}{N} \sum_{n=0}^{N-1} \phi_n W^{(\text{Sinc})} \left(\frac{(k-n)L}{N} \right), \quad (\text{B.9})$$

simply because

$$W^{(\text{Sinc})}(kL/N) = \frac{1}{L} \frac{\sin(\pi k)}{\sin(\pi k/N)} = \frac{1}{L} \times \begin{cases} N, & N \text{ divides } k \\ 0, & \text{otherwise} \end{cases} \quad (\text{B.10})$$

and that's independent of the values of $\phi(x)$ at the grid points. That tempt us to use $W^{(\text{Sinc})}$ instead of $W_N^{(\text{low})}$ as a reference *exact* interpolator.

Trouble comes when N is even. In that case $k_N = (N-2)/2$ according to Theorem 1. A signal with mode $k_N + 1$ cannot be completely reconstructed from the N samples, hence $W_N^{(\text{low})}$ has a frequency cut-off at k_N , having $2k_N + 1$ modes, one mode short from N . That has consequences, the first is that the identity (B.9) is not satisfied by $W_N^{(\text{low})}$ unless $\phi = W_N^{(\text{low})} * \phi$. On the other hand, $W^{(\text{Sinc})}$ does satisfy (B.9) but it can be shown that it does not interpolate a function ϕ with modes below k_N . The reason lies in the fact that the modes of $W^{(\text{Sinc})}$ below k_N are not uniform, see figure B.1.

In summary:

- $W^{(\text{Sinc})}$ is an exact interpolator of *grid points* but not necessarily so for functions with k_N modes,
- $W_N^{(\text{low})}$ is an exact interpolator of *functions with k_N modes* but it might fail to exactly interpolate the grid points unless they correspond to values of function with k_N modes,
- when N is odd, hardly the most common case because powers of 2 are usually chosen, both functions $W^{(\text{Sinc})}$ and $W_N^{(\text{low})}$ are identical.

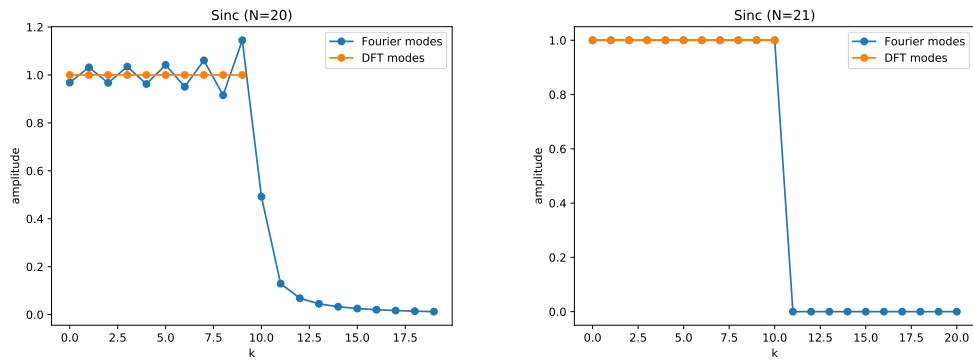


Figure B.1: $W^{(\text{Sinc})}$ power spectrum, discrete FT and true FT. For N odd, we see the same features of $W_N^{(\text{low})}$ that makes $W^{(\text{Sinc})}$ an exact interpolator, while for N even, the modes below k_N are not flat and that makes $W^{(\text{Sinc})}$ fail to be an exact interpolator.

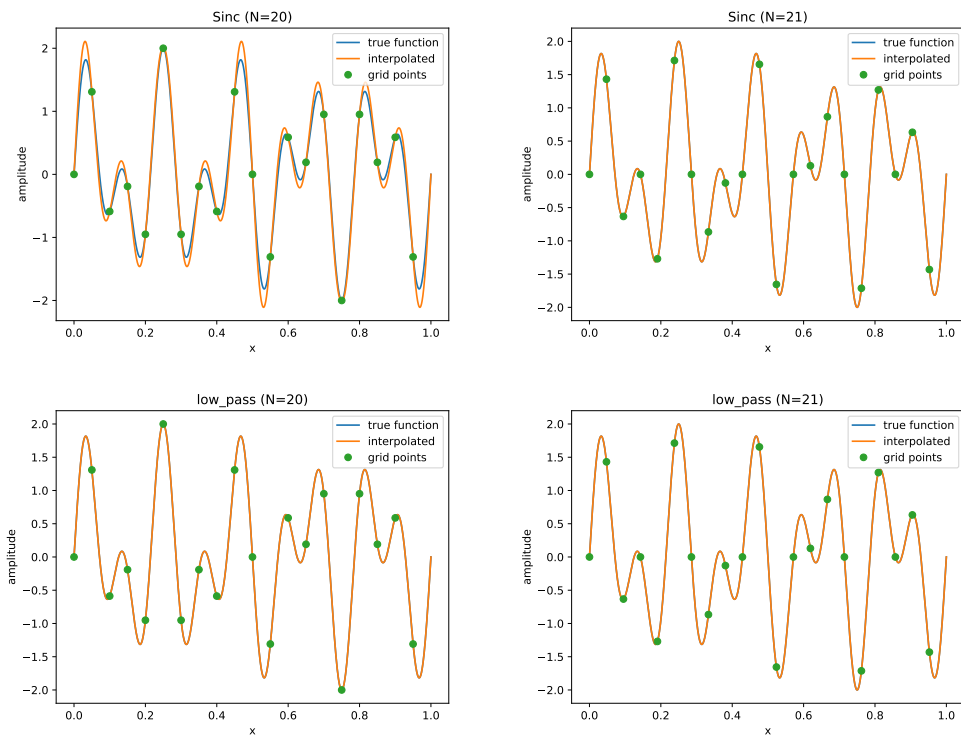


Figure B.2: Trying to interpolate a signal with $W_N^{(\text{low})}$ and $W^{(\text{Sinc})}$. When N is odd, $W_N^{(\text{low})}$ and $W^{(\text{Sinc})}$ are they interpolate the exactly. When N is even, $W^{(\text{Sinc})}$ deviates a little bit from $W_N^{(\text{low})}$ and fails to interpolate exactly.

Bibliography

- Abbott, B. P. et al. (Feb. 2016). “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *PRL* 116.6, 061102, p. 061102. DOI: [10.1103/PhysRevLett.116.061102](https://doi.org/10.1103/PhysRevLett.116.061102). arXiv: [1602.03837](https://arxiv.org/abs/1602.03837) [gr-qc].
- Adamek, Julian, Ruth Durrer, and Martin Kunz (Nov. 2017). “Relativistic N-body simulations with massive neutrinos”. In: *JCAP* 2017.11, 004, p. 004. DOI: [10.1088/1475-7516/2017/11/00410.48550/arXiv.1707.06938](https://doi.org/10.1088/1475-7516/2017/11/00410.48550/arXiv.1707.06938). arXiv: [1707.06938](https://arxiv.org/abs/1707.06938) [astro-ph.CO].
- Adamek, Julian and Christian Fidler (Sept. 2019). “The large-scale general-relativistic correction for Newtonian mocks”. In: *JCAP* 2019.9, 026, p. 026. DOI: [10.1088/1475-7516/2019/09/02610.48550/arXiv.1905.11721](https://doi.org/10.1088/1475-7516/2019/09/02610.48550/arXiv.1905.11721). arXiv: [1905.11721](https://arxiv.org/abs/1905.11721) [astro-ph.CO].
- Adamek, Julian et al. (July 2016). “gevolution: a cosmological N-body code based on General Relativity”. In: *JCAP* 2016.7, 053, p. 053. DOI: [10.1088/1475-7516/2016/07/05310.48550/arXiv.1604.06065](https://doi.org/10.1088/1475-7516/2016/07/05310.48550/arXiv.1604.06065). arXiv: [1604.06065](https://arxiv.org/abs/1604.06065) [astro-ph.CO].
- Adamek, Julian et al. (Sept. 2017). “The effect of early radiation in N-body simulations of cosmic structure formation”. In: *MNRAS* 470.1, pp. 303–313. DOI: [10.1093/mnras/stx115710.48550/arXiv.1703.08585](https://doi.org/10.1093/mnras/stx115710.48550/arXiv.1703.08585). arXiv: [1703.08585](https://arxiv.org/abs/1703.08585) [astro-ph.CO].
- Adamek, Julian et al. (Jan. 2019). “Safely smoothing spacetime: backreaction in relativistic cosmological simulations”. In: *Classical and Quantum Gravity* 36.1, 014001, p. 014001. DOI: [10.1088/1361-6382/aaeca5](https://doi.org/10.1088/1361-6382/aaeca5). arXiv: [1706.09309](https://arxiv.org/abs/1706.09309) [astro-ph.CO].
- Alam, Shadab et al. (2020). “Testing the theory of gravity with DESI: estimators, predictions and simulation requirements”. In: *preprint*. eprint: [Arxiv:2011.05771v1](https://arxiv.org/abs/2011.05771v1). URL: <http://arxiv.org/abs/2011.05771>.
- Amendola, Luca et al. (Apr. 2018). “Cosmology and fundamental physics with the Euclid satellite”. In: *Living Reviews in Relativity* 21.1, 2, p. 2. DOI: [10.1007/s41114-017-0010-3](https://doi.org/10.1007/s41114-017-0010-3). arXiv: [1606.00180](https://arxiv.org/abs/1606.00180) [astro-ph.CO].
- Bardeen, James M. (Oct. 1980). “Gauge-invariant cosmological perturbations”. In: *Physical Review D* 22.8, pp. 1882–1905. DOI: [10.1103/PhysRevD.22.1882](https://doi.org/10.1103/PhysRevD.22.1882).
- Barnes, Josh and Piet Hut (Dec. 1986). “A hierarchical O(N log N) force-calculation algorithm”. In: *Nature* 324.6096, pp. 446–449. DOI: [10.1038/324446a0](https://doi.org/10.1038/324446a0).
- Barrera-Hinojosa, Cristian and Baojiu Li (2020). “GRAMSES: a new route to general relativistic N-body simulations in cosmology. Part I. Methodology and code description”. In: *Journal of Cosmology and Astroparticle Physics* 2020.01, pp. 007–007. DOI: [10.1088/1475-7516/2020/01/007](https://doi.org/10.1088/1475-7516/2020/01/007). URL: <https://doi.org/10.1088/1475-7516/2020/01/007>.
- Barrera-Hinojosa, Cristian, Baojiu Li, and Yan-Chuan Cai (Mar. 2022). “Looking for a twist: probing the cosmological gravitomagnetic effect via weak lensing-kSZ cross-

- correlations”. In: *MNRAS* 510.3, pp. 3589–3604. DOI: [10.1093/mnras/stab3657](https://doi.org/10.1093/mnras/stab3657). arXiv: [2109.02632](https://arxiv.org/abs/2109.02632) [[astro-ph.CO](#)].
- Barrera-Hinojosa, Cristian et al. (2020). *Vector modes in Λ CDM: the gravitomagnetic potential in dark matter haloes from relativistic N-body simulations*. arXiv: [2010.08257](https://arxiv.org/abs/2010.08257) [[astro-ph.CO](#)].
- Berlind, Andreas A. et al. (Aug. 2003). “The Halo Occupation Distribution and the Physics of Galaxy Formation”. In: *ApJ* 593.1, pp. 1–25. DOI: [10.1086/376517](https://doi.org/10.1086/376517). arXiv: [astro-ph/0212357](https://arxiv.org/abs/astro-ph/0212357) [[astro-ph](#)].
- Bertocco, S. et al. (Jan. 2020). “INAF Trieste Astronomical Observatory Information Technology Framework”. In: *Astronomical Data Analysis Software and Systems XXIX*. Ed. by R. Pizzo et al. Vol. 527. Astronomical Society of the Pacific Conference Series, p. 303. arXiv: [1912.05340](https://arxiv.org/abs/1912.05340) [[astro-ph.IM](#)].
- Bonvin, Camille et al. (Feb. 2018). “Redshift-space distortions from vector perturbations”. In: *JCAP* 2018.2, 028, p. 028. DOI: [10.1088/1475-7516/2018/02/028](https://doi.org/10.1088/1475-7516/2018/02/028). arXiv: [1712.00052](https://arxiv.org/abs/1712.00052) [[astro-ph.CO](#)].
- Borzyszkowski, Mikolaj, Daniele Bertacca, and Cristiano Porciani (Nov. 2017). “liger: mock relativistic light cones from Newtonian simulations”. In: *MNRAS* 471.4, pp. 3899–3914. DOI: [10.1093/mnras/stx1423](https://doi.org/10.1093/mnras/stx1423). arXiv: [1703.03407](https://arxiv.org/abs/1703.03407) [[astro-ph.CO](#)].
- Chisari, Nora Elisa and Matias Zaldarriaga (2011). “Connection between Newtonian simulations and general relativity”. In: *Physical Review D* 83.12. ISSN: 1550-2368. DOI: [10.1103/physrevd.83.123505](https://doi.org/10.1103/physrevd.83.123505). URL: <http://dx.doi.org/10.1103/PhysRevD.83.123505>.
- Clifton, Timothy et al. (Mar. 2012). “Modified gravity and cosmology”. In: *Physics Reports* 513.1, pp. 1–189. DOI: [10.1016/j.physrep.2012.01.001](https://doi.org/10.1016/j.physrep.2012.01.001). arXiv: [1106.2476](https://arxiv.org/abs/1106.2476) [[astro-ph.CO](#)].
- Daverio, David, Mark Hindmarsh, and Neil Bevis (2015). *Latfield2: A c++ library for classical lattice field theory*. arXiv: [1508.05610](https://arxiv.org/abs/1508.05610) [[physics.comp-ph](#)].
- Event Horizon Telescope Collaboration et al. (Apr. 2019). “First M87 Event Horizon Telescope Results. I. The Shadow of the Supermassive Black Hole”. In: *APJL* 875.1, L1, p. L1. DOI: [10.3847/2041-8213/ab0ec7](https://doi.org/10.3847/2041-8213/ab0ec7). arXiv: [1906.11238](https://arxiv.org/abs/1906.11238) [[astro-ph.GA](#)].
- Fixsen, D. J. (2009). “The Temperature Of The Cosmic Microwave Background”. In: *The Astrophysical Journal* 707.2, pp. 916–920. DOI: [10.1088/0004-637x/707/2/916](https://doi.org/10.1088/0004-637x/707/2/916). URL: <https://doi.org/10.1088%2F0004-637x%2F707%2F2%2F916>.
- Frigo, Matteo and Steven G. Johnson (2005). “The Design and Implementation of FFTW3”. In: *Proceedings of the IEEE* 93.2. Special issue on “Program Generation, Optimization, and Platform Adaptation”, pp. 216–231.
- Hassani, Farbod et al. (Dec. 2019). “k-evolution: a relativistic N-body code for clustering dark energy”. In: *JCAP* 2019.12, 011, p. 011. DOI: [10.1088/1475-7516/2019/12/01110.48550/arXiv.1910.01104](https://doi.org/10.1088/1475-7516/2019/12/01110.48550/arXiv.1910.01104). arXiv: [1910.01104](https://arxiv.org/abs/1910.01104) [[astro-ph.CO](#)].
- Haverkort, Herman (Sept. 2011). “An inventory of three-dimensional Hilbert space-filling curves”. In: *arXiv e-prints*, arXiv:1109.2323, arXiv:1109.2323. arXiv: [1109.2323](https://arxiv.org/abs/1109.2323) [[cs.CG](#)].
- Hawking, S.W and G.F.R. Ellis (1973). *The large scale structure of space-time*. Cambridge Monographs on Mathematical Physics. ISBN: 0-521-09906-4.
- Hockney, R. W. and J. W. Eastwood (1988). *Computer simulation using particles*. IOP Publishing Ltd. ISBN: 0-852740392-0.
- Jelic-Cizmek, Goran et al. (Sept. 2018). “The generation of vorticity in cosmological N-body simulations”. In: *JCAP* 2018.9, 006, p. 006. DOI: [10.1088/1475-7516/2018/09/00610.48550/arXiv.1806.05146](https://doi.org/10.1088/1475-7516/2018/09/00610.48550/arXiv.1806.05146). arXiv: [1806.05146](https://arxiv.org/abs/1806.05146) [[astro-ph.CO](#)].
- Josuttis, Nicolai (2012). *C++ Standard Library - A Tutorial and Reference*. en. 2nd ed. Addison-Wesley Longman. ISBN: 978-0-321-62321-8.

- (2017). *C++17 The Complete Guide*. en. ISBN: 978-0321563842.
- Krause, E. et al. (June 2017). “Dark Energy Survey Year 1 Results: Multi-Probe Methodology and Simulated Likelihood Analyses”. In: *arXiv e-prints*, arXiv:1706.09359, arXiv:1706.09359. arXiv: [1706.09359 \[astro-ph.CO\]](#).
- Laureijs, R. et al. (Oct. 2011). “Euclid Definition Study Report”. In: *arXiv e-prints*, arXiv:1110.3193, arXiv:1110.3193. arXiv: [1110.3193 \[astro-ph.CO\]](#).
- Lepori, Francesca et al. (Sept. 2022). “The halo bias for number counts on the light cone from relativistic N-body simulations”. In: *arXiv e-prints*, arXiv:2209.10533, arXiv:2209.10533. DOI: [10.48550/arXiv.2209.10533](#). arXiv: [2209.10533 \[astro-ph.CO\]](#).
- Message Passing Interface Forum (June 2021). *MPI: A Message-Passing Interface Standard Version 4.0*. URL: <https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf>.
- Mukhanov, V.F., H.A. Feldman, and R.H. Brandenberger (1992). “Theory of cosmological perturbations”. In: *Physics Reports* 215.5, pp. 203–333. ISSN: 0370-1573. DOI: [https://doi.org/10.1016/0370-1573\(92\)90044-Z](https://doi.org/10.1016/0370-1573(92)90044-Z). URL: <http://www.sciencedirect.com/science/article/pii/037015739290044Z>.
- Pakmor, Ruediger et al. (Oct. 2022). “The MillenniumTNG Project: The hydrodynamical full physics simulation and a first look at its galaxy clusters”. In: *arXiv e-prints*, arXiv:2210.10060, arXiv:2210.10060. arXiv: [2210.10060 \[astro-ph.CO\]](#).
- Peebles, P. J. E. (Feb. 1970). “Structure of the Coma Cluster of Galaxies”. In: *Astrophysics Journal* 75, p. 13. DOI: [10.1086/110933](#).
- Percival, Will J. et al. (Feb. 2010). “Baryon acoustic oscillations in the Sloan Digital Sky Survey Data Release 7 galaxy sample”. In: *MNRAS* 401.4, pp. 2148–2168. DOI: [10.1111/j.1365-2966.2009.15812.x](#). arXiv: [0907.1660 \[astro-ph.CO\]](#).
- Perlmutter, S. et al. (June 1999). “Measurements of Ω and Λ from 42 High-Redshift Supernovae”. In: *ApJ* 517.2, pp. 565–586. DOI: [10.1086/307221](#). arXiv: [astro-ph/9812133 \[astro-ph\]](#).
- Pippig, Michael (2013). “PFFT: An Extension of FFTW to Massively Parallel Architectures”. In: *SIAM Journal on Scientific Computing* 35.3, pp. C213–C236. DOI: [10.1137/120885887](#). eprint: <https://doi.org/10.1137/120885887>. URL: <https://doi.org/10.1137/120885887>.
- Planck Collaboration et al. (Sept. 2020). “Planck 2018 results. VI. Cosmological parameters”. In: *AAP* 641, A6, A6. DOI: [10.1051/0004-6361/201833910](#). arXiv: [1807.06209 \[astro-ph.CO\]](#).
- Quintana-Miranda, Eduardo, Pierluigi Monaco, and Luca Tornatore (2023). “GrGadget: an N-body TreePM relativistic code for cosmological simulations”. In: DOI: [10.48550/ARXIV.2301.11854](#). URL: <https://arxiv.org/abs/2301.11854>.
- Riess, Adam G. et al. (Sept. 1998). “Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant”. In: *AJ* 116.3, pp. 1009–1038. DOI: [10.1086/300499](#). arXiv: [astro-ph/9805201 \[astro-ph\]](#).
- Saha, Prasenjit and Scott Tremaine (Oct. 1992). “Symplectic Integrators for Solar System Dynamics”. In: *Astrophysics Journal* 104, p. 1633. DOI: [10.1086/116347](#).
- Sefusatti, E. et al. (Aug. 2016). “Accurate estimators of correlation functions in Fourier space”. In: *MNRAS* 460.4, pp. 3624–3636. DOI: [10.1093/mnras/stw1229](#). arXiv: [1512.07295 \[astro-ph.CO\]](#).
- Springel, Volker (Dec. 2005). “The cosmological simulation code gadget-2”. In: *Monthly Notices of the Royal Astronomical Society* 364.4, pp. 1105–1134. ISSN: 0035-8711. DOI: [10.1111/j.1365-2966.2005.09655.x](#). eprint: <https://academic.oup.com/mnras/article-pdf/364/4/1105/18657201/364-4-1105.pdf>. URL: <https://doi.org/10.1111/j.1365-2966.2005.09655.x>.

- Springel, Volker, Naoki Yoshida, and Simon D. M. White (Apr. 2001). “GADGET: a code for collisionless and gasdynamical cosmological simulations”. In: *Nature* 6.2, pp. 79–117. DOI: [10.1016/S1384-1076\(01\)00042-2](https://doi.org/10.1016/S1384-1076(01)00042-2). arXiv: [astro-ph/0003162](https://arxiv.org/abs/astro-ph/0003162) [[astro-ph](#)].
- Springel, Volker et al. (Sept. 2021). “Simulating cosmic structure formation with the GADGET-4 code”. In: *MNRAS* 506.2, pp. 2871–2949. DOI: [10.1093/mnras/stab1855](https://doi.org/10.1093/mnras/stab1855). arXiv: [2010.03567](https://arxiv.org/abs/2010.03567) [[astro-ph.IM](#)].
- Stroustrup, Bjarne (Jan. 2013). *The C++ programming language*. en. 4th ed. Boston, MA: Addison-Wesley Educational. ISBN: 978-3-96730-017-8.
- Suzuki, N. et al. (Feb. 2012). “The Hubble Space Telescope Cluster Supernova Survey. V. Improving the Dark-energy Constraints above $z \lesssim 1$ and Building an Early-type-hosted Supernova Sample”. In: *ApJ* 746.1, 85, p. 85. DOI: [10.1088/0004-637X/746/1/85](https://doi.org/10.1088/0004-637X/746/1/85). arXiv: [1105.3470](https://arxiv.org/abs/1105.3470) [[astro-ph.CO](#)].
- Taffoni, G. et al. (Jan. 2020). “CHIPP: INAF Pilot Project for HTC, HPC and HPDA”. In: *Astronomical Data Analysis Software and Systems XXIX*. Ed. by R. Pizzo et al. Vol. 527. Astronomical Society of the Pacific Conference Series, p. 307. arXiv: [2002.01283](https://arxiv.org/abs/2002.01283) [[astro-ph.IM](#)].
- Thomas, D. B., M. Bruni, and D. Wands (Sept. 2015). “Relativistic weak lensing from a fully non-linear cosmological density field”. In: *JCAP* 2015.9, pp. 021–021. DOI: [10.1088/1475-7516/2015/09/021](https://doi.org/10.1088/1475-7516/2015/09/021). arXiv: [1403.4947](https://arxiv.org/abs/1403.4947) [[astro-ph.CO](#)].
- Thomas, D.B. et al. (2015). “f(R) gravity on non-linear scales: the post-Friedmann expansion and the vector potential”. In: *Journal of Cosmology and Astroparticle Physics* 2015.07, p. 051. DOI: [10.1088/1475-7516/2015/07/051](https://doi.org/10.1088/1475-7516/2015/07/051). URL: <https://dx.doi.org/10.1088/1475-7516/2015/07/051>.
- Wald, Robert M. (1984). *General Relativity*. First Edition. University Of Chicago Press. ISBN: 0226870332.
- Weinberg, S. (1972). *Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity*. Wiley. ISBN: 9780471925675.
- (2008). *Cosmology*. OUP Oxford. ISBN: 9780198526827.
- White, S. D. M. (Dec. 1976). “The dynamics of rich clusters of galaxies.” In: *MNRAS* 177, pp. 717–733. DOI: [10.1093/mnras/177.3.717](https://doi.org/10.1093/mnras/177.3.717).
- (July 1978). “Simulations of merging galaxies.” In: *MNRAS* 184, pp. 185–203. DOI: [10.1093/mnras/184.2.185](https://doi.org/10.1093/mnras/184.2.185).
- Xu, Guohong (May 1995). “A New Parallel N-Body Gravity Solver: TPM”. In: *ApJS* 98, p. 355. DOI: [10.1086/192166](https://doi.org/10.1086/192166). arXiv: [astro-ph/9409021](https://arxiv.org/abs/astro-ph/9409021) [[astro-ph](#)].
- Yoo, Jaiyul, A. Liam Fitzpatrick, and Matias Zaldarriaga (Oct. 2009). “New perspective on galaxy clustering as a cosmological probe: General relativistic effects”. In: *Physical Review D* 80.8, 083514, p. 083514. DOI: [10.1103/PhysRevD.80.083514](https://doi.org/10.1103/PhysRevD.80.083514). arXiv: [0907.0707](https://arxiv.org/abs/0907.0707) [[astro-ph.CO](#)].