Constrained local universe simulations from galaxy peculiar velocities

Timur Doumler

N° d'ordre: 76-2012 Année 2012

THÈSE

délivrée par

L'UNIVERSITÉ CLAUDE BERNARD LYON 1, FRANCE

et préparée en cotutelle avec

L'UNIVERSITÉ DE POTSDAM, ALLEMAGNE

ECOLE DOCTORALE
Physique et Astrophysique – PHAST

pour l'obtention

DU DIPLOME DE DOCTORAT

(arrêté du 7 août 2006 / arrêté du 6 janvier 2005)

soutenue publiquement le 22 juin 2012

par

TIMUR DOUMLER

Constrained Local Universe simulations from galaxy peculiar velocities

Directeurs de thèse:
Mme COURTOIS Hélène *(Lyon)* et Mr STEINMETZ Matthias *(Potsdam)*

JURY: Mme COURTOIS Hélène
Mr GASCON Jules
Mr GOTTLÖBER Stefan
Mr HOFFMAN Yehuda
Mr STEINMETZ Matthias
Mr TULLY Brent
Mr YEPES Gustavo

Constrained Local Universe simulations from galaxy peculiar velocities

by

Timur Doumler

Lyon/Potsdam, September 2012

I would like to dedicate this dissertation to my mother, Prof. Inna Doumler. There is no doubt that without her continued support, love, and encouragement, this work would never have been completed.

Abstract

Galaxy peculiar velocities provide valuable information about our motion with respect to the observed large-scale structure and can be used to constrain the underlying dark matter distribution. Based on this approach, the technique of constrained realisations allows us to run numerical simulations that resemble the observed Local Universe. This provides a powerful numerical laboratory to study the dynamics, formation and evolution of structure in the Local Universe. The crucial step is to generate appropriate initial conditions from the observational data.

We present here improvements on the technique of constrained simulations, along with a newly developed highly optimised numerical code that can handle the upcoming large observational datasets. Galaxies evolve from the gravitational collapse of primordial overdensities in the early Universe; their motion leads to a large-scale displacement field. A major source of systematic errors in constrained simulations arises by not accounting for this effect. To overcome this limitation, we develop the Reverse Zeldovich Approximation (RZA) reconstruction method. The RZA allows to reconstruct displacements and initial positions of observed galaxies and generate a significantly better estimate of the initial conditions of the Local Universe. This method is extensively tested on simulation data. We also study the influence of data quality and various observational and systematic errors. We show that with the RZA technique, the reconstruction quality of the density and velocity fields improves significantly. The positions of objects in the evolved constrained simulations are recovered more accurately and over a wider range of masses.

Résumé

Les vitesses particulières de galaxies fournissent des informations importantes sur notre mouvement par rapport aux grandes structures observées et peuvent être utilisées afin de contraindre la distribution de matière noire sous-jacente. En se fondant sur cette approche, la technique des réalisations contraintes permet de calculer des simulations numériques qui ressemblent à l'Univers Local observé. Ceci fourni un laboratoire numérique puissant pour étudier la dynamique, la formation et l'évolution des structures bien connues de l'Univers Local. L' étape cruciale est de générer, à partir des données observationnelles, des conditions initiales appropriées.

Nous présentons ici des améliorations de la technique des simulations contraintes, accompagnées d'un code numérique nouvellement développé et hautement optimisé, qui peut gérer les énormes ensemble de données observationnelles. Les galaxies évoluent à partir du effondrement gravitationnel des surdensités primordiales de l'Univers très jeune ; leurs mouvements créent un champ de déplacement à grande échelle. Une source majeure d'erreurs systématiques dans les simulations contraintes est produite si l'on ne tient pas compte de cet effet. Afin de dépasser cette limitation, nous avons développé la méthode de reconstruction par approximation inverse de Zeldovich (RZA). La RZA permet de reconstruire les déplacements et les positions initiales des galaxies observées et de générer une estimation significativement meilleure des conditions initiales de l'Univers Local. Cette méthode est intensivement testée sur des données de simulations. Nous étudions aussi l'influence de la qualité des données et de diverses erreurs observationnelles et/ou systématiques. Nous démontrons qu'avec la technique RZA, la qualité de la reconstruction des champs de densité et de vitesse est drastiquement améliorée. Les positions des objets dans les simulations contraintes évoluées sont retrouvées plus précisément et sur un plus grand intervalle de masses.

Zusammenfassung

Die Pekuliargeschwindigkeiten von Galaxien liefern wertvolle Informationen über ihre Bewegung relativ zu den beobachteten großräumigen Strukturen und können verwendet werden, um die zugrundeliegende Verteilung der dunklen Materie abzuschätzen. Auf dieser Grundlage können mit der Methode der "Constrained Realisations" numerische Simulationen durchgeführt werden, die die Materieverteilung im Lokalen Universum widerspiegeln und somit leistungsfähige numerische Experimente ermoeglichen, um die Dynamik, Entstehung und Evolution von Strukturen im lokalen Universum zu untersuchen. Eine wichtige Voraussetzung für diese Experimente ist es, korrekte Anfangsbedingungen aus den Beobachtungsdaten abzuleiten.

In dieser Arbeit werden Verbesserungen der "Constrained Simulations"-Technik sowie ein für diesen Zweck neu entwickelter, hochgradig optimierter numerischer Code vorgestellt, welcher es ermöglicht, die kommenden, sehr umfangreichen Beobachtungsdaten zu verarbeiten. Galaxien entwickeln sich aus dem Gravitationskollaps von primordialen Überdichten im frühen Universum. Deren Bewegung führt zu einer großräumigen Dislokation, deren Vernachlässigung die größte systematische Fehlerquelle für "Constrained Simulations" darstellt. An dieser Stelle knüpft die Reverse-Zeldovich-Näherung (RZA) an, die hier vorgestellt wird. Mithilfe der RZA können wir diese Dislokation abschätzen und somit die ursprünglichen Positionen der beobachteten Galaxien rekonstruieren. Dadurch erhält man wesentlich genauere Anfangsbedingungen fuer die Entwicklung des Lokalen Universums. Diese Methode wird ausführlich an Simulationsdaten getestet. Der Einfluß der Datenqualität und verschiedener Beobachtungs- und systematischer Fehler wird eingehend untersucht. Die Rekonstruktionsgenauigkeit der Dichte- und Geschwindigkeitsfelder kann durch den Einsatz der RZA signifikant erhöht werden. In unseren Tests stimmen die Positionen von Objekten in den mit RZA erzeugten "Constrained Simulations" wesentlich besser mit den ursprünglichen überein, als ohne die RZA, und das für einen deutlich größeren Bereich von Massen.

Contents

Not only does God play dice with the world – He does not let us see what He has rolled.

Stanisław Lem

Chapter 1

Motivation

We live in a time where cosmology, the study of the structure, origin, and evolution of the Universe on large scales, has finally reached the stage of a precision science. The ΛCDM (Lambda-Cold-Dark-Matter) model has established itself as the standard model of modern cosmology. In this generally accepted picture, the Universe evolved from an extremely dense and hot state, termed the Big Bang, followed by a stage of expansion and cooling that continues to this day. Contemporary large-scale structure, such as galaxies and clusters, condensed by gravitational amplification of initial small fluctuations in the cosmic density field that are likely of quantum origin. The cosmological parameters that describe the matter-energy content of the Universe and its expansion history are known today to a high precision of a few percent (Komatsu et al. 2011). This consistent image emerged by bringing together theoretical models (Peebles 1980, 1993; Weinberg 2008), observations of the cosmic microwave background radiation (Penzias & Wilson 1965; Smoot et al. 1991; Spergel et al. 2007), the distribution of galaxies on large scales (Tegmark et al. 2004; Cole et al. 2005), estimates of the primordial abundances of the light elements (Alpher et al. 1948), supernovae explosions in distant galaxies (Riess et al. 1998; Perlmutter et al. 1999), and assessments of the total mass of individual galaxies and galaxy clusters (Oort 1930a,b,c; Zwicky 1933, 1937; Freeman 1970; Clowe et al. 2006). The surprising finding was that the Universe is dominated by dark matter, believed to be composed of collisionless particles interacting solely by gravity and thus exremely difficult to observe, and dark energy, which can be understood as a field of non-zero vacuum energy that permeates the whole Universe and accelerates the cosmic expansion. Ordinary baryonic matter, despite being the building blocks of all observable, luminous objects like galaxies, stars, and planets, comprises only a small fraction of the cosmic matter-energy balance. These luminous objects form from baryonic gas accreted into the gravitational potential of dark matter haloes assembled by hierarchical clustering (White & Rees 1978; White & Frenk 1991).

Today, the most successful method of studying the formation and evolution of large-scale structure (LSS) are numerical N-body simulations. Large-scale structure formation is a highly non-linear process and the possibility to describe it analytically is limited. The predictive power of such analytic models was far surpassed by high-resolution cosmological simulations during the advent of high-performance computing in the 1990s, which allowed to numerically resolve the evolution of the LSS on mass and length scales spanning many magnitudes. It is a major achievement of numerical cosmology that these simulations can accurately reproduce the web-like appearance of the large-scale galaxy distribution observed in deep galaxy redshift surveys such as 2dF and SDSS (Springel et al. 2005). The standard approach is to sample the primordial matter density fluctuations with an ensemble of collisionless tracer particles that represent the initial conditions of the dark matter density field. Then, the system can be evolved forward in time by

a step-wise numerical integration of appropriate equations of motion (Efstathiou et al. 1985). Since then, powerful numerical simulation codes for massively parallel computing have been developed (Kravtsov et al. 1997; Teyssier 2002; Springel et al. 2001; Springel 2005, 2010). The N-body method of using collisionless tracer particles is sufficient to accurately model structure formation on super-galactic scales, which is dominated by dark matter clustering. Baryonic physics becomes important at the scale of individual galaxies and below. Modern cosmological simulation codes include different recipes for treating the dynamics of the baryonic component by solving the underlying hydrodynamical equations alongside the dark matter. Numerical recipes have been developed to enhance these methods by adding physics beyond the standard gas hydrodynamics, such as radiative gas cooling and heating, star formation (Springel & Hernquist 2003a,b; Rasera & Teyssier 2006), supernovae feedback (Scannapieco et al. 2005, 2006; Dubois & Teyssier 2008), magnetic fields (Fromang et al. 2006; Dolag & Stasyszyn 2009; Doumler & Knebe 2010), and radiative transfer (Partl et al. 2010; Wise & Abel 2011), which allows one to simulate processes like the formation of galaxies (Steinmetz 1999; Crain et al. 2009; Scannapieco et al. 2011), the epoch of reionisation and first stars (Iliev et al. 2011; Wise et al. 2012), and the intergalactic medium (Cen & Ostriker 1999, 2006; Klar & Mücket 2012). Cosmological simulations have since yielded many interesting insights that could not have been achieved by observational or analytic means alone, and greatly helped to refine our understanding of the formation and evolution of cosmic structure. It must be admitted though that, despite substantial progress in the respective fields, at scales where baryonic – and therefore observable – matter becomes dominant the convergence between observational and theoretical studies is still in its infancy. It is notoriously difficult to reproduce some of the observations in the simulation world, such as the yet not sufficiently understood formation of disk galaxies (Scannapieco et al. 2011), the observed galaxy luminosity function (Binney & Tremaine 2008), or the properties of cosmic voids (Tikhonov & Klypin 2009).

As important as it may be to have a theoretical picture of our Universe's origin and evolution, the focus naturally lies on the specific area of the Universe that we happen to inhabit. It is of special interest to understand the structure and dynamics of the Local Universe that surrounds our own Milky Way galaxy. This encompasses the Local Group (containing the Milky Way, the Andromeda galaxy, and several smaller objects), the Local Supercluster with the Virgo Cluster as its dominant object, and several other clusters at larger distances that govern this region of space (Figure 1.1). The key to an understanding of the dynamics of the Local Universe lies in the peculiar velocity field, observationally accessible through galaxy distance surveys that allow one to determine the galaxies' peculiar velocities (Tully 1988; Willick et al. 1997; Tonry et al. 2001; Karachentsev et al. 2004). Peculiar velocities enable us directly trace the underlying gravitational potential of the total (i.e. both dark and baryonic) matter distribution that determines the evolution of structure in the Local Universe. It is now well established that the Milky Way galaxy has a high peculiar motion of 630 km/s with respect to the CMB rest frame (Fixsen et al. 1996). Several components on different scales are believed to shape this configuration (Tully et al. 2008): the infall towards the nearby Virgo cluster, the push from the Local Void, the large-scale flow towards the region of the Great Attractor, and the possible influence of more distant objects like the massive Shapley concentration, interacting in a complex manner that is still poorly understood. It would be very enlightening to model this intricate system in the framework of cosmological simulations, where one has access to the evolution of the underlying total matter distribution.

If one wishes to do so, a fundamental problem arises. Cosmological simulations – no matter how meticulously they are designed on a technical level – generally are started from initial conditions that represent a random realisation of the known statistics of the primordial density

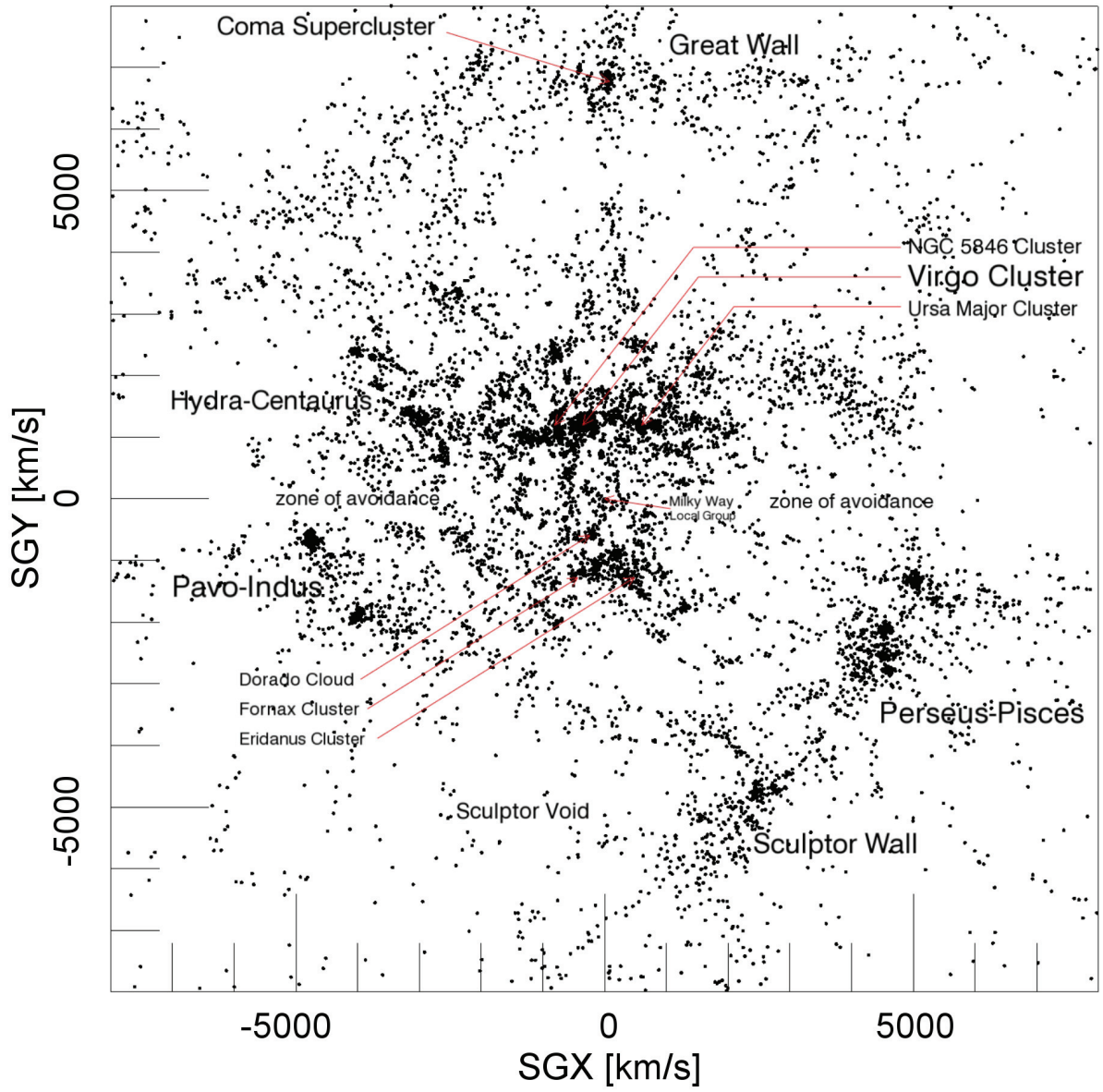

Figure 1.1: The Local Universe: slice of ±1000 km/s through the supergalactic plane from the v8k galaxy redshift catalogue, corrected for redshift distortions. This catalogue is not a volume limited sample and therefore galaxy number density degrades with distance. In the Zone of Avoidance, the view is severely obscured by the galactic disc of the Milky Way and no galaxies are seen. *Figure taken from Courtois, Pomarède & Tully (in prep).*

fluctuations. While the general nature of such a simulated universe will share many similarities with the particular realisation that is our own, it is not practically possible to find by chance structures that resemble the Local Universe in enough detail. This in itself is not necessarily a fundamental problem; a lot can be learned already from random realisations. However, it is currently unclear whether the Local Universe is actually a typical realisation. The peculiar alignment of the observed clusters in the Local Universe and the abnormally high amplitude of the observed peculiar velocity flow suggest that it is not. The Local Universe described by the known observational data is therefore possibly systematically biased.

A solution is to add constraints to the initial conditions that directly come from the observations. By narrowing down the natural cosmic variance of possible universe realisations by imposing a set of constraints, it is possible to obtain simulations that resemble, on large scales, the LSS distribution of the observed Local Universe. The basic idea is to generate a realisation of the initial random field which conforms to the imposed constraints in regions dominated by the available data, while at the same time drawing from a conventional random realisation of the field in regions where the data is insufficient. Observational errors attached to the constraints should be compensated in a way that is consistent with the assumed prior model of the random field. To generate such constrained realisations (CRs), a powerful algorithm was developed by Hoffman & Ribak (1991), but it is only in the last decade that the application of this method has matured to a stage where high-resolution constrained simulations with predictive power about the real Universe can be produced (Klypin et al. 2003). A particularly useful source for constraints that can be used with the CR method are catalogues of galaxy peculiar velocities. They provide a direct and unbiased tracer of the total (i.e. dark matter and baryonic) density field and retain precious information about the large-scale matter distribution and the cosmological initial conditions of the Local Universe due to their high linearity and large-scale correlation. This powerful approach creates a link between the realms of observational and numerical cosmology. This link provides the basis of the CLUES (Constrained Local UniverseE Simulations) project (`www.clues-project.org`), an international collaboration of researchers from both theory and observations with the goal to provide simulations that model the Local Universe as closely as possible. The research presented in this thesis is part of this project.

Within the CLUES project, several constrained simulations with varying boxsizes and cosmological parameters have been performed and investigated, taylored to different scientific applications. Figure 1.2 shows one of the larger CLUES simulations, the BOX160, reproducing several main players that shape the Local Universe, namely the Virgo, Coma, and Perseus-Pisces clusters and the Great Attractor region, and creating in its centre an environment similar to the one of our Local Group. The CLUES simulations can be used as a numerical laboratory to study many aspects of structure formation in the context of this specific environment. Recent projects adress the formation of Milky-Way-like disk galaxies (Scannapieco et al. 2011), the dynamics of satellite galaxies in the Local Group (Libeskind et al. 2010; Knebe et al. 2011b,c), the coldness of the local flow (Martínez-Vaquero et al. 2009), and the reionisation history of the Local Group (Iliev et al. 2011). Of particular interest is the formation history of the Local Group (Forero-Romero et al. 2011). It seems that the Local Group is not a typical object encountered in the Universe, but has evolved with an unusually quiet formation history that could be strongly influenced by the specifics of its large-scale environment.

Despite their success, CLUES simulations have some severe drawbacks which leave much room for improvement. While the major clusters highlighted in Figure 1.2, with virial masses around $10^{15} M_{\odot}$, can be reproduced robustly, the structure on smaller mass and length scales is essentially dominated by the random component. Additionally, the positions of these clusters are subject to large systematic errors (Gottlöber et al. 2010). In practice, it is still necessary to

run many constrained realisations to find one that gives a favourable representation. Structure on much lower mass scales around 10^{12} – $10^{13} M_{\odot}$, which is the mass range that contains the Local Group, appears in a completely random fashion as in unconstrained simulations; to find

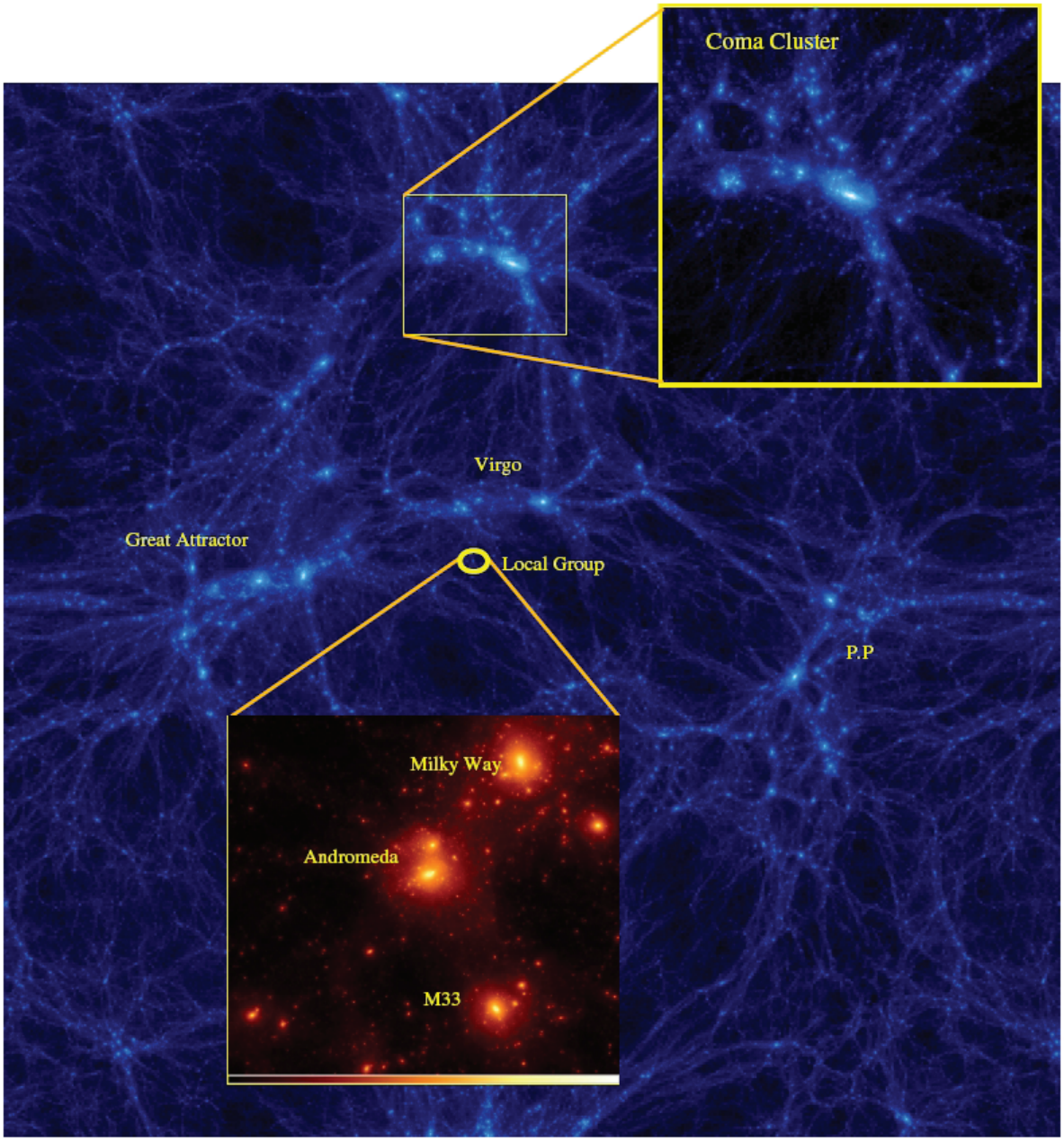

Figure 1.2: Large scale dark matter density distribution of the BOX160 Local Universe simulation. The image covers the full box range of 160 Mpc/h. Several objects reproducing their counterparts in the real Universe are identified, in particular the main galaxy clusters of the Local Universe: Virgo, Perseus-Pisces, Coma and the Norma-Hydra-Centaurus region known as the Great Attractor. The circle shows the position of our Local Group; the inserted panel shows the detailed structure of a simulated Local Group. This small region is 2 Mpc/h wide. *This picture was taken from the CLUES website:* www.clues-project.org

a Local Group-like configuration around the right position, such as the one in the magnified panel in Figure 1.2, a large amount of different constrained realisations have to be searched, defeating in part the idea of constraining cosmic variance. It was previously unclear whether the main reason behind these limitations is the poor quality of the first observational datasets that were used for constraints, flaws in the method to generate constrained initial conditions, the general non-linearity of the system, or a combination of all of the above. In fact, the initial conditions of previous CLUES simulations were constrained using peculiar velocity catalogues that are outdated from today's perspective. Current catalogues (Tully et al. 2009; Courtois et al. 2012), and much more so the peculiar velocity datasets that will become available during the next few years (Courtois et al. 2011a,b; Courtois & Tully 2012; Tully & Courtois 2012), are much less sparse, suffer less from observational errors, cover more volume, and are therefore expected to constrain the simulations much more strongly.

A very important requirement to optimally utilise the available observational data is to refine the current physical method of generating constrained initial conditions, otherwise too much of the additional information obtained from modern observations may be lost in the process. One major problem of the previously used method is that the effects of the cosmic displacement field were not accounted for. This field occurs because the patches of matter that evolve from the initial seed perturbations into observable objects may have travelled long distances up to $10 - 20$ Mpc from the primordial state until today due to the large-scale gravitational field. To recover the initial conditions of the Local Universe based on a present-time snapshot of these objects, one has to follow this movement back in time and reconstruct the positions of their progenitors in the early Universe. Subsequently, these reconstructed initial positions can be used as constraints to generate more accurate initial conditions that can then be, as before, run forward in time with current simulation codes. The lack of such a reconstruction is arguably the main cause of the large systematic errors seen in the CLUES simulations. Simultaneously, a technical challenge has to be surmounted: to our knowledge there is currently no publicly available numerical code for the task of generating constrained initial conditions that could handle with reasonable efficiency the great number of constraints expected to come from upcoming surveys, or would be flexible enough to adapt to changes and refinements to the technique of generating constrained initial conditions from peculiar velocities.

This work picks up at these critical strands. We propose a new reconstruction scheme for the initial positions and displacements of the haloes hosting the observed galaxies, the Reverse Zel'dovich Approximation (RZA), which is conceptually simple and significantly improves the quality of initial conditions constrained with peculiar velocity data. As a prerequisite, the numerical code ICeCoRe (Initial Conditions and COnstrained REalisations) was developed for this work. It is a command-line program that implements a combination of very efficient algorithms to create constrained (as well as conventional, i.e. unconstrained) initial conditions for cosmological simulations. These algorithms are also useful for the related technique of Wiener filter reconstruction from observational data. The ICeCoRe code was designed to be flexible with regards to the type of constraints that one wishes to use, and therefore has applications beyond the procedure of constraints from radial peculiar velocities that is the main topic of this work. Equipped with these newly developed tools, we address several important questions in the context of generating constrained simulations: How accurate is the CLUES technique in producing a simulated recreation of the observed Local Universe? How much do we improve in accuracy by adding the RZA technique of initial conditions reconstruction? What is, in this context, the best way to place constraints to generate constrained initial conditions? What is the effect of observational errors coming from the data, and systematic errors of the method itself? Can we give some hints on optimising the observational data that is used for constraints? What role does

the random component play and how great is the variance across constrained realisations from different random seeds? And based on these results, is it possible to give a precise lower limit on the mass and length scales that can be robustly reproduced with the constrained simulations technique?

To cover all these questions, a systematic study is presented in this work based on mock peculiar velocity catalogues drawn from the BOX160 simulation that serves as the "model Universe". These mock catalogues serve as the input to reconstruct its initial conditions and run constrained re-simulations of it. We decided to work with mock catalogues as opposed to observational data, because in the underlying BOX160 simulation the complete history of the density and velocity fields and the dark matter haloes is accessible, and the generated initial condition reconstructions and constrained simulations can be immediately checked against the "true" result. This is required to draw quantitative statements on the accuracy of the procedure. In principle, it would be possible to carry out such a study using a conventional random simulation as the test universe, such as the state-of-the-art simulations publicly available through the MultiDark database (Riebe et al. 2011). We chose the BOX160 instead, because it is already a constrained simulation, albeit generated with the previous CLUES method, which means that it has some similarity with the observed Local Universe. In this way, quantitative estimates on the reconstruction quality can be easier applied to observational data, since the large-scale velocity field and cluster distribution that we wish to reconstruct here are not too different. In this work, we are interested in the large-scale distribution of matter, i.e. scales above those where baryonic physics are important. We therefore decided to completely neglect baryonic physics for this study and perform the analysis in a dark-matter-only framework. We take the peculiar velocities of dark matter haloes in the simulation as a proxy for the observable peculiar velocities of galaxies that would form inside them and by this remove a lot of complexity.

Figure 1.3 summarises the general idea behind this study. Starting from the BOX160 simulation as our "model universe", we extract a catalogue of the dark matter haloes from the evolved present-time simulation snapshot, and construct different mock catalogues of galaxy peculiar velocities in such a way that they reproduce different properties of the observational data, such as the limited data volume, incompleteness and the observational errors. This mock data is then used as input for the ICeCoRe code to reconstruct the cosmological initial conditions and create a constrained realisation that can be itself simulated forward until present time and compared to the original configuration. We do this with the "classical" CLUES method as well as with adding our newly developed RZA reconstruction. We also test and compare different methods to place these constraints onto the random field. The comparison with the "true field" of BOX160 provides feedback information that can be used to refine the method, thus closing the development cycle (black arrows in Figure 1.3). The outcome is an optimised method of generating constrained intial conditions from peculiar velocity data together with quantitative estimates of its validity and precision. With this, the natural next step for future work is to directly apply the method to observational data (red arrows in Figure 1.3) and yield constrained simulations of the actual Local Universe that should be significantly more accurate than earlier attempts. This will be the subject of future studies.

This thesis is organised as follows. In Chapter 2, we summarise the tools, assumptions and models that are used: the currently available peculiar velocity data, the standard ΛCDM model of structure formation, numerical cosmological simulations, and how they are interrelated. Chapter 3 presents the employed numerical methods: the generation of cosmological initial conditions, Wiener filtering, the constrained realisations algorithm, and their implementation in the ICeCoRe code that was developed for this work. Chapter 4 presents the Reverse Zeldovich Approximation (RZA) reconstruction, a novel method for the reconstruction of cosmological initial

conditions from peculiar velocity catalogues, specifically the initial positions of their underlying dark matter haloes. Chapter 5 applies this method to our simulated test universe, the BOX160 simulation. We construct constrained realisations of cosmological initial conditions with the RZA reconstruction using mock catalogues extracted from BOX160. Then, we perform and analyse a set of constrained simulations. Finally, in Chapter 6, the obtained results are summarised and discussed, and an outlook is given for further development and scientific applications of the work presented here. In the Appendix, we provide a full documentation of the ICECORE code, with the hope that it will be a useful tool for the scientific community.

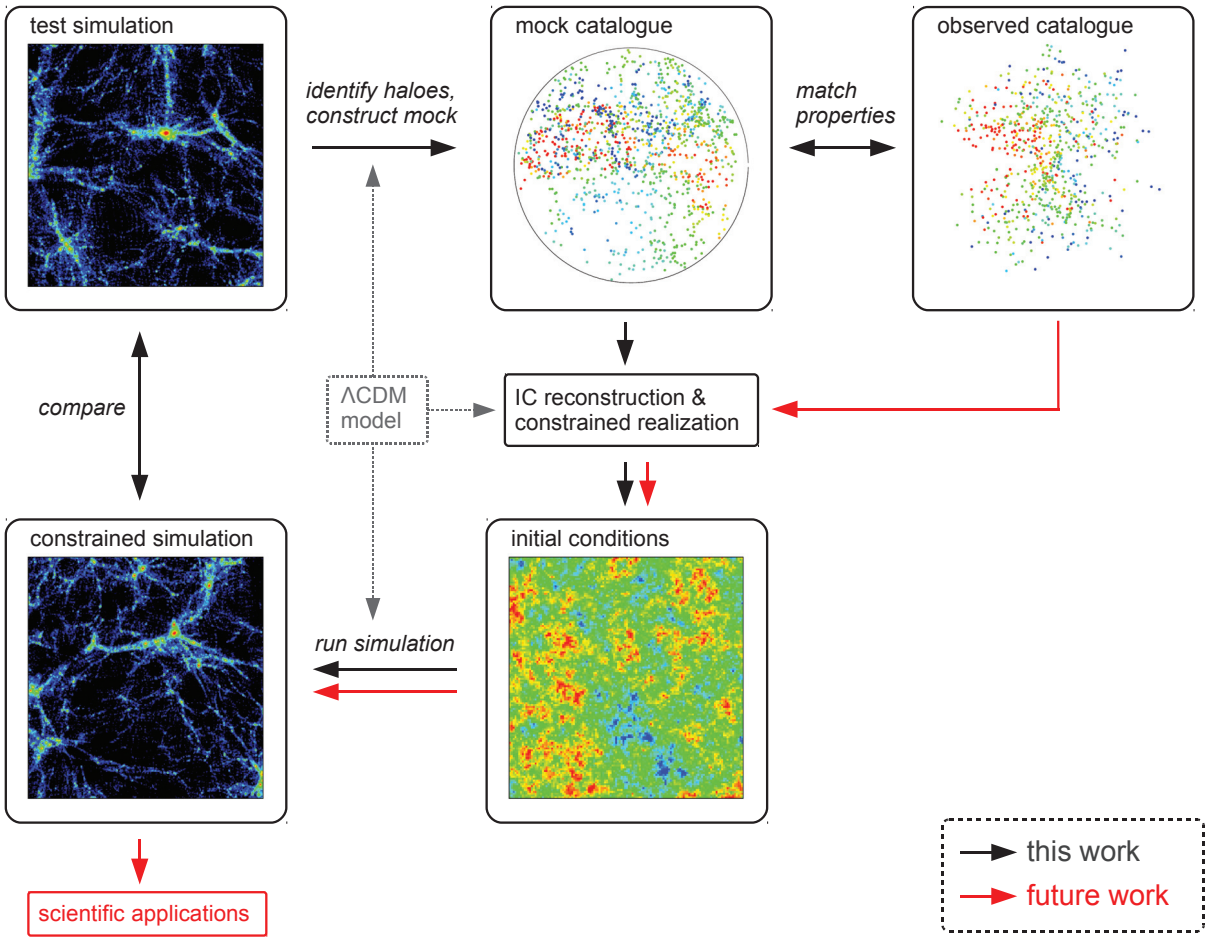

Figure 1.3: Schematic view of the approach underlying this study. Mock galaxy distance catalogues from a previously performed cosmological simulation of structure formation are used to reconstruct its initial conditions and to create constrained realisations. Those are then run forward until present time and compared with the original simulation. This way, an optimal reconstruction method can be found, and the impact of different error sources in the reconstruction process can be studied. In particular, the effects of observational errors introduced in the mocks, mimicking actual observational data, can be quantified. This development cycle is shown with black arrows. After a stable procedure has been developed by iterations over this cycle, it can eventually be directly applied to the observational data to produce constrained simulations of the actual Local Universe that can be used as numerical laboratory for scientific studies (red arrows).

Chapter 2

The Universe

In this chapter, we review the currently available observational peculiar velocity data and the insight they give into the dynamics of our Local Universe. We then summarise the theoretical framework of cosmic structure formation in the standard ΛCDM model that the subsequent chapters rely on. We introduce a mathematical description of the cosmic expansion and the theory of density perturbations. Structure formation is, except at very early times, a highly non-linear process that cannot be described analytically. The tool of choice to model this process are N-body simulations. The non-linear density and velocity fields produced by such simulations have several properties that have to be taken into account if one wishes to construct constrained initial conditions based on present-day data. Different arguments are given why galaxy peculiar velocity catalogues are a good candidate for such input data.

2.1 Galaxy peculiar velocities and the Local Flow

2.1.1 The Hubble Law

The discovery of the cosmic expansion was eventually made possible by the development of a method to measure extragalactic distances. It dates back to Hubble (1929), who used for this purpose observations of Cepheid variable stars as standard candles. He discovered that on sufficiently large scales all galaxies are moving away from us, and that the observed radial velocity $v_r^{\rm obs}$ increases with distance from the observer r proportionally to a constant H_0. This Hubble constant is interpreted as the current value of the expansion rate of the Universe. It can be estimated from the relation of observed radial velocities $v_r^{\rm obs}$ with the measured distances r out to sufficiently large scales, which will be a linear relation with the slope

$$H_0 = \left\langle \frac{v_r^{\rm obs}}{r} \right\rangle \quad . \tag{2.1}$$

The scatter around this relation is due to the peculiar motions $v_r^{\rm pec}$ of galaxies. The observed total radial velocity $v_r^{\rm obs}$ of a galaxy is therefore the sum of its peculiar motion and the "Hubble drag" due to the expansion,

$$v_r^{\rm obs} = v_r^{\rm pec} + H_0\, r \quad . \tag{2.2}$$

The total radial velocity itself is directly observed via the redshift z of the electromagnetic spectrum of a receding object from wavelengths Λ to observed wavelengths $\Lambda^{\rm obs}$, which can be

interpreted as a Doppler shift,

$$v_r^{\rm obs} = c\,z \quad ; \quad 1+z = \frac{\Lambda^{\rm obs}}{\Lambda} \quad , \tag{2.3}$$

where c is the vacuum speed of light.

Since it is very difficult to obtain a high precision on extragalactic distance measurements, the Hubble constant H_0 is not known with as much accucacy as the other cosmological parameters. Current observations constrain it to lie within about $70 < H_0 < 75$ km s^{-1} Mpc^{-1}. The Hubble Space Telescope Key Project gives a generally accepted estimate of 72 ± 8 km s^{-1} Mpc^{-1} (Freedman et al. 2001). Recently, the value of 70.2 ± 1.4 km s^{-1} Mpc^{-1} was derived from the WMAP7 data (Komatsu et al. 2011). In theoretical cosmology, a unit system is often used that parametrises away this uncertainty by defining the Hubble constant as

$$H_0 = 100\,h\,\text{km s}^{-1}\,\text{Mpc}^{-1} \quad , \tag{2.4}$$

and to express distances in Mpc$/h$ and masses in $M_\odot/h$, respectively. We adopt this convention here, although we reverse to the units of the observed scale (Mpc and $M_\odot$, respectively) for observational data (see Appendix D for the numerical values of these units).

2.1.2 Observational data

Peculiar velocities of galaxies are computed from the observation of their redshift and an independent estimate of their distance. The redshift allows one to determine the radial component of their total velocity $v_r^{\rm obs}$ via equations 2.2 and 2.3. If one can measure the luminosity L and therefore the absolute magnitude M of a galaxy, one obtains the distance modulus $\mu = m - M$, where m is the apparent magnitude and M is defined as the apparent magnitude that would be observed at a distance of 10 parsec from Earth. The distance r in units of parsec can then be computed as

$$r = 10^{1+\mu/5} \quad . \tag{2.5}$$

The uncertainty in distance Δr follows from the uncertainty in the distance modulus $\Delta\mu$ by standard error analysis,

$$\Delta r \;=\; \frac{1}{5}\ln(10) \times 10^{1+\mu/5}\Delta\mu \;\approx\; 0.461\,r\,\Delta\mu \quad . \tag{2.6}$$

The radial peculiar velocity $v_r^{\rm pec}$ can then be directly obtained through equation 2.2 based on a generally accepted zero-point scale (Freedman et al. 2001). However, the obtained values for $v_r^{\rm pec}$ are relatively insensitive to a zero-point error in the distance scale, as long as the chosen value of H_0 used in equation 2.2 is consistent with the distance measurements (Tully et al. 2008). The observational error on the radial peculiar velocity, $\Delta v_r^{\rm pec}$, is determined by the observational error on the distance Δr,

$$\Delta v_r^{\rm pec} = -H_0\,\Delta r \quad . \tag{2.7}$$

Several methods exist to measure the absolute magnitude and luminosity of galaxies. The Tully-Fisher relation (Tully & Fisher 1977) is an empirical relationship between the luminosity of a spiral galaxy and the amplitude of its gas rotation speed. This well-established method can provide distances with decent accuracy and a high data density over an appropriately large volume and is appropriate to obtain data that would be suitable for a reconstruction of the

underlying field and eventually the cosmological initial conditions. The relation does not apply to elliptical galaxies, since they are in general not rotationally supported and contain few gas. However, a comparable method can be used with elliptical galaxies: fundamental plane (Faber & Jackson 1976; Djorgovski & Davis 1987; Colless et al. 2001), which establishes a relationship between the luminosity, the central stellar velocity, and the effective radius of the galaxy. An alternative is the surface brightness fluctuation (SBF) method (Tonry et al. 2001). However, he disadvantage of elliptical galaxies is that they are preferentially located in high-density regions (morphology-density relation, e.g. van der Wel et al. 2010) which do not sample the large-scale galaxy flows. Other galaxy distance measurement methods include the Cepheid period-luminosity relation (Freedman & Madore 1990; Freedman et al. 2001), which was already known to Hubble, and the tip of the red giant branch (TRGB) method (Karachentsev et al. 2004; Rizzi et al. 2007), although these methods suffer from limited reaches out to about only 10 – 15 Mpc. Data out to very far distances and independent from the galaxy types can be obtained from observations of type Ia supernovae serving as standard candles (Jha et al. 2007). While this method is quite accurate, it rests on serendipity and thus can provide only very sparse data samples. The Tully-Fisher method of measuring distances to spiral galaxies is the only one that currently combines all necessary assets: probing the space regions where coherent cosmic flows prevail and obtaining an adequate sampling of the Local Universe volume.

Methods for measuring galaxy distances have typical observational errors of $\Delta r \approx 10 - 20\%$. It follows then from equation 2.7 that peculiar velocity values with acceptable accuracy can be obtained only within a relatively limited volume: at a distance of 30 Mpc/h and with a relatively low distance error of $\Delta r = 10\%$, the uncertainty in the peculiar velocity will be 300 km/s, and at 60 Mpc/h the error is already at 600 km/s, so that at such a distance the peculiar velocity datapoints will be extremely noisy with relative errors of 100% and more. However, the error can be reduced if one can obtain independent distance measurements for several galaxies that are known to belong to one galaxy cluster or group.

The current dataset of peculiar velocities used by the CLUES project is the Cosmicflows-1 catalogue. This data was assembled by Tully et al. (2008), extending the data of the Nearby Galaxies Catalog (Tully 1988), and currently contains distances to 1797 galaxies in 742 groups, providing a fairly complete sampling of the sky within 3000 km/s (corresponding to a distance of 30 Mpc/h). This data is publicly available through the Extragalactic Distance Database (EDD, Tully et al. 2009). It is a combination of distances obtained with the Tully-Fisher relation, the Cepheid period-luminosity relation, the TRGB method, and the SBF method. Figure 2.1 illustrates the distribution of the datapoints with distance and the associated errors: the median distance error is at 13% and the maximum error at 20%. Figure 2.2 shows an all-sky projection of the radial peculiar velocities computed from the 742 group distances. The dark grey shading is the Zone of Avoidance (ZoA), a narrow strip devoid of datapoints where the view is severely obscured by the galactic disc of the Milky Way.

The ongoing observational work in the Cosmicflows program is currently directed towards preparing a much deeper and larger sample of peculiar velocities (Courtois et al. 2011a,b; Courtois & Tully 2012; Tully & Courtois 2012). The upcoming data will contain distance measurements out to 6000 km/s, and eventually reach out to 15 000 km/s in the near future, exceeding all presently available data in both data volume and precision.

2.1.3 Cosmic flows in the Local Universe

It is now well established that our Galaxy has a peculiar motion of about 630 km/s relative to the rest frame of the observable Universe (Fixsen et al. 1996), which manifests itself as an observed dipole anisotropy in the cosmic microwave background. To explain the observed amplitude and

direction of this peculiar motion is the subject of a lively debate. The total peculiar motion can be decomposed into components on successively larger scales. Obviously, one has first to correct for the orbit of the Earth-bound observer and the motion of the Sun in the Milky Way disk, which is quite substantial, in order to define a Galactic standard of rest. Then, the most local component of the Milky Way (MW) peculiar velocity is the collision course towards our neighbouring Andromeda (M31) Galaxy (Hoffman et al. 2007) at a distance of approximately 0.75 Mpc (Ribas et al. 2005). On a larger scale, the Local Group, i.e. MW, M31, and the surrounding smaller galaxies, are embedded in a coherently moving Local Sheet with a radius of about 7 Mpc (Tully et al. 2008). In this local region, the variance of peculiar velocities is remarkably low; this property is known as the coldness of the local Hubble flow (Karachentsev et al. 2003; Macciò et al. 2005). The bulk peculiar motion of the Local Sheet is believed to be dominated by two components. The first is the infall towards the nearest massive galaxy cluster and its surrounding structure, i.e. the Virgo cluster about 16 Mpc away and with a mass around $10^{15} M_{\odot}$ (Fouqué et al. 2001). Almost orthogonal to the direction of the Virgocentric infall the Local Sheet experiences a peculiar motion away from the Local Void, which may be perceived as an outwards "push" due to the Void's expansion. This Local Void may be in its entirety as large as 70 Mpc across (Tully et al. 2008), although this is difficult to determine from the current observations.

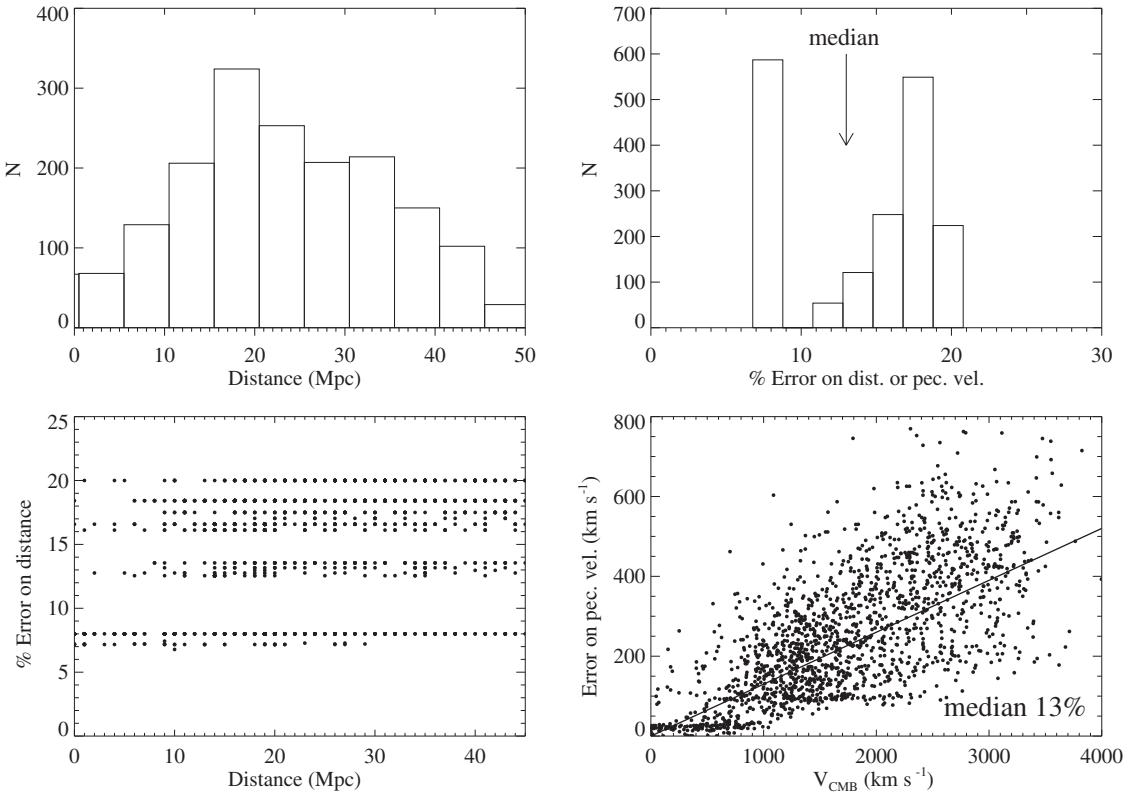

Figure 2.1: Observational data in the Cosmicflows-1 catalogue. Top left: distance distribution of the 1797 individual measurements within 742 groups with accurate distance measurements. Top right: the distribution of percent errors in distances and peculiar velocities in the observed catalogue. Bottom left: percent error in distance as a function of distance. Bottom right: the absolute errors on peculiar velocities over the observed total velocity. Note that here, distances are on the observed scale (Mpc instead of Mpc/h). *Figure taken from Courtois et al. (2012).*

On a larger scale, it is well known since the 1980s that the whole volume encompassing the Local Group and the Virgo Cluster experiences a strong pull towards a massive overdense region ("Great Attractor") at distances of 30 – 50 Mpc, centered near the giant Norma, Hydra and Centaurus clusters (Lilje et al. 1986; Lynden-Bell et al. 1988). At the position of the Local Group, this pull is balanced by the Perseus-Pisces cluster, which is located at a similar distance of ≈ 50 Mpc$/h$ in opposite direction. The resulting large-scale tidal field is believed to have a significant influence on the evolution of the region containing the Local Group (van de Weygaert & Hoffman 1999). However, at the 50 Mpc$/h$ scale the flow does not yet converge to the amplitude and direction of the CMB dipole. The current debate concentrates mainly around the question whether this can be explained with peculiar motions on even larger scales. There seems to be a flow towards the massive Shapley concentration about 150 Mpc away (Kocevski & Ebeling 2006; Kocevski et al. 2007), which is currently believed to be the most massive overdensity in the Local Universe. It has been also theorised that the large-scale flow could continue even further (Feldman et al. 2010; Kashlinsky et al. 2010). Currently, there is still no consensus on the depth of the convergence of these large-scale flows with the CMB dipole. As an alternative to theories based on the peculiar velocity field, a primordial "tilt" of the Universe has been proposed to explain it (Ma et al. 2011; Fixsen & Kashlinsky 2011). To shed more light on the issue, in the last few years significant progress has been made in obtaining reconstructions of the large-scale density and peculiar velocity fields from different observational data. Lavaux et al. (2010) used the 2MASS redshift survey to obtain a reconstruction of the velocity field out to 120 Mpc$/h$, which reaches out to the periphery of the Shapley concentration. Figure 2.3 shows a map of this reconstruction highlighting the most prominent structures. They found that less than half of the amplitude of the CMB dipole is generated within a 40 Mpc$/h$ radius enclosing the Hydra-Norma-Centaurus region, and they do not observe a convergence to the actual direction of the CMB dipole even out to 120 Mpc$/h$. There are claims that this may pose a challenge to the standard ΛCDM model (Lavaux et al. 2010). However, others argue in favour of a quicker

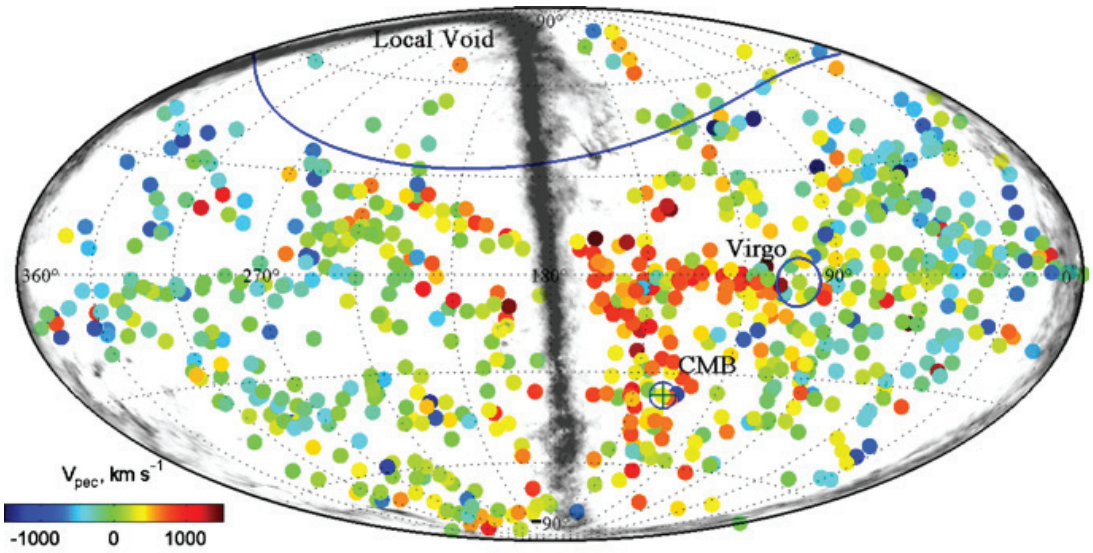

Figure 2.2: An Aitoff sky projection of the 742 grouped peculiar velocities in Cosmicflows-1 with galactic extinction (Zone of Avoidance) overplotted in grey shading. The directions towards Virgo and the CMB dipole and the Local Void are marked in blue. *Figure taken from Courtois et al. (2012).*

conversion of the flow, such as Erdoğdu et al. (2006) using a Wiener Filter reconstruction of the 2MASS catalogue. More recently, Courtois et al. (2012) used the Cosmicflows-1 catalogue of peculiar velocities for a Wiener Filter reconstruction of the large-scale field (see Chapter 3). This allows one to decompose the reconstructed velocity field into local and tidal components. They found that, at a distance of 80 Mpc/h, the velocity field seems to be dominated by the tidal component, i.e. induced by overdensities outside of that radius. They also confirm that the expansion of the Local Void significantly contributes to the peculiar velocity of the Local Group, and that the dynamical role of the Virgo cluster is much less dominant than believed earlier. This is mostly consistent with the study of Nusser & Davis (2011), who used the SFI++ Tully-Fisher catalogue (Springob et al. 2007) and a different method of bulk flow estimation.

The large-scale flows that shape the configuration of the Local Universe have important implications on the Local Group itself. Using constrained simulations of the Local Universe, Forero-Romero et al. (2011) found that the formation of a Local-Group like object seems to require a specific formation history. An early formation time and a quiescent mass accretion history without major merger events provides a favourable environment for the formation of disc galaxies like MW and M31. Their results support the view that the specific large-scale configuration around the Local Group plays a critical role in providing this particular environment. Thus, the formation and evolution of the Local Group is deeply connected to the surrounding large-scale structure that is shaped by cosmic flows.

A crucial step towards the understanding of these complicated dynamics lies in the much more accurate observational data which is expected to become available in the next few years. In order

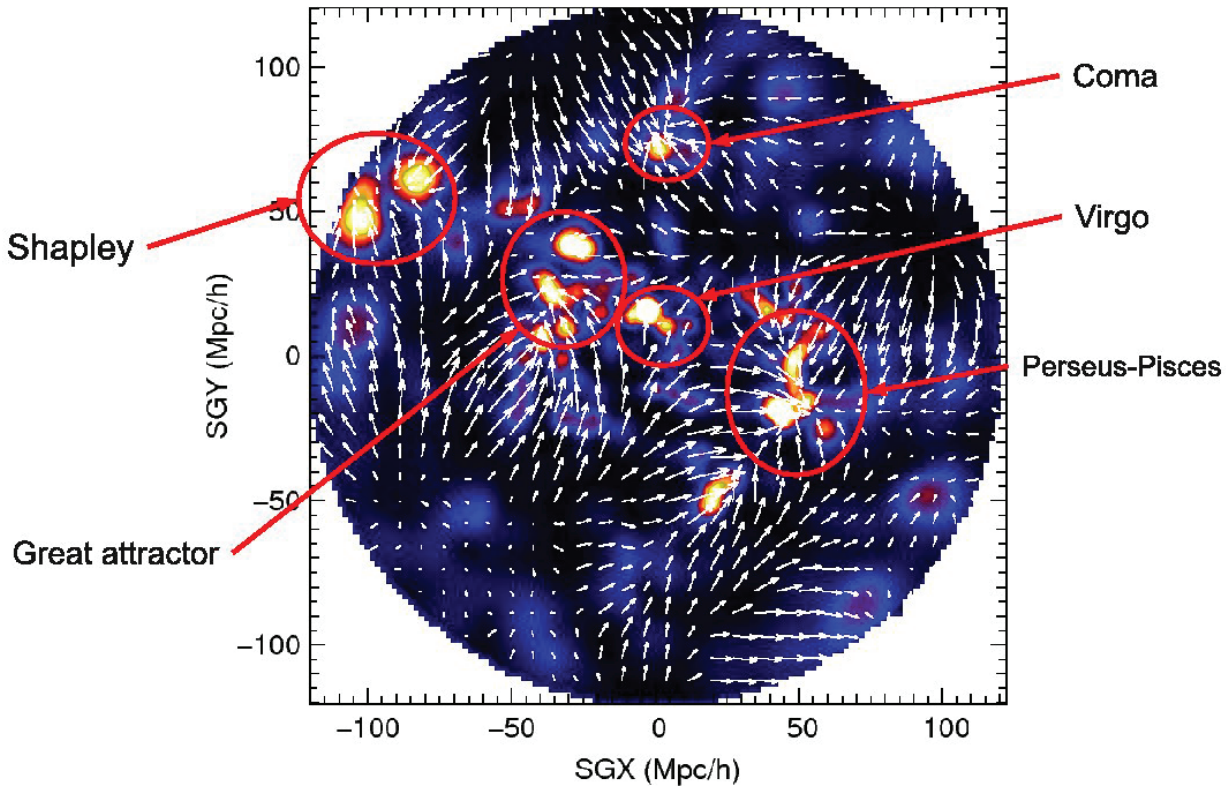

Figure 2.3: Reconstructed density and peculiar velocity field in the Local Universe (SGZ=0 plane) out to a distance of 120 Mpc/h using MAK reconstruction applied to the 2MASS Redshift Survey and constrained with observed distances. *Figure taken from Lavaux et al. (2010).*

to optimally exploit the information provided from this data, a good algorithm to reconstruct the underlying density and velocity fields is required. Furthermore, we need a way to produce precise initial conditions for constrained simulations which then allow us to model the evolution of this system. The focus of this thesis is aimed towards developing these necessary tools.

2.2 The ΛCDM cosmology

In the standard cosmological model, the Big Bang was followed by a phase of inflation, an extremely rapid exponential expansion that amplified primordial quantum fluctuations to macroscopic density fluctuations. In the following era, the contemporary elementary particles and eventually nuclei of the light chemical elements successively condensed from the matter-radiation field as the Universe continued to expand and cool. Eventually, in the transition from this radiation-dominated to the matter-dominated epoch, baryonic matter decoupled from the radiation field. We can observe the photons originating from this surface of last scattering in the CMB. In the following "dark ages", structure began to condense from the relatively homogeneous matter distribution by the gravitational collapse of overdense regions. Eventually, the densest regions became dense enough for the first stars to form, and their radiation ionised the Universe (reionisation). For the subsequent evolution, the ΛCDM model favours a scenario of hierarchical structure formation, where larger structure formed by merging and accretion of smaller structure, leading from the first stars to galaxies and eventually galaxy clusters.

This work considers the matter-dominated epoch where the formation of large-scale structure takes place, i.e. starting from some time after the CMB decoupling and until today. This is also the epoch which is modelled by numerical cosmological simulations. In the remainder of this section, the corresponding theoretical framework is presented. This material is covered in many cosmology textbooks, therefore only a brief summary is given here, limited to the equations and assumptions that will be explicitly used or referenced in this work. For a complete description we refer the reader to the wealth of available literature, e.g. Peebles (1980, 1993); Schneider (2006); Peacock (2007); Weinberg (2008).

2.2.1 The expanding Universe

A generalisation of the Hubble law for arbitrary times t is that the relative separation $\boldsymbol{r}$ and total velocity $\boldsymbol{v}$ between two positions increases with time as

$$\boldsymbol{v} = \boldsymbol{u} + H(t)\,\boldsymbol{r} \quad , \tag{2.8}$$

where $\boldsymbol{u}$ is the time-dependent peculiar velocity, such that (2.2) is the special case of $t = t_0$ (present time). Then, $H(t)$ is the time-dependent expansion rate of the universe, with $H_0 = H(t_0)$. The relative size of the Universe at any time t can be described by the cosmic scale factor $a(t)$, normalised such that at present time t_0, the scale factor is defined as $a(t_0) = 1$. For an expanding Universe, $a(t)$ is then a monotonically increasing function of time with $a(t) < 1$ at all times $t < t_0$. This scale factor is related to the expansion rate via

$$H(t) = \frac{\dot{a}}{a} \quad . \tag{2.9}$$

The fundamental assumption of the cosmological principle states that no point and no direction in the Universe is privileged. From this follows that, on sufficiently large scales, the Universe is homogeneous and isotropic. This assumption defines a unique general metric, the Friedmann-Lemaître-Robertson-Walker (FLRW) metric (Robertson 1933, 1935, 1936a,b; Walker 1937; based

on earlier work by Friedmann 1922, 1924; Lemaître 1927, 1931). This metric does not explicitly depend on time except for the scale factor $a(t)$. In order to derive an expression for the time dependence of $a(t)$, we need an equation of state for each component of the Universe, $p = w\rho c^2$, relating the pressure p and density ρ for matter ($w_\mathrm{m} = 0$, non-relativistic fluid or gas), radiation ($w_\mathrm{r} = 1/3$, relativistic particles), and vacuum energy ($w_\Lambda = -1$). With the FLRW metric and these equations of state, such an expression can be derived directly from Einstein's field equations of general relativity (Einstein 1914, 1915a,b, 1916). The derivation yields the Friedmann equations,

$$H^2 = \left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G\rho}{3} - \frac{kc^2}{a^2} + \frac{\Lambda c^2}{3} \quad , \tag{2.10}$$

$$\dot{H} + H^2 = \frac{\ddot{a}}{a} = -\frac{4\pi G}{3}\left(\rho + \frac{3p}{c^2}\right) + \frac{\Lambda c^2}{3} \quad , \tag{2.11}$$

with the physical constants G (gravitational constant) and c (speed of light), and the vacuum energy or cosmological constant Λ. The total matter density is made up of the non-relativistic (ordinary matter) and relativistic (radiation) component, $\rho = \rho_\mathrm{m} + \rho_\mathrm{r}$. Further, k is the curvature of space, with the three possible cases of the positively curved hyperspherical space $k > 0$ ("open" Universe), negatively curved hyperboloid space $k < 0$ ("closed" Universe), and the ordinary Minkowski space ("flat" Universe).

In cosmology, the different terms in (2.10) are usually encoded in the dimensionless cosmological parameters,

$$\Omega_\mathrm{m} = \frac{8\pi G\rho_{\mathrm{m},0}}{3H_0^2} \qquad \text{matter density parameter,} \tag{2.12}$$

$$\Omega_\mathrm{r} = \frac{8\pi G\rho_{\mathrm{r},0}}{3H_0^2} \qquad \text{radiation density parameter,} \tag{2.13}$$

$$\Omega_k = -\frac{kc^2}{H_0^2} \qquad \text{curvature parameter,} \tag{2.14}$$

$$\Omega_\Lambda = \frac{\Lambda c^2}{3H_0^2} \qquad \text{dark energy density parameter.} \tag{2.15}$$

$\rho_{\mathrm{m},0}$ and $\rho_{\mathrm{r},0}$ denote the present matter and radiation density in the Universe at time t_0. For general times the densities are

$$\rho_\mathrm{m} = \frac{\rho_{\mathrm{m},0}}{a^3} \quad ; \quad \rho_\mathrm{r} = \frac{\rho_{\mathrm{r},0}}{a^4} \quad . \tag{2.16}$$

The matter density evolves with $1/a^3$, which just describes the geometrical dilution of the particle number density due to the expanding volume. On the other hand, the radiation density evolves with $1/a^4$, because photons and relativistic particles suffer an additional energy loss due to the expansion of their wavelength, resulting in a redshift z,

$$1 + z = \frac{1}{a} \quad , \tag{2.17}$$

which allows one to interpret the observed Hubble redshift 2.3 as a direct consequence of cosmic expansion rather than a Doppler shift. The redshift z can therefore be used in the sense of a time scale, with $z = 0$ being the present time t_0.

With the cosmological parameters and the relations 2.16, equation 2.10 can be rewritten as

$$\left(\frac{H}{H_0}\right)^2 = \frac{\Omega_\mathrm{r}}{a^4} + \frac{\Omega_\mathrm{m}}{a^3} + \frac{\Omega_k}{a^2} + \Omega_\Lambda \quad . \tag{2.18}$$

The transition between the early, radiation-dominated era ($\Omega_\mathrm{m} < \Omega_\mathrm{r}$) and the subsequent matter-dominated era $\Omega_\mathrm{m} > \Omega_\mathrm{r}$ is marked by the redshift of equality, $z_\mathrm{eq} \approx 3000$ (Weinberg 2008). As a result of the different evolution of matter and radiation with the cosmic expansion (equation 2.16), the radiation density in the present Universe is very low compared to the matter density. Here, we are only interested in this matter-dominated epoch, at redshifts sufficiently smaller than z_eq, so for our purposes we can neglect Ω_r completely. Then, since $H/H_0 = 1$ at t_0, it follows from equation 2.18 that $\Omega_k = 1 - \Omega_\mathrm{m} - \Omega_\Lambda$.The time evolution of the scale factor a is then determined only by the two parameters Ω_m and Ω_Λ:

$$\dot{a} = \frac{\mathrm{d}a}{\mathrm{d}t} = H_0 \sqrt{\Omega_\mathrm{m} \left(\frac{1}{a} - 1\right) + \Omega_\Lambda \left(a^2 - 1\right) + 1} \quad . \tag{2.19}$$

We now have the desired expression for the time evolution of the scale factor, which is normalised with $a_0 = 1$ and $\dot{a}_0 = H_0$. We can now also compute the age of the Universe at a given scale factor a from the cosmological parameters by integrating over 2.19,

$$t(a) = \int_0^a \frac{\mathrm{d}a}{\dot{a}} \quad . \tag{2.20}$$

The current estimates of the cosmological parameters from the WMAP7 data (Komatsu et al. 2011) are $\Omega_\mathrm{m} = 0.272$ and $\Omega_\Lambda = 0.728$, with the age of the Universe being 13.78×10^9 years. Table 2.1 lists the different sets of parameters that we use at various places in this work. These values have several important implications. The fact that, within the current observational estimate, Ω_m and Ω_Λ exactly add up to 1 implies that we live in a flat Universe without a relevant curvature of space ($\Omega_k = 0$). This also means that the total matter-energy content of the Universe, $\Omega_\mathrm{m} + \Omega_\Lambda$, is very close to the critical density ρ_crit,

$$\rho_\mathrm{crit} = \frac{3H^2}{8\pi G} \quad , \tag{2.21}$$

which is the density of a Universe with $\Omega_k = 0$ and marks the boundary between a Universe that would eventually recollapse due to self-gravitation and a Universe that would continue to expand forever. The mean matter density of the Universe, $\bar{\rho}$, is then given by

$$\bar{\rho} = \Omega_\mathrm{m}\, \rho_\mathrm{crit} \quad . \tag{2.22}$$

This matter density is composed mainly of dark matter and baryonic matter, $\Omega_\mathrm{m} = \Omega_\mathrm{dm} + \Omega_\mathrm{b}$. The current estimate of the baryon density is $\Omega_\mathrm{b} = 0.0455$ (Komatsu et al. 2011), which gives a baryon fraction of $f_\mathrm{b} = \Omega_\mathrm{b}/\Omega_\mathrm{m} \approx 0.167$. Therefore, there is roughly five times more dark matter than baryonic matter in the Universe. This predominance of dark matter is well known from observational evidence (Oort 1930a,b,c; Zwicky 1933, 1937; Freeman 1970; Begeman 1989; Clowe et al. 2006), and it is one of the main puzzles of modern physics to explain its nature.

The high present value of the dark energy density Ω_Λ causes an accelerated expansion of the Universe: if the total equation of state of the Universe drops below $w = 1/3$, which it will with high enough Ω_Λ, it can be shown that $\ddot{a}$ becomes positive, i.e. the expansion of the Universe starts to accelerate. The discovery of this accelerated expansion was again made possible by galactic distance measurements: Riess et al. (1998) and Perlmutter et al. (1999) derived it from observations of distant type Ia supernovae. This decisively proved that we actually live in a phase transition between the matter-dominated and a dark-energy-dominated epoch, which started around $z \approx 1$.

	WMAP3 Spergel et al. (2007)	WMAP5 Komatsu et al. (2009)	WMAP7 Komatsu et al. (2011)
Ω_m	0.24	0.279	0.272
Ω_Λ	0.76	0.721	0.728
n_s	0.95	0.960	0.961
σ_8	0.75	0.817	0.807
h	0.73	0.70	0.702

Table 2.1: The different sets of cosmological parameters used in this work for the computation of LSS reconstructions, cosmological initial conditions, and numerical simulations (except for LSS reconstructions from the Cosmicflows-1 catalogue, where we use $h = 0.74$). The estimated error intervals of the cosmological parameters are not considered here; they are listed in the respective papers.

2.2.2 Comoving coordinates

Equipped with a physical description of the cosmic expansion from the beginning of the matter-dominated epoch until today, we now turn to a description of how the matter in the Universe is distributed.

We continue to use the ideal fluid approach that was already used earlier in the derivation of the Friedmann equations. This is a valid model until baryonic effects become important at high densities. Such a self-gravitating fluid is locally governed by the Euler equations and the Poisson equation. The set of equations is:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{v}) = 0 \quad , \tag{2.23}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \boldsymbol{\nabla})\boldsymbol{v} = -\frac{\boldsymbol{\nabla} p}{\rho} - \boldsymbol{\nabla}\phi \quad , \tag{2.24}$$

$$\boldsymbol{\nabla}^2 \phi = 4\pi G \rho \quad . \tag{2.25}$$

Matter and momentum conservation are expressed by the continuity equation (2.23) and the equation (2.24), respectively. The gravitational force $-\boldsymbol{\nabla}\phi$ acting on the fluid due to its gravitational potential ϕ is determined by the Poisson equation (2.25). The FLRW metric depends on the cosmic expansion only through the scale factor $a(t)$. It is therefore possible to choose a natural system of coordinates that are comoving with the cosmic expansion, such that an observer who perceives the Universe as isotropic will have a constant comoving position $\boldsymbol{x}$. This is achieved with the transformation

$$\boldsymbol{x} = \frac{\mathbf{1}}{a}\boldsymbol{r} \quad , \tag{2.26}$$

where $\boldsymbol{r}$ is the ordinary physical, i.e. non-comoving spatial position. Further, the comoving density is defined by (cf. equation 2.16),

$$\rho_x(\boldsymbol{x}) = a^3 \rho(\boldsymbol{r}) \quad , \tag{2.27}$$

so that in the absence of external forces other than the cosmic expansion, comoving distances and densities stay constant over time. One also defines the comoving or peculiar velocity $\boldsymbol{u}$,

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \frac{1}{a}\boldsymbol{u} \quad , \tag{2.28}$$

which is connected to the physical velocity $\boldsymbol{v} = \mathrm{d}\boldsymbol{r}/\mathrm{d}t$ through

$$\boldsymbol{u} = \boldsymbol{v} - H(t)\boldsymbol{r} \quad , \tag{2.29}$$

so that the peculiar velocity is obtained by subtracting the Hubble drag from the total physical velocity (cf. equation 2.8). If we now replace the partial time derivative $\partial/\partial t$ at constant position $\boldsymbol{r}$ with the one at constant comoving position $\boldsymbol{x}$, and the spatial derivative $\boldsymbol{\nabla}$ with the comoving spatial derivative $\boldsymbol{\nabla}_x$,

$$\left.\frac{\partial}{\partial t}\right|_r = \left.\frac{\partial}{\partial t}\right|_x - \frac{\dot{a}}{a}\boldsymbol{x}\cdot\boldsymbol{\nabla}_x \quad ; \quad \boldsymbol{\nabla}_x = \frac{1}{a}\cdot\boldsymbol{\nabla} \quad , \tag{2.30}$$

we can recast the set of equations 2.23 – 2.25 into comoving form,

$$\frac{\partial\rho_x}{\partial t} + \frac{1}{a}\boldsymbol{\nabla}_x\cdot(\rho_x\boldsymbol{u}) + \frac{3\dot{a}}{a}\rho_x = 0 \quad , \tag{2.31}$$

$$\frac{\partial\boldsymbol{u}}{\partial t} + \frac{1}{a}(\boldsymbol{u}\cdot\boldsymbol{\nabla}_x)\boldsymbol{u} + \frac{\dot{a}}{a}\boldsymbol{u} = -\frac{1}{a}\boldsymbol{\nabla}_x\phi \quad , \tag{2.32}$$

$$\boldsymbol{\nabla}_x^2\phi = 4\pi G a^2(\rho - \bar{\rho}) \quad . \tag{2.33}$$

This is a system of coupled non-linear differential equations. Unfortunately, it is not possible to analytically solve the system for the general case. However, it is possible to obtain a solution for the evolution of the density field in the limit of linear perturbation theory.

2.2.3 Linear theory of density perturbations

Today, the matter distribution in the Universe is very inhomogeneous: dense clumps of matter like galaxies are contrasted by the vast emptiness of space in the large cosmic voids. By contrast, the observed temperature fluctuations of the CMB are very small ($\Delta T/T \approx 10^{-5}$). Since the photons of the CMB originate from the moment of matter-radiation decoupling at $z \approx 1000$, we can infer that the early Universe had an almost homogeneous matter distribution with only very small density fluctuations, which build the seeds for the gravitational condensation of contemporary structure. A theory of structure formation has to be able to explain this transition.

It is useful to describe density fluctuations as a density contrast against the mean density $\bar{\rho}$ of the Universe, given by equations 2.21 – 2.22. The overdensity δ is defined by

$$\delta(\boldsymbol{r}, t) = \frac{\rho(\boldsymbol{r}, t) - \bar{\rho}(t)}{\bar{\rho}(t)} \quad . \tag{2.34}$$

The starting point for linear perturbation theory is the basic assumption that these density fluctuations are small, $\delta \ll 1$. This is valid either on very large scales, where the Universe appears almost homogeneous, or at very early redshift z, where the fluctuations are still small. If we then substitute δ for the density in equations 2.31 – 2.33 and neglect the pressure and all second-order terms $\propto \delta^2$, $\delta\boldsymbol{u}$, and $\boldsymbol{u}^2$, the set of equations becomes:

$$\frac{\partial\delta}{\partial t} + \frac{1}{a}\boldsymbol{\nabla}_x\cdot\boldsymbol{u} = 0 \quad , \tag{2.35}$$

$$\frac{\partial\boldsymbol{u}}{\partial t} + \frac{\dot{a}}{a}\boldsymbol{u} = -\frac{1}{a}\boldsymbol{\nabla}_x\phi_x \quad , \tag{2.36}$$

$$\boldsymbol{\nabla}^2\phi_x = \frac{3}{2}\Omega_\mathrm{m}\frac{\dot{a}}{a}\delta \quad , \tag{2.37}$$

where we also used 2.21 – 2.22 to remove the gravitational constant from the Poisson equation. In this simplified form, the equations can be linearised to obtain an analytical solution for the time evolution of δ. By combining the equations, $\boldsymbol{u}$ and ϕ can be eliminated, yielding a second-order differential equation for the density contrast (Bonnor 1957):

$$\ddot{\delta} + \frac{2\dot{a}}{a}\dot{\delta} = \frac{3}{2}\Omega_{\mathrm{m}}\frac{\dot{a}}{a}\delta \quad . \tag{2.38}$$

This linear second-order equation can be solved by separating the dependences on time and space,

$$\delta(\boldsymbol{x}, t) = D(t)\ \delta_0(\boldsymbol{x}) \quad , \tag{2.39}$$

meaning that in the linear approximation the initial shape $\delta_0(x)$ of the overdensity distribution remains fixed, and its amplitude scales with time proportional to the factor $D(t)$. This factor is then determined by the equation

$$\ddot{D} + \frac{2\dot{a}}{a}\dot{D} = \frac{3}{2}\Omega_{\mathrm{m}}\frac{\dot{a}}{a}D \quad . \tag{2.40}$$

This resembles the equation of a damped harmonic oscillator, with the gravity term as an excitational force, and the cosmological expansion factor as the dampening term. The analogy is somewhat flawed though, since both of these factors are themselves time-dependent in a non-trivial way. The general solution (Heath 1977) is a linear combination of a growing and a decaying mode, which are of the form (Peebles 1980):

$$D_+(a) \propto \frac{\dot{a}}{a}\int_0^a \frac{\mathrm{d}a}{\dot{a}^3} \quad ; \quad D_-(a) \propto \frac{\dot{a}}{a} \quad , \tag{2.41}$$

Since the very early Universe, $D_-(a)$ had sufficient time to decay towards zero, and we can safely assume that it could not have had a significant impact on structure formation. Therefore we can completely neglect the decaying mode for the evolution in the matter-dominated epoch that is studied here. The growth of the linear overdensity field is then given by the simple scaling relation

$$\delta(\boldsymbol{x}, t) = D_+(t)\ \delta_0(\boldsymbol{x}) \quad . \tag{2.42}$$

We use the convention that $D_+(t_0) = 1$ at present time t_0 (or $a = 1$), so that $\delta_0 = \delta(t_0)$ is the linear overdensity distribution extrapolated to present time. With this normalisation, one obtains the following expression for D_+, which from here on will be called the (linear) growth factor:

$$D_+(a) = \frac{1}{D_0}\frac{\dot{a}}{a}\int_0^a \frac{\mathrm{d}a}{\dot{a}^3} \quad ; \quad D_0 = \frac{1}{H_0}\int_0^1 \frac{\mathrm{d}a}{\dot{a}^3} \quad . \tag{2.43}$$

Since for any reasonable set of cosmological parameters D_+ is a smooth, monotonically increasing function of a, the integral can be easily evaluated numerically using equation 2.19.

Here, we considered the equations of linear perturbation theory for the Newtonian case, because this is the theoretical framework within which numerical simulations of structure formation are formulated. It is also possible to derive them in the general relativistic case. The full treatment is given in e.g. Weinberg (2008). Intuitively, relativistic effects such as the finite speed of light are not expected to be negligible on the large length scales that are probed by cosmological simulations (up to several hundred Mpc/h). One can however show that the linear growth of density perturbations in the Newtonian approximation is identical to the corresponding fully relativistic linear perturbation growth equation in the synchronous gauge (Chisari & Zaldarriaga 2011; Gnedin et al. 2011).

2.2.4 Power spectrum and correlation function

To analyse the statistical properties of the overdensity field $\delta(\boldsymbol{x})$, it is useful to consider its Fourier transform $\delta(\boldsymbol{k})$. Since $\delta(\boldsymbol{x})$ is a real-valued scalar field, its Fourier transform is Hermitian, $\delta(\boldsymbol{k}) = \delta^*(-\boldsymbol{k})$. For the definition of the Fourier transform, we follow the same convention that is used in Peacock (2007),

$$\delta(\boldsymbol{k}) = \int \delta(\boldsymbol{x}) e^{i\boldsymbol{k}\cdot\boldsymbol{x}} \mathrm{d}\boldsymbol{x} \quad ; \quad \delta(\boldsymbol{x}) = \frac{1}{(2\pi)^3} \int \delta(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \mathrm{d}\boldsymbol{k} \quad . \tag{2.44}$$

Inflation theory predicts that the distribution of primordial density fluctuations in the Universe should be a Gaussian random field (Kolb et al. 1990). Therefore, for any set of positions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the corresponding overdensities $\delta(\boldsymbol{x}_1), \ldots, \delta(\boldsymbol{x}_N)$ are correlated random variables, and their joint probability distribution function is a multivariate Gaussian. From the cosmological principle one concludes that this field must be homogeneous, i.e. its probability distribution function is translation invariant. For any homogeneous Gaussian random field, it follows (Bardeen et al. 1986) that the Fourier modes $\delta(\boldsymbol{k})$ are mutually independent, or in other words, uncorrelated:

$$\langle \delta(\boldsymbol{k}) \delta^*(\boldsymbol{k}') \rangle = \delta_D(\boldsymbol{k} + \boldsymbol{k}') \langle |\delta(\boldsymbol{k})|^2 \rangle \quad , \tag{2.45}$$

where δ_D is the Dirac delta distribution. Further, all Fourier modes will have random phases, and the probability distribution function of their amplitudes $\mathrm{P}(|\delta(\boldsymbol{k})|)$ is given by the Rayleigh distribution:

$$\mathrm{P}(|\delta(\boldsymbol{k})|)\, \mathrm{d}\boldsymbol{k} = \frac{|\delta(\boldsymbol{k})|^2}{P(\boldsymbol{k})} \exp\left(\frac{-|\delta(\boldsymbol{k})|^2}{2\, P(\boldsymbol{k})} \right) \mathrm{d}\boldsymbol{k} \quad , \tag{2.46}$$

where the power spectrum $P(\boldsymbol{k})$ is defined as the variance of the amplitude for a given Fourier mode $\boldsymbol{k}$,

$$P(\boldsymbol{k}) = \langle \delta(\boldsymbol{k}) \delta^*(\boldsymbol{k}) \rangle \; = \langle |\delta(\boldsymbol{k})|^2 \rangle \quad . \tag{2.47}$$

Moreover, a homogeneous Gaussian random field is *ergodic*, i.e. the average over an ensemble of different random field realisations is equivalent to the average over a sufficiently large volume of a single particular realisation (Adler 1981). This is an important property, since it allows us to make valid statistical statements about the only realisation that we are able to observe: our own Universe.

By definition (2.34), the overall mean overdensity is zero, i.e. for any position $\boldsymbol{x}$ and wavevector $\boldsymbol{k}$ the statistical *mean field* of a random realisation is given by

$$\langle \delta(\boldsymbol{x}) \rangle = \langle \delta(\boldsymbol{k}) \rangle = 0 \quad . \tag{2.48}$$

The properties 2.46 and 2.48 imply that the probability distribution function, and in fact any statistical property of $\delta(\boldsymbol{x})$, is fully determined by the power spectrum $P(k)$. One can also derive the two-point correlation function $\xi(\boldsymbol{x})$ of the overdensity field. The derivation yields that $\xi(\boldsymbol{x})$ is the Fourier transform of the power spectrum $P(k)$,

$$\xi(\boldsymbol{x}) = \langle \delta(\boldsymbol{x}') \delta(\boldsymbol{x}' + \boldsymbol{x}) \rangle = \frac{1}{(2\pi)^3} \int_0^\infty P(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \mathrm{d}\boldsymbol{k} \quad . \tag{2.49}$$

This property of random fields is not restricted to the Gaussian case and also valid for non-Gaussian random processes. In signal processing, it is also known as the Wiener-Khinchin

theorem (Wiener 1930; Khinchin 1934). It is however unique to Gaussian random fields that the knowledge of $\xi(\boldsymbol{x})$, just as $P(k)$, defines all statistical properties of the field. It then follows also that all higher moments, both in Fourier space (skewness, kurtosis, ...) and in real space (3-point, 4-point, ... correlation function) must vanish.

Besides homogeneity, the cosmological principle also requires that the Universe be isotropic. Therefore, the power spectrum and correlation function cannot depend on the direction, so that $P(\boldsymbol{k}) = P(k)$ and $\xi(\boldsymbol{x}) = \xi(x)$, with the comoving wavenumber $k = |\boldsymbol{k}|$ and comoving distance $x = |\boldsymbol{x}|$, respectively. If we then substitute $\mathrm{d}\boldsymbol{k} = k^2\,\mathrm{d}k\,\sin\vartheta\,\mathrm{d}\vartheta\,\mathrm{d}\varphi$ in equation 2.49, this allows to perform the integration over the angles, leaving

$$\xi(x) = \frac{1}{2\pi^2}\int_0^\infty k^2\, j_0(kx)\, P(k)\,\mathrm{d}k \quad , \tag{2.50}$$

with the spherical Bessel function $j_0(kx) = \sin(kx)/kx$. By setting $x = 0$ in equation 2.50, we can obtain the autocorrelation, i.e. the total variance of the overdensity field, from the power spectrum:

$$\sigma^2 = \langle|\delta|^2\rangle = \xi(0) = \frac{1}{2\pi^2}\int_0^\infty k^2\, P(k)\,\mathrm{d}k \quad . \tag{2.51}$$

In equation 2.51, we are integrating over the power of all wavelengths. Depending on the shape of the power spectrum $P(k)$, this integral may diverge. However, in cosmological applications one is most often not interested in the full statistical properties of δ down to arbitrarily small scales, but only down to some characteristic length scale R that depends on the application. It is therefore common to convolve, or "smooth" the field with some window function W. Since convolution in real space is equivalent to multiplication in Fourier space, the smoothed field δ_W is related to the original field δ via

$$\delta_W(\boldsymbol{x}) = \frac{1}{(2\pi)^3}\int \delta(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}}\, W(kR)\,\mathrm{d}\boldsymbol{k} \quad . \tag{2.52}$$

The total variance of such a smoothed overdensity field is then

$$(\sigma_W)^2 = \frac{1}{2\pi^2}\int_0^\infty k^2 P(k)\, W^2(kR)\,\mathrm{d}k \quad . \tag{2.53}$$

Popular choices for the smoothing kernel are

$$W_{\mathrm{G}}(kR) = e^{-(kR)^2/2} \qquad \text{Gaussian window,} \tag{2.54}$$

$$W_{\mathrm{sth}}(kR) = \frac{3j_1(kR)}{kR} \qquad \text{spherical top-hat window,} \tag{2.55}$$

with the spherical Bessel function $j_1(kx) = \sin(kx)/(kx)^2 - \cos(kx)/kx$. An important property of zero-mean Gaussian random fields states that any linear function of $\delta(\boldsymbol{x})$, such as the smoothed field $\delta_W(\boldsymbol{x})$, will be likewise a zero-mean Gaussian random field (Bardeen et al. 1986).

With this theoretical arnamentarium at hand, we can now consider the primordial density perturbations of the Universe. In the current standard model, they are believed to originate from quantum fluctuations around a thermal equilibrium and therefore consist of a Gaussian random field (Landau et al. 1980). The cosmic inflation then amplified them to macroscopic size. Inflation theory predicts that the primordial power spectrum produced by inflation should be scale-invariant, $P(k) \propto k^{n_{\mathrm{s}}}$ (Harrison-Zeldovich spectrum), with the primordial scale factor n_{s} close to 1 (Harrison 1970). The current estimate from WMAP7 data is $n_{\mathrm{s}} = 0.961$ (Komatsu

et al. 2011). Subsequently, the power spectrum undergoes changes through the various interaction processes between dark matter, baryons, and radiation during the transition from the radiation-dominated to the matter-dominated phase. These changes can be encoded in a transfer function $T(k)$, so that the resulting power spectrum can be described with

$$P(k) \propto T^2(k)\, k^{n_\mathrm{s}} \quad . \tag{2.56}$$

In the epoch of matter domination, which we consider here, the interacting components have decoupled; furthermore, the linear approximation implies that the different modes of the Gaussian field $\delta(k)$ evolve independently without interacting, so that the shape 2.56 of the power spectrum stays fixed in time. Its amplitude increases in time along with the growth factor, and from 2.42 and 2.47 it is obvious that

$$P(k,t) = D_+^2(t) P_0(k) \quad , \tag{2.57}$$

where $P_0(k)$ is the linear power spectrum extrapolated to present time $z = 0$.

To define the normalisation of $P_0(k)$, the additional parameter σ_8 was introduced. Historically, this stems from the observation that at present time, the variance (2.53) of the overdensity δ is approximately equal to 1 when smoothed with a spherical top-hat filter with $R = 8$ Mpc$/h$, so the normalisation parameter will likewise have the convenient property of being approximately equal to 1. It is defined by

$$\sigma_8 = \left[\frac{1}{2\pi^2} \int_0^\infty k^2\, P(k)\, W_{\mathrm{sth}}^2(kR)\, \mathrm{d}k \right]^{1/2} \quad \text{with} \quad R = 8\ \mathrm{Mpc}/h \quad . \tag{2.58}$$

The current estimate from WMAP7 data is $\sigma_8 = 0.807$ (Komatsu et al. 2011).

Fitting formulae for $T(k)$ for different cosmological models can be found e.g. in Bardeen et al. (1986); Eisenstein & Hu (1998). In practice, $P_0(k)$ (including the transfer function) can be computed in tabulated form from cosmological parameters using numerical codes such as CMBFAST (Seljak & Zaldarriaga 1996), CAMB (Lewis et al. 2000), and ICOSMO (Refregier et al. 2011). The shape of this theoretical power spectrum is in good agreement with the observed angular power spectrum of the CMB and other observational probes (Tegmark & Zaldarriaga 2002), proving the validity of the theory in the linear phase of cosmic structure formation. Figure 2.4 shows $P_0(k)$ computed with the CAMB code from the current WMAP7 cosmological parameters. Figure 2.5 shows the associated two-point correlation function, plotted in logarithmic scale (left) and a magnification in linear scale (right). The prominent peak at 100 Mpc$/h$ corresponds to the baryon acoustic oscillations (BAO), induced by a clustering of baryonic matter at this length scale due to acoustic waves which propagated in the early universe. The same feature can be seen in the form of wiggles at the low-k end of the power spectrum. The BAO peak can be used as a cosmological standard ruler at these very large scales (Eisenstein & Hu 1998).

2.2.5 The linear peculiar velocity field

Since the focus of this work is the cosmological peculiar velocity field, it is particularly important to describe it alongside the overdensity in the context of linear perturbation theory. For overdensities, the linear approximation is valid only for the early structure formation. The condition of small perturbations will invariably be violated as soon as dense structures emerge. On the other hand, as we will see later, for velocity fields the linear approximation is much more useful even up to present time and will be heavily used in the remaining chapters.

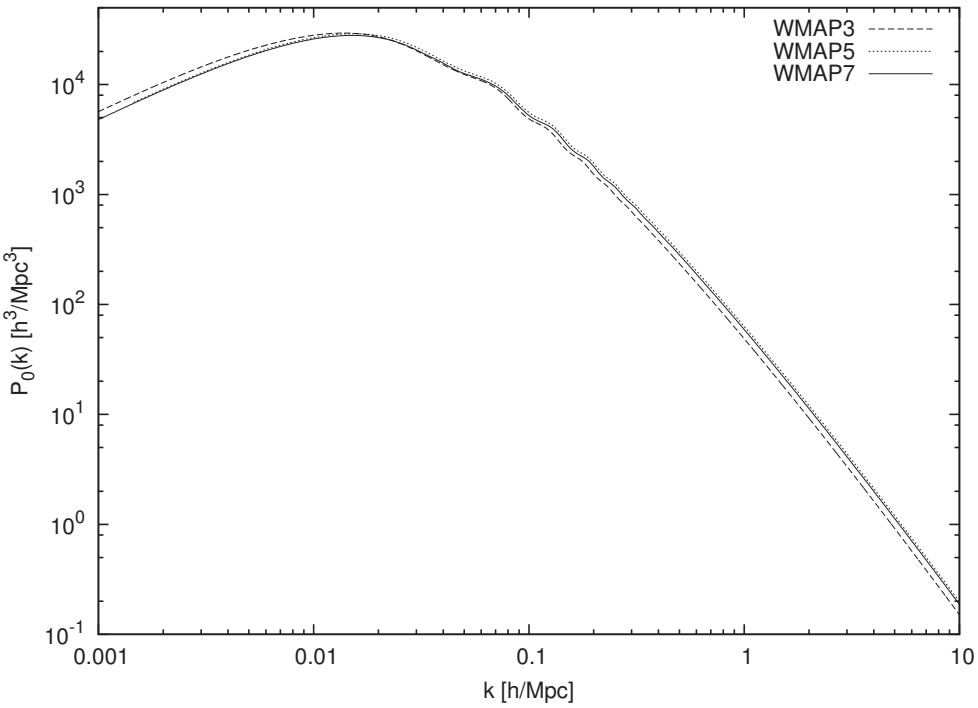

Figure 2.4: Linear matter power spectrum $P_0(k)$ at $z = 0$ for three different sets of ΛCDM parameters, WMAP3, WMAP5, and WMAP7 (see Table 2.1), normalised to $\sigma_8 = 0.75$ (WMAP3), 0.817 (WMAP5), and 0.807 (WMAP7).

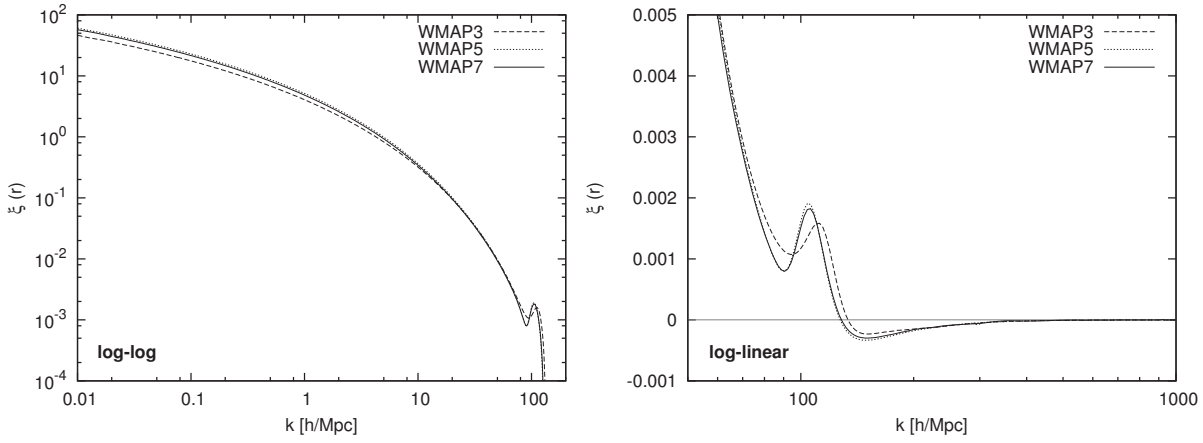

Figure 2.5: Linear two-point matter correlation function for three different sets of ΛCDM parameters. Shown is a full plot in logarithmic scale (left) and a cutout from the region around the BAO peak in linear scale (right).

We continue to use the comoving coordinates, so that the comoving velocity is $\boldsymbol{u} = \mathrm{d}\boldsymbol{x}/\mathrm{d}t$. At $z = 0$ (present time), $\boldsymbol{x}$ is equal to the position $\boldsymbol{r}$, and $\boldsymbol{u}$ to the peculiar velocity $\boldsymbol{v}_{\mathrm{pec}} = \boldsymbol{v} - H_0\boldsymbol{r}$. Whenever we use $\boldsymbol{r}$ instead of $\boldsymbol{x}$, and v_{pec} instead of $\boldsymbol{u}$ later, we mean the special case of $z = 0$ for the following equations.

To obtain a solution for the comoving peculiar velocity $\boldsymbol{u}$, the solution for the overdensity 2.42 can be combined with the continuity equation 2.35, resulting in

$$\boldsymbol{\nabla}_x \cdot \boldsymbol{u} = -a\dot{D}\delta_0 = -a\dot{D}D\delta = -a\frac{\mathrm{d}D}{\mathrm{d}a}\dot{a}\delta = -\frac{\mathrm{d}(\ln D)}{\mathrm{d}(\ln a)}\dot{a}\delta \quad . \tag{2.59}$$

The cosmological growth rate f is defined as

$$f(a) = \frac{\mathrm{d}(\ln D)}{\mathrm{d}(\ln a)} \quad , \tag{2.60}$$

which describes the growth of the peculiar velocity field through the growth of the overdensity growth factor D relative to the scale factor a. Different approximation formulae for f exist, such as the simple $f(t_0) \approx \Omega_\mathrm{m}^{0.6}$ (Peebles 1980), or with a slightly different exponent (Lightman & Schechter 1990), also with carrying higher-order terms involving Ω_Λ (Lahav et al. 1991). These approximations are valid to a varying degree depending on the cosmological parameters (Hamilton 2001) and formulated mostly for t_0 ($a = 1$). In this work, we use the exact expression instead, with f as a function of a and hence usable within our comoving description. It can be derived from the equations above and evaluated by numerical integration:

$$f(a) = \frac{1}{\dot{a}^2}\left(\Omega_\Lambda a^2 - \frac{\Omega_\mathrm{m}}{2a}\right) + a\left(\dot{a}^3 \int_0^a \frac{\mathrm{d}a}{\dot{a}^3}\right)^{-1} - 1 \quad . \tag{2.61}$$

With this growth rate f, the equation for comoving velocity becomes

$$\boldsymbol{\nabla}_x \cdot \boldsymbol{u}(\boldsymbol{x}) = -\dot{a}f\delta(\boldsymbol{x}) \quad ; \quad \boldsymbol{u}(\boldsymbol{x}) = \frac{\dot{a}f}{4\pi}\int_V \delta(\boldsymbol{x}')\frac{\boldsymbol{x}-\boldsymbol{x}'}{|\boldsymbol{x}-\boldsymbol{x}'|^3}\mathrm{d}\boldsymbol{x}' \quad . \tag{2.62}$$

The Fourier-space equivalent of equation 2.62 is

$$i\boldsymbol{k}\cdot\boldsymbol{u}(\boldsymbol{k}) = -\dot{a}f\delta(\boldsymbol{k}) \quad ; \quad \boldsymbol{u}(\boldsymbol{k}) = i\,\frac{\boldsymbol{k}}{k^2}\,\dot{a}f\delta(\boldsymbol{k}) \quad . \tag{2.63}$$

This means that, in the limit of linear perturbation theory, the velocity and density fields are completely determined by each other at any given moment in time, without the need to explicitly solve the continuity equation. Because of equation 2.63, if δ is Gaussian distributed, then so must be each component u_x, u_y, u_z of the velocity field. Due to isotropy and homogeneity, each component must also have mean zero (the Universe has no net velocity in any preferred direction), which means that the absolute velocity $|\boldsymbol{u}|$ will be Maxwell-Boltzmann distributed[1]. Further, the power spectrum of each component of $\boldsymbol{u}$ receives an additional factor of $1/k^2$ compared to $P(k)$ of the overdensity. This means that the velocity field will be influenced much less by small scales (high k), and instead have a larger correlation length with more structure on large scales (low k) compared to the overdensity field. In the following, we further investigate this important fact.

Similar to the two-point correlation function of the overdensity field 2.50, one can construct the two-point correlation function of two comoving velocity vectors (which will be a 3×3 tensor), as well as the two-point correlation function between the overdensity and one velocity vector

[1] Due to the high correlation length of the linear peculiar velocity field, this is in general only true if one considers a large enough volume with a radius of the order of at least a few hundred Mpc/h, cf. Section 3.1.3.

(which will be a vector). One can combine equation 2.49 with the additional factors from equation 2.63 to form expressions for these correlations:

$$\langle \delta(\boldsymbol{x}') u_\alpha(\boldsymbol{x}'+\boldsymbol{x})\rangle = \frac{(\dot{a}f)^2}{(2\pi)^3}\int_0^\infty \frac{ik_\alpha}{k^2} P(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \mathrm{d}\boldsymbol{k} \quad , \tag{2.64}$$

$$\langle u_\alpha(\boldsymbol{x}') u_\beta(\boldsymbol{x}'+\boldsymbol{x})\rangle = \frac{\dot{a}f}{(2\pi)^3}\int_0^\infty \frac{k_\alpha k_\beta}{k^4} P(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \mathrm{d}\boldsymbol{k} \quad , \tag{2.65}$$

where $\alpha, \beta \in \{x, y, z\}$ index the three cartesian components. Because of isotropy, one can again reduce these to functions depending on the comoving distance $x = |\boldsymbol{x}|$ only. The calculation is somewhat more complicated due to the presence of the different cartesian components, and yields (Monin & Yaglom 1965; Gorski 1988; Zaroubi et al. 1999):

$$\langle \delta(\boldsymbol{x}') \boldsymbol{u}(\boldsymbol{x}'+\boldsymbol{x})\rangle_\alpha = -\dot{a}f\, \hat{x}_\alpha\, \zeta(x) \tag{2.66}$$

for the density-velocity correlation vector and

$$\langle \boldsymbol{u}(\boldsymbol{x}') \boldsymbol{u}(\boldsymbol{x}'+\boldsymbol{x})\rangle_{\alpha\beta} = (\dot{a}f)^2 \Psi_{\alpha\beta} \tag{2.67}$$

for the velocity-velocity correlation tensor, with

$$\Psi_{\alpha\beta} = \left\{ \psi_\mathrm{T}(x)\delta^K_{\alpha\beta} + [\psi_\mathrm{R}(x) - \psi_\mathrm{T}(x)]\, \hat{x}_\alpha \hat{x}_\beta \right\} \quad . \tag{2.68}$$

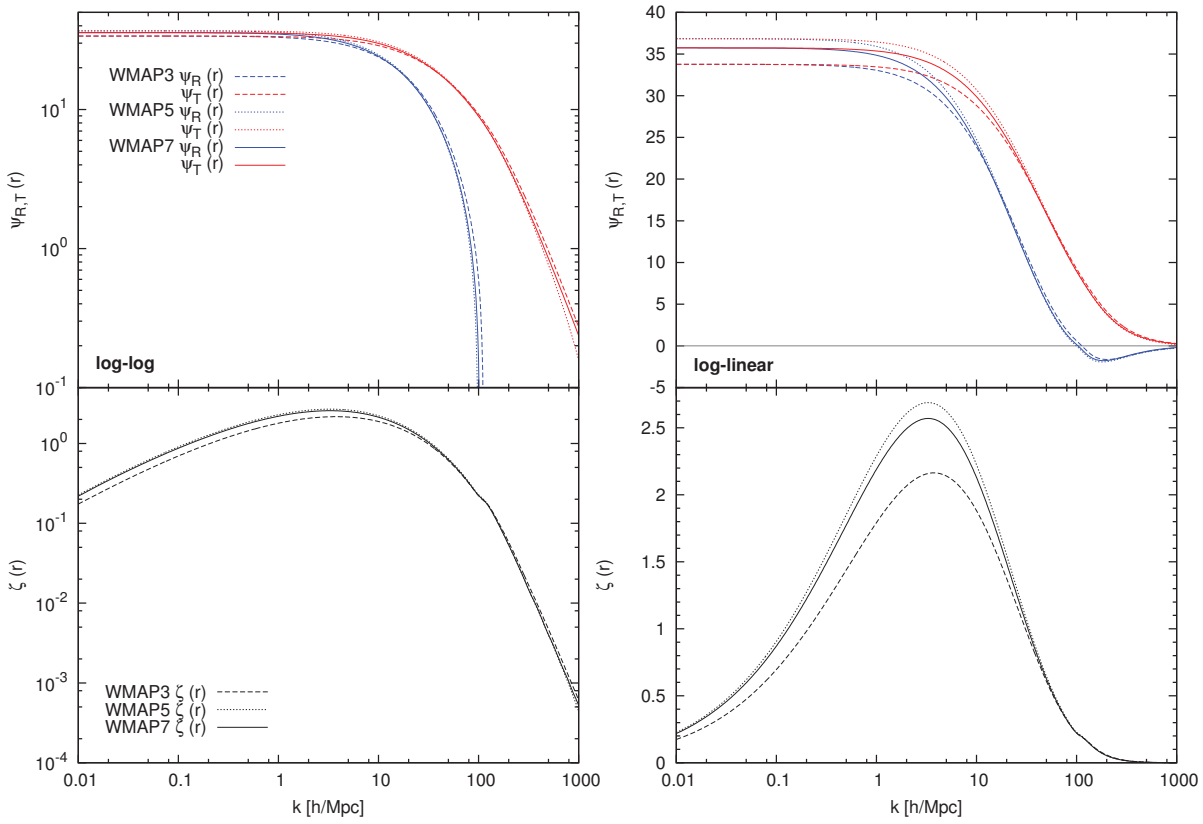

Figure 2.6: Linear two-point radial and transverse velocity-velocity (top row) and density-velocity (bottom row) correlation functions for three different sets of ΛCDM parameters, shown in logarithmic scale (left) and linear scale (right).

Here, $\delta^K_{\alpha\beta}$ is the Kronecker delta and $\hat{\boldsymbol{x}} = \boldsymbol{x}/x$ is the normalised distance vector. Then, $\hat{x}_\alpha$ is one cartesian component of this normalised vector, with $\alpha, \beta \in \{x, y, z\}$. The underlying one-dimensional functions are of the form:

$$\psi_{\mathrm{R}}(x) = \frac{1}{2\pi^2} \int_0^\infty \left[j_0(kx) - \frac{2j_1(kx)}{kx} \right] P(k)\mathrm{d}k \quad , \tag{2.69}$$

$$\psi_{\mathrm{T}}(x) = \frac{1}{2\pi^2} \int_0^\infty \frac{j_1(kx)}{kx} P(k)\mathrm{d}k \quad , \tag{2.70}$$

$$\zeta(x) = \frac{1}{2\pi^2} \int_0^\infty k j_1(kx) P(k)\mathrm{d}k \quad . \tag{2.71}$$

In the following we call ψ_{R}, ψ_{T}, and ζ radial velocity correlation function, transverse velocity correlation function, and density-velocity correlation function, respectively.

Figure 2.6 shows a plot of the velocity-velocity and density-velocity correlation functions for ΛCDM parameters. Considering the velocity-velocity correlation functions, it becomes evident that velocities are correlated to much larger distances than overdensities and feature much less information on small scales. Velocities are still correlated to 90% at a distance of 6 Mpc/h, to 50% at a distance of 40 Mpc/h, and to 10% at a distance of 200 Mpc/h, with still noticeable correlations out to even larger distances. This property of the linear peculiar velocity field will be of great importance in the following chapters.

The variance of the linear velocity field (per cartesian component α) is

$$\sigma^2(u_\alpha) = \langle |u_\alpha|^2 \rangle = (\dot{a}f)^2 \psi_{\mathrm{R,T}}(0) = \frac{\dot{a}^2 f^2}{2\pi^2} \int_0^\infty P(k)\,\mathrm{d}k \quad , \tag{2.72}$$

which corresponds to a peculiar velocity of about 300 km/s at $z = 0$ for a WMAP7 cosmology. This is not too different from the observed variance of galaxy peculiar velocities, and a first hint that the linear approximation may be useful for peculiar velocities at present time.

2.2.6 The Zeldovich approximation

The assumption of a "frozen" density distribution made by linear perturbation theory, where only the overall amplitude grows in time, is a sufficiently valid description for times and scale lengths where the perturbations are small, $\delta \ll 1$. However, when the perturbations grow to a sufficiently large amplitude, the dynamical motion of matter induced by its gravitational potential causes a displacement of the density distribution that cannot be neglected anymore. A more accurate description of the system can be formulated in the framework of Lagrangian perturbation theory. We can assign a Lagrangian coordinate $\boldsymbol{q}$ to each moving patch of matter, which corresponds to its unperturbed position. Its Eulerian coordinate $\boldsymbol{x}$, which corresponds to the physical position of the patch in configuration space, is then

$$\boldsymbol{x}(t) = \boldsymbol{q}(\boldsymbol{x}) + \boldsymbol{\psi}(\boldsymbol{x}, t) \quad , \tag{2.73}$$

where $\psi(\boldsymbol{x}, t)$ is the displacement from the initial state $\boldsymbol{x} = \boldsymbol{q}$. To describe the time evolution of this displacement, Zeldovich (1970) proposed the very useful approximation,

$$\boldsymbol{x}(t) = \boldsymbol{q}(\boldsymbol{x}) + \boldsymbol{D}_+(t)\boldsymbol{\psi}_0(\boldsymbol{x}) \quad , \tag{2.74}$$

which is the first-order solution of Lagrangian perturbation theory, neglecting the acceleration term. Then, the direction of the displacement vector stays constant in time and its amplitude

evolves with the growth factor D_+. The equation for the peculiar velocity $\boldsymbol{u} = a\,\mathrm{d}\boldsymbol{x}/\mathrm{d}t$ can be obtained by taking the time derivative of equation 2.74,

$$\boldsymbol{u}(\boldsymbol{x},t) = \dot{a} f \boldsymbol{\psi}(\boldsymbol{x},t) \quad , \tag{2.75}$$

which means that each patch continues to move with a constant peculiar velocity that is proportional to the displacement field. Using the mass conservation in Lagrangian coordinates, $\rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \bar{\rho}(\boldsymbol{q})\,\mathrm{d}\boldsymbol{q}$, one can now obtain a relation for the overdensity δ, which is the Lagrangian extension of the simple linear theory equation 2.42,

$$\begin{aligned} \delta(\boldsymbol{x}) &= \left|\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}\boldsymbol{x}}\right| - 1 \\ &= \left|\mathbf{I} - \frac{\mathrm{d}\boldsymbol{\psi}}{\mathrm{d}\boldsymbol{x}}\right| - 1 \\ &= \left|\mathbf{I} - D(t)\frac{\mathrm{d}\boldsymbol{\psi}_0}{\mathrm{d}\boldsymbol{x}}\right| - 1 \end{aligned} \tag{2.76}$$

(e.g. Nusser et al. 1991), where $\mathbf{I}$ is the unit matrix. This connects the overdensity to the deformation tensor $\mathrm{d}\boldsymbol{\psi}/\mathrm{d}\boldsymbol{x}$. The overdensity becomes positive, if the medium is contracting relative to the comoving frame ($\mathrm{d}\boldsymbol{\psi}/\mathrm{d}\boldsymbol{x} < 1$), and negative if it is expanding ($\mathrm{d}\boldsymbol{\psi}/\mathrm{d}\boldsymbol{x} > 1$), while the deformation grows in time with the linear growth factor.

The Zeldovich approximation is a very powerful tool to analytically describe structure formation beyond the linear regime. Let us consider a single Fourier mode $\delta(\boldsymbol{k})$ of the full overdensity field, which corresponds to a one-dimensional sinusoidal primordial density perturbation $\delta(z) \propto \cos(\boldsymbol{k}\cdot\boldsymbol{x})$ at some early redshift z which is still in the linear regime. With the Zeldovich approximation, it is possible to obtain an analytical solution[2] for the gravitational collapse of this perturbation (Zeldovich 1970),

$$\boldsymbol{x}(z) = \boldsymbol{q} + \left(\frac{1+z_\mathrm{c}}{1+z}\right)\frac{\boldsymbol{k}}{k^2}\sin(\boldsymbol{k}\cdot\boldsymbol{q}) \quad , \tag{2.77}$$

$$\boldsymbol{v}(z) = \dot{a} f \left(\frac{1+z_\mathrm{c}}{\sqrt{1+z}}\right)\frac{\boldsymbol{k}}{k^2}\sin(\boldsymbol{k}\cdot\boldsymbol{q}) \quad . \tag{2.78}$$

The time evolution of this single-mode perturbation is illustrated in Figure 2.7 (numerical simulation). It can be shown that the Zeldovich solution is exact up to redshift z_c (e.g. Shandarin & Zeldovich 1989), which is the moment of caustic formation. The value of z_c depends only on the amplitude of the initial perturbation. At z_c, the deformation tensor becomes infinite, which results in a singularity of the density distribution. The resulting structure is commonly referred to as the Zeldovich pancake. At the caustic, the two matter streams from both sides collide and eventually cross each other. After this shell crossing, the Zeldovich approximation is not valid any longer, since it predicts that the two streams would just pass through each other and continue to move with constant velocity. The actual behaviour is that the Zeldovich pancake forms an equilibrium between its own gravity and its internal velocity dispersion. The resulting structure is a stable density concentration with the matter oscillating around its peak. This leads to the formation of additional, secondary density peaks. The time evolution of this non-linear system cannot be described analytically, and a solution can be obtained only by numerical simulation.

[2] The amplitude factors in this solution are only analytically correct for an Einstein-de-Sitter Universe, $\Omega_\mathrm{m} = 1$, $\Omega_\Lambda = 0$. However, for other cosmologies the results are only slightly different. At high z, the difference for cosmologies with non-zero Λ is negligible.

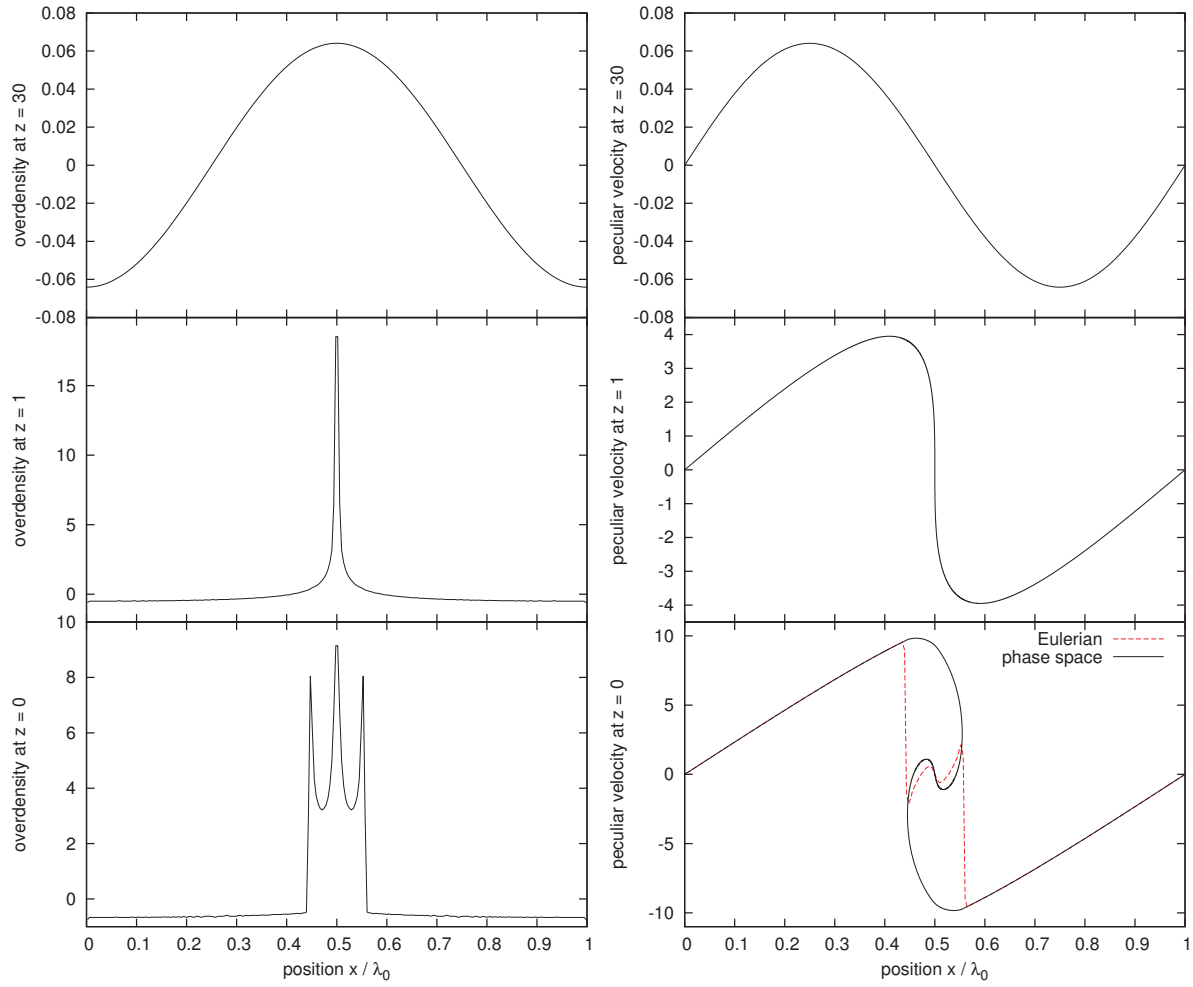

Figure 2.7: Gravitational collapse of a one-dimensional Zeldovich wave with collapse time $z_c = 1$; numerical simulation performed with the AMIGA code (Doumler & Knebe 2010). Overdensity field (left) and peculiar velocity field (right) in the linear regime ($z = 30$, top), at the moment of collapse ($z = 1$, middle), and in the non-linear phase after shell-crossing ($z = 0$, bottom). The peculiar velocity field in the non-linear phase shows multistreaming; the average velocity at any position (dashed red line) does not coincide with the actual velocities of the different particle streams (solid black line). This setup does not depend on the wavelength Λ_0 of the initial perturbation, so the position x is shown in units of Λ_0.

The multistreaming nature of the distribution after shell crossing breaks the mapping between Lagrangian coordinates $\boldsymbol{q}$ and Eulerian coordinates $\boldsymbol{x}$. From the position $\boldsymbol{x}$ and velocity $\boldsymbol{u}$ of a particle in the non-linear state after shell crossing it is not possible anymore to obtain an estimate of its initial position $\boldsymbol{q}$. This means that the non-linear evolution of such density peaks erases information about its initial conditions. This aspect of non-linear structure formation will become very important when we come to the problem of reconstructing initial conditions in Chapter 4.

2.2.7 Hierarchical structure formation

In the "quasi-linear regime" where the Zeldovich approximation is valid, the peculiar velocity field can be reasonably approximated as a curl-free potential flow (in the linear regime, this is

strictly true: $\boldsymbol{u} \propto -\boldsymbol{\nabla}\phi$ from equations 2.62 and 2.37). The deformation tensor $\mathrm{d}\boldsymbol{\psi}/\mathrm{d}\boldsymbol{x}$ is then symmetric and can be diagonalised with eigenvalues $\mu_1(\boldsymbol{x}) \geq \mu_2(\boldsymbol{x}) \geq \mu_3(\boldsymbol{x})$, which all grow proportional to $D_+(z)$. The overdensity is then

$$\delta(\boldsymbol{x}) = \{[1-\mu_1(\boldsymbol{x})]\,[(1-\mu_2(\boldsymbol{x})]\,[(1-\mu_3(\boldsymbol{x})]\} - 1 \quad , \tag{2.79}$$

which, for a single three-dimensional perturbation, is equivalent to a superposition of three orthogonal Zeldovich waves with different amplitudes, aligned along its fundamental axes. When the first eigenvalue μ_1 becomes infinite, the three-dimensional perturbation will collapse into a two-dimensional sheet. The next singularity at μ_2 leads to the formation of a one-dimensional filament; finally, the collapse along the third dimension leads to the formation of a halo. This model gives an explanation how the web-like appearance of large-scale structure in the Universe, a network of sheets and filaments with dense haloes at their intersections, could evolve from initial conditions that lack any geometrical features.

This collapse model of large-scale density perturbations suggests a "top-down" scenario of structure formation: The largest known objects, on the scales of galaxy clusters, could be formed first from the gravitational collapse of large-scale Zeldovich waves. Substructure would then emerge in the non-linear phase from the fragmentation of these objects (Bode et al. 2001). The assumption that large-scale waves would dominate the primordial power spectrum $P(k)$ is closely tied to the hot dark matter (HDM) and warm dark matter (WDM) models, which assume that dark matter particles have large thermal velocities and cool down to the non-relativistic, matter-dominated phase at relatively late times. This introduces an exponential cutoff at small scales in the transfer function $T(k)$. The current cold dark matter (CDM) paradigm favours instead a transfer function without this suppression of small-scale perturbations. The scale of this damping mostly depends on the assumed properties of the dark matter particles. The CDM model leads to the "bottom-up" scenario of structure formation: small scales collapse first, forming mini-haloes that host the first stars. Larger objects on galaxy and eventually cluster scales are subsequently formed by merging of smaller structure and matter accretion (White & Frenk 1991).

Despite its simplicity, the Zeldovich approximation describes the general large-scale texture of the density field quite well. One can construct a realisation of the primordial Gaussian random field, i.e. a set of cosmological initial conditions, sampled by particles at discrete Lagrangian positions $\boldsymbol{q}$, and advance these particles forward in time using the Zeldovich approximation. This will produce a surprisingly good prediction about the position and orientation of haloes, filaments, and voids in the evolved stage (Pauls & Melott 1995). The density field created by this particle distribution will be accurate when smoothed on a sufficiently large scale (Bouchet et al. 1995). For a ΛCDM Universe, this scale is about 5 Mpc/h at $z = 0$, although this varies locally and one has to go to larger smoothing scales in regions with high overdensity where shell-crossing occurs earlier. Of course, when going to smaller scales, the approximation fails to describe the inner structure of haloes, filaments, and sheets where shell-crossing has occured. Due to its nature of assuming a linear flow with constant velocities, the Zeldovich approximation is not able to describe any of the processes like merging, accretion, and the inner dynamics of virialised objects, which are driven by the local gravitational interaction of matter; all these processes are dominated by non-linear dynamics happening after shell crossing. The impossibility to follow non-linear gravitational dynamics analytically persists with all extensions of the Zeldovich approximation, such as second-order and higher-order Lagrangian perturbation theory (Buchert et al. 1994; Bouchet et al. 1995; Bernardeau et al. 2002), or the exclusion of path crossing by introducing artificial adhesion or viscosity (Gurbatov & Saichev 1984), which leads to Burgers' equation (Burgers 1973). In order to study structure formation down to galactic scales, one

has to resort to solving the time-evolution of the system numerically by means of cosmological simulations.

2.3 Cosmological N-body simulations

The basic idea of cosmological simulations is to numerically solve for the time evolution of the dynamical quantities of the Universe's matter components. Neglecting the baryonic component, these will be the density and velocity fields of collisionless matter. These fields are considered inside a finite computational volume that is statistically representative for the whole Universe. The standard choice for the computational domain is a three-dimensional, cubic box with a boxsize L that is large enough so that the Universe can be approximated as being homogeneous on scales larger than this boxsize, typically of the order of 100 Mpc/h or larger. To reflect the infinite, homogeneous and isotropic geometry of the Universe, the box is taken to have periodic boundary conditions, so that all positions inside the box are statistically equivalent. This setup has several advantages that make it ideal for a numerical treatment.

The starting point of a cosmological simulation is to construct the initial conditions, i.e. a random realisation of the primordial density fluctuations at some intial redshift $z_{\rm init}$. Since $z_{\rm init}$ must lie in the linear phase of the matter-dominated epoch, this is relatively straightforward, since the Gaussian probability distribution functions and the primordial power spectrum $P(k)$ are known for this case. Typical values are $100 \leq z_{\rm init} \leq 30$, depending on the chosen boxsize and resolution. This distribution is then evaluated forward in time until the desired final redshift, for example $z = 0$, which corresponds to today's time t_0, in order to follow structure formation into the current evolutionary epoch. However, simulations are also a very popular tool to study the Universe at high redshift, in the epoch of reionisation and the formation of first stars (e.g. Iliev et al. 2011; Romano-Diaz et al. 2011).

Large-scale structure formation is dominated by the dark matter component, which is assumed to be composed of some type of collisionless particles. This means that, on all scales of interest, they interact only with the gravitational field of the Universe. The mean free path of dark matter particles is assumed to be many orders of magnitude higher than the boxsizes of interest (e.g. in Bertone 2010), so that two-particle interactions can be safely neglected. The standard approach is to sample the distribution of dark matter with an ensemble of discrete particles that obey the collisionless, gravitational behaviour. Every such simulation particle corresponds to a very large number of physical dark matter particles. The numerical treatment of this particle ensemble is then equivalent to the classical N-body problem, with the added cosmological expansion which is determined by equation 2.19 and formulated in comoving coordinates. A large deal of additional complexity arises if one wishes to model the baryonic component as well, which we neglect here.

2.3.1 The N-body method

For an accurate modelling of the dynamical evolution of collisionless dark matter, the Euler equations 2.31 – 2.33 do not provide a sufficient description. In Eulerian coordinates, i.e. any fixed discretisation of the three-dimensional Euclidean space, the multi-streaming behaviour of collisionless dark matter particles cannot be treated properly, a problem that is already apparent for the non-linear phase of the Zeldovich pancake (bottom right panel of Figure 2.7). A complete description can only be obtained by considering the whole phase space. With the phase-space density $f(\boldsymbol{r}, \boldsymbol{v}, t)\mathrm{d}\boldsymbol{r}\mathrm{d}t$, the dynamical evolution of a system of gravitationally interacting collisionless matter is defined by the collisionless Boltzmann equation (here in physical coordinates):

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} f - (\boldsymbol{\nabla}\phi)\frac{\partial f}{\partial \boldsymbol{v}} = 0 \quad , \tag{2.80}$$

where the gravitational potential ϕ is determined by Poisson's equation (2.25). Unfortunately, the collisionless Boltzmann equation is in practice impossible to solve numerically. However, the equation can be reformulated with Hamiltonian dynamics using Poisson bracket notation (Leeuwin et al. 1993),

$$\frac{\partial f}{\partial t} + \{f, H\} = 0 \quad ; \quad H = \frac{1}{2}v^2 + \phi(\boldsymbol{r}) \quad . \tag{2.81}$$

The existence of a Hamiltonian H implies that, according to the Liouville theorem, the phase-space distribution function f stays constant along any trajectory through phase space. This means that if one can obtain a discrete, uniform sampling of the distribution function $f(\boldsymbol{r}, \boldsymbol{v}, t_0)$ at some initial time t_0 and solve the equations of motion for the trajectory of each sample, the solution will continue to be a uniform sampling of the phase-space distribution function f at all times.

The N-body method accomplishes this via a sampling of the density and velocity fields with a set of discrete particles, each having a definite position $\boldsymbol{x}_i$, velocity $\boldsymbol{v}_i$ and mass m_i. First we need to obtain a particle sampling of the initial conditions generated at $z_{\rm init}$. The task of constructing such initial conditions will be discussed in Chapter 3. Solving for the time evolution of the system then reduces to solving the Newtonian equations of motion for each particle i, in comoving coordinates

$$\frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t} = \frac{1}{a}\boldsymbol{v}_i \quad , \tag{2.82}$$

$$\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = -\frac{1}{a}\boldsymbol{\nabla}\phi(\boldsymbol{x}_i) \quad , \tag{2.83}$$

forwards in time until $z = 0$. This is accomplished by integrating them with discrete time steps using symplectic time integrators (e.g. Zemp et al. 2007). The gravitational force on each particle is determined by

$$-\boldsymbol{\nabla}\phi(\boldsymbol{x}_i) = G\sum_{j\neq i} m_j \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{|\boldsymbol{x}_j - \boldsymbol{x}_i|^3} \quad , \tag{2.84}$$

which would have to be evaluated for every particle and timestep. These sums are very expensive to compute for a large set of millions or even billions of particles. Several approaches have been developed as an alternative to direct summation. The tree method (Barnes & Hut 1986) arranges particles in a hierarchy of groups, and when the force on a particular particle is computed, the force exerted by distant groups is approximated by their lowest multipole moments (Appel 1985; Springel et al. 2001). Particle-mesh (PM) codes instead solve the Poisson equation on a discretised mesh, which for a regular cubic grid can be done very fast by using the Fast Fourier Transform. The required density field can be obtained by a mapping of the particle density to the grid using mass assignment schemes (Hockney & Eastwood 1992). The force at each particle location can then be obtained by grid interpolation. The finite resolution of the regular grid can be refined in regions with high particle density by either carrying out direct summation locally ($\mathrm{P^3M}$ codes, Efstathiou et al. 1985), or by adaptive mesh refinement (AMR, Berger & Colella 1989). With all these approaches, many numerical subtleties have to be considered in practice, which we will not discuss here; see e.g. the review article by Dehnen & Read (2011).

Popular numerical codes used for cosmological simulations are ART (Kravtsov et al. 1997), ENZO (Bryan & Norman 1997), and RAMSES (Teyssier 2002), who are designed around AMR, and GADGET (Springel et al. 2001; Springel 2005), which is a tree code. We do not go into the details of how these codes treat the baryonic component (see e.g. Tasker et al. 2008) and execute them in dark-matter-only mode. The N-body dark matter particle solvers of all these codes give essentially the same results down to the resolution limit of the chosen sampling, and the numerical differences are negligible (see Heitmann et al. 2005, 2008). We verified this by comparing dark-matter-only runs of the same initial conditions until $z = 0$ (BOX160, discussed below) with RAMSES, ART, and GADGET, and found no significant differences in the properties of the resulting structure, such as the statistics of the evolved fields and the positions and peculiar velocities of the evolved dark matter haloes at $z = 0$. We then chose GADGET as the code of choice for all simulations presented in this work because of its fast performance and ease of use.

2.3.2 Structure formation in simulations

As an example for a cosmological simulation, we consider here the BOX160 simulation, a constrained simulation of the Local Universe conducted within the CLUES project. This simulation will be used as a reference simulation in Chapters 4 and 5, but here we will focus on the general statistics of the evolved non-linear density and velocity fields. The BOX160 simulation was set up with the WMAP3 cosmological parameters $\Omega_\mathrm{m} = 0.24$, $\Omega_\Lambda = 0.76$, $\sigma_8 = 0.75$ (cf. Table 2.1), and a ΛCDM input power spectrum $P_0(k)$ consistent with these parameters. The boxsize used is $L = 160$ Mpc$/h$. The original simulation contains 1024^3 dark matter particles with particle masses of $m = 2.54 \times 10^8 M_\odot/h$ and was performed with the ART code. For the study presented in Chapters 4 and 5, we re-ran the same initial conditions with the GADGET code and a lower resolution of 256^3 particles, with particle masses of $m = 1.63 \times 10^{10} M_\odot/h$ from a starting redshift of $z_\mathrm{init} = 30$ until the final redshift $z = 0$. The two runs show only negligible differences at scales above the resolution of the 256^3 run.

Figure 2.8 shows a map of the density and velocity fields used as initial conditions at $z_\mathrm{init} = 30$ (left) and the final snapshot of the evolved simulation at $z = 0$ (right). While the initial conditions are a realisation of the initial Gaussian random field, lacking any distinct geometrical features, the evolved state shows the characteristic web-like appearance of contemporary large-scale structure. The density field has evolved into a complicated network of sheets and filaments, with dense dark matter haloes at their intersections and large cosmic voids in between. The dark matter haloes are of special interest, since they are the hosts of observable galaxies, which are believed to form by accretion and subsequent cooling of gas inside the haloes' gravitational potential wells (White & Rees 1978; White & Frenk 1991; Binney & Tremaine 2008). The distribution of dark matter haloes follows a hierarchical pattern. The largest objects lie in the mass range of massive galaxy clusters ($\approx 10^{15} M_\odot/h$) and are located in the regions of highest density. These regions are also the ones that contribute the most to the shape of the peculiar velocity field (bottom panels of Figure 2.8). Objects of lesser mass are also found along the filaments and, in smaller numbers, in the underdense regions. Assembled through hierarchical structure formation, merging and accretion processes, dark matter haloes typically show a rich substructure of smaller orbiting sub-haloes or satellite haloes which are gravitationally bound to the main object. These have substructure themselves, and this hierarchy continues down over many orders of magnitude in mass (limited by the numerical resolution of the simulation). This self-similar appearance can be also seen in the 2 Mpc$/h$ wide Local Group region in Figure 1.2. There is an ongoing debate whether this plethora of substructure across many orders of mass scales is compatible with the relatively small number of observed satellite galaxies in the Local Group (Klypin et al. 1999b; Moore et al. 1999; Simon & Geha 2007; Wang et al. 2012).

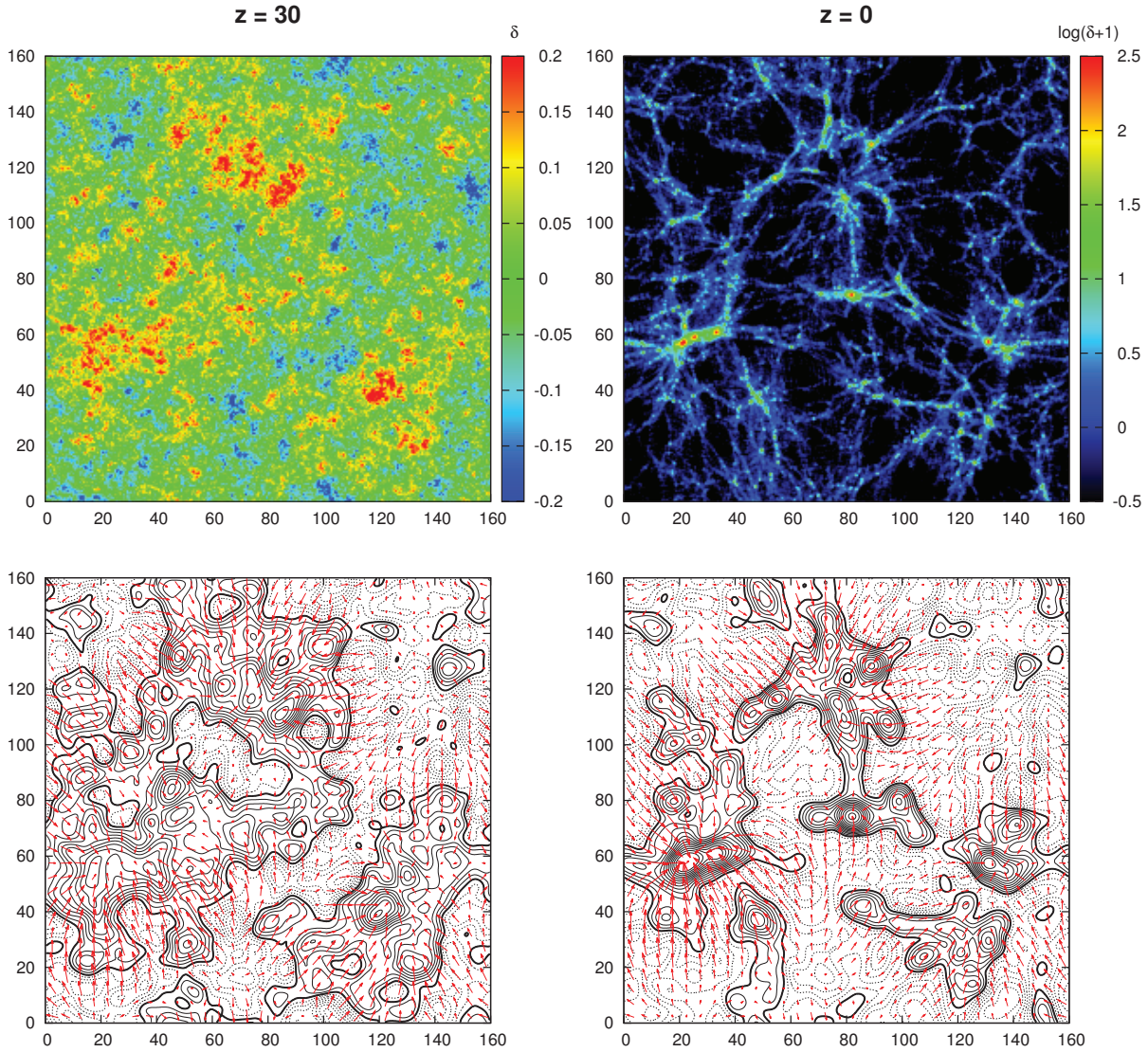

Figure 2.8: Overdensity field (top) and velocity field (bottom) at the initial redshift $z = 30$ (left) and final redshift $z = 0$ (right) of the BOX160 cosmological simulation with 160 Mpc/h boxsize, obtained with TSC mass assignment onto a 256^3 numerical grid. The contours underlying the velocity field are again the overdensity field, but smoothed with a Gaussian kernel with radius $R_G = 2.5$ Mpc/h. Note the different scale for the overdensity maps: the initial distribution consists of relatively small fluctuations and is shown in linear scale, while the final distribution is shown in log scale due to the very high overdensities of several hundred inside haloes. This map is a XY-plane projection of a 10 Mpc/h thick slice located at $87 < Z < 97$ Mpc/h.

The formation of individual haloes can be traced back to the gravitational collapse of overdense regions in the initial conditions. A first model of this collapse process was developed by Press & Schechter (1974), who approximated the halo formation process by the collapse of a spherical overdense region. The model was later generalised to triaxial ellipsoidal collapse by Sheth et al. (2001). As an overdense region collapses, it will eventually decouple from the overall cosmic expansion and contract gravitationally, until an equilibrium between gravitational contraction and outward pressure from the internal velocity dispersion of the dark matter particles is

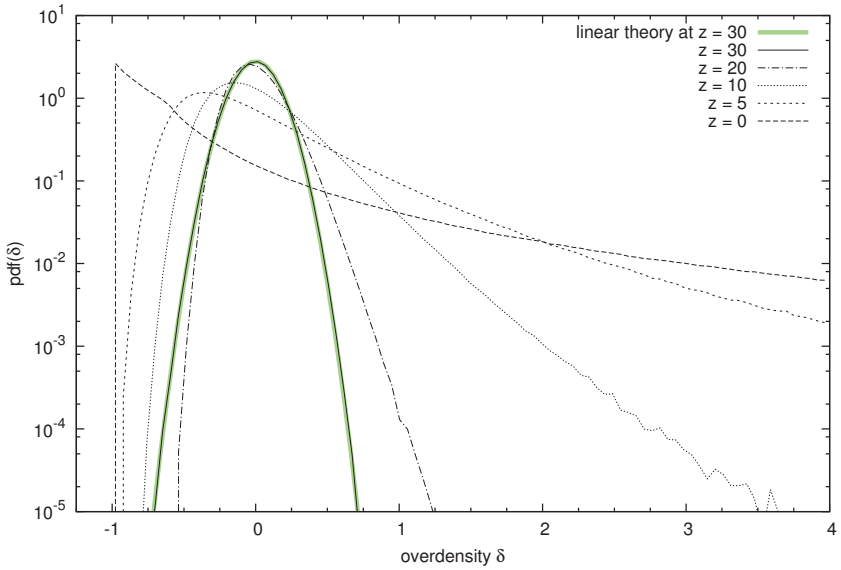

Figure 2.9: Distribution function of overdensities δ in the BOX160 simulation at different redshifts. The $z = 0$ histogram (solid black curve) depicts the original initial conditions density field; the histograms at later z were obtained from simulation snapshots by mapping all dark matter particles to a 256^3 grid using TSC mass assignment. The green curve shows a linear Gaussian distribution with mean 0 and standard deviation $\langle|\delta|\rangle$ as defined by the parameters at initial redshift $z = 30$.

reached and the formed halo keeps an approximately constant size. At this point, the halo is virialised, i.e. the ratio of the mean kinetic and potential energy is determined by the virial theorem, $2\langle E_{\rm kin}\rangle = -\langle E_{\rm pot}\rangle$. At virialisation, the mean overdensity of the region is around $\delta_{\rm vir} \approx 200$; one can then define a virial radius $R_{\rm vir}$, and compute the mass $M_{\rm vir}$ inside this radius (virial mass). The virial radius is often defined such that the enclosing mass has an overdensity of 200 relative to the cosmic mean density (Binney & Tremaine 2008), although other numbers are used as well (Knollmann & Knebe 2009), and some authors prefer to use the maximum of the halo's circular velocity profile to characterise a halo (Knebe et al. 2011a). One has to keep in mind that in a cosmological simulation, the virial equilibrium is never exactly reached, because merging and accretion is an ongoing process. The mass function of dark matter haloes in the simulations, i.e. the number of haloes per mass interval, is generally in good agreement with theoretical models (Tinker et al. 2008). It was also found that the spatial mass distribution within virialised dark matter haloes follows a general mass profile (Navarro et al. 1997). The intricate properties of dark matter haloes are a major subject of ongoing research.

To identify dark matter haloes in a simulation, a wide selection of codes using different algorithms is available; a good overview and comparison is given in Knebe et al. (2011a). For the simulations in this work, we use Amiga's Halo Finder (AHF; Knollmann & Knebe 2009), because it offers a fast performance and is able to identify subhaloes down to many hierarchical levels.

2.3.3 The evolved density and velocity fields

While the density and velocity fields are of linear Gaussian nature in the high-redshift initial phase, their statistical properties change drastically in the course of the structure formation process. Figure 2.9 shows the probability distribution function of δ at different redshifts. The physical restriction that densities cannot become negative, and therefore $\delta \geq -1$ always, breaks

the Gaussian symmetry quite early, and a significant skewness develops. The resulting distribution in the quasi-linear regime is reasonably approximated by a lognormal distribution (Coles & Jones 1991). In the non-linear phase at $z = 0$, the overdensity distribution has a very sharp peak just above $\delta = -1$ and a long tail of very high overdensities that are of the order of 10^3 – 10^4 in the inner regions of dark matter haloes, depending on the numerical resolution.

A majority of the dark matter haloes found at $z = 0$ can be associated with peaks in the initial conditions (Ludlow & Porciani 2011), although this mapping is not unambiguous in general, and the peaks undergo significant changes in both shape and position during structure formation. This cosmic displacement, i.e. the distance that a given patch of matter travels from the initial linear field until it finds oneself in an object at $z = 0$, is quite substantial, and in the case of BOX160 it is about 10 Mpc/h on average. One can see this movement more clearly when comparing the positions of the most prominent density peaks in the smoothed density field in Figure 2.8; although one also can see a clear similarity of the initial conditions and the evolved field on scales above that of the displacement. We also find that the size and amplitude of the initial peaks does not completely determine the properties of the haloes that evolve from them, which are influenced to a great degree by non-linear processes like merging and accretion. Conversely, the non-linear evolution of the density field irretrievably erases information about the initial conditions, as already discussed in Section 2.2.6. Following Crocce & Scoccimarro (2006) and Wagner (2009), we consider the propagator $g(z)$, which is defined as the Fourier-space cross-correlation coefficient between the initial field δ_0 and the evolved field $\delta(z)$ at different redshifts z as a function of scale,

$$g(z) = \frac{\langle \delta(z,k)\, \delta_0^*(k) \rangle}{\langle |\delta(z,k)| \rangle \, \langle |\delta_0(k)| \rangle} \quad . \tag{2.85}$$

For BOX160, the propagator is shown in Figure 2.10 for snapshots at different redshifts; the shape is in very good agreement with the analytical model in Crocce & Scoccimarro (2006), although the finite-volume effect of the 160 Mpc/h introduces some noise because the larger scales are sampled by relatively few discrete Fourier modes (see Section 3.1.3). The correlation with the primordial linear field degrades further as the simulation evolves. At $z = 0$, at wavenumbers above $k = 0.5$ h/Mpc, equivalent to length scales below 10 Mpc/h, in the range of the cosmic displacement field, there is virtually no correlation left. To a limited degree, some of the correlation with the initial field can be improved, for which it is necessary to devise some estimate of the shape of the displacement field and correct for its effect. This is the main component of reconstruction schemes discussed in Chapter 4.

It is striking that the evolved peculiar velocity field at $z = 0$ is affected much less by non-linear structure formation than the density field and has a very similar appearance to the primordial peculiar velocity field in the initial conditions. The similarity is even more obvious if one compares the velocity field at z_{init} and $z = 0$ component-wise (Figure 2.11). This is due to several reasons. First, the values of the velocity vector components are not subject to the positiveness-constraint that breaks the Gaussian symmetry in the density case; second, the velocity field has a high large-scale correlation and is therefore much less sensitive to scales below the quasilinear regime which are most strongly affected by non-linear gravitational dynamics. It is, on the other hand, affected to a similar degree by the cosmic displacement. There is also some non-linearity bias that enhances the velocity amplitude in regions of large density. We will analyse this further in the next section.

In Figure 2.12, we consider the power spectrum $P(k)$ of the evolved field at $z = 0$, compared to the linear input power spectrum $P_0(k)$. Except some fluctuations on the largest modes (i.e. small k) due to the discrete Fourier sampling (the excess power there is not typical and will

be discussed in Section 3.3.3), the initial conditions of BOX160 (black symbols) follow – by construction – very closely the input WMAP3 power spectrum (solid black line). On the other hand, gravitational clustering causes a strong enhancement of the density power spectrum at $z = 0$ (red symbols). At $z = 0$, this enhancement (studied e.g. by Peacock & Dodds 1994, 1996; Smith et al. 2003) starts to become noticeable at scales below ≈ 60 Mpc$/h$, which is around the scale where linear theory ceases to be valid. The non-Gaussianity also breaks the independence of the Fourier modes $\delta(\boldsymbol{k})$: gravitational collapse induces a transport of power from large modes to smaller modes (Ma 2007).

On the other hand, the power spectrum of fluctuations in the velocity field evolves very differently, an aspect that to our knowledge has not been analysed in detail before. In the linear regime of the initial conditions, the divergence of the velocity field $\boldsymbol{\nabla}\cdot\boldsymbol{u}$ is equal to the overdensity δ except for a factor (equation 2.62), so that (neglecting that factor) the power spectrum of $\nabla\cdot\boldsymbol{u}$ is identical to the initial $P(k)$ (black symbols in Figure 2.12). As the field evolves out of the linear regime by gravitational clustering, this equality breaks, and the two power spectra evolve differently[3]. Interestingly, at $z = 0$ the power spectrum of $\nabla\cdot\boldsymbol{u}$ has *decreased* for scales below the linear regime, which means that gravitational collapse has *smoothed* the peculiar velocity field on scales roughly between the onset of the quasilinear regime at a few tens of Mpc$/h$ and scales of a few Mpc$/h$ (green symbols in Figure 2.12). This smoothing of medium-scale fluctuations is also seen in Figure 2.11. Our interpretation is that both the gravitational collapse of haloes and the expansion of voids "stretch out" the velocity field: both the infall towards haloes and the outflow from voids are marked by quite smooth, laminar flows. Additionally, initial velocity fluctuations

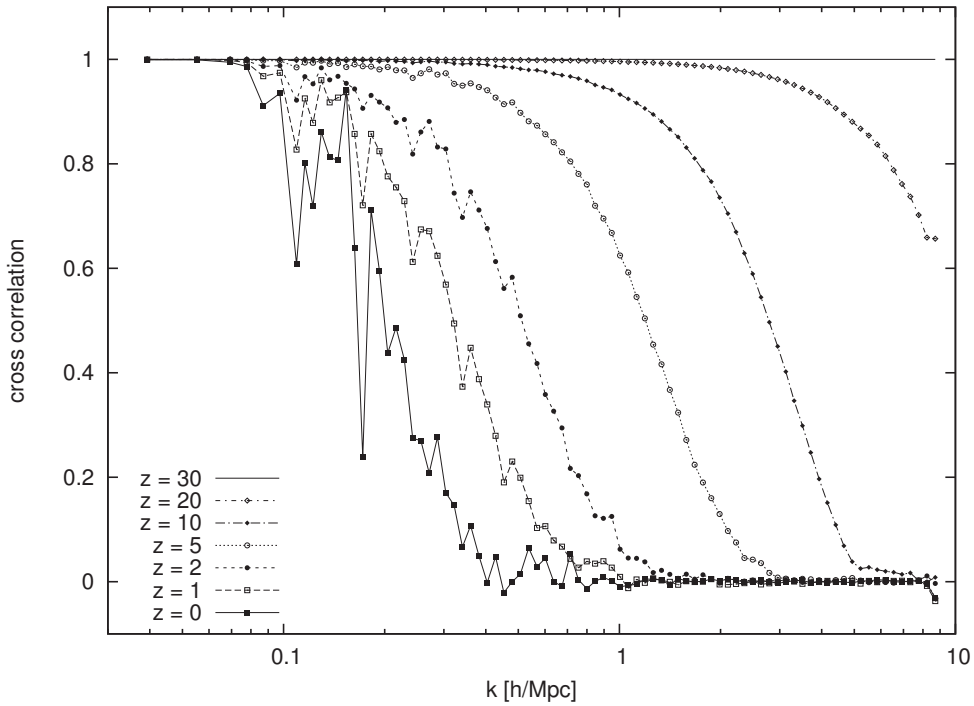

Figure 2.10: Fourier-space cross-correlation of the evolved density field $\delta(z)$ at different redshifts z with the initial field at $z = 30$, computed from different snapshots of the BOX160 simulation. The wavelength of $k = 1$ h/Mpc corresponds to a scale length of about 6 Mpc$/h$.

[3]It is possible to construct relations between the overdensity δ and the divergence of the peculiar velocity $\boldsymbol{\nabla}\cdot\boldsymbol{u}$ that hold in the quasilinear regime as well; an overview is given in Kitaura et al. (2012).

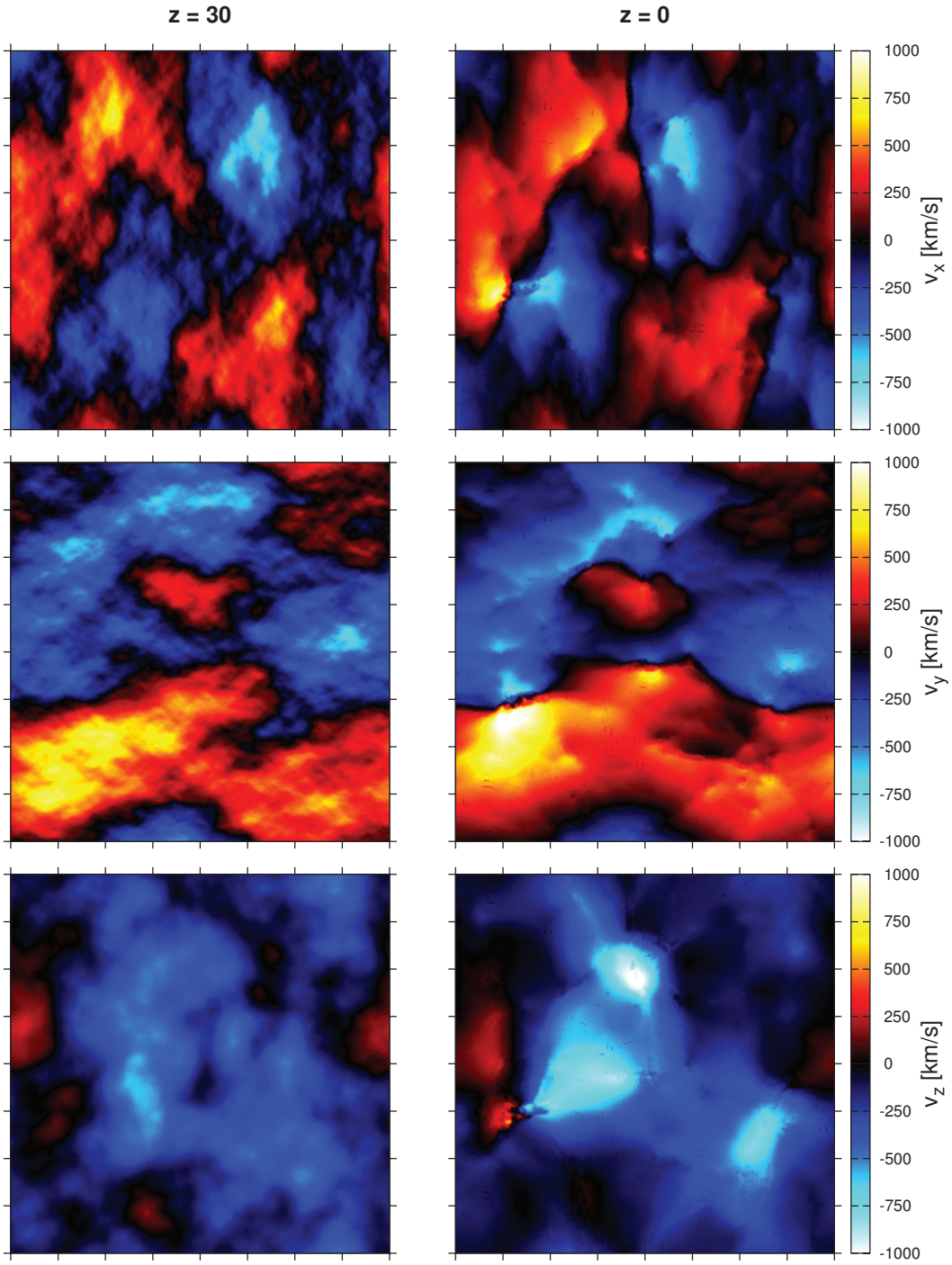

Figure 2.11: Same slice as Figure 2.8, at the initial redshift $z = 30$ (left) and final redshift $z = 0$ (right), but the peculiar velocity field is shown component-wise. Note that the v_z component (bottom) lies perpendicular to the plane of the figure, while the other two lie on it. The varying type of anisotropy in the noise patterns of the three components visualises the difference between radial (ψ_R) and transverse (ψ_T) peculiar velocity correlations. The right panels were generated by TSC binning of the particle velocities to a $N = 256^3$ grid (the same resolution was used in the left panels), without further smoothing.

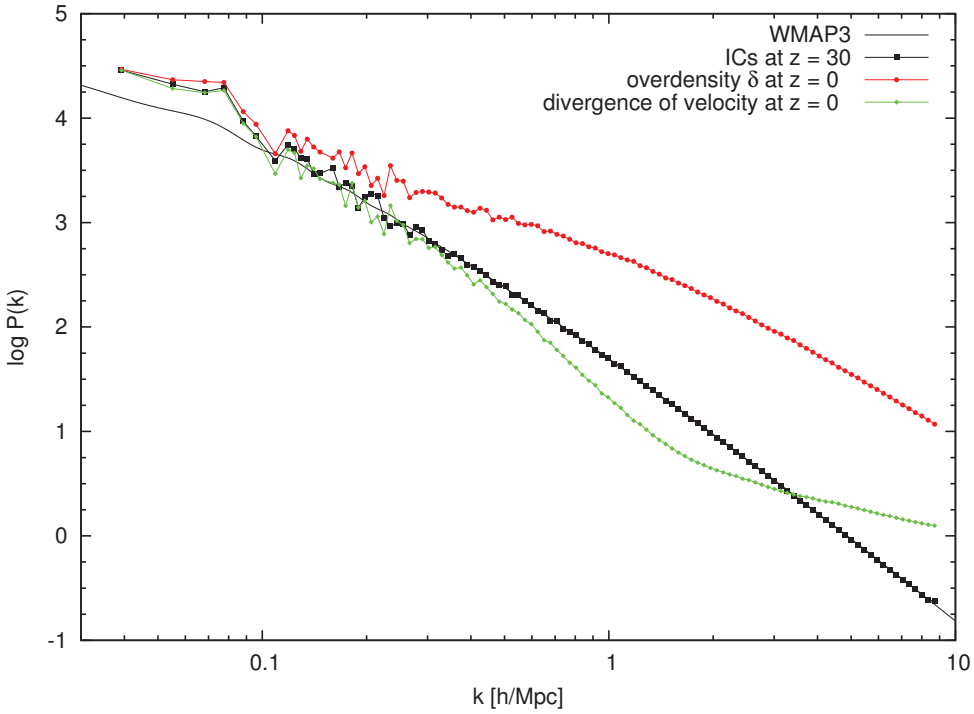

Figure 2.12: Periodogram estimate of the power spectrum $P(k)$ of the BOX160 initial conditions at $z = 30$ (black), the evolved density field δ at $z = 0$ (red), and the divergence of the evolved velocity field $\boldsymbol{\nabla} \cdot \boldsymbol{v}$ at $z = 0$ (green). The theoretical WMAP3 input power spectrum is shown as a solid black line. The power spectra at $z = 0$ have been obtained from a TSC mass assignment of the density and velocity fields to a 256^3 grid and corrected for the errors resulting from mass assignment using the scheme of Jing (2005).

become trapped in the virial motion inside haloes after their collapse, so that power is removed from the scales above that of individual haloes. At the halo scale (below ≈ 2 Mpc$/h$, or above $k \approx 3\ h/\mathrm{Mpc}$), the power spectrum of the velocity divergence is again above the linear theory prediction because of a non-linear enhancement of velocities, which we will analyse further in the next section.

2.3.4 Halo peculiar velocities

The peculiar velocity field has statistical properties that are closer to its initial conditions than it is the case for the density field; this makes the velocity field a potentially good source of information about the cosmological initial conditions. We now analyse the evolved peculiar velocity field at $z = 0$ in more detail. The present-day peculiar velocity field is accessible observationally through the peculiar velocities of galaxies (see Section 2.1). In the current theoretical picture of galaxy formation, all these galaxies reside within the potential wells of dark matter haloes. It is therefore a reasonable assumption that the observed galaxy peculiar velocities follow, in fact, the peculiar velocities of their surrounding dark matter haloes, and in this way provide a direct tracer of the underlying peculiar velocity field at $z = 0$. This in turn provides a direct and unbiased tracer of the underlying total gravitational potential (cf. equation 2.32) and therefore the total matter distribution (i.e. including dark matter).

Hence, we want to study how peculiar velocities of dark matter haloes are distributed; the results will be, to a reasonable degree, also applicable to galaxy peculiar velocity data. As a

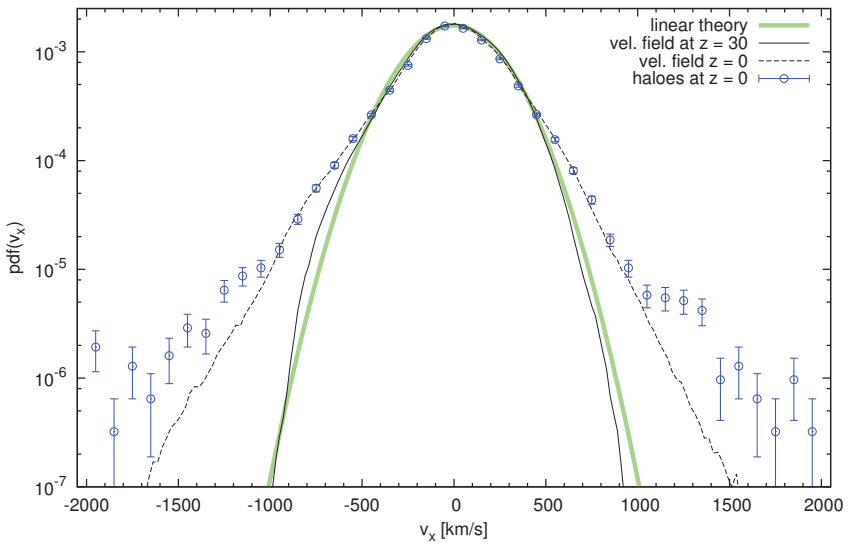

Figure 2.13: Distribution function of one velocity field component (v_x) in BOX160 at the initial conditions $z = 30$ and from the simulation snapshot at $z = 0$ (normalised to the same growth rate). The blue points show the distribution function of dark matter halo velocities at $z = 0$. The green curve shows a theoretical Gaussian distribution, with the mean and standard deviation adjusted to $\langle v_x \rangle$ and $\langle |v_x| \rangle$, respectively.

specimen, we use the AHF halo catalogue of the BOX160 simulation at $z = 0$. One can define the peculiar velocity of a halo as the bulk motion of all particles that belong to this halo. In AHF terms, it is computed as the average motion of all dark matter particles within the virial radius $R_{\rm vir}$.

Figure 2.13 shows the distibution function of the initial velocity field at $z_{\rm init} = 30$ (solid black line), the evolved velocity field at $z = 0$ (dashed black line), and the peculiar velocities of haloes identified in the BOX160 (blue points). The field distributions have been obtained from sampling on a 256^3 computational grid; the halo velocities come from the AHF halo catalogue. We consider one component only, here v_x, because this should be a Gaussian distributed variable in linear theory (the total velocity should be Maxwellian, see Section 2.2.5). The green curve in Figure 2.13 shows the theoretical Gaussian distribution as given by linear theory. There is some deviation from this distribution in the initial v_x. It may seem surprising that the velocity vector components in the BOX160 initial conditions are not perfectly Gaussian distributed despite the fact that the initial conditions are the realisation of a Gaussian random field. This is an effect of the finite box volume which we can ignore for now; it will be discussed again in Section 3.1.3.

We are specifically interested in how far the evolved peculiar velocity field at $z = 0$ (dashed curve in Figure 2.13) deviates from the statistics of the linear initial conditions. Similar studies have been presented in Sheth & Diaferio (2001) and Hamana et al. (2003, 2005), who propose theoretical models for the non-linear deviations. Comparing the final distribution at $z = 0$ to the initial conditions, we see that the outer parts show tails of enhanced velocity. This is in part due to the effect of non-linear bias: in regimes of higher density, the velocity distribution remains approximately Gaussian, but the rms variance is higher with increasing density. This dependence of the velocity field on the local density cannot be predicted from linear theory. The resulting non-linear distribution is therefore a sum of several Gaussian distributions with different standard deviations. However, the central part of the distribution, containing the most points, does not deviate much from the initial conditions. Overall, non-linear gravitational collapse has a much

smaller effect on the velocity field than for the drastically altered distribution of overdensities, so we can consider the peculiar velocity field at $z = 0$ to be closer to the linear theory than overdensities in this sense. In Chapter 4, we will see that it also conserves more information about the cosmological initial conditions.

The peculiar velocities of haloes at $z = 0$ (blue points in Figure 2.13) further deviate from the distribution of the whole field, with more strongly enhanced wings in the distribution. This is due to the fact that haloes trace the velocity only at the highest peaks of the density field and thus do not provide a completely unbiased sample. Even within linear theory, the rms peculiar velocity distribution at the peaks deviates somewhat from the general distribution (Bardeen et al. 1986), although this effect is much smaller than the non-linear effect. The latter is due to the fact that haloes are preferentially located in dense environments, where the rms variance of velocities is higher, so that the non-linear wings are enhanced. Additionally, a non-negligible fraction of haloes consists of substructure embedded in larger haloes, so virial motions are contributing as well; they likewise have a higher rms velocity than the linear field.

Figure 2.14 shows the halo peculiar velocities in BOX160 over the halo mass (left panel) and the local density (right panel). While the increased peculiar velocities in high-density regions is obvious, the peculiar velocities are practically independent of the halo mass; both results are consistent with Sheth & Diaferio (2001); Hamana et al. (2003, 2005). The halo mass dependence slightly varies with different halo finders and different realisations. We found that in BOX160 there is a slight tendency of higher velocities for more massive clusters above $\approx 10^{13} M_\odot/h$, whereas another simulation, a random realisation of the same cosmology with the same boxsize, showed a slight tendency of lower velocities at the high-mass end. This will be discussed further in Chapter 3. We also found that the BDM halo finder (Klypin et al. 1999a) tends to higher estimates of halo peculiar velocities at the high-mass end, while the friends-of-friends algorithm (we used the FoF implementation discussed in Knebe et al. 2011a) tends to lower estimates. This is similar to the findings of Suhhonenko & Gramann (2003). We continue to use AHF, which gives halo velocities lying in between those of BDM and FoF.

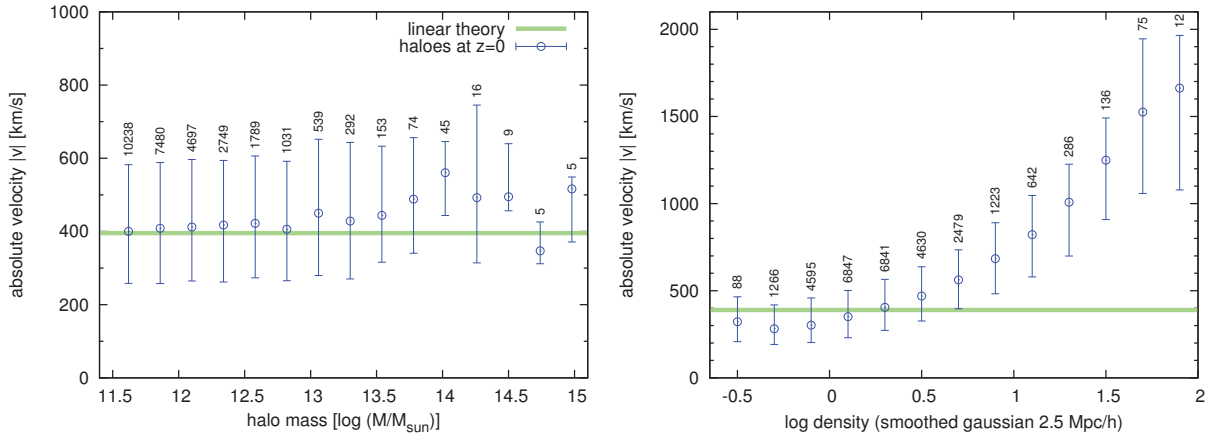

Figure 2.14: Absolute velocity $|\boldsymbol{v}|$ of dark matter haloes in BOX160 at $z = 0$ depending on their mass (left) and the underlying density field (smoothed with a 2.5 Mpc$/h$ Gaussian). For each bin, the point is placed at the median, and the bar shows the interval between the 25$^{\text{th}}$ and 75$^{\text{th}}$ percentile. Additionally, the number of haloes in each bin is given. The green curve shows the overall standard deviation $\langle|\boldsymbol{v}|\rangle$. Note the different scales for the $|\boldsymbol{v}|$ axis.

Overall we can conclude that the peculiar velocity field at $z = 0$ is remarkably closer to the Gaussian statistics of the initial conditions than the corresponding non-linear density field. Even if we restrict ourselves to the velocities of dark matter haloes as tracers of the underlying field, serving as a simulation analogue of the peculiar velocity distribution of observed galaxies, the effects of non-linearity on their velocity distribution are relatively mild. The strongest deviations from linear theory are found in regions of high density, where the halo abundance is higher. We will minimise these biasing effects later in this study by using mock catalogs that are sampled randomly in mass and space. This procedure also mimicks the observational data of spiral galaxy peculiar velocities which are not located at the highest density peaks of the galaxy distribution, but are selected on the random basis of their inclination on the sky being greater than 45 degrees, which is an absolute requirement for the Tully-Fisher method.

2.3.5 Peculiar velocity data as a tracer of the underlying field

We can now connect the observed peculiar velocites of galaxies with our study of the peculiar velocities of haloes. We continue to use the assumption that, since galaxies reside at the bottom of their host haloes' potential wells, they should follow the same peculiar motion. Of course, this motion can also be the virial motion within a bound group or cluster, which is decoupled from the large-scale velocity field. This should be considered in the data by an adequate galaxy grouping or a similar procedure. It is also interesting to relate the non-linear effects to different types of peculiar velocity data. We saw that the non-linear enhancement is stronger in regions with high local overdensity. We therefore expect that the peculiar velocities of different types of galaxies would have different statistical properties. Elliptical galaxies are preferentially located in high-density environments (van der Wel et al. 2010) and should be affected more strongly by the non-linear enhancements. Also, within a cluster, ellipticals are more strongly concentrated towards the cluster centre than the spirals (Sheth & Diaferio 2001). In other words, the peculiar velocity field traced by ellipticals should have a higher variance than that traced by spiral galaxies. On the other hand, spiral galaxies are more often found in less dense environments and should yield more linear statistics. We therefore expect that, in order to obtain a good estimate of the linear peculiar velocity field, data obtained from spiral galaxies (such as with the Tully-Fisher method) could be more useful than data obtained from early-type galaxies, although to our knowledge this has not been studied in detail in this context. Related to the morphology-density relation is the fact that data obtained with the Tully-Fisher method provides a more homogeneous sampling of the sky, which in turn leads to a better estimate of the linear peculiar velocity field: due to the high large-scale correlations of the linear velocity field, more homogeneously distributed data will convey more information about the underlying field than a locally high concentration of datapoints.

Despite the relative sparseness, noisiness and limited volume of peculiar velocity data, it offers several important advantages when compared to galaxy redshift surveys, not only because of the favourable statistical properties of the velocity field itself. Redshift data, while providing quite complete samples of galaxy positions in redshift space, feature geometrical distortions, because the physical distance is not known, and the observed redshift distance is a sum of the actual distance and the peculiar motion (2.2). This leads to the finger-of-god effect (Jackson 1972), elongating the apparent shape of galaxy clusters because of their velocity dispersion, and the bull's eye effect (Kaiser 1987; Thomas et al. 2004), leading to a transversal geometrical distortion due to coherent peculiar motions. Peculiar velocity datasets, on the other hand, feature three-dimensional positions in physical real space. Further, galaxy positions in redshift space are not an unbiased tracer of the underlying total matter distribution, because light does not follow mass. To derive the underlying total matter and velocity field from redshift data is

extremely difficult: relating galaxy positions and luminosities to the underlying field is subject to a significant galaxy bias which is not yet fully understood (McBride et al. 2011). For both density and velocity fields, elaborate reconstruction methods from redshift data have been developed in the last years, e.g. MAK reconstruction (Lavaux et al. 2010), bayesian methods (Kitaura & Enßlin 2008; Kitaura et al. 2010), or from models of the underlying halo density profiles (Muñoz-Cuartas et al. 2011). However, such methods are usually strongly affected by galaxy bias and redshift distortion (Lavaux et al. 2008) and complicated to handle due to the high non-linearity of the data. We argue that in order to obtain a reconstruction of the underlying density and velocity fields that is suitable for a reconstruction of the linear cosmological initial conditions (see Chapter 4), peculiar velocities are the more convenient and natural choice, especially if the aim is to model the observed peculiar velocity field of the Local Universe. This is enhanced by the already discussed property of peculiar velocities being highly correlated on large scales, which greatly supports both reconstruction from the sparse and noisy data and the extrapolation to unsampled regions. In Chapter 3 we will discuss the method of Wiener Filter, which is formulated within the Gaussian linear theory. This is a natural approach because, as we saw, the statistics of the peculiar velocity field are quite close to Gaussian on supergalactic scales.

Chapter 3

Initial conditions, Wiener filter, and constrained realisations

The development of the ICeCoRe *software package for generating constrained initial conditions was a substantial part of the work presented in this thesis. In this chapter, we summarise the numerical algorithms that were implemented in* ICeCoRe *and the necessary theoretical background. The functionality of* ICeCoRe *covers the generation of random initial conditions for cosmological N-body simulations, linear reconstruction of density and velocity fields from sparse and noisy data points with the Wiener filter (WF) technique, and constraining initial conditions with the constrained realisations (CR) method. The flexible and highly optimised implementation is able to handle the upcoming large observational datasets containing several tens of thousands of constraints. Important aspects of the implementation and variations of the standard WF/CR technique are discussed, with focus on the CLUES technique of generating CRs from radial peculiar velocity data. We will also point out inconsistencies of the method that could lead to systematic errors and how they can be treated.*

3.1 Random initial conditions for N-body simulations

For dark-matter-only N-body simulations, which we will discuss here, the initial conditions (ICs) consist of a set of N particles, each with a position $\boldsymbol{x}_i$, a velocity $\boldsymbol{u}_i$, and a particle mass m_i. They are set up such that the particles sample the underlying density and velocity fields, i.e. the phase space, of a random realisation of the Universe at some initial redshift z_{init} in the linear regime of the matter-dominated epoch. The simulation then performs the task of evolving them forward in time by numerically solving equations 2.82 and 2.83. To obtain ICs for such a simulation, we first have to generate a realisation of the initial density and velocity fields. This is possible, because at z_{init} the distribution is a Gaussian random field with a given linear power spectrum $P(k)$, and therefore all statistical properties of the field are known. In the second step, the initial field can be sampled with particles. Subsequently, this particle data can be fed into an N-body code to carry out the simulation.

3.1.1 Initial overdensity and velocity fields

The standard technique to generate the initial fields employs as the computational box a regular cubic grid with N grid cells and periodic boundary conditions. The boxsize L should be chosen

such that the fundamental mode, with wavenumber $k_L = 2\pi/L$, stays linear throughout the simulation. If the simulation will run until $z = 0$, then it follows from the scale of non-linearity that L should be at least around 60 Mpc/h for a ΛCDM simulation. Since we always assume a cubic box shape and a regularly spaced discretisation, the spatial resolution of the grid (i.e. the distance between the discrete grid cells) is given by $\Delta x = L/N^{1/3}$. On such a grid, the Fourier transform (2.44) reduces to a discrete Fourier transform (DFT) in three dimensions,

$$\delta(\boldsymbol{k}) = \frac{1}{N}\sum_{\boldsymbol{x}} \delta(\boldsymbol{x}) e^{i\boldsymbol{k}\cdot\boldsymbol{x}} \quad ; \quad \delta(\boldsymbol{x}) = \frac{1}{(2\pi)^3}\sum_{\boldsymbol{k}} \delta(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \quad . \tag{3.1}$$

The DFT is by design consistent with the assumed periodic boundary conditions and can be computed very efficiently using the well-known Fast Fourier Transfom (FFT) algorithm (Cooley & Tukey 1965)[4]. The Fourier-space positions $\boldsymbol{k}$ are then discrete wavevectors.

For the initial overdensity field δ, we now have to construct a realisation of a Gaussian random field from a known power spectrum $P(k)$. This will be the linear power spectrum $P_0(k)$ of the chosen cosmological model and has to be provided in tabulated form as input (see Section 2.2.4). It has then to be normalised with the chosen σ_8 parameter (equation 2.58) and the chosen redshift $z_{\rm init}$ (equation 2.57). The latter requires to numerically evaluate the linear growth factor D_+ (equation 2.43). Then, a simple way to generate a realisation of δ on the grid is to create random phases and amplitudes with the right variance for each wavevector in Fourier space, since we know that the phases are uniformly random and the amplitudes are Rayleigh distributed (equation 2.46). The fastest way to accomplish that is to draw for each $\boldsymbol{k}$ two random numbers from a zero-mean Gaussian distribution with variance $P(k)$ and assign them to the real and complex part of $\delta(\boldsymbol{k})$, respectively (Knebe 2007).

However, we choose the slightly different approach of convolution from white noise, following Bertschinger (2001) and Prunet et al. (2008). This method produces a mathematically equivalent result but is more flexible in practice. We first create a Gaussian white noise field $w(\boldsymbol{x})$, such that

$$\langle w(\boldsymbol{x})\rangle = 0 \quad ; \quad \langle |w(\boldsymbol{x})|^2\rangle = 1 \quad , \tag{3.2}$$

by assigning a zero-mean, unity-variance Gaussian random number to each grid cell $\boldsymbol{x}$ in real space[5]. We then perform a forward FFT. In Fourier space, the white noise field will have random phases and amplitudes, with a flat power spectrum $\langle |w(\boldsymbol{k})|^2\rangle = 1$. Since the Fourier modes are independent, we can now "imbue" the field with the power spectrum $P(k)$, which produces a realisation of the overdensity field $\delta(\boldsymbol{k})$, by performing the multiplication

$$\delta(\boldsymbol{k}) = \sqrt{P(k)} \cdot w(\boldsymbol{k}) \quad . \tag{3.3}$$

for each grid cell. This multiplication is equivalent to a convolution of the white noise field $w(\boldsymbol{x})$ with the corresponding real-space correlation function. On the discrete periodic grid, this correlation function will be the DFT of the power spectrum $P(k)$,

$$\xi(\boldsymbol{x}) = \langle \delta(\boldsymbol{x}')\, \delta(\boldsymbol{x}'+\boldsymbol{x})\rangle = \frac{1}{(2\pi)^3}\sum_{\boldsymbol{k}} P(k) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \quad . \tag{3.4}$$

[4] In the ICeCoRe code, this is performed using the FFTW3 library (Frigo & Johnson 2005), available at www.fftw.org .

[5] The ICeCoRe code uses the GNU scientific library, available at www.gnu.org/gsl , to compute these Gaussian random numbers with the fast Ziggurat algorithm (Marsaglia & Tsang 2000).

It is easy to see that the Gaussian random field produced by 3.3 is a realisation of the condition $\langle|\delta(\boldsymbol{k})|^2\rangle = P(k)$. The real-space density field $\delta(\boldsymbol{x})$ can be obtained by performing the backwards FFT.

The advantage of the white noise field is that all necessary information about the phases and amplitudes of the ICs is encoded in the white noise field $w(\boldsymbol{x})$, yet it does not depend on the cosmological parameters or the shape and normalisation of the power spectrum $P(k)$. It is therefore possible to create, for example, realisations with different parameters, or different power spectra, from the same white noise field with the same phase structure. The white noise field allows for many other ways of manipulating the obtained realisation. Most importantly, the approach greatly aids the interoperability with other codes that generate initial conditions. This will be important in the context of generating high-resolution initial conditions, which is discussed in Section 3.5.

We note here that any realisation $\delta(\boldsymbol{x})$ can be easily converted to the corresponding white noise field $w(\boldsymbol{x})$, if one has access to the tabulated power spectrum $P(k)$ that was originally used to generate $\delta(\boldsymbol{x})$. This is performed in Fourier space by inverting equation 3.3. We will refer to this procedure as "whitening". Analogously, we will refer to the evaluation of equation 3.3 as "colouring" (following the terminology in signal processing, where noise with a non-flat power spectrum is said to have a noise colour).

The initial overdensity field $\delta(\boldsymbol{x})$ completely defines the initial conditions. Since linear theory is valid at z_{init}, the velocity field $\boldsymbol{u}(\boldsymbol{x})$ is defined by $\delta(\boldsymbol{x})$ through equation 2.63. It is also proportional to the linear displacement field $\boldsymbol{\psi}(\boldsymbol{x})$ through $\boldsymbol{u}(\boldsymbol{x}) = \dot{a} f \boldsymbol{\psi}(\boldsymbol{x})$, where $-\boldsymbol{\nabla}\cdot\boldsymbol{\psi}(\boldsymbol{x}) = \delta(\boldsymbol{x})$. The linear displacement field on the discrete grid can therefore be rapidly computed by performing three FFTs:

$$\psi_\alpha(\boldsymbol{x}) = \frac{1}{(2\pi)^3}\sum_{\boldsymbol{k}} \frac{ik_\alpha}{k^2}\delta(\boldsymbol{k})e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \quad ; \quad \alpha \in \{x, y, z\} \quad . \tag{3.5}$$

3.1.2 Particle sampling

Having the density and displacement fields, it is now possible to construct a particle discretisation. We will consider the case that this is a uniform discretisation, such that the particles have equal mass, and therefore the Lagrangian patches that they sample have equal volume. The standard technique to set up such a discretisation uses the Zeldovich approximation to obtain the positions x_i and u_i of each particle $i = 1, \ldots, N$ (Efstathiou et al. 1985). If we choose some Lagrangian positions $\boldsymbol{q}_i$ (the "pre-initial conditions"), which homogeneously sample the computational box, the x_i and u_i can be obtained by

$$\boldsymbol{x}_i = \boldsymbol{q}_i + \boldsymbol{\psi}_i \tag{3.6}$$

$$\boldsymbol{u}_i = \dot{a} f \boldsymbol{\psi}_i \quad , \tag{3.7}$$

This displaces the particles in such a way that their resulting distribution is a good approximation of the underlying density field δ if z_{init} is low enough. The mass of each particle follows from the comoving mean density (2.22) and the volume of the associated Lagrangian patch sampled by the particle,

$$m_i = \bar{\rho}\,(\Delta x)^3 \quad . \tag{3.8}$$

The natural choice of the particle pre-initial conditions $\boldsymbol{q}_i$ is such that they correspond to the positions of the computational grid, i.e. there is exactly one particle centered on each grid cell. Then, the values $\psi_i = \psi(\boldsymbol{q}_i)$ are directly given by the displacement field grid (3.5). This way

of generating particle ICs is directly implemented into ICECORE. It can be shown that for this (and only this) type of particle lattice, one can exactly recover all $3N$ Fourier amplitudes of the underlying grid from the coordinates of the N particles (Knebe et al. 2001). If the displacement field is computed in Fourier-space (3.5), which is exact for the discrete Fourier modes of the grid, one also completely avoids numerical interpolation errors that would occur with other methods (e.g. by solving the displacement field in real space). For this reason, the cubic lattice particle sampling is optimal for ΛCDM initial conditions, where one wants to sample the high-k end of the power spectrum as accurately as possible. The "artefacts" introduced by the lattice pattern are seen throughout the simulation in underdense regions that do not undergo gravitational collapse, but they do not affect the properties of gravitationally collapsed objects in the case of hierarchical ΛCDM structure formation. Therefore, alternative pre-initial conditions, like the glass method (White 1994), which replaces the cubic lattice pattern with an amorphous sampling, or the "quaquaversal" method of Hansen et al. (2007), are only interesting for WDM models (Wang & White 2007) and not considered here.

The Zeldovich approximation (ZA) is a first-order Lagrangian approximation and therefore the resulting particle sampling at $z_{\rm init}$ is not exactly equal to the theoretical density field $\delta(z_{\rm init})$. Obviously $z_{\rm init}$ should be chosen such that the particle displacement is not shell crossed at any particle position. Further, the ZA slightly underestimates the skewness of δ and excites artificial transient decaying modes (Crocce et al. 2006). This can be alleviated by displacing the particles instead with second-order Lagrangian perturbation theory (2LPT), which allows one to start the simulation at later initial redshifts $z_{\rm init}$. However, Knebe et al. (2009) found that these effects have no influence on the statistics of the evolved simulation at $z = 0$ and that it is safe to use the ZA approach with $z_{\rm init}$ such that the total variance of $\delta(z_{\rm init})$ is around $\langle|\delta|\rangle \approx 0.2$ or even slightly larger. With this criterion, the optimal $z_{\rm init}$ is completely determined by the chosen boxsize L and resolution N. Initial conditions with 2LPT could be interesting for simulations designed to probe high redshifts in detail, instead of evolving until $z = 0$; however, we do not consider such simulations in this work.

3.1.3 Finite-volume effects

Since in this work we are not interested in small-scale effects, we are not going to further discuss the numerical errors that occur at high k close to the Nyquist frequency of the grid $k_{\rm Ny} = \pi/\Delta x$ due to the finite resolution; see Hockney & Eastwood (1992); Bertschinger (2001); Prunet et al. (2008); Hahn & Abel (2011). Much more important for the following are the large-scale effects (i.e. low k) of choosing a finite cubic volume with periodic boundary conditions as the computational domain. Setting up a Gaussian random field in a finite box truncates the power spectrum at large scales below a minimum wavenumber $k_{\rm min}$. This will be the fundamental frequency of the box, $k_L = 2\pi/L$, where L is the boxsize. As a result, large-scale correlations and the clustering of massive objects are suppressed, and the halo mass function of the resulting simulation will be affected (Power & Knebe 2006). The effective variance of the created density field will be lower than the theoretical variance defined by $P(k)$, since all of the power beyond $k_{\rm min}$ is missing. For a ΛCDM power spectrum, one finds that the σ_8 of the ICs is systematically too low compared with the input value by as much as 40% for a boxsize of 50 Mpc/h (Pen 1997), but the error quickly becomes smaller with increasing boxsize. In the BOX160 simulation ($L = 160$ Mpc/h), the error on the σ_8 variance is only 0.8%. It is possible to enforce that the *effective* σ_8 of the produced grid, i.e. considering only modes between k_L and $k_{\rm Ny}$, will be equal to the input σ_8. This is achieved by setting up the ICs using a power spectrum $P^*(k)$ with the adjusted normalisation

$$P^*(k) = \left(\frac{\sigma_8}{\sigma_8^*}\right)^2 P(k) \quad ; \tag{3.9}$$

$$\sigma_8^* = \left[\frac{1}{2\pi^2}\int_{k_{\min}}^{k_{\max}} k^2\, P(k)\, W_{\rm sth}^2(kR)\, \mathrm{d}k\right]^{1/2} \quad ; \quad R = 8\ \mathrm{Mpc}/h \quad ;$$

where $k_{\min} = k_L$ and $k_{\max} = k_{\rm Ny}$. This is the convention used for most CLUES simulations, such as the BOX160 simulation[6]. Another convention is used by the initial conditions generator of Klypin & Holtzman (1997), where $k_{\min} = k_L/\sqrt{2}$ and $k_{\max} = k_{\rm Ny}$. Some simulators also prefer not to use this correction at all, because then, although the σ_8 measured on the grid will be too small, the amplitudes of the modes will match the assumed $P(k)$ in the sampled frequency range. For compatibility reasons, all three possibilities are available in ICeCoRe.

Although the finite-volume effects on the variance of the density field are well known, usually there is little attention given to the effect on the peculiar velocity field. Due to the large-scale correlations of the peculiar velocities, the truncation of $P(k)$ at large scales reduces the total variance of the peculiar velocity field much more strongly than it is the case for the overdensity field. Figure 3.1 illustrates how the truncation of $P(k)$ at a frequency k_L affects the shape of the radial velocity-velocity correlation function $\psi_{\rm R}$. While in the unlimited case, the variance is around $\langle|\psi_\alpha|^2\rangle \approx 34\ \mathrm{Mpc}^2/h^2$, corresponding to $\langle|u_\alpha|^2\rangle \approx (300\ \mathrm{km/s})^2$ per peculiar velocity component, this is strongly reduced by finite boxsizes. At a boxsize of $L = 160\ \mathrm{Mpc}/h$, the resulting variance of the peculiar velocity field is only 58% of the theoretical value: $(230\ \mathrm{km/s})^2$ instead of $(300\ \mathrm{km/s})^2$ per component (cf. green curve/line in Figures 2.13 and 2.14, respectively, which show the standard deviation of the *absolute* peculiar velocity $\sqrt{3}\cdot 230\ \mathrm{km/s} \approx 400\ \mathrm{km/s}$). This strong underestimation of the peculiar velocity variance is therefore not negligible even for relatively large boxsizes and will become relevant for setting up constrained simulations from peculiar velocity data (see Section 3.3.3). Even for a very large boxsize of $L = 1000\ \mathrm{Mpc}/h$, there is still a slight diminution of the peculiar velocity variance of about 4%. The truncation of $P(k)$ also significantly alters the shape of the velocity correlation function down to scales of at least $0.2\ L$.

Besides the truncation of $P(k)$, the computational box introduces a strong discretisation effect on the large scales. As already discussed, a Gaussian random field can be regarded as a superposition of Zeldovich waves. Each wave corresponds to one grid cell in Fourier space, which has a Rayleigh-distributed random amplitude (2.46). As a result, the large scales are very sparsely sampled by the discrete grid Fourier modes. There are only three fundamental modes of wavelength L (or wavenumber k_L), one along each cartesian axis. The next largest modes are those of the face diagonals, with wavelength $L/\sqrt{2}$, of which there are six. Next there are four modes along the cube diagonals with wavelength $L/\sqrt{3}$, and after that three modes which correspond to the second harmonics along the three axes, with wavelength $L/2$. One can see that the large modes exhibit a very sparse and anisotropic sampling. Since the large scales of the box are defined only by the phase and amplitude of a few discrete modes, they exhibit a large cosmic variance, and the effective power spectrum of a realisation will show a large scatter around the input $P(k)$. This well-known effect is significant down to scales of about $0.1 - 0.2\ L$ (cf. Figure 2.12). It means also that the effective correlation function on the grid (3.4) will show significant anisotropies down to these scales, which is not the case for the correlation function in the non-volume-limited, non-discretised case (2.50).

[6] This way of computing the variance of the grid is not entirely accurate. The integral in 3.9 should instead go over the discrete Fourier modes $\boldsymbol{k}$. However, for the boxsizes and resolutions considered here, the difference is not very large.

Pen (1997) suggested an alternative method to set up initial conditions, using the isotropic one-dimensional correlation function $\xi(r)$ instead of the isotropic one-dimensional power spectrum $P(k)$ for the convolution from white noise (see also Bertschinger 2001; Sirko 2005; Hahn & Abel 2011). This is equivalent to convolving the power spectrum $P(k)$ with the box geometry, instead of just renormalising it. His method produces a correct variance for the field, a correct correlation functions up to $L/2$, and reduces the effect on the halo mass function (Gnedin et al. 2011). It also offers the possibility of a non-zero DC mode to account for power on scales larger than the boxsize. Since this method started to become popular only quite recently, and its implications are not yet well-tested, we do not follow this approach here, although we want to point out that it is a very interesting alternative.

We can now also understand why the components of the initial velocity field u_α do not appear to be exactly Gaussian distributed in the initial conditions (Figure 2.13, solid black line). Due to the strong large-scale correlations, the velocity field is influenced most strongly by large-scale flows, which are in turn defined by the values of the large-scale modes. Each discrete large-scale density wave causes a corresponding velocity flow with a variance determined by the wave amplitude. Thus, the overall distribution is a sum of distributions with different variances, of which a few discrete ones contribute more strongly to the overall velocity field than the rest. The total distribution is then not necessarily Gaussian and strongly depends on the random seed. The total variance of the u_α can also fluctuate considerably. According to the central limit theorem, the overall distribution of the u_α will be Gaussian if there are enough of the modes that most strongly define the shape of the velocity field. Therefore, the distribution will approach an exact Gaussian with the correct variance $\langle |u_\alpha|^2 \rangle$ on a large enough boxsize. Figure 2.13 shows that for a boxsize of $L = 160$ Mpc$/h$, this is not yet the case.

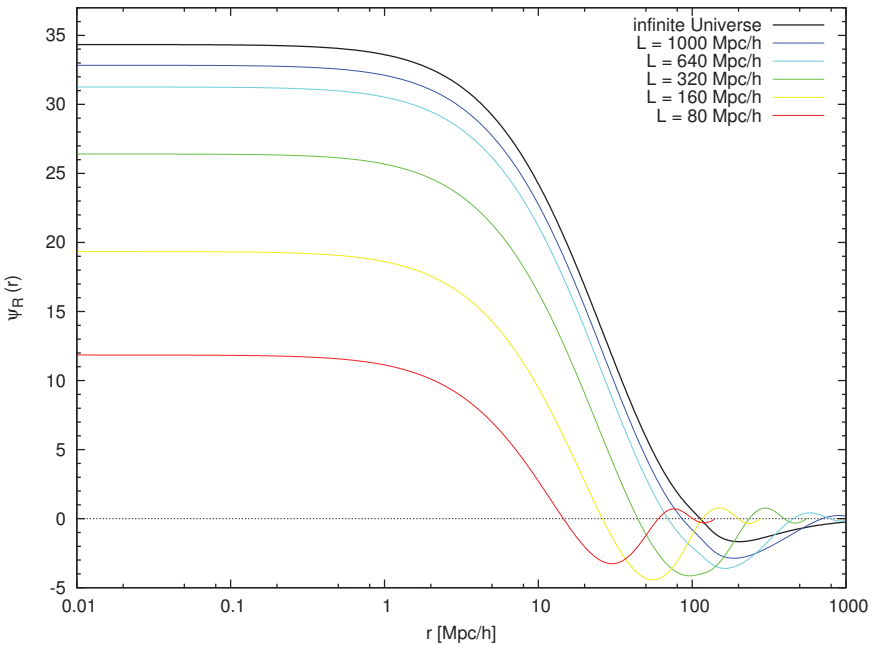

Figure 3.1: Finite-volume effect on the radial velocity-velocity correlation function ψ_R (with $R_\mathrm{G} = 0$). Analytic correlator with different k_min cuts: for an infinite Universe (black) and different limited boxsizes L, where the smallest allowed wavenumber k_min corresponds to the fundamental frequency of that boxsize, $k_L = 2\pi/L$.

3.2 Estimation and prediction of Gaussian random fields

In this section, we turn to the problem of reconstructing cosmological density and velocity fields from observational data. For this purpose, the ICeCoRe code implements the technique of Wiener filtering. At first, this seems unrelated to the problem of generating initial conditions. However, the Wiener Filter is a technique formulated in the framework of linear perturbation theory and Gaussian random fields, and as such, it is ideally suited to be applied to the reconstruction and generation of cosmological initial conditions. This leads to the constrained realisations (CR) technique discussed in Section 3.3.

3.2.1 The Wiener Filter

The Wiener Filter was developed by Wiener (1949) in the context of signal processing. It was designed to reduce the amount of noise present in a signal and to generate an estimation of the underlying true, i.e. noiseless signal. This is accomplished by assuming statistical prior information on the true signal. Eventually the Wiener filter found its way into astrophysics (Rybicki & Press 1992) and is now widely used for linear estimation and prediction in various astrophysical applications. The application of the Wiener Filter to cosmology was pioneered by Zaroubi et al. (1995, 1999), who developed a theoretical framework that allows one to reconstruct the three-dimensional cosmological density and velocity fields from observational data like galaxy redshift surveys or galaxy peculiar velocity catalogues. The material presented in this section and its implementation in the ICeCoRe code are heavily based on their work (see also Hoffman 2009).

For the sake of simplicity, we assume here that the underlying field that we want to reconstruct is the cosmological overdensity field δ, although the formalism can be applied in the same way to any other random field. Let us assume that we have a set of M datapoints $c_1, \ldots, c_M$ at discrete positions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$ that sample the true field $\delta_1, \ldots, \delta_N$ defined at positions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. The data are further corrupted by statistical errors ϵ_i. This can be expressed as[7]

$$c_i = \sum_{j=1}^{N} R_{ij}\delta_j + \varepsilon_i \quad . \tag{3.10}$$

The matrix R_{ij} is a response function that encodes the mapping from the true field to the observed quantity and is assumed to be known. In general, this can include the response of the measuring device, such as blurring or other distortions. Here, we consider the specific case where the observed quantity that goes into a datapoint c_i is a linear functional $V_i(\delta)$ of the overdensity field δ. Then, we can write

$$c_i = V_i(\delta) + \varepsilon_i \quad ; \quad V_i(\delta) = \sum_{j=1}^{M} R_{ij}\delta_j \quad , \tag{3.11}$$

where the observed quantity $V_i(\delta)$ is statistically related to the full overdensity field $\delta(\boldsymbol{x})$ via

$$V_i(\delta) = \frac{1}{(2\pi)^3} \int \delta(\boldsymbol{k}) Y_i(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}_i} \,\mathrm{d}\boldsymbol{k} \quad , \tag{3.12}$$

[7] For clarity, we write all matrix operations component-wise, and for all matrix/vector multiplications, we write out the sums instead of using the summation convention. This notation differs somewhat from the notation used in Zaroubi et al. (1995, 1999); Hoffman (2009). The vector notation is reserved here for spatial vectors with three cartesian components, for which we continue to use symbols in bold italics.

with a kernel $Y_i(\boldsymbol{k})$ that characterises the type of the functional. If the data are values of the density field itself, $V_i(\delta) = \delta$, then we have simply $Y_i(\boldsymbol{k}) = 1$. Here, it is of special interest to consider the case that $V_i(\delta)$ is the linear peculiar velocity field $\boldsymbol{u}$ with $-\boldsymbol{\nabla} \cdot \boldsymbol{u} = \dot{a} f \delta$. In this case, for any cartesian component u_α of $\boldsymbol{u}$, we know from Section 2.2.5 that

$$V_i(\delta) = u_\alpha \quad \Longrightarrow \quad Y_i(\boldsymbol{k}) = \dot{a} f \left(\frac{-ik_\alpha}{k^2} \right) \quad ; \quad \alpha \in \{x, y, z\} \quad . \tag{3.13}$$

The task of reconstructing the underlying density field from such a measurement can now be expressed as an inversion or deconvolution problem. The naive approach would be to try to directly invert the response matrix R_{ij} and evaluate

$$\delta_i = \sum_{j=1}^{M} R_{ij}^{-1} \, c_j \quad , \tag{3.14}$$

but this approach fails for two reasons. First, normally the number M of datapoints is much less than the number N of locations that define the underlying field, and for $M \ll N$ the equation 3.14 is underdetermined and the inversion cannot be performed. But even if one has a complete sampling, $M \approx N$, the inversion is not stable if one has observational errors ε_i. Any attempt to directly evaluate equation 3.14 would greatly amplify the noise from the errors and render the obtained solution unusable. In order to stabilise the inversion, one needs a procedure that is capable of suppressing the noise. We want to find a filtering operator F, which in the simplest case is a linear filter,

$$\delta_i^F = \sum_{j=1}^{M} F_{ij} \, c_j \quad , \tag{3.15}$$

where F is an $N \times M$ matrix. The optimal linear filter is the one that minimises the mean squared error of the estimated field δ^F in comparison with the true field δ. In other words, if one defines the residual field D as the difference between the estimated and the true field,

$$D_i = \delta_i - \delta_i^F \quad , \tag{3.16}$$

then the variance of this residual,

$$\langle D_i D_j \rangle = \langle (\delta_i - \delta_i^F)(\delta_j - \delta_j^F) \rangle \quad , \tag{3.17}$$

must become minimal. Carrying out the minimisation, one finds the minimal variance estimator or the Wiener Filter (WF),

$$\mathrm{WF}_{ij} = \sum_{k=1}^{M} \langle \delta_i c_j \rangle \, \langle c_j c_k \rangle^{-1} \quad , \tag{3.18}$$

where $\langle c_i c_j \rangle$ is the autocorrelation matrix of the data[8] and $\langle \delta_i c_j \rangle$ is the cross-correlation matrix of the original field with the data. Then, the estimated field, termed the Wiener filter mean field, is given by

$$\delta_i^{\mathrm{WF}} = \sum_{j=1}^{M} \sum_{k=1}^{M} \langle \delta_i c_j \rangle \, \langle c_j c_k \rangle^{-1} c_k \quad . \tag{3.19}$$

[8] The notation $\langle c_i c_j \rangle^{-1}$ means that we take the ij component of the *inverse* autocorrelation matrix (and not the inverse of the ij component of the matrix).

The method relies on the prior knowledge of the involved correlations. We can greatly simplify the problem if we assume that the errors ε_i consist of purely statistical noise, i.e. they are not correlated with the data c_i and therefore also not correlated with the field δ_i. Then, the cross-correlation and autocorrelation matrices become:

$$\langle \delta_i c_j \rangle = \langle \delta_i V_j(\delta) \rangle \quad , \tag{3.20}$$

$$\langle c_i c_j \rangle = \langle V_j(\delta) V_k(\delta) \rangle + \langle \varepsilon_j \varepsilon_k \rangle \quad . \tag{3.21}$$

In order to compute the correlations, we can use the fact that the correlation function is completely determined by the known power spectrum $P(k)$ (equation 2.49). Then we can directly compute the autocorrelation function and cross-correlation function for any linear functional V_i of δ, using the corresponding kernel $Y_i(\boldsymbol{k})$ of the functional,

$$\langle \delta_i V_j(\delta) \rangle = \frac{1}{(2\pi)^3} \int Y_j(\boldsymbol{k}) P(k) e^{-i\boldsymbol{k}\cdot(\boldsymbol{x}_j - \boldsymbol{x}_i)} \, \mathrm{d}\boldsymbol{k} \quad , \tag{3.22}$$

$$\langle V_i(\delta) V_j(\delta) \rangle = \frac{1}{(2\pi)^3} \int Y_i(\boldsymbol{k}) Y_j(\boldsymbol{k}) P(k) e^{-i\boldsymbol{k}\cdot(\boldsymbol{x}_j - \boldsymbol{x}_i)} \, \mathrm{d}\boldsymbol{k} \quad . \tag{3.23}$$

If we now additionally assume that the observational errors ε_i are Gaussian distributed, which works in a lot of cases[9], and if we know their variance, we can also evaluate the autocorrelation matrix of the errors, $\langle \varepsilon_i \varepsilon_j \rangle$. This is straightforward if we add yet another assumption, namely that the errors are not correlated with each other, i.e.

$$\langle \varepsilon_i \varepsilon_j \rangle = \delta^K_{ij} \varepsilon_j \quad , \tag{3.24}$$

where δ^K is the Kronecker delta. Then, we have all needed quantities and are able to evaluate the WF operator and perform the WF reconstruction. Numerically, the method relies on a (potentially large) matrix inversion and an efficient way to compute the correlation functions for all necessary variable combinations. In Section 3.4 we will discuss how that can be accomplished in practice and present the implementation in the ICECORE code.

If the locations $x_1, \ldots, x_N$ where the WF is evaluated correspond to the locations $x_1, \ldots, x_M$ where data is available, then the WF performs an estimation, i.e. it just filters the noise from the data. However, the formalism also enables us to evaluate the WF at positions with no data. In this case, it performs a prediction of the field, interpolating or extrapolating the data into unsampled regions. In the following, we will evaluate the WF on a cubic computational grid similar to the one used for initial conditions. This way we can obtain an estimate of the full field $\delta(\boldsymbol{x})$, regularly sampled at all discrete grid cell positions $\boldsymbol{x}$, by evaluating the WF operator for all $\boldsymbol{x}$,

$$\delta^{\mathrm{WF}}(\boldsymbol{x}) = \sum_{i=1}^{M} \sum_{j=1}^{M} \langle \delta(\boldsymbol{x}) c_i \rangle \, \langle c_i c_j \rangle^{-1} c_j \quad . \tag{3.25}$$

In practice, the WF operator is evaluated in four steps. First, we construct and invert the data autocorrelation matrix $\langle c_i c_j \rangle$. Then, we compute the correlation vector η_i, a vector of rank M,

$$\eta_i = \sum_{j=1}^{M} \langle c_i c_j \rangle^{-1} c_j \quad , \tag{3.26}$$

[9] Strictly speaking, it is not entirely accurate to treat the observational errors of observed peculiar velocities as Gaussian distributed: they are actually expected to be lognormal distributed, because the measured quantities, the galaxy luminosity and the corresponding distance modulus μ, are logarithmic quantities. Likewise, it is not expected that the observational errors would be entirely uncorrelated. It was however not investigated yet whether a more accurate treatment of the errors could improve the reconstruction.

using the mock and actual constraints and the inverted data autocorrelation matrix. In the last step, we can evaluate

$$\delta^{\mathrm{WF}}(\boldsymbol{x}) = \sum_{i=1}^{M} \langle \delta(\boldsymbol{x}) c_i \rangle \, \eta_i \tag{3.27}$$

for each grid cell $\boldsymbol{x}$. The values of the correlation vector η_i can provide interesting additional information about how much each datapoint c_i contributes to the full solution.

Analogously we obtain the reconstructed velocity field $\boldsymbol{u}$. For each cartesian component u_α, the Wiener filter mean field is

$$u_\alpha^{\mathrm{WF}}(\boldsymbol{x}) = \sum_{i=1}^{M} \sum_{j=1}^{M} \langle u_\alpha(\boldsymbol{x}) \, c_i \rangle \, \langle c_i c_j \rangle^{-1} c_j \quad ; \quad \alpha \in \{x, y, z\} \quad . \tag{3.28}$$

This allows one, for example, to obtain an estimate of the full peculiar velocity field $\boldsymbol{u}$ from discrete measurements of only its radial components, such as in galaxy peculiar velocity catalogues. This particular technique will be heavily used in Chapters 4 and 5.

So far we have placed no restrictions on the statistics of δ except that its power spectrum or two-point correlation function is known: by minimising the variance, the WF neglects all higher statistical moments. In the general case, for instance in signal processing and data analysis, the WF is considered to be a rather simple and 'naive' tool for this reason, and usually much better filters exist that are adapted to the specific problem. However, this limitation turns into a virtue for Gaussian random fields, where the two-point correlation function defines all statistical properties, and all higher statistical moments vanish. In this case, it can be shown that the WF is the optimal possible estimator. In linear Gaussian random field theory, the WF can also be regarded as a Bayesian reconstruction method, and derived using Bayesian probability theory (Zaroubi et al. 1995). If we know the probability distribution function of both the data and the underlying field (the *prior model*, which in the linear case is completely determined by the power spectrum), we can compute the conditional probability of a field given the data. This leads to two possible estimators: the maximum a posteriori estimator, which gives the most probable field given the data, and the conditional mean field, which averages over all possibe fields weighted with their respective probability. For Gaussian random fields, it can be shown that both estimators coincide and are equal to the WF estimator which minimises the variance of the residual. The WF is therefore the optimal reconstruction method on data for which the nature of the underlying field is Gaussian. For this reason, the WF is an established and widely used method for the reconstruction of the large-scale structure from both galaxy peculiar velocity catalogues (Zaroubi et al. 1999; Courtois et al. 2012) and galaxy redshift surveys (Fisher et al. 1995; Erdoğdu et al. 2006; Kitaura et al. 2009)[10]. Even more important for the work presented here, the WF is the perfect starting point for a reconstruction of the cosmological initial conditions, which are Gaussian random fields by nature.

The Wiener filter is a very conservative estimator, in the sense that it always tends to strongly suppress the present noise and not to create additional noise. One can see from equations 3.18 and 3.20 – 3.21 that the WF operator effectively invokes a weighting of the type signal/(signal+noise). Therefore, in the presence of strong noise, the estimation will tend to the mean field of a random realisation, i.e. the null field (equation 2.48). The same will happen for prediction, i.e. the extrapolation into unsampled regions, if they are not sufficiently correlated with the datapoints.

[10] In the case of galaxy redshift surveys, the WF is often formulated in spherical harmonic space, instead of the formulation in cartesian real and Fourier space discussed here; see Fisher et al. (1995).

For this reason, the Wiener filter cannot be applied iteratively: each reconstruction will by definition have less power than the used input, and any such iteration will converge towards the null field.

3.2.2 Reconstruction of the LSS from peculiar velocity data

The application of the WF to peculiar velocity data is a powerful reconstruction method for the large-scale structure and in particular the peculiar velocity field of the Local Universe. Due to the long correlation distance of peculiar velocities, the WF reconstruction is able to extrapolate the field out to distances significantly larger than the data zone. This allows one to study the cosmography and the large-scale velocity field of the Local Universe in detail and decompose it into the different contributing flows. A related property of the WF when applied to peculiar velocities is that it can interpolate into unsampled regions, such as the Zone of Avoidance, in a physically consistent way. The WF reconstruction from peculiar velocities is not affected by incompleteness of the data as long as the underlying individual distances are not subject to biases

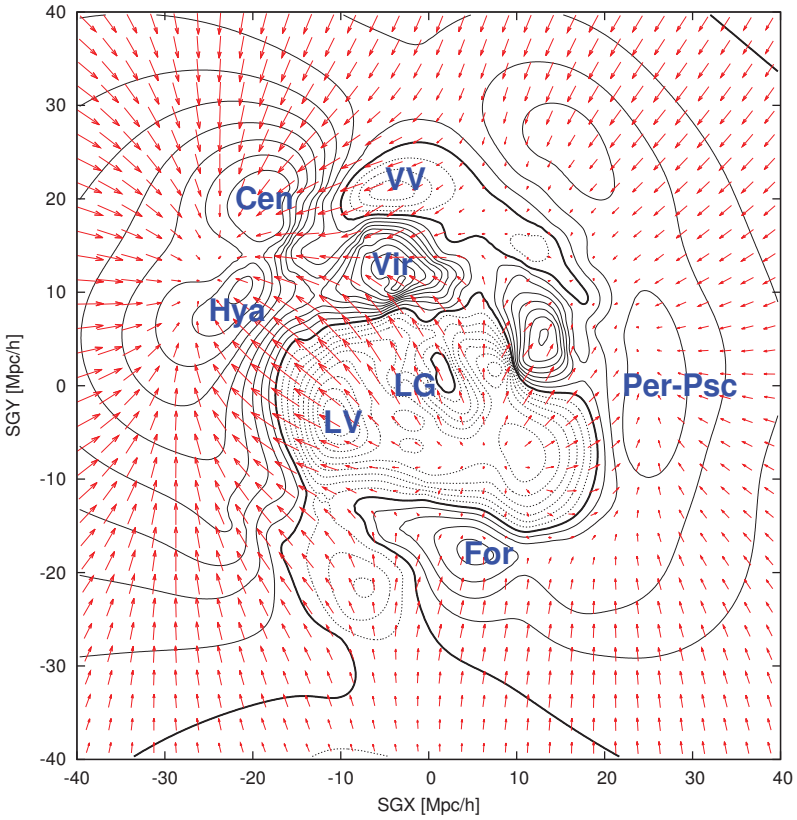

Figure 3.2: Wiener Filter reconstruction from radial peculiar velocities computed with ICECORE and WMAP7 cosmological parameters from the 742 galaxy group radial peculiar velocities in the Cosmicflows-1 catalogue. Shown is a slice of the overdensity field (contour lines) and velocity field (arrows) through the $SGZ = 0$ plane. The thick contour line is placed at the cosmic mean density ($\delta = 0$); the solid lines are isocontour lines in overdense regions; the dashed lines follow underdense regions. Peaks and voids associated with known objects are labelled as follows: LG = Local Group, LV = Local Void, Vir = Virgo cluster, VV = Virgo Void, For = Fornax cluster, Hya = Hydra cluster, Cen = Centaurus cluster, Per-Psc = Perseus-Pisces clusters.

in the distance measurements (Courtois et al. 2012). It can therefore self-consistently operate on data that are sparse in a non-homogeneous way. This is not possible with galaxy redshift data.

Figure 3.2 shows the WF reconstruction of the density field (contour lines) and velocity field (arrows) from the Cosmicflows-1 catalogue discussed in Section 2.1.2. The reconstructed field recovers the main structures very well, such as the Virgo cluster (at about $SGX = -5$; $SGY = 10$), the Great Attractor region with the Hydra and Centaurus clusters (towards negative SGX from Virgo), the Local Void, and the Perseus-Pisces cluster. In the centre of the reconstruction, an environment similar to the neighbourhood of the Local Group with its cold Hubble flow is created. We refer the reader to Courtois et al. (2012) for a detailed analysis of the WF reconstruction from Cosmicflows-1.

A major motivation for the development of ICECORE was to provide a numerical package that is able to perform this kind of analysis for the large upcoming datasets, in particular those that are expected to come out from the Cosmicflows program in the next few years (Courtois et al. 2011a,b; Courtois & Tully 2012; Tully & Courtois 2012). These will reach out to 6000 km/s and eventually even 15 000 km/s; the data will comprise of the order of 10^4 – 10^5 galaxy distances and radial peculiar velocities. The performance of ICECORE's WF implementation is discussed in Section 3.4. Here, we want first to note some details about how to handle this type of observational data in the chosen numerical framework.

Galaxy peculiar velocities provide only information on the radial component $v_r^{\rm pec}$ of the three-dimensional velocity vector $\boldsymbol{v}^{\rm pec}$ (which at $z = 0$ means the same as $\boldsymbol{u}$) at some position $\boldsymbol{r}$ (which at $z = 0$ is equivalent to the comoving position $\boldsymbol{x}$ that we use here). Each datapoint therefore constrains the value of $u_\mu = \boldsymbol{u} \cdot \hat{\boldsymbol{e}}_\mu$, the component of $\boldsymbol{u}$ along some arbitrarily directed unit vector $\hat{\boldsymbol{e}}_\mu$. In an observational dataset, this direction will be always $\hat{\boldsymbol{e}}_\mu = (\boldsymbol{r} - \boldsymbol{r}_0)/|\boldsymbol{r} - \boldsymbol{r}_0|$, where $\boldsymbol{r} - \boldsymbol{r}_0$ is the distance vector of a galaxy relative to the position of the observer $\boldsymbol{r}_0$ at the centre of the data volume. However, here we do not want to restrict the algorithm to this specific case, so we allow $\hat{\boldsymbol{e}}_\mu$ to be any arbitrary direction. This also means that if in the data $\hat{\boldsymbol{e}}_\mu$ is, in fact, restricted to the radial direction with respect to the observer $\boldsymbol{r}_0$, we are still free to place $\boldsymbol{r}_0$ at any position inside the computational volume. To allow an arbitrary direction of $\hat{\boldsymbol{e}}_\mu$ gives ICECORE a greater flexibility. For example, as a recent development it is expected that one can obtain estimates of the 2D transverse peculiar velocities of distant galaxies (Nusser et al. 2012) from measurements of proper motions within the Gaia survey (e.g. Robin et al. 2012). In this case, the data will contain pairs of values u_μ, u_ν, where $\hat{\boldsymbol{e}}_\mu, \hat{\boldsymbol{e}}_\nu$ are both orthogonal to the radial direction $(\boldsymbol{r} - \boldsymbol{r}_0)$. If one then sets the $\hat{\boldsymbol{e}}_\mu, \hat{\boldsymbol{e}}_\nu$ accordingly, ICECORE will be able to directly generate a WF reconstruction from such data without changing anything in the code. The main reason though for keeping the $\hat{\boldsymbol{e}}_\mu$ arbitrary is that this is required for the method of generating ICs that we will use throughout Chapter 5.

For any constraint on one peculiar velocity component u_μ in arbitrary direction $\hat{\boldsymbol{e}}_\mu$, we can formulate the necessary correlation functions (Zaroubi et al. 1999):

$$\langle \delta(\boldsymbol{x}') u_\mu(\boldsymbol{x}' + \boldsymbol{x}) \rangle = \hat{\boldsymbol{e}}_\mu \cdot \langle \delta(\boldsymbol{x}') \boldsymbol{u}(\boldsymbol{x}' + \boldsymbol{x}) \rangle \tag{3.29}$$

for the density-velocity correlation and

$$\langle u_\mu(\boldsymbol{x}') u_\nu(\boldsymbol{x}' + \boldsymbol{x}) \rangle = \hat{\boldsymbol{e}}_\mu \cdot \langle \boldsymbol{u}(\boldsymbol{x}') \boldsymbol{u}(\boldsymbol{x}' + \boldsymbol{x}) \rangle \cdot \hat{\boldsymbol{e}}_\nu \tag{3.30}$$

for the velocity-velocity correlation. We can therefore compute them from the already known correlations of the type 3.22 and 3.23 with constraint kernels 3.13. Further, the linear theory assumption means that the correlations are given by the known linear power spectrum $P_0(k)$ of the Universe, serving as the prior model. The WF reconstruction assumes that the data is tracing

the linear density and velocity fields of the Universe, which is infinite, homogeneous and isotropic. Therefore, those correlations only depend on the distance $x = |\boldsymbol{x}|$ and we can reduce them to one-dimensional functions of x. In order to apply the WF operator to the data we thus have to evaluate the equations 2.66 – 2.71 already discussed in Chapter 2. Further, even though we evaluate the WF on a finite computational volume, the underlying field – the observed Universe – is not subject to boundary conditions, therefore the resulting WF reconstructed field will not have boundary conditions either, in contrast with the properties of cosmological simulations and their ICs.

We want to mention some additional details that have to be considered in order to practically handle observational peculiar velocity data within the numerical framework discussed here. Galaxy distances are given on the observed scale (Mpc, not Mpc/h), and therefore the peculiar velocity that we can compute from it explicitly depends on the value of the Hubble constant $H_0 = 100\ h$ km s^{-1} Mpc^{-1}. For the conversion to the comoving scale, one has to use a reasonable value for h. For Cosmicflows-1, we followed Tully et al. (2008) and chose a value of $h = 0.74$. This value is consistent with the distance measures and calibrated such that there is no net inflow or outflow within the observational volume, although it does not imply that this is the case in reality.

In extragalactic surveys, galaxy positions on the sky are usually given in supergalactic coordinates (latitude SGB and longitude SGL). These are defined by the supergalactic plane (see Lahav et al. 2000 for a precise definition), where $SGB = 0$, and a zero point $SGL = 0$ defined by the intersection of this plane with the galactic plane. Since for a given datapoint in a galaxy distance survey, we have an estimate of the physical distance r in real space, we can convert the galaxy positions to cartesian coordinates SGX, SGY, SGZ,

$$\begin{aligned} SGX &= r\ \cos(SGB)\ \cos(SGL) \\ SGY &= r\ \cos(SGB)\ \sin(SGL) \\ SGZ &= r\ \sin(SGB) \end{aligned} \quad , \tag{3.31}$$

where $SGZ = 0$ is the supergalactic plane and the position of the observer $\boldsymbol{r}_0$ is at $SGX = SGY = SGZ = 0$. It is important to remember that these positions have error bars in radial direction due to the observational distance errors. We can then associate the supergalactic coordinate axes with the axes x, y, z of the computational volume, and move the observer to some position inside the box, conveniently to the box centre at $\boldsymbol{r}_0 = \left(\frac{L}{2}, \frac{L}{2}, \frac{L}{2}\right)$.

Knowledge of the three-dimensional coordinates also allows for a transformation of the radial peculiar velocity $v_r^{\rm pec}$ to different rest frames, such as the rest frame of the Galaxy, the Local Sheet, or the rest frame with respect to the CMB dipole. The conversion formulae are given in Tully et al. (2008); we will not go into detail here. The chosen rest frame of the $v_r^{\rm pec}$ will correspond to the rest frame of the computational box. It depends on the application which rest frame one should use. For example, if the aim is to reconstruct the cosmic flows out to the largest possible distances, it would be natural to adopt the CMB rest frame, while one could use the rest frame of the Local Sheet or Local Supercluster for more local reconstructions.

The WF reconstructed density field is not to be directly compared to the true underlying density field at $z = 0$, at least not to the extent it is possible with the reconstructed velocity field. The reason is that the actual density field of the Universe at $z = 0$ is highly non-linear, and only adequately described by linear theory at scales of several tens of Mpc/h and above (see Section 2.2.7). So what the WF actually produces is an estimate of the *linear* component of the density field, or in other words, an estimate of the density field as it would be if linear theory would be valid at all scales until $z = 0$. The fact that the estimated $\delta^{\rm WF}$ can take values < -1, which would be equivalent to negative densities, is therefore not an issue if we interpret the WF field

in this context. The estimated linear field can still provide very valuable information even on scales where linear theory is, strictly speaking, not a valid assumption. As observed by Courtois et al. (2012), the WF density field accurately reproduces features like the Local Void and the density peaks produced by galaxy clusters at their actual positions, and is therefore a powerful tool for investigating the cosmography of the Local Universe.

3.3 Constrained realisations

We now turn to the main subject of this thesis: the generation of cosmological initial conditions for constrained simulations of the Local Universe. Essentially we want to obtain an estimate of the Gaussian random field $\delta_0(\boldsymbol{x})$ that describes the matter distribution of the observed Local Universe at some early redshift $z_{\rm init}$ in the linear phase of the matter-dominated epoch. Let us assume now that we have a set of constraints $c_1. \ \ldots \ c_M$ that define, at discrete positions $\boldsymbol{x}_1. \ \ldots \ \boldsymbol{x}_M$, the value of a linear functional of the overdensity field $\delta_0(\boldsymbol{x})$ that we seek to reconstruct. It will be discussed later how such constraints can be obtained. For now, given the constraints, the problem consists of generating a Gaussian random field subject to the constraints c_i, i.e. it must adhere to the constrained values at the constrained positions $\boldsymbol{x}_i$, while at the same time being a valid realisation of the linear power spectrum $P_0(k)$ as discussed in Section 3.1, so that this field can be used to set up cosmological initial conditions.

At first sight, the problem seems analogous to WF reconstruction, with the constraints c_i being the same thing as the data points c_i discussed before. This is not too far from the truth. However, the field produced by WF reconstruction is not a typical realisation of a Gaussian random field with power spectrum $P_0(k)$, since it is a non-power-preserving filter and tends towards the null field in regions not covered by constraints. The same holds in Fourier space: the WF mean field will have a vanishing power spectrum at small scales that are unconstrained by the data. In other words, the WF mean field is not statistically homogeneous, because the variance of the residual (3.33), although minimised by the WF, is definitely not negligible. It follows from the definition 3.16 that the true field $\delta(\boldsymbol{x})$ is the sum of the WF mean field, $\delta^{\rm WF}(\boldsymbol{x})$, and the residual $D(\boldsymbol{x})$. Of course, the true actual residual is unknown. But if we can obtain a realisation of $D(\boldsymbol{x})$ that has the same *statistical* distribution as the true residual, it follows that we can construct a constrained realisation $\delta^{\rm CR}$ with

$$\delta^{\rm CR}(\boldsymbol{x}) = \delta^{\rm WF}(\boldsymbol{x}) + D(\boldsymbol{x}) \quad , \tag{3.32}$$

with the desired property of statistical homogeneity. The problem of constructing a constrained Gaussian random field therefore reduces to the task of generating an adequate residual.

3.3.1 The Hoffman-Ribak algorithm

The properties of constrained Gaussian random fields were first studied by Bardeen et al. (1986). It was then recognised by Bertschinger (1987) that CRs could be used to set up constrained initial conditions to produce custom-tailored cosmological simulations. He suggested the first practical algorithm to generate CRs, an iterative method that relies on path integrals to describe the statistics of the Gaussian field. However, due to the slow convergence his method could be applied in practice only with a very limited number of constraints. A more efficient iterative method was developed later by Hoffman & Ribak (1992). The first exact, non-iterative algorithm to generate CRs was described by Binney & Quinn (1991), who formulated the problem in spherical harmonics space, but with the downside that constraints of the form $c_i = V_i(\delta)$ could be placed only localised around a single location (the origin of the spherical harmonics expansion).

The exact optimal algorithm was finally discovered by Hoffman & Ribak (1991). Their key observation was that the residual field $D(\boldsymbol{x})$ that one needs to generate is independent of the values of the constraints c_i. Namely, the variance of the residual is (Hoffman & Ribak 1991, 1992):

$$\langle D(\boldsymbol{x}_1)D(\boldsymbol{x}_2)\rangle = \langle\delta(\boldsymbol{x}_1)\delta(\boldsymbol{x}_2)\rangle - \sum_{i=1}^{M}\sum_{j=1}^{M}\langle\delta(\boldsymbol{x}_1)\,c_i\rangle\,\langle c_i c_j\rangle^{-1}\langle\delta(\boldsymbol{x}_2)\,c_j\rangle \quad , \tag{3.33}$$

where $\langle\delta(\boldsymbol{x}_1)\delta(\boldsymbol{x}_2)\rangle$ is the variance of the unconstrained density field determined by the prior model $P_0(k)$. The residual $D(\boldsymbol{x})$ is itself a zero-mean Gaussian random field, so the variance 3.33 fixes all its statistical properties. Note that it is completely determined by the cross-correlation of the data and the field and their respective autocorrelations. Nowhere in equation 3.33 do the actual values of the constraints c_i enter. We can therefore easily construct a constrained realisation. First we generate a random realisation (RR) on the full computational grid, $\delta^{\mathrm{RR}}(\boldsymbol{r})$, as described in Section 3.1. We then create a set of mock constraints $\tilde{c}_i$ by "observing" the random realisation at the constrained positions:

$$\tilde{c}_i = V_i(\delta^{\mathrm{RR}}) \quad . \tag{3.34}$$

Then the WF mean field of δ^{RR} for this set of mock constraints is

$$\tilde{\delta}^{\mathrm{WF}}(\boldsymbol{x}) = \sum_{i=1}^{M}\sum_{j=1}^{M}\langle\delta(\boldsymbol{x})c_i\rangle\,\langle c_i c_j\rangle^{-1}\tilde{c}_j \quad , \tag{3.35}$$

where we use the tilde to distinguish it from the WF mean field of the actual constraints. The associated residual is

$$\tilde{D}(\boldsymbol{x}) = \delta^{\mathrm{RR}}(\boldsymbol{x}) - \sum_{i=1}^{M}\sum_{j=1}^{M}\langle\delta(\boldsymbol{x})c_i\rangle\,\langle c_i c_j\rangle^{-1}\tilde{c}_j \quad , \tag{3.36}$$

which can be easily evaluated because the correlations and $\tilde{c}_i$ are known. Since $\tilde{D}(\boldsymbol{x})$ does not depend on the values of the constraints $\tilde{c}_i$, it follows that $\tilde{D}(\boldsymbol{x})$ is a valid realisation of $D(\boldsymbol{x})$, because c_i and $\tilde{c}_i$ constrain the same quantities $V_i(\delta)$ at the same positions and therefore their autocorrelations and field cross-correlations are identical. Therefore, a valid constrained realisation can be obtained by

$$\delta^{\mathrm{CR}}(\boldsymbol{x}) = \delta^{\mathrm{WF}}(\boldsymbol{x}) + \tilde{D}(\boldsymbol{x}) \quad , \tag{3.37}$$

or in full form

$$\delta^{\mathrm{CR}}(\boldsymbol{x}) = \delta^{\mathrm{RR}}(\boldsymbol{x}) + \sum_{i=1}^{M}\sum_{j=1}^{M}\langle\delta(\boldsymbol{x})c_i\rangle\,\langle c_i c_j\rangle^{-1}\,(c_j - \tilde{c}_j) \quad . \tag{3.38}$$

Note that in the absence of data, equation 3.38 reduces to $\delta^{\mathrm{CR}} = \delta^{\mathrm{RR}}$, i.e. the constrained realisation is dominated by the random modes. Conversely, at the positions $\boldsymbol{x}_i$ of the constraints, $\delta^{\mathrm{RR}}(\boldsymbol{x}_i)$ cancels against $\tilde{c}_i$, and the CR is completely determined by the constraint ($\delta^{\mathrm{CR}} = \delta^{\mathrm{WF}}$). At all other positions, there will be a smooth transition between both regimes, depending on the data quality. Similar to the WF reconstruction, this equation is in practice decomposed into computing the correlation vector η_i, which is now of the form

$$\eta_i = \sum_{j=1}^{M}\langle c_i c_j\rangle^{-1}\,(c_j - \tilde{c}_j) \quad , \tag{3.39}$$

and then evaluating

$$\delta^{\mathrm{CR}}(\boldsymbol{x}) = \delta^{\mathrm{RR}}(\boldsymbol{x}) + \sum_{i=1}^{M} \langle \delta(\boldsymbol{x}) c_i \rangle \, \eta_i \quad . \tag{3.40}$$

In a completely analogous way, we can also generate the velocity field of the CR, given by

$$u_\alpha^{\mathrm{CR}}(\boldsymbol{x}) = u_\alpha^{\mathrm{RR}}(\boldsymbol{x}) + \sum_{i=1}^{M} \sum_{j=1}^{M} \langle u_\alpha(\boldsymbol{x}) \, c_i \rangle \, \langle c_i c_j \rangle^{-1} \, (c_j - \tilde{c}_j) \quad ; \quad \alpha \in \{x, y, z\} \quad . \tag{3.41}$$

Figure 3.3 illustrates the RR, WF, CR, and residual, in this example for the case of constraining one cartesian component of the peculiar velocity field $\boldsymbol{u}$, with a very simple setup of only two constraints at different positions.

The Hoffman-Ribak method can be straightforwardly applied to constraints with statistical uncertainties, $c_i = V_i(\delta) + \varepsilon_i$. Then, $\langle c_i c_j \rangle$ is defined by 3.21. In this case, at the constrained positions $\boldsymbol{x}_i$ the CR stays fixed to the value of the WF mean field, but this is not anymore the exact input value c_i, but instead an estimate of the underlying de-noised true field, which is the desired behaviour.

The imposed constraints can be of any type that is a linear functional of the overdensity. Besides constraints on the overdensity itself and the peculiar velocity field, this can also include for example the density gradient, curvature, the flow shear, and the shape of the gravitational potential at the constrained positions $\boldsymbol{x}_i$. Each such constraint can be described as a convolution of δ with a function V_i, with a corresponding convolution kernel $Y_i(\boldsymbol{k})$ in Fourier-space. A table of the kernels for 18 different constraint types can be found in van de Weygaert & Bertschinger (1996). A particularly interesting possibility is to constrain functionals of the *smoothed* field instead of the exact field. As discussed in Section 2.2.4, the density field has a lot of power on small scales, and most often one is interested in features of the field smoothed on a particular length scale, which can also be constrained using the Hoffman-Ribak algorithm. A smoothing filter is likewise a linear functional of the field, which can be combined with the constraint kernel. Therefore, we can extend the definition of a constraint (3.12) to

$$V_i(\delta) = \frac{1}{(2\pi)^3} \int \delta(\boldsymbol{k}) Y_i(\boldsymbol{k}) W(\boldsymbol{k}) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}_i} \, \mathrm{d}\boldsymbol{k} \quad , \tag{3.42}$$

where $W(\boldsymbol{k})$ is a smoothing kernel. Here, we will exclusively use the Gaussian smoothing kernel (2.54) with smoothing radius R_{G} for this purpose, although in principle any other linear filter would be possible. It is straightforward to compute the autocorrelation and the cross-correlation for such smoothed constraints by replacing $Y_i(\boldsymbol{k})$ with $Y_i(\boldsymbol{k})W(\boldsymbol{k})$.

3.3.2 Constrained simulations and the CLUES method

The Hoffman-Ribak algorithm provides a practical and powerful method to generate custom-tailored initial conditions. Running such initial conditions forward with N-body simulations gives rise to constrained simulations. With such constrained simulations it is possible to study features of the large-scale structure that would be otherwise rare to find in ordinary N-body simulations due to the high cosmic variance of random realisations.

Applications of this powerful machinery can be generally divided into two categories. One approach is to set up ICs by introducing particular features like peaks and voids with predefined properties and at predefined positions and then study their evolution in the context of non-linear structure formation. This is an interesting approach for the theoretical study of the formation of

haloes, filaments and other objects and how their formation and evolution is influenced by their cosmological environment; see e.g. van de Weygaert & Bertschinger (1996); Antonuccio-Delogu et al. (2002); Romano-Diaz et al. (2006, 2011).

The other direction is to generate constrained simulations that resemble the observed Local Universe. These can serve as an ideal numerical laboratory to study the formation and evolution of the observed structure. This approach entails the task to obtain a set of constraints c_i from observational data. This involves several steps: first, one has to cast the data into a form such that the datapoints are a tracer of the underlying field in the sense of equation 3.10. Then, this field has to be taken backward in time to the desired initial redshift $z_{\rm init}$ of the simulation. Finally, a set of constrained ICs for the simulation can be constructed. They may then be evolved

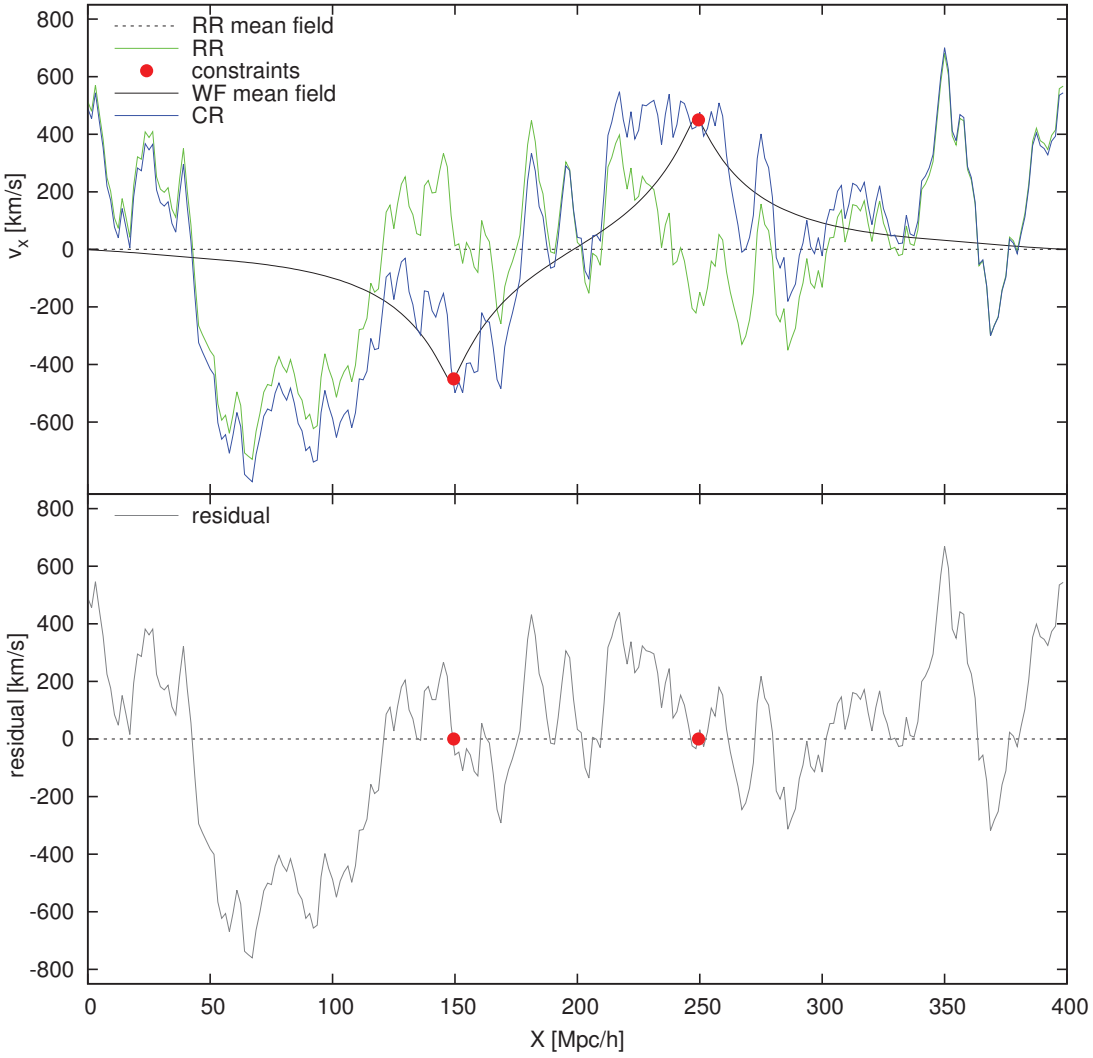

Figure 3.3: Constraining a Gaussian random field (here, v_x component of a cosmological peculiar velocity field at two positions). Top panel: two constraints (red points) are placed on top of a random realisation (RR, green), which results in a constrained realisation (CR, blue). The CR is equal to the Wiener Filter mean field (WF, black) at the positions of the constraints. At the same time, the CR is dominated by the RR in unconstrained regions where the WF falls off to zero. Bottom panel: the residual (grey) equals zero at the positions of the constraints and approaches the RR in unconstrained regions.

forward until $z = 0$ using an N-body code. This procedure was first explored by Kolatt et al. (1996) and Bistolas & Hoffman (1998), who used galaxy redshift data. It was soon discovered that peculiar velocity data may provide a better candidate for generating constraints, since they constrain the underlying field more strongly and are closer to the Gaussian statistics assumed by the WF/CR algorithm. Such constrained simulations using peculiar velocities, namely the MARK III catalogue (Willick et al. 1997), were first carried out by van de Weygaert & Hoffman (1999, 2000) and used to study the origin of the cold local Hubble flow. The first high-resolution simulations that were aimed at reproducing a significant portion of the observed Local Universe with reasonable accuracy were finally presented by Klypin et al. (2003). This method finally gave rise to the CLUES project, where constrained simulations from peculiar velocity data are applied to study various aspects of the Local Universe (see the overview in Gottlöber et al. 2010). In this work, we are building on the CLUES method in order to add several improvements. First, we briefly review the current method.

The method rests on the assumption that the peculiar velocity field of the Local Universe, observed through peculiar velocities of galaxies, does not deviate too much from the linear velocity field of the ICs at some early redshift $z_{\rm init}$, so that we can directly use the values of the radial peculiar velocities $v_r^{\rm pec}$ as constraints c_i. This approach differs from the WF reconstruction from such data by the requirement that the result must be a linear Gaussian random field suitable for initial conditions. One obvious source of incompatible non-linearities in the data are the virial motions of galaxies gravitationally bound to larger objects such as galaxy clusters. This can be overcome by an appropriate grouping of the data points, which effectively "linearises" the data. Virial motions are, of course, not the only source of non-linearity. Another example of non-linearities in the observable peculiar velocity field at $z = 0$ is the general enhancement of peculiar motions due to local overdensities that we studied in Section 2.3.4. A useful quantity to gauge the linearity of the data is

$$\chi^2 = \sum_{i=1}^{M} \sum_{j=1}^{M} c_i \left\langle c_i c_j \right\rangle^{-1} c_j \quad . \tag{3.43}$$

Linear theory states that if the data are sampling a typical realisation of the assumed prior model $P(k)$, then χ^2 must be close to the number of degrees of freedom (dof),

$$\chi^2/\mathrm{dof} \approx 1 \quad , \tag{3.44}$$

where dof is equal to the number of datapoints M. This requirement is generally not fulfilled by peculiar velocity data at $z = 0$. Although theoretical models for some of the non-linear effects exist (Sheth et al. 2001; Hamana et al. 2003), unfortunately the task of obtaining an accurate mapping from the non-linear to the desired linear field seems unfeasible. However, the non-linearities can be compensated for in a statistical sense. Considering the non-linear effects as a form of statistical scatter, one can add a new non-linearity term to the autocorrelation matrix of the data (3.21). In the absense of a better model, this term is considered to be a scalar matrix, and is defined by introducing the non-linearity parameter $\sigma_{\rm NL}$ (Bistolas & Hoffman 1998):

$$\left\langle c_i c_j \right\rangle = \left\langle V_i(\delta) V_j(\delta) \right\rangle + \delta_{ij}^K \varepsilon_j^2 + \delta_{ij}^K \sigma_{\rm NL}^2 \quad , \tag{3.45}$$

where δ^K is the Kronecker delta. The value of $\sigma_{\rm NL}$ is chosen such that the condition 3.44 is fulfilled. For observational radial peculiar velocity data, we find a typical value of $\sigma_{\rm NL} \approx 200$ km/s. The initial conditions are then created as follows. First one constructs and inverts the autocorrelation matrix of the data (3.45). Then an appropriate boxsize must be chosen such that

the data zone lies well within the computational volume. Then, the WF/CR operator can be evaluated on this volume, leading to a linear density field $\delta_0^{\mathrm{CR}}(\boldsymbol{x})$. This field can then be scaled with the growth factor D_+ to the desired starting redshift z_{init} of the simulation (2.39) and used to set up N-body ICs.

For some CLUES simulations such as the BOX160 simulation, additional constraints were used to obtain representations of the clusters that lie outside the peculiar velocity data zone, such as the Coma and Perseus-Pisces clusters. The WF mean field from peculiar velocities does reproduce the overdense regions associated with clusters even if they lie outside the data zone (see Figure 3.2). However, their exact positions and masses are not constrained well in this case. To improve the situation, additional constraints were obtained from the catalogue of nearby X-ray selected clusters (Reiprich & Böhringer 2002). From this data it is possible to obtain estimates on the virial masses of the clusters. Then, one can derive the linear overdensities of the respective clusters assuming the spherical top-hat model. These linear overdensities were then imposed as constraints of the density type with a Gaussian smoothing radius R_{G} that corresponds to the mass scale of the cluster (Gottlöber et al. 2010). How the radius R_{G} for a constraint can be obtained from the mass will be discussed in Section 5.1.1. We will however not consider this type of constraints in the following chapters and continue to work exclusively with peculiar velocities.

3.3.3 Drawbacks of the CLUES method

With the described method of setting up constrained initial conditions, the constrained simulations robustly reproduce the configuration of observed clusters in the Local Universe, like the Local Supercluster, Perseus-Pisces and the Great Attractor region, and create a large-scale environment similiar to that of the Local Group. However, the structure on scales smaller than those clusters seems to be completely dominated by the random component. Additionally, the positions of the recovered clusters are subject to systematic errors and do not appear at their actual positions in the evolved $z = 0$ simulations. It is the aim of this work to develop methods to both reduce systematic errors and increase the range of scales where the cosmic structure can be effectively constrained.

One drawback of the method is the non-linearities in the observed peculiar velocity field are not handled well. While we can filter out statistical contributions and force the resulting CR to adhere to the prior model by using the additional σ_{NL} parameter, the result will still be influenced by systematic errors induced by non-linearity, i.e. the differences between the linearly extrapolated initial field and the full non-linear field. These differences are much smaller for the peculiar velocity field than for the density field, but still not negligible. As we saw in Section 2.3.3, the most prominent effect is the cosmic displacement $\boldsymbol{\psi}$ on a typical scale of 10 Mpc/h. We therefore expect that the quality of the constrained ICs and the resulting simulations can be improved considerably if we replace the simple linear theory scaling of the CR from $z = 0$ to z_{init} (equation 2.42) with a Lagrangian reconstruction scheme that actually follows the cosmic displacement field back in time. Such a method will be presented in Chapter 4. Here, we will first focus on other inconsistencies of the method.

The CLUES method of setting up constrained simulations historically developed from the WF reconstruction of the large-scale structure from peculiar velocities. Just as for the WF reconstruction, the method uses the one-dimensional correlation functions ψ_{R}, ψ_{T} and ζ which describe – in the linear approximation – the homogeneous, isotropic and not volume-limited observed Universe. This WF mean field is then combined with the residual of a random realisation on the computational volume, which has periodic boundary conditions, is non-isotropic (because of the cubic shape) and volume-limited. Therefore, the computed WF mean field $\delta^{\mathrm{WF}}(\boldsymbol{x})$ and residual $D(\boldsymbol{x})$ are not statistically compatible. The non-periodicity of $\delta^{\mathrm{WF}}(\boldsymbol{x})$ poses no major

problem if the computational volume is chosen such that the WF mean field falls off to zero towards the boundaries. Then, δ^{CR} is dominated by δ^{RR} at the box edges, which has the required periodic boundary conditions. However, it can become very problematic for smaller boxsizes, and as we saw the bulk flow of the Local Universe is not expected to fall off to zero at least within 150 Mpc/h or so of the Local Group. To enforce periodic boundary conditions on the resulting constrained ICs even for smaller boxes, the following approach was used for CLUES simulations. First, the constrained overdensity field δ^{CR} is evaluated through 3.38 using the isotropic correlation functions. Then, the initial displacement field $\boldsymbol{\psi}^{\mathrm{CR}} = \boldsymbol{u}^{\mathrm{CR}}/\dot{a}f$ is not computed using equation 3.41, but instead with the FFT method 3.5. This effectively imposes periodic boundary conditions on $\boldsymbol{\psi}^{\mathrm{CR}}$ by subtracting the tidal component, i.e. the contribution due to overdensities outside of the box, from the displacement field. Since the displacement field at z_{init} completely defines the N-body particle distribution, one then obtains valid periodic ICs. Note that then, the velocity field $\boldsymbol{u}^{\mathrm{CR}}$ obtained by FFT does not follow the values of the constraints c_i anymore, because the tidal component was subtracted. The effective density field is less affected by the tidal component, so the CR can still produce a meaningful density distribution inside the box. This approach is not self-consistent though, and this way of imposing periodic boundary conditions is believed to cause further systematic errors.

Besides the periodic boundary conditions, two other effects become important: the finite-volume effect and the anisotropy of the effective correlation function. This means that the correlation functions imposed on δ^{WF} (given by 2.50) and on δ^{RR} (given by 3.4), respectively, are different. This inconsistency is not expected to be significant if the constraints are well within the box and fill only a small subvolume. But we saw in Section 3.1.3 that the finite-volume effect also strongly decreases the total variance of the peculiar velocity field in the box. We argue that this can explain the fact that constrained realisations from peculiar velocities often feature an excess of power in the largest modes of the box (see Figure 2.12). The variance of the observational data is not affected by a finite-volume effect. This discrepancy is only partially moderated by subtracting the tidal component from the data. When setting up a constrained realisation, we force all large-scale power still present in the data (and inconsistent with the finite volume) to be assigned to a few discrete Fourier modes at predefined wavelengths of L, $L/\sqrt{2}$, $L/\sqrt{3}$, $L/2$ etc. We therefore fold all power from scales in between and above those wavelengths into the discrete modes dictated by the finite box, resulting in an excess of power inside these modes. We believe that the same finite-volume effect and excess of power could also cause the otherwise untypical slight increase of halo peculiar velocities with mass in BOX160 (Figure 2.14, left panel). We ran several random realisations with the same parameters, the same simulation code (Gadget-2), and used the same halo finder (Ahf), but each time saw at $z = 0$ a slight decrease of $|\boldsymbol{v}_{\mathrm{pec}}|$ with halo mass instead.

The systematic effects due to the boxshape and the periodic boundary conditions can become severe if the boxsize becomes so small that it is not significantly larger than the data zone, such as the 64 Mpc/h boxsize CLUES simulations discussed in Gottlöber et al. (2010). For this reason, we included in ICeCoRe an alternative way to evaluate the WF/CR operator which is more self-consistent with the box geometry. We discuss the implementation of WF/CR in ICeCoRe in the following section.

3.4 ICeCoRe implementation

The construction of constrained realisations quickly becomes numerically expensive for large numbers of constraints. We are currently unaware of any constrained realisation that was constructed with more than the 4000 constraints used in the simulations of Lavaux (2010). However,

the upcoming Cosmicflows catalogues will provide at least an order of magnitude more peculiar velocity constraints. Computationally, it is not feasible to handle such datasets with either the method used by Lavaux (2010) or the numerical tools, written in IDL, that were up to now used for CLUES initial conditions. An efficient implementation of the WF/CR method would be especially useful if one wants to construct a large ensemble of different constrained realisations. The ICeCoRe code that was developed for this thesis is designed to overcome the present numerical limitations. ICeCoRe is written in C++, utilising several high-performance numerical libraries (details are given in Appendix A). But the task of a new code for WF/CR is not only a question of numerical optimisation, but also requires a careful design of the underlying algorithms.

The WF/CR method rests on correctly evaluating the different correlation functions and constructing and inverting the data autocorrelation matrix $\langle c_i c_j \rangle$. In this chapter, we saw that the WF reconstruction of large-scale structure for cosmography studies on the one hand, and the construction of constrained initial conditions for N-body simulations on the other hand, may require different methods to compute these correlations due to the specifics of the limited volume and periodic boundary conditions required for ICs. In ICeCoRe, we use the approach of pre-computing the correlator, i.e. all necessary correlation functions, from the input power spectrum. This enables a much faster evaluation of the WF/CR operator than to explicitly compute each correlation every time it is needed. Then, for WF reconstruction, the whole process can be decomposed into the following computational steps: computing the correlator, constructing the data autocorrelation matrix, inverting this matrix, and then evaluating the WF/CR operator for each grid cell. In order to generate a CR, two other steps have to be completed beforehand: the construction of a random realisation (section 3.1.1), i.e. the four grids $\delta^{\mathrm{RR}}(\boldsymbol{x})$ and $u_\alpha^{\mathrm{RR}}(\boldsymbol{x})$, $\alpha = x, y, z$, and evaluating the mock constraints $\tilde{c}_i$ on these grids. Most of these steps are straightforward, but computing an appropriate correlator deserves special attention. We implemented two different methods in the ICeCoRe code, the "analytic" and "grid" correlators.

3.4.1 Analytic correlator

We consider here constraints, or data points, of the density and peculiar velocity types. The analytic correlator continues to use the assumption of an infinite, homogeneous and isotropic Universe and therefore continues to use the 1D correlation functions (equations 2.50 and 2.69 – 2.71). As discussed before, this is consistent with the linear-theory statistics of the observed Universe, but inconsistent with the geometry of N-body initial conditions. The analytic correlator is therefore the appropriate approach for a WF reconstruction of the large-scale structure from observational data, such as for the cosmography of the Local Universe. It can also be used to set up constrained ICs if the subvolume containing constraints (data zone) is well within the computational box, so that the absence of periodic boundary conditions on the δ^{WF} can be neglected, and one decides to neglect the inconsistencies between the underlying correlation functions of δ^{WF} and δ^{RR}. The procedure of evaluating the WF/CR operator differs for these two applications. For a WF reconstruction without periodic boundary conditions, we evaluate the WF operator for both the density and the velocity fields to obtain $\delta^{\mathrm{WF}}(\boldsymbol{x})$ (3.38) and $\boldsymbol{u}^{\mathrm{WF}}(\boldsymbol{x})$ (3.41), respectively. For constrained ICs, we evaluate the WF/CR operator only for $\delta^{\mathrm{CR}}(\boldsymbol{x})$, and then compute the displacement field $\boldsymbol{\psi}^{\mathrm{CR}}(\boldsymbol{x})$ with three FFTs (3.5), which removes the tidal component and enforces periodic boundary conditions.

The analytic correlator of ICeCoRe builds upon the same method for computing correlations as the previous IDL code used for CLUES initial conditions, but is more efficient and more flexible with regards to the possible constraints. Currently it supports two types of constraints, density-type (δ) and displacement-type (ψ_μ), which can also be mixed. It is possible to constrain a

particular component of the displacement along an arbitrary unit vector $\hat{\boldsymbol{e}}_\mu$, which also includes the possibility to constrain all three components ψ_α with $\alpha = x, y, z$. It is also possible to assign to each constraint an arbitrary Gaussian smoothing radius R_G. Peculiar velocity constraints are expressed with the displacement type[11]. Therefore, each constraint is defined by ten numbers (seven for the δ constraint type), which are summarised in Table 3.1. The input set of constraints for ICeCoRe consists of a table containing these ten quantities, one line per constraint.

The analytic correlator relies on pre-tabulated correlation functions. From the provided tabulated power spectrum $P(k)$, we prepare four two-dimensional lookup tables over the whole required range of distances x and smoothing radii R_G, evaluating a one-dimensional Fourier integral for each point in the lookup space:

$$\xi(x, R_\mathrm{G}) = \frac{1}{2\pi^2} \int_{k_\mathrm{min}}^{k_\mathrm{max}} k^2\, j_0(kx)\, W_\mathrm{G}(kR_\mathrm{G})\, P(k)\, \mathrm{d}k \quad , \tag{3.46}$$

$$\psi_\mathrm{R}(x, R_\mathrm{G}) = \frac{1}{2\pi^2} \int_{k_\mathrm{min}}^{k_\mathrm{max}} \left[j_0(kx) - \frac{2j_1(kx)}{kx} \right] W_\mathrm{G}(kR_\mathrm{G})\, P(k)\mathrm{d}k \quad , \tag{3.47}$$

$$\psi_\mathrm{T}(x, R_\mathrm{G}) = \frac{1}{2\pi^2} \int_{k_\mathrm{min}}^{k_\mathrm{max}} \frac{j_1(kx)}{kx}\, W_\mathrm{G}(kR_\mathrm{G})\, P(k)\mathrm{d}k \quad , \tag{3.48}$$

$$\zeta(x, R_\mathrm{G}) = \frac{1}{2\pi^2} \int_{k_\mathrm{min}}^{k_\mathrm{max}} k j_1(kx)\, W_\mathrm{G}(kR_\mathrm{G})\, P(k)\mathrm{d}k \quad . \tag{3.49}$$

where $W_\mathrm{G}(kR_\mathrm{G})$ is the Gaussian smoothing kernel and k_min, k_max go by default over the whole k range of the tabulated $P(k)$, although it is possible to specify another range, e.g. $k_\mathrm{min} = k_L$ to explicitly consider the finite-volume effect. Some codes (e.g. Hahn & Abel 2011) use the FFTLog algorithm (Talman 1978; Hamilton 2000) to numerically integrate this type of integrals, which allows one to reduce them to one-dimensional FFTs. However, we instead evaluate the integrals by exact numerical integration, since this has to be done only once for a given $P(k)$ and therefore numerical accuracy is more important than speed. We use the adaptive 61-point Gauss-Kronrod algorithm from the GSL[12], which is able to evaluate these integrals with relative numerical errors less than 10^{-6}.

Figure 3.4 shows plots of the resulting four functions of x and R_G that are precomputed and stored. Then, whenever a particular correlation value is needed, we interpolate on these precomputed tables using the GSL cubic spline interpolation. For M constraints, we have to evaluate $M(M+1)/2$ such interpolations to compute the symmetric matrix $\langle c_i c_j \rangle$. This is extremely fast, taking only about 4 seconds for $M = 4000$ and 5 minutes for $M = 50\,000$ in serial on a laptop with a 2.4 GHz Intel Core 2 Duo CPU. In order to perform the WF/CR algorithm on a grid with N cells, we then have to evaluate $N \times M$ interpolations, which is similarly fast if the grid size is not too large. This allows to generate a CR on a $N = 256^3$ grid within minutes using an ordinary consumer-grade computer. Additionally, the evaluation of the WF/CR operator (equation 3.38) is straightforward to parallelise, since the computation of $\delta^\mathrm{WF}(\boldsymbol{x})$ or $\delta^\mathrm{CR}(\boldsymbol{x})$ in one grid cell is independent from any other grid cells. In ICeCoRe, we included a parallelisation with OpenMP for shared-memory machines. For larger grid sizes, the method quickly becomes computationally expensive. But in practice one does not have to

[11] The ICeCoRe code consistently uses internal units of Mpc/h. Constraints on the peculiar velocity $\boldsymbol{u}$ are treated as constraints on the linear displacement field $\boldsymbol{\psi}$ in units of Mpc/h. This system of units keeps the correlation functions free from cosmology-dependent factors and has numerical advantages if one mixes density and velocity constraints. Peculiar velocity data in km/s has to be converted to units of Mpc/h by dividing the values by the factor $\dot{a}f$ (equation 2.75). At $z = 0$, this equals $H_0 f$, with $f(z = 0) \approx 0.48$ for WMAP7 parameters. The unit conversion factor from km/s to Mpc/h is therefore $100f$, since $H_0 = 100\, h$ km s^{-1} Mpc^{-1}.

[12] GNU scientific library, available at www.gnu.org/gsl .

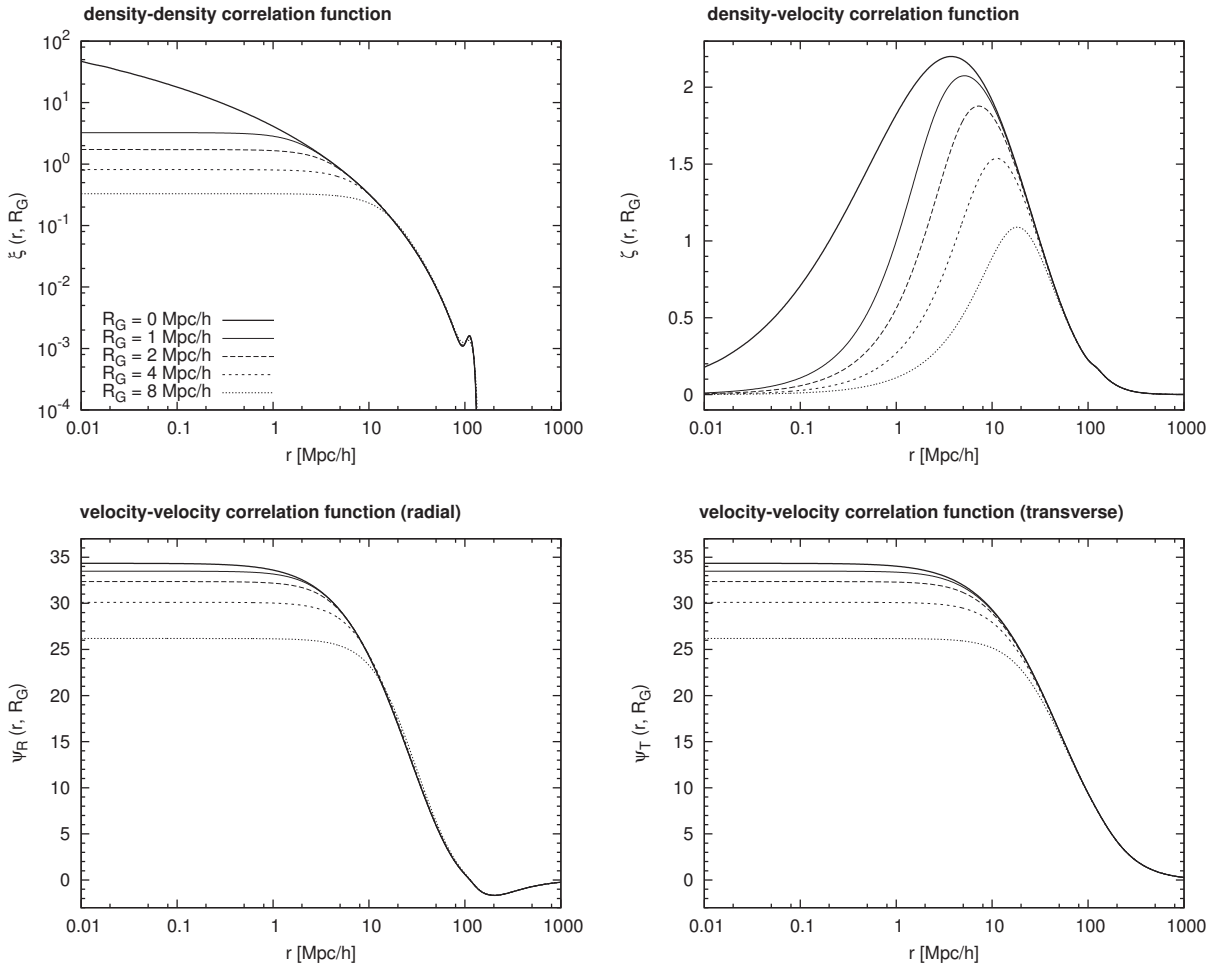

Figure 3.4: Correlation functions $\xi(r)$, $\zeta(r)$, $\psi_{\mathrm{R}}(r)$, $\psi_{\mathrm{T}}(r)$ for the analytic correlator assuming an isotropic and infinite Universe, computed with ICECORE from the WMAP3 power spectrum for different smoothing radii R_{G} (in the code, a much finer sampling in R_{G} is used). Note that the density-density correlation function $\xi(r)$ is plotted in log-log scale, while the other three functions are plotted in log-linear scale.

evaluate a CR on a high-resolution grid anyway, since the small scales below a few Mpc$/h$ are not constrained by the usual kind of data and can be added later from a random realisation. This procedure will be discussed in Section 3.5.

For CRs, we also have to obtain a set of mock constraints $\tilde{c}_i$ from the pre-generated random realisation (RR). This is performed by a simple trilinear interpolation on the density and velocity grids of the RR, respectively. In case the constraints c_i feature non-zero smoothing radii R_{G}, the mock constraint must be the value of the smoothed RR fields, as well. We found that for common smoothing radii not larger than a few Mpc$/h$, it can be considerably faster to perform this smoothing locally by the real-space convolution of a grid patch with the Gaussian window, than to do the smoothing via FFT on the whole grid. This lookup of the $\tilde{c}_i$ is always performed in accordance with the periodic boundary conditions of $\delta^{\mathrm{RR}}(\boldsymbol{x})$ and $\boldsymbol{u}^{\mathrm{RR}}(\boldsymbol{x})$.

3.4.2 Grid correlator

The grid correlator provides an alternative implementation, evaluating all correlation functions for the WF/CR operator such that they are fully consistent with the box geometry, discretisation and periodic boundary conditions. This should be the preferred method for setting up constrained initial conditions, if the boxsize is not considerably larger than the box subvolume that contains constraints, and/or if there are constraints close to the box boundary, in which case the periodic boundary conditions and the large-scale discretisation effects cannot be neglected anymore for δ^{WF}.

In order to be consistent with the properties of ICs discussed in Section 3.1, the correlation functions are not evaluated as one-dimensional integrals, such as in the case of WF reconstruction, but instead with FFTs on a grid that is identical to the grid used for the ICs themselves. This grid is not isotropic, therefore the correlations will depend not only on the distance, but also on the spatial direction, i.e. on the full distance vector $\boldsymbol{x}$ between two points. The equations 3.22 – 3.23 defining the correlations then become

$$\langle \delta_i V_j(\delta) \rangle = \frac{1}{(2\pi)^3} \sum_{\boldsymbol{k}} Y_j(\boldsymbol{k}) P(k) e^{-i\boldsymbol{k}\cdot(\boldsymbol{x}_j - \boldsymbol{x}_i)} \quad , \tag{3.50}$$

$$\langle V_i(\delta) V_j(\delta) \rangle = \frac{1}{(2\pi)^3} \sum_{\boldsymbol{k}} Y_i(\boldsymbol{k}) Y_j(\boldsymbol{k}) P(k) e^{-i\boldsymbol{k}\cdot(\boldsymbol{x}_j - \boldsymbol{x}_i)} \quad . \tag{3.51}$$

Each correlation value for a particular $\boldsymbol{x}$ can then be obtained by an FFT on the whole grid and a subsequent lookup. The strength of this method is that the constraints can have completely different and arbitrary constraint kernels Y_i and smoothing kernels W, since all correlations are explicitly computed each time they are needed. This method was first implemented by van de Weygaert & Bertschinger (1996) as an extension of the original Hoffman-Ribak algorithm. The approach is mostly used for constrained ICs that do not use observational data, but in-

Column	Quantity	Meaning
1	Y_i	Constraint type: δ or ψ_μ
2	x_i	Constraint position: cartesian components of $\boldsymbol{x}_i$
3	y_i	
4	z_i	
5	c_i	constraint value
6	ε_i	estimated constraint error
7	$\hat{e}_{\mu,x}$	Direction vector for ψ_μ: cartesian components of $\hat{\boldsymbol{e}}_\mu$
8	$\hat{e}_{\mu,y}$	
9	$\hat{e}_{\mu,z}$	
10	$R_{i,\mathrm{G}}$	Gaussian smoothing radius of constraint

Table 3.1: Ten quantities that define a single constraint c_i in the constraints input file for ICeCoRe. Possible constraint types are the density, the three cartesian components of the linear displacement $\boldsymbol{\psi}$ (each component is one constraint), or a specific component ψ_α of the linear displacement $\boldsymbol{\psi}$ defined by the direction unit vector $\boldsymbol{e}_\alpha$. Linear peculiar velocity constraints follow from the linear theory equation $\boldsymbol{v} = \dot{a} f \boldsymbol{\psi}$. For density-type constraints, the values inside columns 7 – 9 are ignored.

stead artificially defined peaks and other features; in this case, it is a completely self-consistent method. The same method was also used by Lavaux (2010) to set up a constrained simulation of the Local Universe with ≈ 4000 constraints of the displacement type generated from a MAK reconstruction of the 2MASS galaxy redshift catalogue. The problem of the method is its very high computational cost. In order to construct the autocorrelation matrix of the constraints, one needs $M(M+1)/2$ FFTs (where M is the number of constraints), and then another M FFTs to evaluate the WF/CR operator on the grid. This is only practical if one has very few constraints, for example if one constrains the shape and orientation of just a few peaks in the ICs (van de Weygaert & Bertschinger 1996). With a large set of non-local constraints, the computational cost quickly explodes: Lavaux (2010) reports that the computation of the correlation matrix with $M = 3942$ took a little less than a week on 128 processors on the Mercury cluster at the National Center for Supercomputing Applications (NCSA). Since the computational cost for creating a CR scales as $O(M^2 N \log N)$, it is currently not feasible to apply the method to the upcoming large peculiar velocity datasets that provide tens of thousands of constraints.

With the grid correlator of ICeCoRe, we here present an alternative implementation of the method that reduces the required computational cost by a large factor, while yielding mathematically equivalent results. Van de Weygaert & Bertschinger (1996) already noted that the required computational effort can be greatly reduced if the constraints are all of the same type. We extend this simple case by allowing the same types of constraints as the analytic correlator, i.e. density and displacement types (Table 3.1), which can be mixed. We therefore need to evaluate density-density, density-displacement and displacement-displacement correlation functions. Similar to the analytic correlator, the method rests on pre-computed correlation functions, but because of the added anisotropy and periodic boundary conditions, in this case they consist of three-dimensional grids instead of simple interpolation tables. These three-dimensional correlation functions will now depend not only on the input power spectrum $P(k)$, but also on the chosen boxsize L and grid resolution Δx of the constrained ICs. We found that in order to compute all necessary correlations, it is sufficient to prepare only the four following correlation grids:

$$\xi(\boldsymbol{x}) = \frac{1}{(2\pi)^3} \sum_{\boldsymbol{k}} P(k) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \tag{3.52}$$

$$\zeta_x(\boldsymbol{x}) = \frac{1}{(2\pi)^3} \sum_{\boldsymbol{k}} \left[\frac{-ik_x}{k^2}\right] P(k) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \tag{3.53}$$

$$\psi_{xx}(\boldsymbol{x}) = \frac{1}{(2\pi)^3} \sum_{\boldsymbol{k}} \left[\frac{k_x^2}{k^4}\right] P(k) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \tag{3.54}$$

$$\psi_{xy}(\boldsymbol{x}) = \frac{1}{(2\pi)^3} \sum_{\boldsymbol{k}} \left[\frac{k_x k_y}{k^4}\right] P(k) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \quad . \tag{3.55}$$

They are the three-dimensional analogues of the isotropic analytic functions ξ, ζ, ψ_{R} and ψ_{T}. A visual representation of these grid functions is shown in Figure 3.5 for a relatively small boxsize of $L = 64$ Mpc$/h$, where the deviation from the analytic functions is quite large. Due to the underlying symmetries, all correlation values that will be possibly needed for the WF/CR procedure can be obtained by simple lookups on these four grids, if the lookup indices x_α are permuted accordingly:

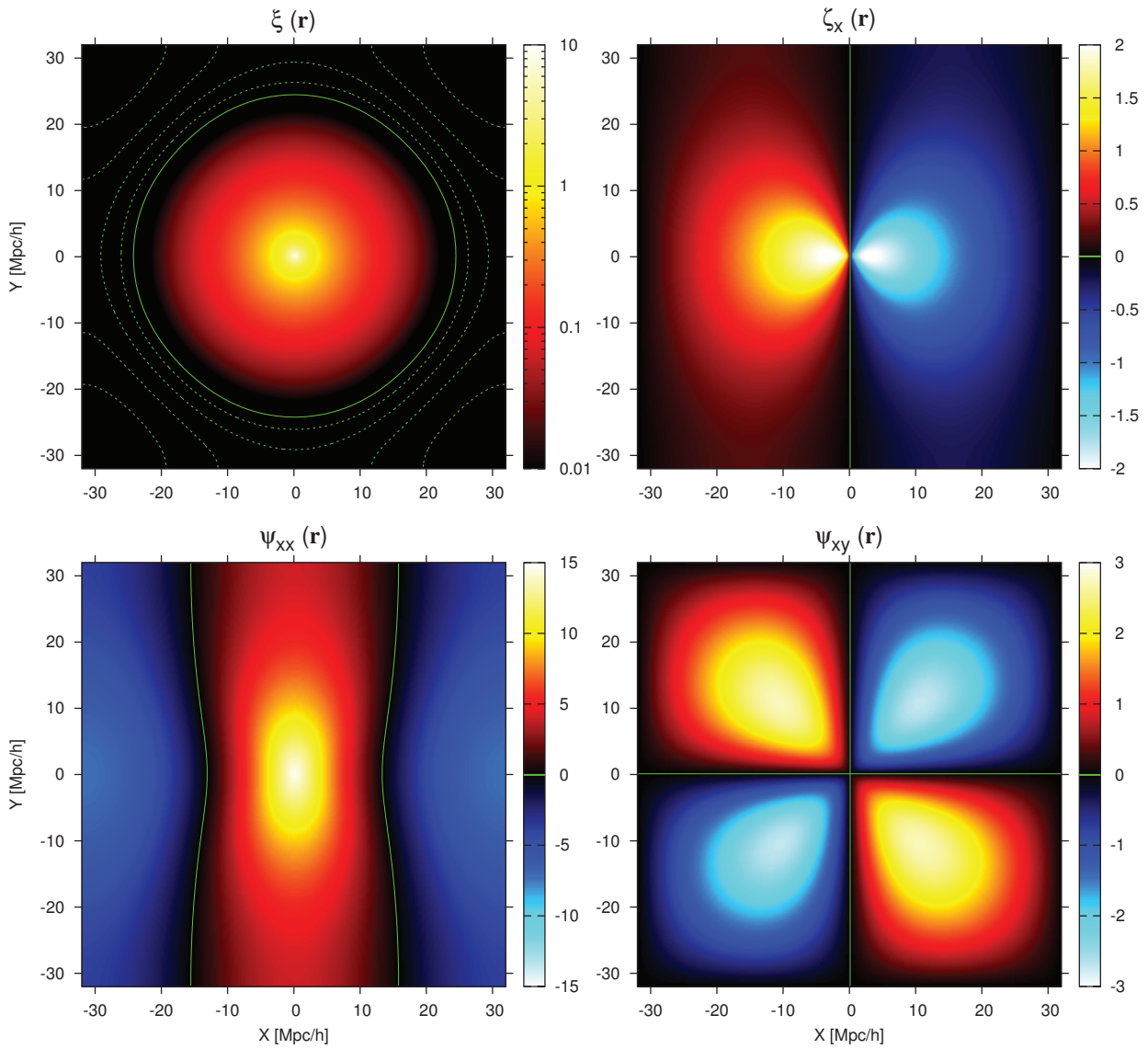

Figure 3.5: Three-dimensional correlation functions $\xi(\boldsymbol{r})$, $\zeta_x(\boldsymbol{r})$, $\psi_{xx}(\boldsymbol{r})$, $\psi_{xy}(\boldsymbol{r})$ for the grid correlator assuming a boxsize of $L = 64$ Mpc$/h$ and periodic boundary conditions, computed with ICeCoRe from the WMAP3 power spectrum (no smoothing: $R_\mathrm{G} = 0$) on a $N = 256^3$ grid. The plots show the XY-plane through $Z = 0$. The solid green contour lines show where the functions have value zero. Note that the density-density correlation function $\xi(\boldsymbol{r})$ is shown in logarithmic scale. Here, the additional dashed green contour lines show the regions where ξ is negative and are positioned at $\xi = -0.005,\ -0.01,\ -0.015,\ -0.02$, which reveals the anisotropy of $\xi(\boldsymbol{r})$ close to the box boundaries.

$$
\begin{aligned}
&\langle\delta(\boldsymbol{x}')\delta(\boldsymbol{x}'+\boldsymbol{x})\rangle = \xi(x_x, x_y, x_z) && \langle\psi_x(\boldsymbol{x}')\psi_x(\boldsymbol{x}'+\boldsymbol{x})\rangle = \psi_{xx}(x_x, x_y, x_z)\\
& && \langle\psi_x(\boldsymbol{x}')\psi_y(\boldsymbol{x}'+\boldsymbol{x})\rangle = \psi_{xy}(x_x, x_y, x_z)\\
&\langle\delta(\boldsymbol{x}')\psi_x(\boldsymbol{x}'+\boldsymbol{x})\rangle = \zeta_x(x_x, x_y, x_z) && \langle\psi_x(\boldsymbol{x}')\psi_z(\boldsymbol{x}'+\boldsymbol{x})\rangle = \psi_{xy}(x_x, x_z, x_y)\\
&\langle\delta(\boldsymbol{x}')\psi_y(\boldsymbol{x}'+\boldsymbol{x})\rangle = \zeta_x(x_y, x_z, x_x) && \langle\psi_y(\boldsymbol{x}')\psi_x(\boldsymbol{x}'+\boldsymbol{x})\rangle = \psi_{xy}(x_y, x_x, x_z)\\
&\langle\delta(\boldsymbol{x}')\psi_z(\boldsymbol{x}'+\boldsymbol{x})\rangle = \zeta_x(x_z, x_x, x_y) && \langle\psi_y(\boldsymbol{x}')\psi_y(\boldsymbol{x}'+\boldsymbol{x})\rangle = \psi_{xx}(x_y, x_z, x_x)\\
& && \langle\psi_y(\boldsymbol{x}')\psi_z(\boldsymbol{x}'+\boldsymbol{x})\rangle = \psi_{xy}(x_y, x_z, x_x)\\
& && \langle\psi_z(\boldsymbol{x}')\psi_x(\boldsymbol{x}'+\boldsymbol{x})\rangle = \psi_{xy}(x_z, x_x, x_y)\\
& && \langle\psi_z(\boldsymbol{x}')\psi_y(\boldsymbol{x}'+\boldsymbol{x})\rangle = \psi_{xy}(x_z, x_y, x_x)\\
& && \langle\psi_z(\boldsymbol{x}')\psi_z(\boldsymbol{x}'+\boldsymbol{x})\rangle = \psi_{xx}(x_z, x_x, x_y) \quad ,
\end{aligned}
\tag{3.56}
$$

where the x_α ; $\alpha \in \{x, y, z\}$ are the cartesian components of $\boldsymbol{x}$. Through equations 3.29 and 3.30 we also cover the case of ψ_μ constraints in arbitrary direction $\hat{\boldsymbol{e}}_\mu$. Generally, the constraints do not follow the grid cells, so that the lookup will actually be an interpolation on the grid, which we implemented as a simple trilinear interpolation. Then, the computation of each correlation value will involve eight grid cell lookups and seven linear interpolations. This is many orders of magnitude faster than if each time we would have to perform an FFT on the whole grid, as in van de Weygaert & Bertschinger (1996); Lavaux (2010). With the current implementation we find that the grid correlator computes correlations about a factor of 7 slower than the analytic correlator; constructing a 4000×4000 correlation matrix takes 29 seconds on a laptop with a 2.4 GHz Intel Core 2 Duo CPU. This is an impressive speed-up compared to Lavaux (2010). We want to mention that this current numerical implementation in ICECORE leaves room for further optimisations, which are the subject of ongoing work.

When generating CRs from observational peculiar velocity data, we must keep in mind that the underlying correlation functions of the grid correlator assume the periodic limited-volume behaviour on both the random realisation and the provided set of constraints. Therefore, the results obtained with the analytic and the grid correlator will be different. Figure 3.6 compares the WF density and velocity fields reconstructed from the Cosmicflows-1 galaxy groups, computed on a finite box with $L = 80$ Mpc$/h$ with three different methods: the analytic correlator for both the density and velocity fields (such as for LSS reconstruction), the analytic correlator for the density field and FFT for the velocity field (such as for generating ICs with a periodic displacement field), and the grid correlator for both (for generating self-consistent ICs). While the density and velocity fields are, in the first case, consistent with the observed Local Universe, in the second case we force periodicity on the velocity/displacement fields in order to be able to generate ICs. This strongly changes the shape of the velocity field, e.g. the pull from the GA is now felt from the opposite edge of the box as well. It is obvious that the underlying WF density field is however *not* periodic, and if the WF mean field does not vanish at the box size boundaries, this will introduce systematic errors in the CR that could corrupt the simulation. The third method uses the grid correlator and one now obtains density and displacement fields that are fully consistent with the periodic box geometry. This comes at a price, namely that the WF mean field is not anymore accurate compared to the Local Universe in the outer parts of the box. The reconstruction is still accurate in the central part of the box, approximately inside a centered sphere of radius $L/4$. Unfortunately, there is no optimal method for setting up constrained ICs in this scenario: squeezing the Local Universe into a periodic box will always involve systematic inconsistencies with the observed configuration. These can be quantified if one compares the fields produced by the analytic and grid correlators, respectively.

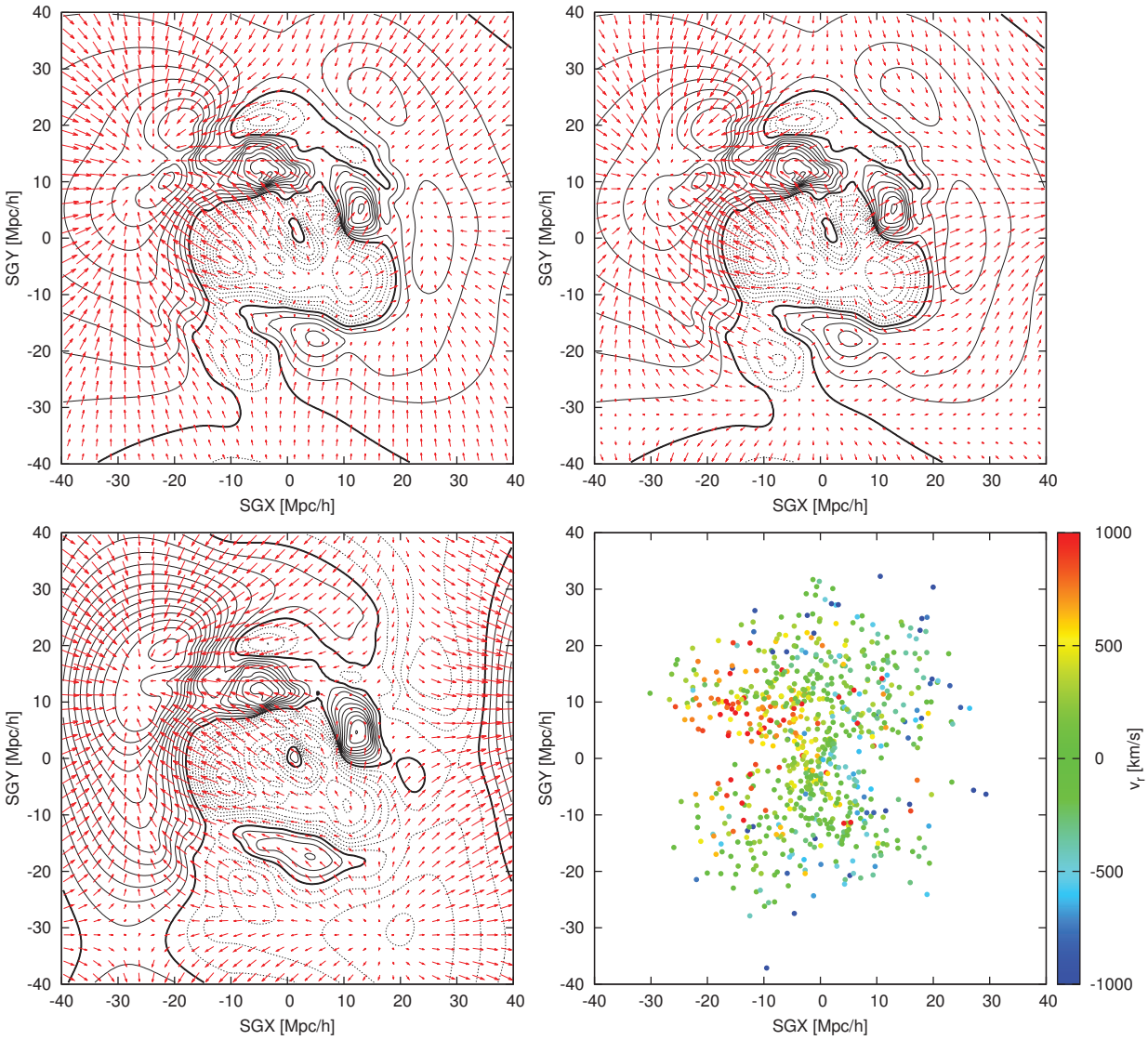

Figure 3.6: Same slice and orientation as Figure 3.2: no periodic boundary conditions (top left), periodic boundary conditions assumed when computing the velocity field from the overdensity (top right), and with the grid correlator (bottom left) assuming periodic boundary conditions on all quantities. The bottom right panel shows the input constraints: radial peculiar velocities from the 742 Cosmicflows-1 galaxy groups.

We want to point out that the already mentioned method of Pen (1997) could provide an interesting alternative. This method uses the isotropic real-space correlation functions to set up ICs and simultaneously reduces the problem of variance underestimation by the finite volume. A combination of this method with the WF/CR algorithm could lead to constrained simulations that are closer to the statistics of the observed Universe, while still being self-consistent with the simulation geometry like the grid correlator presented here.

3.4.3 Matrix inversion

In between the computation of the data autocorrelation matrix $\langle c_i c_j \rangle$ and the WF/CR operator, who both rely on computing correlation functions, the only other computationally expensive step of the WF/CR method is the inversion of the matrix to yield $\langle c_i c_j \rangle^{-1}$. With the upcoming large peculiar velocity datasets, this $M \times M$ matrix can become very large, so we have to make sure that the inversion is stable in each case.

The matrix $\langle c_i c_j \rangle$ is real-valued, non-sparse, symmetric and positive-definite. It is especially well-formed because, due to the shape of the correlation functions, the diagonal entries $\langle c_i c_i \rangle$ (the autocorrelations of the datapoints) will be always the largest-valued entries in each row and column. Because of these properties, the matrix is always invertible[13]. The optimal algorithm to invert a matrix with these properties is the Cholesky decomposition (see e.g. Golub & Loan 1996), which is more efficient than the traditional Gauss-Jordan elimination or LU decomposition method, because it explicitly uses the symmetry of $\langle c_i c_j \rangle$. This is performed in two steps. First, the matrix is decomposed into an upper triangular matrix U and a lower triangular matrix L, which is simply $L = U^T$ because of the symmetry,

$$\langle c_i c_j \rangle = U^T U \quad , \tag{3.57}$$

where the T superscript denotes the matrix transpose. The two triangular matrices can be inverted very efficiently by forward substitution, and then the inverse autocorrelation matrix is obtained by factorisation with

$$\langle c_i c_j \rangle^{-1} = U^{-1} U^{-T} \quad . \tag{3.58}$$

Besides the faster execution speed, the most important advantage of the Cholesky decomposition is its high numerical stability. The ordinary LU decomposition is an unstable algorithm and requires some sort of pivoting strategy to reduce the growth of numerical roundoff errors and stabilise the inversion of large matrices (Kreyszig 2006). By contrast, the Cholesky decomposition requires no pivoting or other regularisation and the numerical errors will always be small. Thus, the concern raised by Lavaux (2010) that the matrix inversion could become unstable for ranks larger than a few thousand, is something that we did not observe. For the Cholesky inversion in ICeCoRe, we utilise the LAPACK[14] library. We obtained the best results with the LAPACK implementation contained in the MKL[15], which is also parallelised (multi-threaded). The computational cost of the inversion scales with $O(M^3)$, and the maximum rank M of the matrix that can be inverted is limited in principle only by the available memory. For $M = 4000$, the inversion takes only a couple of seconds in serial. For matrices larger than $M \approx 10\,000$ – $20\,000$, it is recommended to switch to parallel machines. For $M = 50\,000$, the inversion takes

[13] The only exception would occur if there are two constraints at the same position that directly contradict each other. But even in this case the matrix would be invertible if a constant term is added to the diagonal, i.e. if the errors ε_i or the σ_{NL} parameter are non-zero (equation 3.45), acting as a stabilising regularisation term.

[14] Linear Algebra PACKage (Anderson et al. 1999), available at www.netlib.org/lapack

[15] Intel Math Kernel Library, see www.intel.com.

around 60 minutes with eight OpenMP threads on two quad-core 2.4 GHz Intel Xeon processors, with 18 GB of memory required for the inversion. In all cases, we observed that both the matrix inversion and the solutions subsequently obtained with the WF/CR operator were numerically stable. We therefore acknowledge that the ICeCoRe code is able to handle significantly larger sets of constraints than used before for constrained realisations in a very fast and numerically efficient way and is therefore the ideal tool for WF reconstructions and generation of CRs from the large upcoming peculiar velocity datasets.

3.5 High-resolution, multi-scale, and baryonic initial conditions

Modern cosmological simulations, running on massively parallel high-performance computing clusters, follow the simultaneous evolution of billions of particles. This poses high requirements on IC generation codes to generate initial conditions with matching high resolution. One kind of simulations starts from a uniform high-resolution mass sampling and aims to obtain a statistically representative sample of structure formation over a large cosmological volume. Examples are the Millennium Simulation (Springel et al. 2005), with $L = 500$ Mpc$/h$ and $N = 2160^3$, the Horizon Simulation (Teyssier et al. 2009), with $L = 2000$ Mpc$/h$ and $N = 4096^3$, and the more recent Bolshoi simulation (Klypin et al. 2011), with $L = 250$ Mpc$/h$ and $N = 2048^3$. There are also simulations focusing on the detailed evolution of one particular object (or a few objects), such as a dark matter halo hosting a galaxy, emdedded in the large-scale environment (Springel et al. 2008; Crain et al. 2009). They use the popular "zoom-in" technique, with an increased mass resolution to accurately sample the object of interest, and a lower mass resolution for the rest of the volume. This is sufficient to accurately follow the gravitational interaction of the large-scale environment with the objects of interest. This efficient technique is particularly interesting in constrained CLUES simulations, where one can study in high resolution the formation and evolution of Local-Group-like galaxies, with the large-scale environment defined by a constrained realisation of the Local Universe (Gottlöber et al. 2010). The usual method for setting up these zoomed simulations is to first perform the desired simulation with uniform coarse resolution. Then, from the finished simulation at $z = 0$ the regions are selected where the resolution should be increased. The particles corresponding to that region are identified in the initial conditions, and then the resolution of the initial conditions is increased around the corresponding Lagrangian patch to yield the multi-scale ICs for the final simulation run. It is also possible to create several nested shells of decreasing resolution centered on the zoomed region, and several unconnected zoomed regions within the box.

Different methods and numerical codes exist to generate this kind of high-resolution and multi-scale initial conditions. The first method, introduced by Katz et al. (1994), consists of generating a full box with the high resolution and then degrade it everywhere except for the zoomed region. The advantage of this method is that one has full control over the numerical errors. The degrading procedure can be performed either exact in real space (by averaging over high-resolution cells) or exact in Fourier space (by a resampling of the Fourier modes). The downside of this approach is that the numerical cost of performing FFTs on the full box with the maximum resolution can be quite demanding. The recently developed code Ginnungagap[16] (Knollmann 2012, private communication) uses especially optimised FFTs to achieve resolutions of up to $N = 16384^3$. The alternative approach is to generate local refinements on the coarse initial conditions only where needed. This technique was developed by Bertschinger (2001). His Grafic-2 code, as well as the parallelised version Mpgrafic (Prunet et al. 2008), starts from the white noise field of the coarse initial conditions and then creates refinements that

[16] available at code.google.com/p/ginnungagap .

are consistent with the coarse field, leading to a multi-scale white noise field. To perform the colouring and thereby obtain the full multi-scale realisation with power spectrum $P(k)$, they combine local FFTs of different resolution to the multi-scale white noise field. This however adds aliasing and oscillatory errors at the few percent level. The recently developed code MUSIC by Hahn & Abel (2011) improves on this method by convolving the white noise field with the real-space correlation function instead, therefore extending the method of Pen (1997) to multi-scale initial conditions. Their implementation virtually removes the numerical errors.

To extend these complex methods to constrained realisations by combining them with the WF/CR algorithm in a self-consistent manner is a daunting task. Even for non-zoomed, uniform initial conditions, the computational cost of carrying out the WF/CR operator on the whole grid quickly explodes for large N. The ICECORE code is comparatively limited in this respect, since we did not yet implement a distributed-memory parallelisation of the WF/CR operator; with the currently present shared-memory parallelisation (OpenMP) it is not practical to generate initial conditions with $N > 1024^3$. However we argue that one would not need to generate a CR with higher resolution anyway. In all current applications of the CR method, the constraints are placed either on the peculiar velocity/displacement field or on functionals of the smoothed density field, and constrain the realisation only on scales above a few Mpc/h. Smaller scales are completely dominated by the random component. The most efficient approach is therefore to create a CR on a coarse grid, where the grid spacing does not have to be finer than 1 – 2 Mpc/h. Even for a very large boxsize of $L = 1000$ Mpc/h, a CR with $N > 1024^3$ or even $N = 512^3$ resolution would be sufficient. The smaller scales (and additional local refinements) can then be added from a random realisation by processing with another IC generating code. This technique is used for CLUES simulations: for example, the initial conditions of BOX160 with $L = 160$ Mpc/h and $N = 1024^3$ were generated by adding random small scales to a coarse CR with $N = 256^3$. This was done by first generating the coarse CR and a high-resolution RR, and then directly replacing the numerical values of the large-scale Fourier modes of the RR with those of the CR. However, their implementation of this procedure is comparatively limited in maximum resolution (Gottlöber et al. 2010). Additionally, in practice it is very difficult to ensure that the coarse CR and high-resolution RR are statistically fully consistent with each other and the procedure does not introduce, for example, artefacts or discontinuities on the resuting combined grid in either Fourier or real space. The ICECORE code presented in this work provides a more flexible and powerful strategy. We use the fact that ICECORE generates ICs using the white noise field $w(\boldsymbol{x})$, which for any constrained realisation handled within ICECORE can be easily computed by whitening (see Section 3.1.1). The white noise field $w^{\mathrm{CR}}(\boldsymbol{x})$ of a CR completely defines the full fields δ^{CR} and $\boldsymbol{u}^{\mathrm{CR}}$ for a given prior model $P(k)$. Therefore, we can directly feed $w^{\mathrm{CR}}(\boldsymbol{x})$ to another code that can then generate high-resolution small scale modes and/or refinements on top of the $w^{\mathrm{CR}}(\boldsymbol{x})$ and from there create the full realisation of the initial conditions. This approach is very natural, since several high-resolution IC generators, specifically the GRAFIC-2, MPGRAFIC, MUSIC and GINNUNGAGAP codes, start off from a coarse-grid white noise field anyway. By limiting the communication between ICECORE and the follow-up code to the initial white noise field, we also avoid any inconsistencies in the $P(k)$, normalisation, or any other parameters between the CR and the RR: the whole colouring procedure is performed self-consistently inside the follow-up code. Of course, to be completely consistent with the original constraints c_i, this code should use the same $P(k)$ and cosmological parameters as ICECORE.

The whole recipe then consists of creating $\delta^{\mathrm{CR}}(\boldsymbol{x})$ on a uniform grid with ICECORE, whitening it to obtain $w^{\mathrm{CR}}(\boldsymbol{x})$, passing this on to another code, and perform all subsequent steps like up-sampling, placing refinements, and colouring there. We thoroughly and successfully tested this procedure with the MUSIC and GINNUNGAGAP codes. Figure 3.7 shows the generation of

incremental refinements with GINNUNGAGAP starting from a very coarse $N = 64^3$ constrained realisation produced with ICECORE. With the power and accuracy of these codes, we acknowledge that this strategy is ideal to provide accurate initial conditions for the next generation of high-resolution constrained simulations.

We close this chapter with a remark on other types of initial conditions. It is of special interest to obtain ICs for the baryonic component of the universe as well. These are required for hydrodynamic cosmological simulations modelling the formation of stars and galaxies. Depending on the type of the simulation code, they consist of either SPH particles (Hernquist & Katz 1989) or a grid sampling of the baryonic density, velocity and pressure distributions (Toro 1999). The common approach is to assume that at the initial redshift $z_{\rm init}$ the baryon distribution is following the dark matter distribution[17], except that now the mean density is $\Omega_{\rm b}\,\bar{\rho}$ for baryons and $\Omega_{\rm dm}\,\bar{\rho}$ for the dark matter. Since $\Omega_{\rm m} = \Omega_{\rm dm} + \Omega_{\rm b}$, the total matter distribution will be equal to $\delta(\boldsymbol{x})$. ICECORE does not contain routines for directly generating such multi-phase ICs. To obtain them from a constrained realisation generated with ICECORE, the same procedure based on the white noise field can be used. We first generate $w^{\rm CR}(\boldsymbol{x})$ with ICECORE which completely defines both the dark matter and baryon ICs for a given power spectrum, $\Omega_{\rm dm}$ and $\Omega_{\rm b}$. We then feed this into an IC generator that is capable of producing the desired hydrodynamic ICs, such as MPGRAFIC and MUSIC. An exception is the RAMSES simulation code, which can initialise and start a hydrodynamic simulation directly from the overdensity field $\delta^{\rm CR}(\boldsymbol{x})$ generated by ICECORE if no zoom-in regions are desired (we successfully tested this procedure as well). We also want to point out that the white noise field $w^{\rm CR}(\boldsymbol{x})$ completely defines the initial conditions if one wishes to use different pre-initial conditions for the particle sampling, such as a glass distribution, or a different method for the initial particle displacement, such as 2LPT (see Section 3.1.2).

[17] We note that this approach is not entirely accurate. Baryons decouple from the radiation field at $z \approx 1000$, i.e. the redshift of the CMB. As a result, for typical initial redshifts $z_{\rm init}$, for which initial conditions are produced, baryon density fluctuations do not yet exactly trace the dark matter distribution. The transfer function $T(k)$ of the baryonic component, and therefore also its initial power spectrum $P(k)$, will in general be different from the $T(k)$ and $P(k)$ of the dark matter due to the different underlying physics. See Yamamoto et al. (1998); Yoshida et al. (2003); Hahn & Abel (2011).

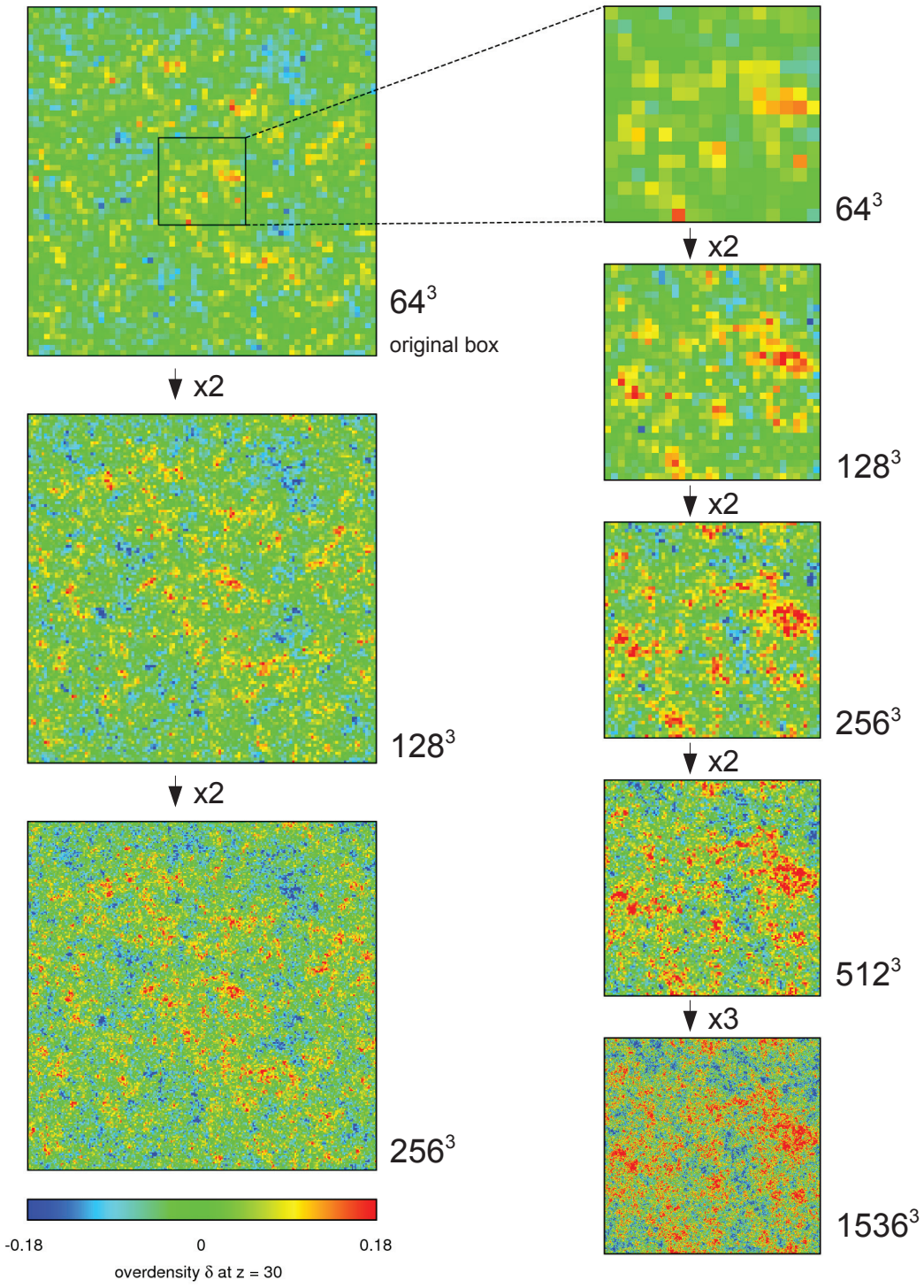

Figure 3.7: Refinements generated incrementally with the GINNUNGAGAP code from a low-resolution (64^3) constrained realisation of initial conditions at $z = 30$ created with ICECORE (top left). The full box (left) has a boxsize of $L = 64$ Mpc/h. The zoom-in region (right) is 16 Mpc/h across.

Chapter 4

Lagrangian reconstruction from peculiar velocity data

In this chapter, we develop a method to reconstruct the cosmological displacement field using the peculiar velocity of dark matter haloes at $z = 0$ as a proxy. We apply a reverse Zeldovich approximation (RZA) to link the velocity and displacement fields and to trace the haloes back to their initial position. We quantify the error of this procedure depending on different properties of the haloes and their environment. We then investigate how this method can be applied to realistic radial peculiar velocity data. For this, we build a set of mock catalogues from the BOX160 simulation, varying different observational limits like the distance error and the data volume to quantify their impact. We reconstruct the three-dimensional velocity field using the Wiener Filter and then apply the RZA. The results suggest that, compared to the previous method used for CLUES simulations, the RZA leads to a much better estimate of the initial positions of the haloes. The method performs very well throughout most of the data volume and is especially efficient on isolated objects found in less dense environments. However, the approximation fails in high-density regions that are dominated by virial motions and accretion onto massive clusters. This can be alleviated by an adequate grouping or selection of data points. The displacements reconstructed by the RZA can be subsequently used for constraining cosmological initial conditions. We also extend the RZA to second-order Lagrangian perturbation theory (2LPT) and compare the results.

4.1 Reconstructing initial conditions

4.1.1 The general problem

As we saw in Chapter 2, according to the ΛCDM model the observable structure in the Universe evolved from initial conditions in the form of a linear Gaussian random field through gravitational collapse, which in general is a highly non-linear process. If we wish to perform a simulation that resembles the Local Universe as much as possible, it is first necessary to obtain the underlying initial conditions from the present-day state of the Local Universe. But even if we know this state, i.e. the overdensity and velocity fields $\delta(\boldsymbol{r})$ and $\boldsymbol{v}(\boldsymbol{r})$ at present-day redshift $z = 0$, to very high precision, we cannot just integrate the equations of motion back in time, such as by running a cosmological N-body simulation backwards, to recover the initial conditions. Although gravity is in principle invariant under time reversal, any such approach will fail even in the linear regime

of structure formation. The linear theory of density perturbations has a growing mode $D_+(t)$ and a decaying mode $D_-(t)$ (2.41). If we now reverse the time direction, the decaying mode will instead rapidly increase and amplify any uncertainties in the data or even slight numerical errors until they eventually dominate the solution. With such a procedure, the probability of recovering the highly ordered state of homogeneous and almost uniform initial conditions will be infinitely small.

In the non-linear regime of structure formation, i.e. on the scales of gravitationally collapsed and virialised objects such as galaxy clusters, it is even more apparent that the memory of its linear initial conditions will be irretrievably erased. If an object is accreted into such a cluster at some earlier time and enters an orbit, from observing the present-day motion there is in general no way to tell from which direction it came and how many orbits it has completed; there are many different initial configurations possible that would evolve into an almost identical result. We saw already in the simple single-wave case of the Zeldovich pancake collapse that from a non-linearly evolved state it is not possible anymore to reconstruct the mapping between Eulerian coordinates $\boldsymbol{x}$ at present time and the Lagrangian coordinates $\boldsymbol{q}$ that define the initial conditions. This loss of information about the ordered initial state is essentially a manifestation of the second law of thermodynamics. If one wishes to employ a scheme to reconstruct the linear initial conditions, it should therefore be restricted to the growing mode D_+ and to length scales above the scale of shell-crossing, i.e. to the so-called quasi-linear regime. A very useful tool in this context, exploited by many reconstruction methods in some way or the other, is the Zeldovich approximation, which was already introduced in 2.2.6. For a particle sampling of δ it connects the Lagrangian coordinates $\boldsymbol{q}$ of those particles (corresponding to the comoving positions in the homogeneous initial state at $z \to \infty$) to their comoving positions $\boldsymbol{x}(z)$ at a later redshift z via the displacement field $\boldsymbol{\psi}$,

$$\boldsymbol{x}(z) = \boldsymbol{q} + D_+(z)\boldsymbol{\psi}_0(\boldsymbol{q}) \quad , \tag{4.1}$$

naturally restricting itself to the growing mode D_+ and implementing the simplification that all tracers of the field move on straight paths with constant velocities. As we already saw, this seemingly oversimplifying method works surprisingly well for scales above shell crossing.

4.1.2 Density-based methods

In the last two decades, different schemes have been designed to perform a reconstruction of cosmological initial conditions from data at $z = 0$. Many rely on galaxy positions in redshift space as an estimate of the underlying density field $\delta(\boldsymbol{r})$, while others use peculiar velocity data. Not all of the schemes actually construct run-capable initial conditions that could be directly fed into an N-body code; some recover just the initial power spectrum $P(k)$ to some degree, for example in order to reconstruct the BAO peak, while others generate an estimate of the initial field smoothed on some scale. Many require input data that can not be realistically obtained from observations, such as a complete samling of $\delta(\boldsymbol{r})$ inside some computational box. One of the first such methods is the Gaussianisation procedure of Weinberg (1992), which assumes that individual grid cells that sample $\delta(\boldsymbol{r})$ preserve their rank order during structure formation, and then assigns to them density values consistent with a linear Gaussian random field while keeping that order. While this procedure produces run-capable ICs by design, it does not attempt to perform a Lagrangian reconstruction and could therefore only be useful on scales above that of the cosmological displacement. A simple Lagrangian reconstruction scheme has been proposed by Eisenstein et al. (2007), which computes the displacement field $\boldsymbol{\psi}$ from the density field using the linear-theory equation $\delta = -\boldsymbol{\nabla} \cdot \boldsymbol{\psi}$ and then essentially to move a particle sampling

of δ backwards using $\boldsymbol{\psi}$. Because δ is highly non-linear, it has to be smoothed with a Gaussian kernel of at least 10 Mpc/h, which still significantly overestimates ψ in regions with high δ due to the large skewness of the non-linear density field even at that smoothing scale. This scheme is useful to reconstruct the BAOs (Noh et al. 2009) but too crude for usable initial conditions. Improvements on this method replace the linear theory equation with higher-order formulae connecting δ and $\boldsymbol{\nabla} \cdot \boldsymbol{\psi}$ (e.g. Falck et al. 2012).

Another method, the MAK reconstruction (Frisch et al. 2002; Brenier et al. 2003; Mohayaee et al. 2003), exploits the Zeldovich approximation more directly. Assuming that all $\boldsymbol{q}_i$ and $\boldsymbol{x}_i(z = 0)$ of a particle sampling are connected by straight lines $\boldsymbol{\psi}_i$, and that the displacement field is irrotational, it can be shown that there is a unique solution minimising the total action $\sum_i |\boldsymbol{\psi}|^2$. This solution, given by the Monge-Ampère equation (Monge 1781; Ampère 1820), is a good estimate of the actual displacement field on scales above orbit crossing. Mathematically this is equivalent to solving the Monge-Kantorovich problem of optimal transport (Kantorovich 1942). Using methods like the auction algorithm of Bertsekas (1991), this unique solution can be numerically computed. An alternative is the PIZA method (Croft & Gaztañaga 1997), giving a good approximate solution to the same numerical problem by iteratively minimising the action without the assumption of uniqueness. Wagner (2009) has shown that this approach yields better results than the simpler linear theory method of Eisenstein et al. (2007). Lavaux et al. (2010) have applied the MAK reconstruction to the 2MASS redshift survey, and in a second step, these reconstructed displacements have been used by Lavaux (2010) to generate constrained cosmological initial conditions and run a constrained simulation. Their simulation recovers the main features of the large-scale structure of the utilised observational data, such as the Virgo, Fornax, Perseus-Pisces, Norma and Centaurus clusters and the Local Void, although the correlation of the simulated velocity field with the observed radial peculiar velocity data is relatively poor.

Also building upon the least action principle are numerical action methods (NAM). Here, the Zeldovich approximation is replaced by a more general description of gravity, and the orbits are allowed to be curved. Then, for a set of galaxies, such orbits are computed iteratively, again minimising the total action. This approach was pioneered by Peebles (1989), at first restricted to the objects in the Local Group. Later, NAM was expanded to the Local Supercluster (Shaya et al. 1995) and more recently to all-sky galaxy redshift surveys (Branchini et al. 2002; Phelps et al. 2006). However, like many other methods not mentioned here, NAM is geared more towards reconstructing the objects' present-day velocities rather than their initial conditions, and to our knowledge the full orbit reconstruction of a cosmological volume all the way into the linear regime is not feasible with this method.

A very recent development is the method of Kitaura (2012), who first employ a Gaussianisation step of the density field at $z = 0$, as traced by galaxy positions, by Hamitonian sampling with a Gaussian-Poisson model. In the second step, they then use an iterative approach based on a 2LPT structure formation model in order to recover an estimate of the initial density field in the linear regime.

Several of these density-based methods can provide useful reconstructions of the initial conditions down to scales of a few Mpc/h, if the input data consists of a complete sampling of the density field $\delta(\boldsymbol{x})$ at $z = 0$, typically from a test N-body simulation. However, the situation drastically worsens if one is presented realistic observational data. A fundamental problem of all these methods is that they are strongly affected by observational biases (Lavaux et al. 2008), such as redshift distortions, the poorly determined relation between mass and luminosity, edge and finite-volume effects, and especially data incompleteness. Indeed, contributions to the displacement field that come from outside of the data zone or from gaps in the data like the Zone of Avoidance (ZoA), cannot be recovered by any of those methods. This is particularly unfortunate

if one is interested in precisely those contributions, such as the large-scale flows of the Local Universe and the yet unsolved problem of the peculiar velocity field convergence towards the CMB dipole.

4.1.3 Velocity-based methods

The first method to reconstruct cosmological initial conditions using peculiar velocity data is the time machine of Nusser & Dekel (1992). They first reconstruct the full three-dimensional velocity field $\boldsymbol{v}(\boldsymbol{r})$ from the radial velocity data using the POTENT method (Bertschinger et al. 1990), although other reconstruction procedures can be used as well, and derive from it the gravitational potential using the linear theory equation. Assuming the Zeldovich approximation with no orbit crossing, and an irrotational velocity field, they cast the Zeldovich equation into Eulerian coordinates, yielding the Zeldovich-Bernoulli equation, and then integrate this back in time. This gives a result that resembles the original initial conditions when smoothed on a scale of ~ 10 Mpc$/h$. However, the method suffers from the non-linearity bias and systematically overestimates overdensities and underestimates underdensities, so that no run-capable initial conditions can be produced. Also, being an Eulerian method, it does not attempt a full Lagrangian reconstruction.

The method currently used in the CLUES project was already described in Chapter 3. Under the assumption that the peculiar velocity field traced by the data is sufficiently close to the linear velocity field at the initial conditions, the data is directly used as constraints to generate constrained initial conditions with the WF/CR algorithm. By using velocity data instead of the density field, the non-linearity bias is significantly reduced, and the WF/CR algorithm by design generates run-capable ICs that are a valid statistical realisation of the linear power spectrum $P(k)$. On the other hand, the result is only valid for scales larger than that of the displacement field (~ 10 Mpc$/h$), because no Lagrangian reconstruction is performed and the velocity constraints for the ICs are placed at their observed $z = 0$ position.

It is clear that to fully exploit the possibilities of peculiar velocity data for constrained simulations, one should use them together with a Lagrangian reconstruction method. It is however surprising that all methods of Lagrangian reconstruction, or actually any Gaussianisation procedure that goes beyond simple linear theory, have so far concentrated exclusively on density fields (traced by galaxy redshift data or given by a simulation) as the input. In this chapter, we present a novel self-consistent method for Lagrangian reconstruction from radial peculiar velocity data, based on the Zeldovich approximation, that can be used to generate run-capable initial conditions. The remainder of this chapter is organised as follows: in Section 4.2 we summarise the method and test how well it can be applied to peculiar velocities at $z = 0$; in Section 4.3 we then construct a set of mock radial velocity catalogues from a cosmological simulation, and use them in Section 4.4 to extensively test the method and quantify the impact of different observational errors and limits. Finally, in Section 4.5 we extend the method to second-order Lagrangian perturbation theory (2LPT) to see if further improvements on the reconstruction quality can be obtained.

4.2 The Reverse Zeldovich Approximation

In order to incorporate a Lagrangian reconstruction scheme into the CLUES method, the procedure can be divided in two parts. First, we reconstruct the displacement field at discrete positions using the data, which is described here. Then, in an approach similar to Lavaux (2010), we use these values of the displacement fields to generate run-capable constrained initial conditions and produce constrained simulations, which will be discussed in the next chapter.

4.2.1 Halo displacements and velocities

Let us assume again that we have a particle discretisation of the overdensity field $\delta(\boldsymbol{r}, z)$ at some redshift z, with particles at comoving positions $\boldsymbol{x}(z)$, which evolved from the homogeneous initial conditions $\boldsymbol{q}$ by

$$\boldsymbol{x}(z) = \boldsymbol{q} + \boldsymbol{\psi}(z) \quad , \tag{4.2}$$

where $\boldsymbol{\psi}(z)$ is the displacement field. The velocity of each particle can be described as $\boldsymbol{u} = \mathrm{d}\boldsymbol{x}/\mathrm{d}t$. The remainder of this chapter is formulated in comoving coordinates, so that by "velocity" we always mean the comoving peculiar velocity. Then, the Zeldovich approximation gives an equation connecting the displacement field and the velocity field based on the assumption of straight paths,

$$\boldsymbol{u}(z) = \dot{a} f \boldsymbol{\psi}(z) \quad , \tag{4.3}$$

where $\boldsymbol{\psi}(z) = D_+(z)\boldsymbol{\psi}_0$. At redshift $z = 0$, comoving and real-space coordinates coincide, and further $a = D_+ = 1$, so that $\dot{a} = H_0$. The equations then reduce to

$$\boldsymbol{r} = \boldsymbol{q} + \boldsymbol{\psi} \quad , \tag{4.4}$$

$$\boldsymbol{v}(\boldsymbol{r}) = H_0 f \boldsymbol{\psi} \quad . \tag{4.5}$$

Whenever we use $\boldsymbol{r}$ instead of $\boldsymbol{x}$, and $\boldsymbol{v}$ instead of $\boldsymbol{u}$ (we drop the subscript "pec" for the following), we mean the same quantities but in the special case of $z = 0$, where comoving and proper coordinates coincide. By $z = 0$ the approximation will, in general, break down in overdense regions, but we can assume it for now. Thinking in the other direction, this enables us to obtain an estimate of the displacement field $\boldsymbol{\psi}$ and the Lagrangian position $\boldsymbol{q}$, if the velocity $\boldsymbol{v}$ is known. For the following, the main idea is that we are not interested in recovering the full displacement field $\boldsymbol{\psi}(\boldsymbol{r})$ everywhere in the computational box; rather, it is enough to recover it at the positions of discrete data points. We can then use these discrete displacement field values and their reconstructed Lagrangian positions to constrain initial conditions, utilising the WF/CR algorithm. This approach works well with the first-order Zeldovich approximation, since it is completely local and can be performed at discrete locations, whereas higher-order extensions depend on integrals over the computational volume.

The data points where velocity data is available observationally correspond to galaxies, each of which is embedded in a dark matter halo. In order to test the validity of the Zeldovich approximation for such a sampling of the velocity field at $z = 0$, we can thus apply it to dark matter haloes in a cosmological simulation. For the remainder of this chapter we use the BOX160 simulation already discussed before. Dark matter halo positions and velocities have been identified from the $z = 0$ snapshot using the AHF halo finder. Only haloes of mass $\log(M/M_\odot) \geq 11.5$ from the simulation are considered; we want to discard haloes that are either too poorly resolved to obtain reasonable estimates on their peculiar velocity, or too small to host galaxies that would be observable in a galaxy distance survey.

Figure 4.1 illustrates a typical medium-size halo from the simulation. The positions of particles within the virial radius R_{vir} that have been identified by AHF are shown with black dots. The halo has a position $\boldsymbol{r}$, which is defined by AHF as the position of the most bound particle, and a velocity $\boldsymbol{v}$, which is the mean velocity of all of the halo's particles. In the initial conditions at $z_{\mathrm{init}} = 30$, the same particles, identified by their IDs, occupy the positions marked with the blue dots. Most of them form a coherent patch in space, the protohalo[18]. This patch covers an

[18] Note that a small fraction of the particles at z_{init} is not connected to the protohalo patch, i.e. the Lagrangian volume is disjoint. Those particles mostly ended up in the virialised halo after accretion or merging processes and

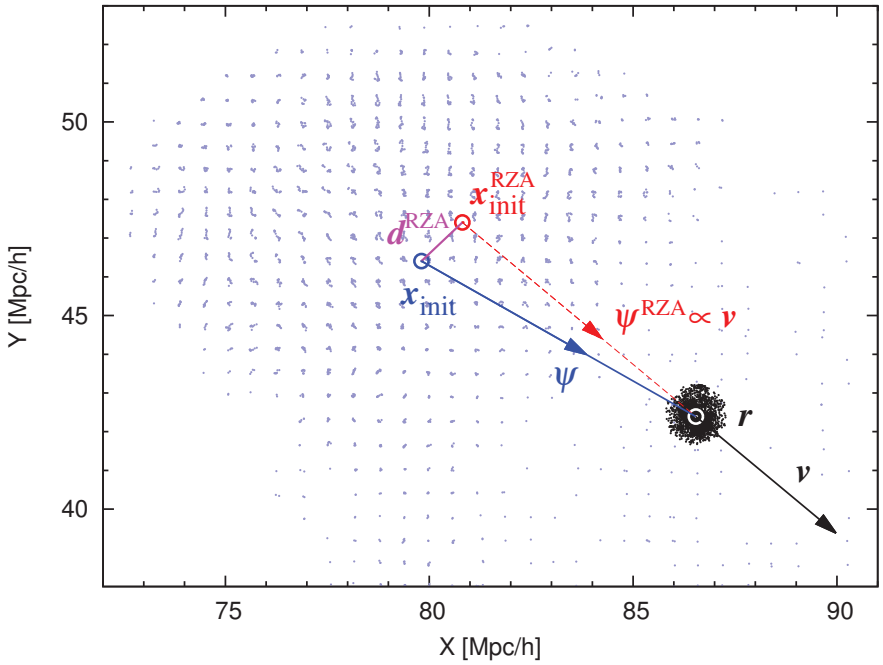

Figure 4.1: RZA on a simulated dark matter halo identified at $z = 0$ with virial mass $M = 7.9 \times 10^{12} M_\odot$, virial radius $R_{\rm vir} = 410$ kpc$/h$. Black dots: positions of all particles inside $R_{\rm vir}$ at $z = 0$, with mean velocity $\boldsymbol{v}$ and the halo centre at position $\boldsymbol{r}$. Blue dots: positions of the same particles in the initial conditions at $z_{\rm init} = 30$, with centre-of-mass at $\boldsymbol{x}_{\rm init}$ ("initial position"). $\boldsymbol{\psi}$ (blue arrow): actual displacement, with $|\boldsymbol{\psi}| = 8.36$ Mpc$/h$. $\boldsymbol{\psi}^{\rm RZA}$ (red arrow): RZA reconstructed displacement. $\boldsymbol{x}_{\rm init}^{\rm RZA}$: RZA reconstructed initial position. $d^{\rm RZA}$ (purple): the RZA error.

overdense region in the initial conditions that will later collapse to form the halo. If we neglect the tiny initial displacements of the particles $\boldsymbol{\psi}_{\rm init}$ at the initial conditions by approximating $\boldsymbol{x}_{\rm init} \approx \boldsymbol{q}$ for each particle, then the volume $V_{\rm init}$ occupied by the protohalo corresponds to the Langrangian volume of the halo. This volume can be quite big, measuring about 10 Mpc$/h$ in diameter in this case or even more than 20 Mpc$/h$ for a massive cluster. Since the initial density distribution is almost uniform, this volume depends directly on the mass via $V_{\rm init} \propto M^3$. We then define the "initial position" of the halo to be $\boldsymbol{x}_{\rm init}$, averaged over all particles, and the displacement of the halo as $\boldsymbol{\psi} = \boldsymbol{r} - \boldsymbol{x}_{\rm init}$. It is important that those are now quantities averaged over the whole halo, rather than taken for individual particles.

Let us assume now that this halo hosts a galaxy with an observable velocity $\boldsymbol{v}$. We continue to use the assumption introduced in Section 2.3 that the peculiar velocity of any observed galaxy follows the mean peculiar velocity of its surrounding dark matter halo host. Then we can directly use the velocities $\boldsymbol{v}(r)$ of the haloes in a simulation as a simple model for the observational data. In what we call the Reverse Zeldovich Approximation (RZA), we can now simply reverse

subsequent relaxation that happened much later than $z_{\rm init}$ during the non-linear structure formation process. We will not attempt to track this process for each individual halo and choose to not treat those particles separately, although they may affect the estimation of $\boldsymbol{x}_{\rm init}$ and $\boldsymbol{\psi}$. We also add that while the halo in Figure 4.1 is the most common case, there are some more extreme cases with more disjoint Lagrangian regions; the corresponding haloes at $z = 0$ are mostly the result of violent major merger events. For a thorough study of protohaloes, see Ludlow & Porciani (2011).

equations 4.4 and 4.5 and estimate the displacement of a halo by

$$\boldsymbol{\psi}^{\mathrm{RZA}} = \frac{\boldsymbol{v}}{H_0 f} \quad , \tag{4.6}$$

and the initial position of the halo by

$$\boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}} = \boldsymbol{r} - \frac{\boldsymbol{v}}{H_0 f} \quad . \tag{4.7}$$

Both $\boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}}$ and $\boldsymbol{\psi}^{\mathrm{RZA}}$ could then be used to place a constraint for generating initial conditions. (This procedure constitutes the second step of our algorithm and will be described in Chapter 5.) This reconstructed initial position will be, in general, at some distance d^{RZA} from the actual initial position, which we define to be the "RZA error" for this object,

$$d^{\mathrm{RZA}} = |\boldsymbol{x}_{\mathrm{init}} - \boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}}| = |\boldsymbol{x}_{\mathrm{init}} - \boldsymbol{r} + \frac{\boldsymbol{v}}{H_0 f}| \quad . \tag{4.8}$$

The RZA error d^{RZA} provides an easy way to quantify the scale length down to which the RZA is valid for different kinds of haloes and environments. Such a study will be presented in the next section.

4.2.2 RZA application to velocities at $z = 0$

The Zeldovich approximation breaks down when shell crossing occurs on the scales that we consider, marking the transition to the non-linear phase of structure formation. We expect that the RZA is not valid at all for haloes gravitationally bound to more massive haloes, i.e. orbiting, infalling, and merging substructure, since the magnitude and direction of their velocities at $z = 0$ have been significantly altered by those processes. So in order to obtain a good estimate of the displacement field, those objects should be discarded. Physically, this would mean to detect and remove haloes that are gravitationally bound to more massive host objects, as well as ongoing major mergers and possibly other scenarios. This is quite a complicated task in itself, so we use a simplified scheme instead, which does not consider the underlying physics but works well for our purpose. Rather than properly finding actual subhaloes (such as in e.g. Knollmann & Knebe 2009), we simply detect haloes that share at least one dark matter particle with another halo and therefore presumably interact in some way or another, without investigating the nature of this interaction. We then call a halo a "subhalo", if it shares at least one dark matter particle with another halo more massive than itself. Conversely, we call a halo a "main halo" if it shares only particles with less massive haloes or with no other haloes at all. In this way, we divide all the identified haloes in the simulation into two groups, subhaloes and main haloes, and keep only main haloes for the RZA analysis. This simple and rather conservative scheme is very effective in filtering out virial motions from the peculiar velocity data, since from clusters or other gravitationally bound systems only the main object will be retained.

In observational data, it is likewise not always possible to determine whether some galaxies are gravitationally bound to each other or otherwise involved in non-linear interactions. However, in the case of observational data we need another scheme, since the surrounding dark matter haloes are not directly observable. We hence introduce an alternative selection method that should also work with galaxies: we determine the isolation radius R_{iso} of a halo to be the distance to the next more massive halo, or in other words, the radius within which a halo is the most massive; then we discard all "non-isolated haloes" with R_{iso} less than some critical value, and keep the rest as "isolated haloes". We choose $R_{\mathrm{iso}} = 2.5$ Mpc$/h$, which ensures that all isolated haloes

will be well outside of each other's virial radii. Thus, by construction, all isolated haloes are also main haloes, and we will likewise filter out all virial motions from the data.

Figure 4.2 shows the absolute halo displacement $|\boldsymbol{\psi}|$ over the halo velocity $|\boldsymbol{v}|$ and the angle between those vectors at $z = 0$ for different sets of haloes. As expected, subhaloes (green) contain little to no information about their cosmological displacement from their momentary velocities at $z = 0$. On the other hand, for the main haloes (red) and even more so the isolated haloes (blue) the approximation holds reasonably well. In the low-velocity regime, which mostly corresponds to the low-overdensity regime, the relation is satisfied nearly perfectly; further up the plots reproduce the well-known tendency of the Zeldovich approximation to underestimate the total velocities (e.g. Bouchet et al. 1995), since it neglects the additional gravitational acceleration and deflection created by the dynamically changing density distribution. Nevertheless, the direction of displacement is in general conserved very well in the velocity vector: for the majority of identified haloes the angle between the two, $\alpha = \mathrm{acos}(\frac{\boldsymbol{v}\cdot\boldsymbol{\psi}}{|\boldsymbol{v}|\cdot|\boldsymbol{\psi}|})$, lies below 10°.

After conducting this study, we learned that a similar investigation was already carried out by Sheth & Diaferio (2001), who compared the peculiar velocities of haloes at $z = 0$ with their *initial* velocities in the linear regime, instead of their displacement as we did here. They likewise found that halo peculiar velocities at $z = 0$ retain the information from the initial conditions relatively well, with deviations in the angle of motion of typically only 10°, if one thoroughly filters out virial motions. Our results are in very well agreement with theirs.

The remainder of this analysis concentrates on the main haloes, since we now established that subhaloes should not be considered for RZA. This leaves us with a total of 29122 objects with $\log(M/M_\odot) \geq 11.5$ within the BOX160. We focus on the error d^{RZA} of the RZA on the initial halo position guess (equation 4.8), a practical quantity to estimate the validity of the approximation. Figure 4.3 shows d^{RZA} (blue) depending on several properties of the haloes,

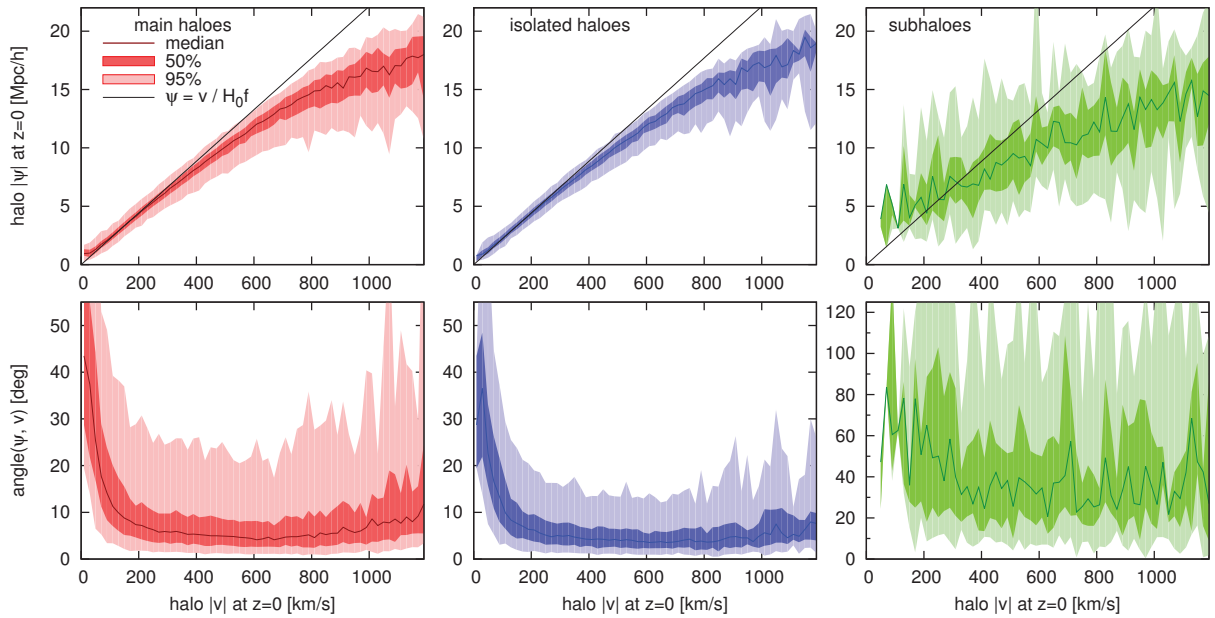

Figure 4.2: Absolute displacement $|\boldsymbol{\psi}|$ vs. absolute peculiar velocity $|\boldsymbol{v}|$ (top row) and the angle between these two vectors (bottom row) at $z = 0$ for different sets of haloes. Main haloes (red): haloes after all their subhaloes (if any) have been discarded. Isolated haloes (blue): haloes which are the most massive ones within a radius of $R_{\mathrm{iso}} = 2.5$ Mpc$/h$. Subhaloes (green): haloes that have been classified as substructure of a more massive halo. Note the different scale for the subhaloes' angles.

compared to the error we would make without any Lagrangian reconstruction (red), in which case the error is simply the displacement $|\boldsymbol{\psi}|$ itself.

The analysis reveals that a majority of the main haloes have a surprisingly low d^{RZA}: the median is at 1.36 Mpc/h, the mean at 2.3 Mpc/h. This is well below the scale on which a quasi-linear approximation is normally considered to be valid. Above all, it is a significant improvement over the "0th order" approximation that the overdensity peaks traced by the haloes do not move at all, leading to a mean error of $\langle|\boldsymbol{\psi}|\rangle = 8.7$ Mpc/h. The distribution of d^{RZA} is highly skewed: for most of the haloes, d^{RZA} is within a few Mpc/h, but at the same time a small fraction of objects has a very high d^{RZA}. Because of this skewness, the median and the upper and lower quartiles are shown, being more meaningful than the mean and the 1σ interval.

The top row of Figure 4.3 demonstrates the expected behaviour: the success of RZA depends highly on the underlying overdensity. In higher-density regions, the non-linear enhancement of peculiar velocities is stronger, and the regions are certainly shell-crossed, so that the Zeldovich approximation is not an optimal description of the dynamics. Although we discarded the sub-

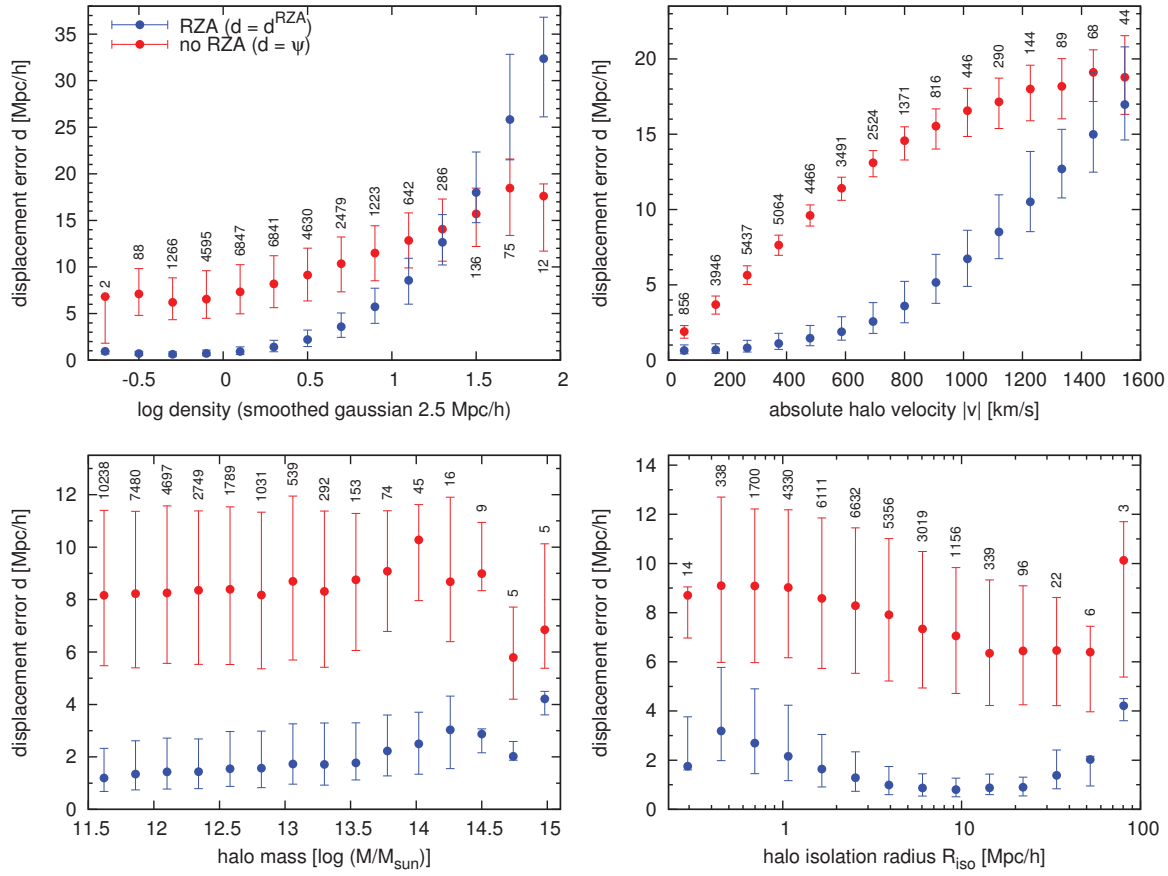

Figure 4.3: Displacement error with RZA reconstruction (d^{RZA}, blue) and without RZA reconstruction ($|\boldsymbol{\psi}|$, red), for all main haloes with mass $M > 10^{11.5} M_{\odot}$ inside BOX160, depending the underlying density, the absolute halo velocity, the halo mass, and the haloes' isolation radius (distance to the closest more massive halo). For each bin, the point is placed at the median, and the bar shows the interval between the 25$^{\mathrm{th}}$ and 75$^{\mathrm{th}}$ percentile. Additionally, the number of haloes in each bin is given. Note the different scales for the d axis.

haloes, still some shell crossing will occur for the main haloes in the overdense regions. The dependence of d^{RZA} on the total velocity is also strong because high velocities are associated with dense environments. A couple hundred objects even have $d^{\mathrm{RZA}} > |\boldsymbol{\psi}|$, meaning that RZA completely fails there. They stick out in the top left panel of Figure 4.3 as the three last bins with the highest density. All of those outliers are relatively low-mass objects travelling at high velocities in the immediate vicinity of one of the most massive clusters in the box and thus experiencing significant non-linear contributions to their peculiar velocities. They can be removed by a scheme that is more rigorous than our shared-particles approach, such as the isolation criterion with a high enough R_{iso}. In order to catch those extreme outliers, it is thus sufficient to apply this more rigorous scheme to the most dense environments only. In observational data, this effectively means reducing rich galaxy clusters to a single data point, while keeping field galaxies ungrouped. From the lower right panel it can be seen what happens if an additional cut in R_{iso} is applied instead to the whole set of main haloes: the retained objects (on the right side) will have a significantly lower d^{RZA} in average, but at the same time we would remove a substantial fraction of data points.

The lower left panel reveals that there is only a very weak dependence of d^{RZA} on the actual mass of haloes. Furthermore, the displacement $|\boldsymbol{\psi}|$ shows no significant correlation with the mass either. This echoes the weak dependence of halo peculiar velocities on their mass (cf. Section 2.3.4). A simulation with larger boxsize would provide better statistics on the most massive objects, where a slight effect can be seen, but they are not a focus of this work.

From this theoretical study on identified haloes in a cosmological simulation it is clear that the RZA can provide a reasonable estimate of the cosmological displacement and the initial position for most of the objects. The primary interest is now how well this scheme can be applied to more realistic observational data. This is discussed in the following section.

4.3 Mock catalogues

So far the study on RZA was a purely theoretical one, using a complete dark matter halo catalogue extracted from a simulation at $z = 0$. The goal however is to eventually apply RZA to observational peculiar velocities. We completely neglected so far that for any galaxy's peculiar velocity $\boldsymbol{v}^{\mathrm{pec}}$, only the radial component v_r^{pec} is observable, that there might be a significant error on such measurements, and that they can be obtained only in a limited volume and with a significant incompleteness. A powerful tool to reconstruct the full three-dimensional velocity field from such data, the Wiener Filter, has been introduced in Chapter 3, along with its implementation in the ICeCoRe code. The idea presented here is to combine Wiener filtering on the data, reconstructing the velocity field, with the RZA reconstruction.

First we generate realistic radial velocity mock catalogues, with different properties, from simulation data. We can then feed this data into the ICeCoRe code as constraints and compute for each mock the three-dimensional Wiener filter mean field $\boldsymbol{v}^{\mathrm{WF}}$. We then use the $\boldsymbol{v}^{\mathrm{WF}}$ vector at the position of each object as an estimate of its three-dimensional velocity, and via the RZA, its displacement field $\boldsymbol{\psi}^{\mathrm{RZA}}$ and initial position $\boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}}$. We can then compare it to the original $\boldsymbol{\psi}$ and $\boldsymbol{x}_{\mathrm{init}}$, which is known from the underlying simulation, to quantify how well the RZA reconstruction from each mock performs, and compute the RZA error d^{RZA}. The impact of different observational errors implemented in the mocks and the overall performance of the procedure can then be quantified.

4.3.1 The BOX160 mock volume

As the "test universe" and the source for mock catalogues we choose again the BOX160 simulation. It contains reproductions of the main structures of the Local Universe: The Virgo cluster within the Local Supercluster, the Coma and Perseus-Pisces superclusters, the Hydra and Centaurus superclusters, and a Great Attractor. The reproduction is reasonably well but not quite exact; notably, the "Virgo" cluster of BOX160 with virial mass $M_{\rm vir} = 3.25 \times 10^{14} M_\odot/h$ is somewhat less massive than its observed counterpart ($M_{\rm vir} \approx 7 \times 10^{14} M_\odot$, Fouqué et al. 2001).

We choose a galaxy group identified in BOX160, which consists of three main haloes with virial masses of 4.9, 6.0 and $6.7 \times 10^{11} M_\odot/h$; we choose the middle one, calling it the simulated "Milky Way". The position of this halo, $\boldsymbol{r}_{\rm MW}$, marks the fixed position of the mock observer. The halo is at a distance of 17 Mpc/h to the simulated BOX160 "Virgo", the next massive cluster. This distance is somewhat larger than the actually observed distance from the Milky Way to the centre of the Virgo Cluster ($\approx$ 16 Mpc or 11 Mpc/h, Fouqué et al. 2001). However, this is not important here, because we want to test the RZA reconstruction method itself, without explicitly comparing the specific cosmography of the chosen mock volume to the observed Universe. In fact, one could carry out this study on a completely random realisation; by choosing the BOX160, we just make sure that the large-scale configuration of our chosen proving ground is not too different from the observed Local Universe, so that we can later apply the method to observational data without expecting completely different results. For now, we treat the BOX160 just as a realisation of some "test universe", without explicitly considering its specific cosmography.

To construct a mock catalogue, we cast a sphere with a fixed radius $R_{\rm max}$ around $\boldsymbol{r}^{\rm MW}$, and consider only haloes within this sphere – the "observational volume". The default value is $R_{\rm max} = 30$ Mpc/h, which mimics a redshift cut at 3000 km/s, similar to the Cosmicflows-1 catalogue (Tully et al. 2009; Courtois et al. 2012). The largest mock volume used in this study has $R_{\rm max} = 60$ Mpc/h, concurring with upcoming Cosmicflows data that will extend to 6000 km/s. Since $\boldsymbol{r}^{\rm MW}$ is located near the centre of the 160 Mpc/h simulation box, even at $R_{\rm max} = 60$ Mpc/h we are sufficiently far away from the box edge, so we will not suffer from the effects of the periodic boundary conditions discussed in Section 3.3.3.

The standard choice of the sphere within 30 Mpc/h of the chosen MW candidate places the BOX160 Virgo cluster and the thick, overdense filament surrounding it well inside this volume. There, Virgo is the third object by mass after the Hydra and Centaurus clusters. The other massive clusters are all outside of this sphere, which is only 2.8% of the total box volume. The larger $R_{\rm max} = 60$ Mpc/h sphere, which encompasses to 22% of the total box, completely contains the BOX160 Great Attractor, and touches the BOX160 Coma and Perseus-Pisces clusters at its edge.

Figure 4.4 shows the distribution functions of the displacement field components $\psi_{x,y,z}$ for all main haloes in the whole box (left) and only inside the chosen $R_{\rm max} = 30$ Mpc/h subvolume (right). In the whole simulation, each of the components is very close to a Gaussian distribution with zero mean, i.e. there is no net bulk flow when averaging over the whole simulation volume. The fact that the ψ_x, ψ_y, and ψ_z components have slightly different standard deviations is a finite-volume effect typical for such a boxsize (cf. Section 3.1.3). It is interesting to note that this sampling of $\boldsymbol{\psi}$ with main haloes is so close to Gaussian statistics, although technically it introduces a sampling bias by tracing only the peaks of the overdensity field. This means that data from such a main halo sampling provides an adequate input for Wiener Filter reconstruction.

On the other hand, the right side of Figure 4.4 in the relatively small mock volume reveals that there is a significant net displacement of about 4 Mpc/h. Also, the mock volume as a whole

is overdense compared to the box average, which is a well established fact by observations and means that it is further non-linearily evolved than the average field. While the main haloes in the box have a median displacement $|\boldsymbol{\psi}|$ of 8.7 Mpc/h and a median RZA error d^{RZA} of 1.36 Mpc/h, for the 1243 main haloes inside the mock volume the median $|\boldsymbol{\psi}|$ is at 11.7 Mpc/h and the median d^{RZA} at 2.8 Mpc/h. The net overdensity also means that there is a net inflow into the observational volume. It is interesting to see how the Wiener Filter reconstruction will handle such a difficult case. In the Cosmicflows-1 data, H_0 is chosen such that there is no net inflow/outflow with respect to the data zone (Tully et al. 2008; Courtois et al. 2012; see also section 3.2.2). This restriction may not be required in general. For the mock data, we keep the value of $h = 0.73$ as given by the simulation parameters.

4.3.2 Generating mock data

The AHF catalogue gives the positions $\boldsymbol{r}$, velocities $\boldsymbol{v}$, and virial masses M for all identified haloes in the BOX160 simulation. Since the AHF velocities are in comoving coordinates, the Hubble flow is not present; comoving velocities at $z = 0$ correspond to the observed peculiar velocity and comoving positions at $z = 0$ correspond to physical positions in real space. To generate a mock radial velocity catalogue, we first shift these halo positions so that they are centered on the mock observer, $\boldsymbol{r} - \boldsymbol{r}^{\mathrm{MW}} \to \boldsymbol{r}$. Recall that we consider only main haloes and only those with mass $\log(M/M_\odot) > 11.5$. We then have for all haloes their distance $r = |\boldsymbol{r}|$ and their radial velocity $v_r = \boldsymbol{v} \cdot \boldsymbol{r}/r$ relative to $\boldsymbol{r}^{\mathrm{MW}}$. For the mock datapoints, we consider only haloes within $r \leq R_{\mathrm{max}}$.

Observationally, radial peculiar velocities v_r^{pec} and their errors are obtained from the measured galaxy distance r, some estimate of the absolute distance error Δr, and the observed redshift v_r^{obs} via

$$v_r^{\mathrm{pec}} = v_r^{\mathrm{obs}} - r \cdot H_0 \quad , \tag{4.9}$$

$$\Delta v_r^{\mathrm{pec}} = -\Delta r \cdot H_0 \quad . \tag{4.10}$$

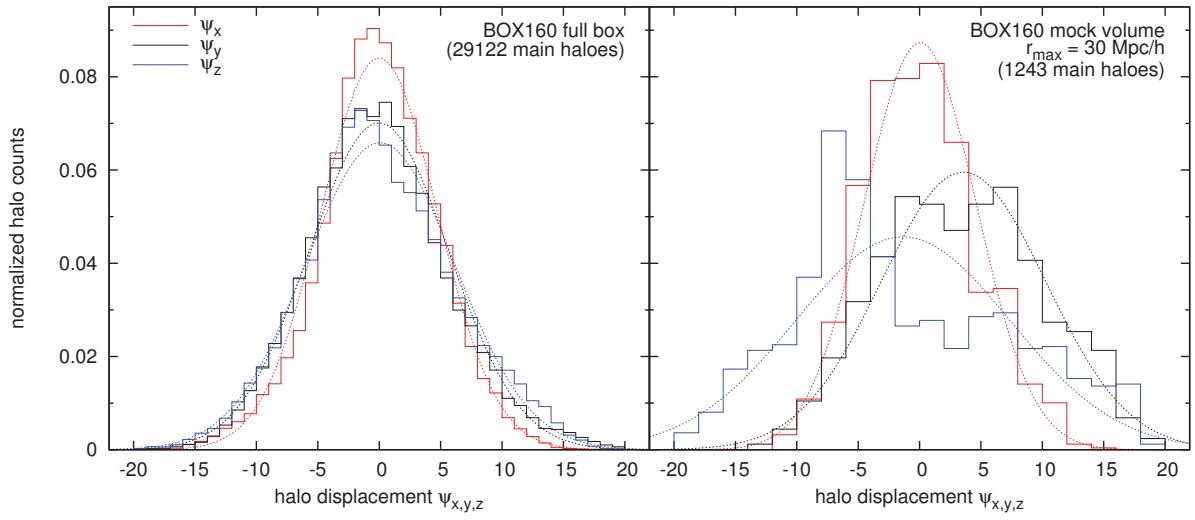

Figure 4.4: Distribution of halo displacements ψ_x, ψ_y, ψ_z for all main haloes with $\log(M/M_\odot) \geq 11.5$ in the BOX160 simulation box (left) and in the $R_{\mathrm{max}} = 30$ Mpc/h mock volume (right). The dotted lines show a Gaussian fit for each component.

To mimic these errors, we first take the known radial velocity of a halo from the AHF catalogue, $v_r^{\rm AHF}$, and the known distance $r^{\rm AHF}$ to the mock observer. In the Cosmicflows-1 catalogue, the different points come from different types of measurements with differently distributed errors between 7 and 20 % (cf. Figure 2.1); here, we simplify the situation by assuming relative distance errors δr that are Gaussian distributed with a constant rms, $(\delta r)_{\rm rms}$. If we choose a fixed value for this rms accuracy of the distance, then the absolute mock distance error is generated via

$$\Delta r^{\rm mock} = {\rm G}(0,1) \cdot (\delta r)_{\rm rms} \cdot r^{\rm AHF} \quad , \tag{4.11}$$

where G(0,1) is a random number drawn from a Gaussian distribution with mean 0 and variance 1. Then, the radial velocity with the added error for the mock catalogue is computed as follows:

$$v_r^{\rm mock} = v_r^{\rm AHF} + \Delta v_r^{\rm mock} \quad , \tag{4.12}$$

$$\Delta v_r^{\rm mock} = -\Delta r^{\rm mock} \cdot H_0 \quad . \tag{4.13}$$

These datapoints then enter the mock catalogue and form the set of input constraints c_i for the WF/CR algorithm. The data are of the radial velocity type with the direction vector $\hat{\boldsymbol{e}}_\mu = \boldsymbol{r}/r$, i.e. the radial component with respect to the mock observer is constrained.

In observational data, $v_r^{\rm obs}$ is observed in the rest frame of the observer; for a reconstruction of the cosmic displacement field, it is more sensible to transform these velocities to a larger rest frame such as the rest frame with respect to the CMB dipole. For the mocks, we achieve a similar setup by not considering the peculiar motion of the simulated MW halo where we placed the mock observer. We rather take directly the velocities in the fixed rest frame of the simulation box as computed by AHF.

4.3.3 The mock catalogue set

From the BOX160 AHF halo catalogue, we created a total of 20 different mock peculiar velocity catalogues in order to test how the observational distance error, the amount and distribution of data points, and the size of the observational volume are going to affect RZA reconstruction. Table 4.1 summarises the basic parameters of all mocks. Each of the mocks is referred to by a name encoding its properties. The first letter characterises the method of halo selection (A – E: by mass cut; L,I,R by other criteria); the next two digits show the radius of the observational volume $R_{\rm max}$ in Mpc/h; and the last two digits are the rms distance error $(\delta r)_{\rm rms}$ in percent. The last two mock catalogues do not feature distance errors but instead contain 3D peculiar velocity data, which will be discussed below. In this case the last two digits are 3D.

The "standard" catalogue is the C30_10, which we consider a "typical" sparse peculiar velocity dataset. We take the procedure of considering only main haloes as a proxy for the "grouping" performed on observational data. The C30_10 contains all main haloes above a mass cut $M_{\rm min} = 10^{11.9} M_\odot/h$ within $R_{\rm max} = 30$ Mpc/h, yielding 588 radial velocity datapoints. This choice gives the C30_10 similar properties to the grouped Cosmicflows-1 catalogue, but somewhat more sparse[19]. The standard choice of 10% rms distance error is also similar to the observational data: while the median rms distance error is somewhat higher at 13% on the individual galaxies in Cosmicflows-1, this error reduces when the galaxies are arranged in groups. The C30_10 is interesting because even if the data are improving in terms of the number of individual galaxy distances, the number of galaxy groups in a radius of 30 Mpc/h is probably not going to vary by much, nor is the accuracy on the most nearby distances.

[19] Choosing this sample was motivated by the fact that at the time we commenced this study, we had a preliminary version of the Cosmicflows-1 available that contained exactly the same number of 588 galaxy groups within 30 Mpc/h. The current catalogue is less sparse with 742 galaxy groups within the same volume.

mock name	constraint type	$R_{\max}$ [Mpc/h]	$(\delta r)_{\rm rms}$	mass cut [$\log M/M_\odot$]	halo selection	M	$\sigma_{\rm NL}$ [km/s]	
C30_00	v_r	30	0 %	> 11.9	by mass	588	221	I
C30_05	v_r	30	5 %	> 11.9	by mass	588	228	
C30_10	v_r	30	10 %	> 11.9	by mass	588	235	
C30_15	v_r	30	15 %	> 11.9	by mass	588	242	
C30_20	v_r	30	20 %	> 11.9	by mass	588	246	
A30_10	v_r	30	10 %	> 12.3	by mass	282	242	II
B30_10	v_r	30	10 %	> 12.1	by mass	413	235	
D30_10	v_r	30	10 %	> 11.7	by mass	898	216	
E30_10	v_r	30	10 %	> 11.5	by mass	1243	198	
C40_10	v_r	40	10 %	> 11.9	by mass	1256	179	III
C50_10	v_r	50	10 %	> 11.9	by mass	2184	176	
C60_10	v_r	60	10 %	> 11.9	by mass	3518	190	
E40_10	v_r	40	10 %	> 11.5	by mass	2614	156	IV
E50_10	v_r	50	10 %	> 11.5	by mass	4701	154	
E60_10	v_r	60	10 %	> 11.5	by mass	7637	165	
L30_10	v_r	30	10 %	< 11.9	by mass	588	208	V
I30_10	v_r	30	10 %	—	by isolation	588	180	
R30_10	v_r	30	10 %	—	random	588	219	
C30_3D	v_x, v_y, v_z	30	0 %	> 11.9	by mass	1764	183	– VI
E60_3D	v_x, v_y, v_z	60	0 %	> 11.5	by mass	22911	158	

Table 4.1: Overview over the performed Wiener Filter reconstructions of the $z = 0$ peculiar velocity field on mock catalogues extracted from the BOX160 simulation. The data describes the mock catalogue that was used as a set of constraints c_i for the reconstruction. From left to right: an abbreviation used in this chapter to refer to each mock and the reconstruction computed from it; whether only radial or full 3D velocity information was used for the constraints; the radius of the spherical data zone in Mpc/h; the mock rms distance error that was added; the mass limit above which haloes are selected; the selection method; the total number M of constraints (for v_r this is equal to the number of data points); and the $\sigma_{\rm NL}$ parameter that was used for each reconstruction to enforce $\chi^2/{\rm dof} = 1$.

With the C30_10 mock as the starting point, we vary the rms distance error in five steps between none and 20%, yielding the mocks C30_00 through C30_20. We also vary the mass cut from $M_{\rm min} = 10^{12.3} M_\odot/h$ to the minimum of $10^{11.5} M_\odot/h$ in five steps, yielding the mocks A30_10 through E30_10. A30_10, the sparsest sample, has the fewest data points of all mocks with only 282 radial velocities. We also construct mocks with larger observational volumes around $\boldsymbol{r}_{\rm MW}$, varying $R_{\rm max}$ from 30 to 60 Mpc/h in four steps, for two different mass cuts, yielding the mocks C30_10 through C60_10 and E30_10 through E60_10. Although the RZA method itself places no restrictions on the allowed size of the data volume, we do not consider $R_{\rm max} > 60$ Mpc/h here in order to avoid problems with the periodic boundary conditions of the box. The E60_10 mock has the most data points with 7637 radial velocities. We estimate that this mock is comparable to the upcoming Cosmicflows-2 catalogue in terms of data quality.

Next, we want to explore how other halo selection criteria rather than a mass cut will affect the reconstruction. Regarding again the C30_10 mock as a "standard" one, we fix the amount of data points constant at 588, as well as the distance error and the data volume. Then, for the L30_10 mock, we take the 588 next most massive points after the ones in C30_10, so that they all have a mass below $10^{11.9} M_\odot/h$, to check the reconstruction quality if only less massive objects are considered. For the I30_10 mock, we consider the 588 most isolated objects (those with $R_{\rm iso} > 2.1$ Mpc/h) leading to a yet different sampling of the same volume. In the last variation of data selection, we randomly pick 588 points from all main haloes in the data volume, regardless of their mass or other properties, yielding the R30_10 mock. Randomly picking haloes mimicks the observational data of spiral galaxy peculiar velocities obtained with the Tully-Fisher method, which are not located at the highest density peaks of the galaxy distribution (as the haloes selected by mass), but are selected on the random basis of their inclination on the sky being greater than 45 degrees.

Finally, we want to quantify by how much the reconstruction quality is degraded by the fact that only the radial component v_r is observable, rather than the full three-dimensional velocity vector $\boldsymbol{v}$. This is interesting, since as we mentioned in Chapter 3, the transverse peculiar velocities of galaxies may become accessible to observations in the future (Nusser et al. 2012). We construct the mock C30_3D, which has its data points at the same positions as the C30_00 mock, but for each object it lists all three components v_x, v_y, v_z of the velocity instead of v_r. The last mock, E60_3D, is the most extreme case, containing three-dimensional velocities in the largest volume with the lowest mass cut, which makes $3 \cdot 7637 = 22911$ data points in total. Although such a mock has little meaning when compared to current observational data, we use it to test the capability of the ICECORE code to yield a significantly better reconstruction in the case of a very large number of high-quality constraints.

Figure 4.5 shows the peculiar velocity datapoints contained in the mocks within $R_{\rm max} = 30$ Mpc/h as an Aitoff projection centred on the mock observer (without the mock errors). In the dense regions around the massive clusters, there is a lot of variance in the velocities due to the non-linear multistreaming from orbit crossing. Outside of those regions, the flow is quite smooth, making the large coherence length of peculiar velocities apparent. Near the centre of the Aitoff map, there is a patch of galaxies coherently moving away from the observer towards the negative x direction at significant velocities. This is the large-scale flow towards the Great Attractor (GA) of BOX160, which extends into the data zone, although the GA itself is more than 20 Mpc/h outwards from the data zone boundary. Above that region, there is a Local Void (although not as pronounced as in the observed Local Universe) that is devoid of objects. There are several other underdense regions that are only mapped by relatively low-mass objects below the C mass

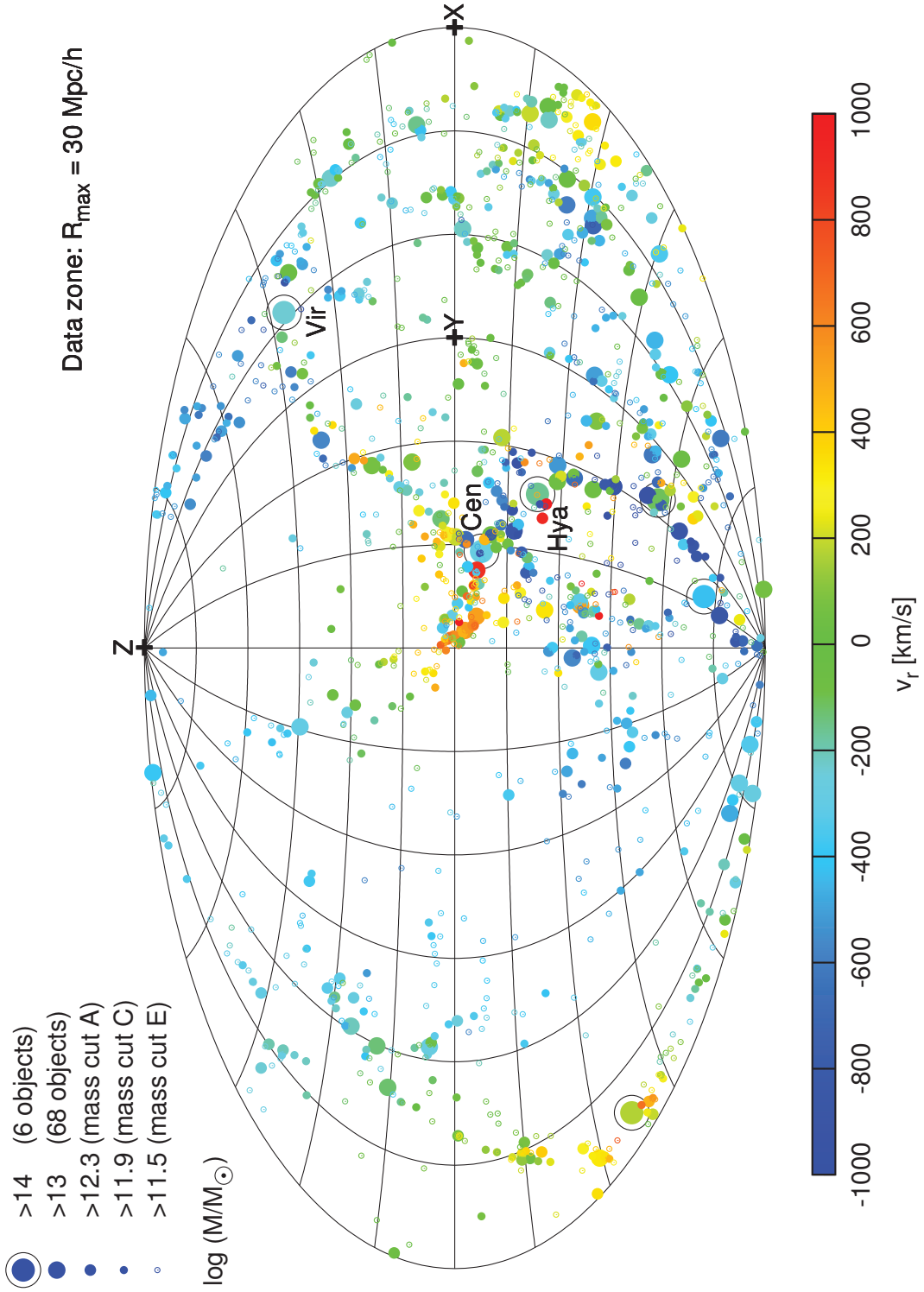

Figure 4.5: Aitoff projection map of the BOX160 mock catalogue radial velocities v_r relative to the mock observer inside a sphere of radius 30 Mpc/h. The labels indicate the three most prominent objects in the data zone (simulated Hydra, Centaurus, Virgo clusters) and the directions of the x, y, z axes.

cut (11.9). Therefore, in the C mocks, there will be huge gaps in the mock volume not mapped by any objects. This means that the mocks based on mass cuts will show a galaxy distribution strongly clustered around the denser regions, while mocks with mass-independent halo selection (L, I, R) will have a more homogeneous distribution, providing a more complete sampling of the mock sky while not actually containing a larger number of data points.

Of course, there are more observational features that could be incorporated in the mocks. For example, one could add a Zone of Avoidance. However, it has been already established that the Wiener Filter mean field successfully extrapolates into such unsampled regions and handles datasets well that are sparse in an inhomogeneous and/or anisotropic way (Courtois et al. 2012). We already exploit this behaviour by using sparse mock catalogues with a very limited observational volume and huge gaps in the data in underdense regions; adding an additional gap will not fundamentally change the situation. One could also think of mimicking the increase of incompleteness with distance, or modelling the different galaxy types and measurement methods in more detail. For this, it would be necessary to populate the haloes with galaxies of different morphologies and luminosities by setting up a full semianalytic model on top of the N-body simulation. This would make the situation unnecessary complex without additional insight into the validity of the RZA and our method of generating constrained initial conditions.

4.4 RZA on Wiener Filter reconstructions

Having generated the different mock catalogues, it is now possible to reconstruct the three-dimensional velocity field with the Wiener Filter and use it as a proxy for the displacement field to perform the RZA reconstruction.

4.4.1 Reconstruction details

Since for RZA we are interested in $\boldsymbol{v}^{\mathrm{WF}}$ only at the discrete positions of the data, we can either compute the full Wiener mean field first and then interpolate the result on the desired positions, or solve for the Wiener filter solely on those discrete positions. The second version is computationally faster and avoids interpolation errors; on the other hand, it does not provide the fully sampled mean field if one chooses to analyse it as well. We found that if we keep the resolution of the simulation box ($L = 160$ Mpc/h; $N = 256^3$) for the full-box WF reconstruction, interpolation errors on the peculiar velocity field are negligible, so we reconstruct the full field first. To filter out non-linear contributions, we choose σ_{NL} for each reconstruction such that $\chi^2/\mathrm{dof} = 1$. For the power spectrum, we use the original WMAP3 $P(k)$ of the BOX160 simulation.

For the ICECORE runs, we choose the analytic correlator, but cut the Fourier integrals for the correlations function at the fundamental frequency of the box, $k_{\mathrm{min}} = k_L = 2\pi/L$, since we know that larger modes are not present within BOX160. The grid correlator would be strictly consistent with the linear correlation function of the periodic cubic grid; on the other hand, this way we are using the same correlator for the mock data as later for the observational data, except the k_{min} cut. Test reconstructions with the grid correlator on the same mocks revealed no significant differences for this choice of the observational volume. This is expected as long as the volume is chosen such that periodic boundary conditions are not important and the constraints probe only scales larger than the box resolution, so that the Nyquist frequency k_{Ny} is not relevant.

4.4.2 Comparison

Figure 4.6 shows the reconstructed velocity field $\boldsymbol{v}^{\mathrm{WF}}(\boldsymbol{r})$ compared to the original field $\boldsymbol{v}(\boldsymbol{r})$ for all three components v_x, v_y, v_z for the C30_10 reconstruction inside the $R_{\mathrm{max}} = 30$ Mpc/h mock

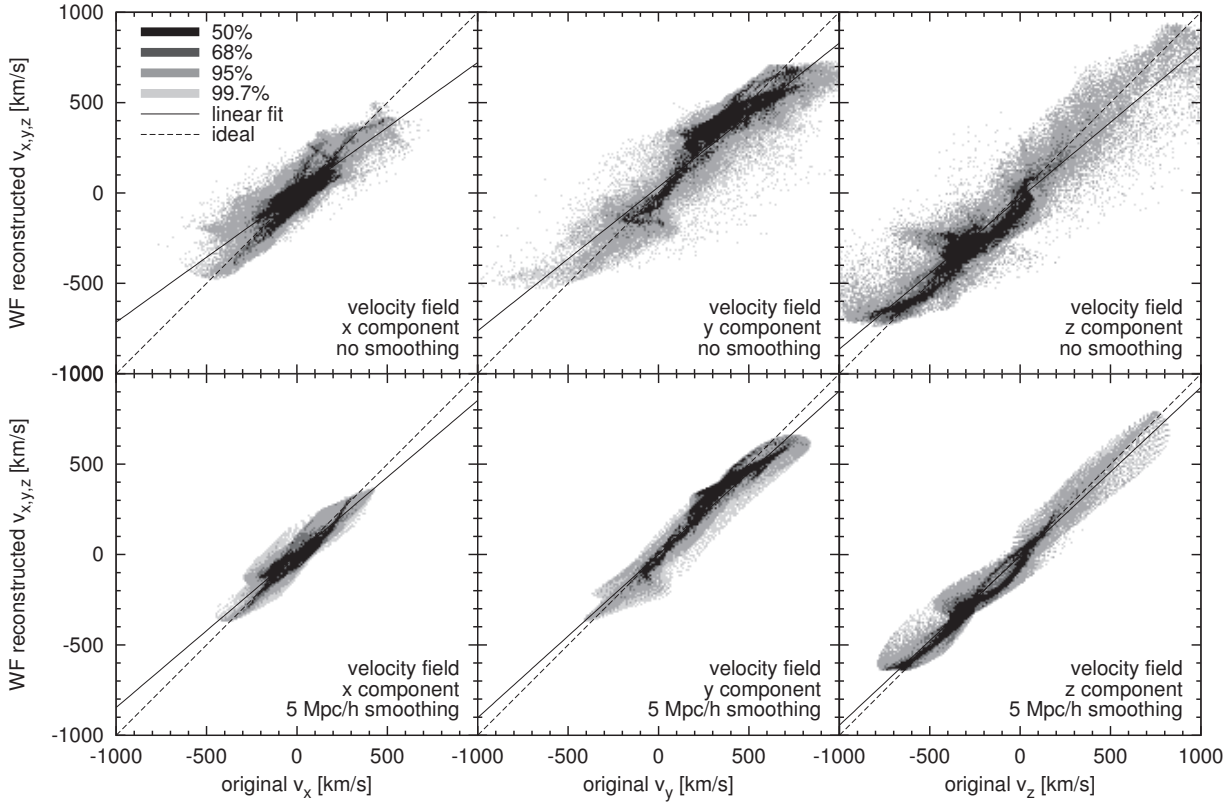

Figure 4.6: Cell-by-cell comparison of WF reconstructed vs. actual velocity field $\boldsymbol{v}(\boldsymbol{r})$ within 30 Mpc/h of the observer, using the C30_10 mock for the reconstruction. Top row: Without smoothing. Bottom row: with 5 Mpc/h Gaussian smoothing on both fields. The solid line shows a linear regression fit $v_i^{\mathrm{WF}} = \beta \cdot v_i^{\mathrm{orig}} + \varepsilon$; the dashed line would be the ideal result $v_i^{\mathrm{WF}} = v_i^{\mathrm{orig}}$.

volume. The upper half of Figure 4.6 is a scatter plot of all the grid cells inside this sphere, together with a linear regression fit showing the resulting slope (solid) versus the ideal slope of 1 (dashed). Averaged over all three components $i = x, y, z$, the rms error per component is $\sqrt{\langle \left(v_i^{\mathrm{WF}}(\boldsymbol{r}) - v_i(\boldsymbol{r})\right)^2 \rangle} = 172$ km/s, the slope is $\beta = 0.67$, and the Pearson correlation coefficient is 0.91. The y offset ε of the linear fit is negligibly small. The slope flattening is due to the typical filtering bias of the WF: where the field is underdetermined by the data, the result will tend towards zero. The filtering bias is discussed in more detail in Section 4.4.7.

The lower half shows a scatter plot of the same grids after both have been smoothed with a 5 Mpc/h Gaussian filter. On this smoothing scale, the reconstruction is very accurate, despite that only 588 data points with significant errors have been used for the reconstruction input: the rms error is just 63 km/s, the slope is 0.92, and the correlation factor is 0.977. This reproduces the known result that on scales above ~ 5 Mpc/h the Wiener Filter yields an excellent reconstruction of the three-dimensional velocity field from sparse and noisy radial velocity data. The protruding or arc-shaped artefacts in the scatter plots are caused by specific local features of the flow structure that the WF failed to reproduce.

However, more interesting for the RZA is a comparison of the Wiener Filter results with the halo displacements. Following the RZA approach, we assume that the WF velocity field is an estimate of the displacement field, $\boldsymbol{\psi}^{\mathrm{WF}} = \boldsymbol{v}^{\mathrm{WF}}/H_0 f$. We compare the displacements $\boldsymbol{\psi}$ of all

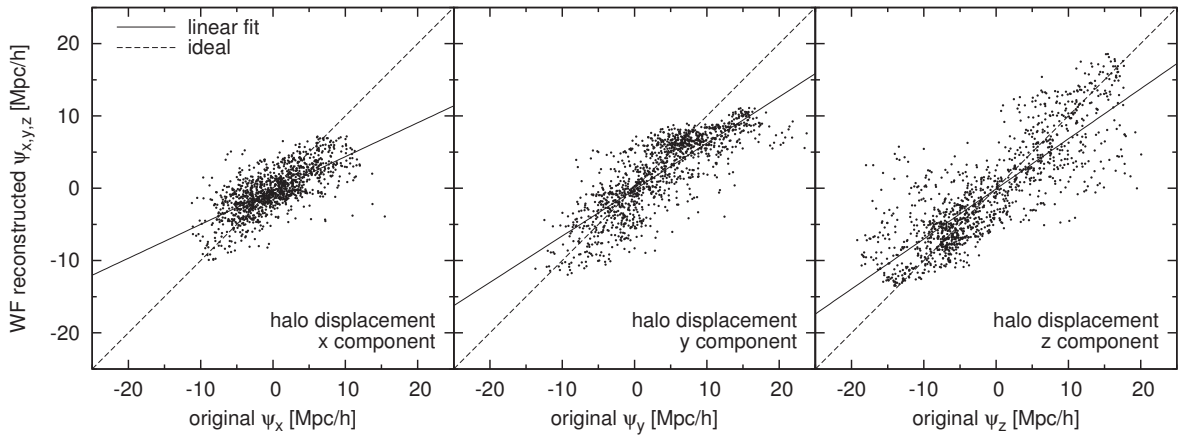

Figure 4.7: Comparison of WF reconstructed vs. actual displacement $\boldsymbol{\psi}$ for all haloes within 30 Mpc/h of the observer at the discrete positions of those haloes, using the C30_10 mock for the reconstruction. The solid line shows a linear regression fit $\psi_i^{\mathrm{WF}} = \beta \cdot \psi_i^{\mathrm{orig}} + \varepsilon$; the dashed line would be the ideal result $\psi_i^{\mathrm{WF}} = \psi_i^{\mathrm{orig}}$.

main haloes within the $R_{\mathrm{max}} = 30$ Mpc/h mock volume with the values of the reconstructed displacement field $\boldsymbol{\psi}^{\mathrm{WF}}$ at the positions of those haloes. Such a scatter plot is given in Figure 4.7 for the C30_10 reconstruction. Here, the average rms error per component is 4.17 Mpc/h, the slope is 0.64, and the correlation factor is 0.82. The correlation is poorer because now, the RZA error (the intrinsic scatter between $\boldsymbol{\psi}$ and $\boldsymbol{v}$ due to non-linearity) is added on top of the error from the imperfect WF reconstruction. Still, we obtain a reasonable correlation.

We now perform the same procedure with all 20 mocks. The data points are fed to ICeCoRe as input, the appropriate σ_{NL} is determined (cf. Table 4.1), and the Wiener Filter mean field for velocity/displacement is computed. Considering only the result inside the 30 Mpc/h sphere, the WF velocity field and halo displacements are then compared with the original simulation. This is fitted with a linear regression line. However it would be tedious to handle such a huge amount of scatter plots. To present the results in a more compact way, we concatenate all three cartesian components for the linear regression and consider the resulting slope and rms error per component. Figure 4.8 shows the slope (black) and rms error (blue) for the velocity field, $\boldsymbol{v}^{\mathrm{WF}}(\boldsymbol{r})$ vs. $\boldsymbol{v}(\boldsymbol{r})$ (dashed lines, open symbols), and the halo displacements $\boldsymbol{\psi}^{\mathrm{WF}}$ vs. $\boldsymbol{\psi}$ (solid lines, filled symbols). The velocity rms has been converted to Mpc/h for comparison through division by the factor $H_0 f$.

The different panels in Figure 4.8 divide the mock reconstructions into groups by the different mock observational parameters that are varied. The roman numerals correspond to the mock groups in Table 4.1. Panel I shows the dependance of the reconstruction quality on the rms distance error; panel II varies the mass cut (and therefore the amount of data points) for the halo mass-selected mocks; panel III shows the effect of increasing the data volume beyond the default $R_{\mathrm{max}} = 30$ Mpc/h for the default mass cut at 11.9; panel IV repeats the same for lower mass cut mocks at 11.5; panel V shows mocks with different selection methods while keeping the number of datapoints constant; finally, panel VI compares radial with three-dimensional data.

4.4.3 Error sources of RZA reconstruction

In general, the slope β is a measure for how well the underlying field is constrained by the data. Ideally it should be 1; the WF filtering bias reduces it to a lower value. The rms error is a

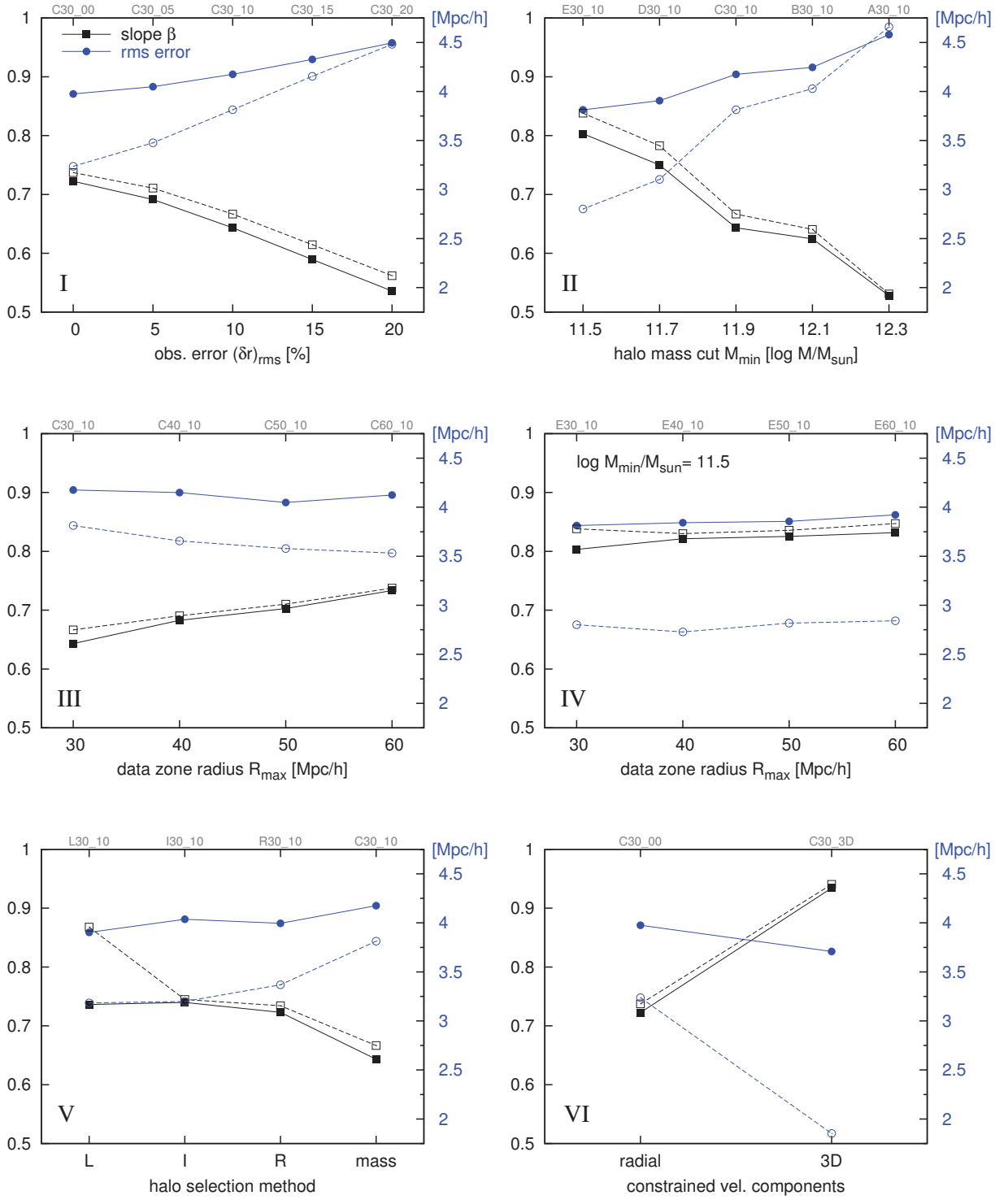

Figure 4.8: Slope β (black; left scale) and rms error (blue; right scale in Mpc$/h$) for a linear regression comparison of the different reconstructed fields with the original one within 30 Mpc$/h$ of the observer. The solid lines represent fits to the original displacement $\boldsymbol{\psi}(\boldsymbol{r})$ of the haloes; the dashed lines are for fits to the velocity field $\boldsymbol{v}(\boldsymbol{r})$ at $z = 0$. The latter have been converted to Mpc$/h$ through division by the factor $H_0 f$, in order to display them in the same scale as the displacement.

measure for how well the WF result reproduces the "true" solution for the velocity field and the halo displacements, respectively.

Generally, for the velocity, the reconstruction can become almost arbitrarily accurate if we sufficiently increase the data quality. On the other hand, the halo displacements (and therefore the RZA reconstruction) are dominated by the RZA error rather than by the WF reconstruction quality; varying the mock properties has a lesser effect. There is a "wall" at around 3.5 Mpc/h rms error per component that cannot be penetrated even with the best-quality mocks. This is the scale at which, averaged over the mock volume, the $\boldsymbol{\psi}$ and $\boldsymbol{v}$ fields themselves disagree, because the quasi-linear assumption is not valid on these scales. Of course, this disagreement varies from region to region, as shown in the analysis in Chapter 4.2.2.

As expected, the distance errors and the mass cut both have a significant influence on the quality of the reconstruction (groups I and II). The latter seems to be more important: starting from the C30_10 mock, a higher improvement is obtained when the number of data points is increased than if the distance errors are decreased. This is interesting when considering the upcoming observational data, where the sparseness of the data will be reduced more effectively than the observational distance errors compared to present observations. The next lower mass cut mock D30_10 with 10% distance errors but 898 radial velocities instead of 588 gives a better reconstruction than the mock that keeps the 588 points but has no distance errors at all. This is remarkable because an rms error of 10% in distance at a distance of 30 Mpc/h leads to a rms error of 300 km/s on the radial velocities, and because of the Gaussian distance error distribution, some velocities in the mocks have error bars up to 100% and higher. It demonstrates that due to their coherence on large scales, peculiar velocities are an excellent input data source for the WF despite the large errors: even a few coherent data points with 100% errors and separated by a few Mpc/h represent a strong measurement of the local velocity field.

Increasing the data volume (groups III and IV) has a much lesser effect. This is expected for peculiar velocity data as opposed to redshift data, where a larger data volume would lead to a significantly better overall result. If we increase the total volume of the mock catalogue out to a distance of 60 Mpc/h, the improvement on the reconstruction inside the 30 Mpc/h is minimal. There is some effect for the high 11.9 mass cut mocks, since the additional information partly compensates for the sparse samling. But for the low mass cut mocks at 11.5 there is no significant improvement, although the E60_10 mock contains already 7637 data points in total. This reflects a known favourable property of the WF: it successfully reconstructs the tidal component of the velocity and displacement fields, i.e. the part that is induced by the mass distribution outside the data zone (e.g. Courtois et al. 2012). This also includes the dipole term, i.e. the bulk motion of the data volume due to the external field, which in our case is significant. Adding more detail on this outside field does not significantly change the tidal component on the inner volume. The RZA reconstruction with WF can therefore work with small data volumes compared to density-based methods. Any such method is by design not able to reconstruct the tidal component: information outside of the data zone cannot be inferred from galaxy positions alone. A density-based Lagrangian reconstruction from a catalogue volume of only 30 Mpc/h would therefore be of little use. On the other hand, Lagrangian reconstruction from peculiar velocities are an ideal tool to study the tidal flows on large scales.

Panel V in Figure 4.8 compares mocks created from different data selection criteria while keeping the number of data points constant at 588 inside 30 Mpc/h. The mass cut C, which picks the 588 most massive objects, is compared against selecting lower-mass objects (L), the most isolated (I) and randomly picked (R) objects. Any of the alternative selection methods works better than selecting by mass. Mass selection is biased towards a higher sampling the overdense regions and a poorer sampling of the less dense regions. Any other selection will sample the

less dense regions more completely and therefore conserve more information about the large-scale modes of the cosmic matter distribution. Additionally, a more homogeneous sampling not biased towards the denser regions will be less affected by the non-linear enhancement bias of the peculiar velocity field. Therefore, as expected, such a sampling creates a WF solution that is better constrained and has a lower rms error. This detail is interesting in the context of observational data. It confirms the prediction we made in Section 2.3.5, that a galaxy sample more evenly distributed around the sky and sufficiently probing less dense regions would lead to better reconstrutions than a galaxy sample preferentially located in dense regions. It suggests that galaxy distance samples of spiral galaxies (such as those derived from the Tully-Fisher relation) may be more ideal for RZA reconstruction than those primarily containing early types, which are biased towards massive surrounding haloes and dense environments.

Finally, Group VI addresses the question how much of the information is missed due to the fact that only radial components of the velocity are observable. For this sake, we constructed the C30_3D mock, pretending that the full 3D velocity vector would be accessible in some way. It has three-dimensional velocity data on the usual 588 haloes inside 30 Mpc/h. In C30_3D, there are no added errors because it has little meaning to try to derive mock 3D velocity errors from some distance error. We therefore compare it to C30_00, which has no errors either. As expected, the increase in reconstruction quality for the $\boldsymbol{v}$ field is dramatic if the full 3D data is used as input. For the WF solution of the velocity field, the contribution to the total rms scatter due to the limitation to radial components is over 100 km/s and thus larger than any other individual error source. On the other hand, for the displacement reconstruction, removing this additional scatter has less effect, because the total scatter is dominated by the RZA error. The filtering bias is almost completely removed by adding three-dimensional information to the data, but the rms error on the displacement components is reduced only from 4.0 to 3.7 Mpc/h. This means that even for very high-quality data, the RZA estimate of the displacement $\boldsymbol{\psi}$ is still significantly limited in precision by the disagreement between the large-scale velocity field and $\boldsymbol{\psi}$ due to non-linear motions that do not follow the Zeldovich approximation.

4.4.4 RZA error distribution

To obtain a more detailed view at the reconstruction quality, we consider the displacement error d^{RZA} for the haloes inside the mock volume. Figure 4.9 shows the distribution of d^{RZA} for different mocks. Recall that d^{RZA} is computed for *all* haloes inside the mock volume, regardless of whether they were part of the particular mock from which a reconstruction was computed, so that we can directly compare the overall reconstruction quality from all mocks.

Compared to the distribution of halo position error without RZA reconstruction (black dashed line), the RZA gives a significant improvement. While half of the haloes have an x_{init} position error above 11 Mpc/h without RZA, this is the case for only around 10% of the haloes with RZA reconstruction. Used as constraints for initial conditions, this leads to much more exact constrained simulations, as we will see in Chapter 5. The distribution of course depends on the quality of the mocks. Typically, the reconstructions from radial velocities have median d^{RZA} values around 5 Mpc/h and a skewed d^{RZA} distribution. For comparison we plot also the "ideal" RZA (solid black), directly using all 3D halo velocities with equations (4.6) and (4.7). As already mentioned, this gives a median d^{RZA} of 2.8 Mpc/h inside the mock volume and a similarly skewed d^{RZA} distribution. The mock groups I and II are represented in red lines in the left and right panels of Figure 4.9, respectively.

The trend of degrading quality with increasing distance errors and decreasing number of datapoints is repeated here. However, the differences between the different d^{RZA} distributions do not seem very huge considering that the respective quality of the mocks differ from each

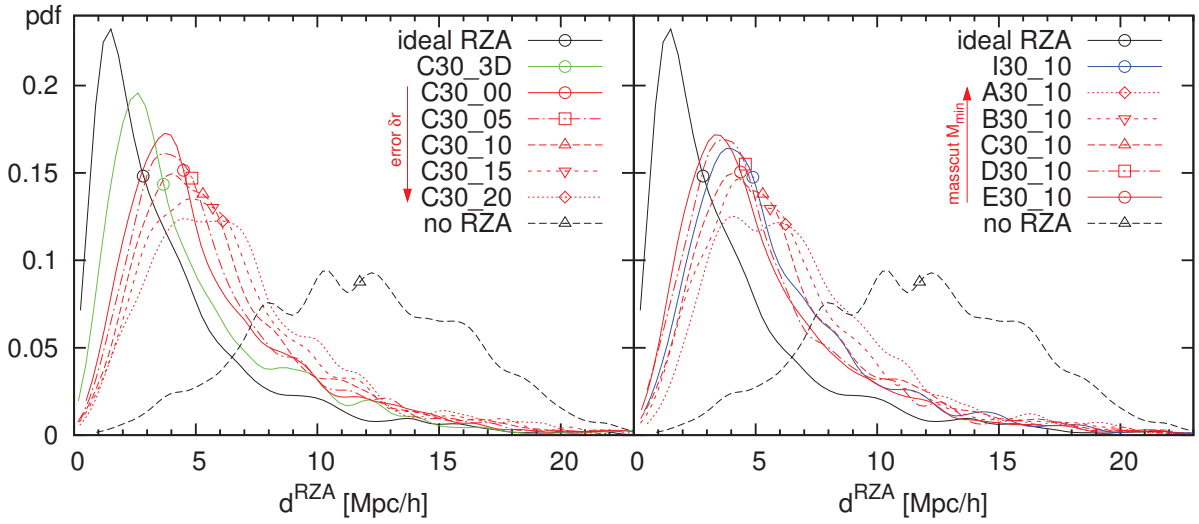

Figure 4.9: Probability distribution function (pdf) of the RZA displacement error d^{RZA} for different mock reconstructions, as well as a theoretical reconstruction using the exact 3D halo velocity of all haloes within the data zone, i.e. $d = |\boldsymbol{\psi} - \boldsymbol{v}/H_0 f|$ (black solid), and no reconstruction at all, i.e. $d = |\boldsymbol{\psi}|$ (black dashed). The symbols are placed at the median of each distribution. Each pdf was convolved with a 0.5 Mpc/h Gaussian kernel to obtain a smoother plot.

other considerably: the distance errors are varied between none and 20%, and the amount of datapoints inside the same volume between 282 and 1243. Comparing the mass cuts in the right panel, there is a hint that the reconstruction quality starts to saturate: the difference between the D30_10 and E30_10 catalogues is relatively small. Indeed, if the data quality increases, the overall reconstruction error is more dominated by the RZA error. The RZA reconstruction quality would thus not significantly increase if we add data mapping scales below the d^{RZA} scale. This scale changes locally but it is 3.5 Mpc/h per cartesian component on average. Therefore, about one constraint per $(3.5\ \mathrm{Mpc}/h)^3$ on average would provide a sufficient data density.

The blue line in the right panel uses the I30_10 mock, with 588 points like the C30_10 mock (red long dashed) but more evenly distributed, which shows a noticeable improvement in the d^{RZA} distribution. The green line in the left panel uses the three-dimensional data of C30_3D. The distribution is much closer to the "ideal RZA" than all the radial velocity mocks. We can derive from this that for RZA reconstruction, actually more accuracy is lost by having only the radial component than suggested by the rms errors.

Figure 4.10 is conceptually similar to Figure 4.3, but comparing different reconstructions. The d^{RZA} error is binned against the underlying density and the absolute velocity. The standard C30_10 mock (red) is contrasted with the I30_00 mock having a more uniform data distribution (green), the E30_10 mock having about twice the amount of data points (blue), and the C30_00 mock having no data errors (purple). "Ideal" RZA (black) and no RZA (dashed black) is included for comparison. One can again see the main result: no matter how the data quality is in detail, the RZA reconstruction greatly reduces the distance errors for most of the objects in the data compared with no Lagrangian reconstruction of their initial positions (dashed black). The left panel of Figure 4.10 highlights again the strong dependance of d^{RZA} on the underlying density. For the displacement field in underdense regions, it seems particularly important to have a more homogeneous sampling of the data, mapping this region well and with a distance error as low as possible: the C30_10 mock performs noticeably worse here than both the more homogeneous isolated halo mock I30_10 and the mock C30_00 with no errors. Conversely, in high-density

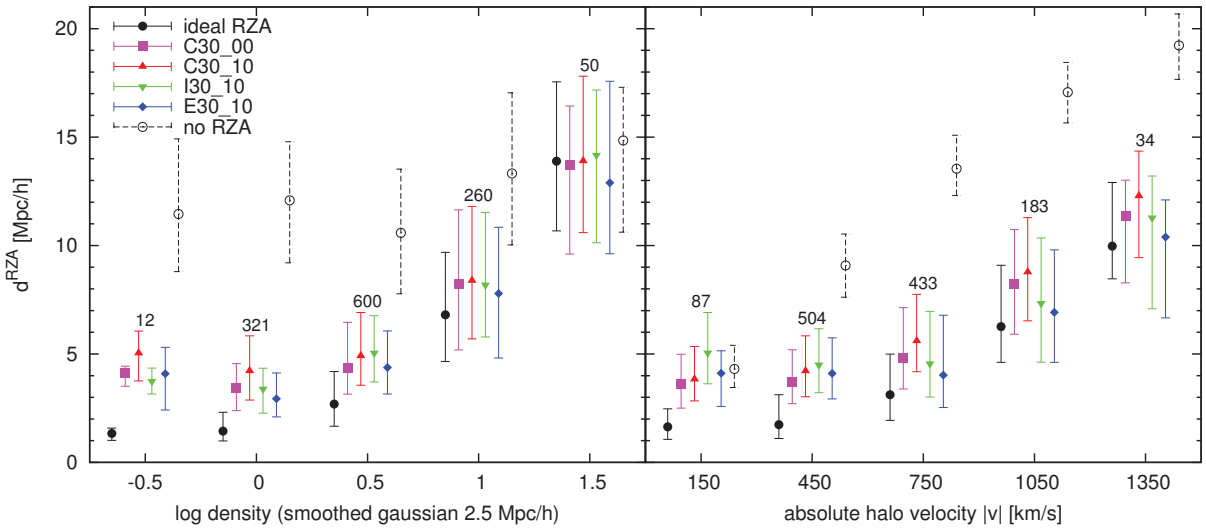

Figure 4.10: RZA displacement error d^{RZA} of haloes inside the data zone binned over the underlying density and the halo velocity for different mock reconstructions, as well as the ideal RZA (solid black) and no RZA (dashed black). The number of haloes in each bin is also given.

regions, d^{RZA} is high and errors on the individual velocities have little influence, since those velocities do not track the displacement field well due to the high non-linearity. In this case, it seems better to have more data points in general: the E30_10 mock performs somewhat better than the others. Fed with more data, the WF algorithm has more chance to filter out the noise. It is interesing that in the densest bin (around 1.5 smoothed density), the WF result from E30_10 performs even better than the theroretical "ideal" RZA. This can be understood from the properties of WF as a linear filter. The true halo velocities, even when perfectly known, are a bad tracer of the displacement field in the densest regions. But if the surrounding field is mapped sufficiently well, the WF can create a solution for the displacement field that is closer to the more linear actual displacements. The difference is quite small though. We can argue that when there is a subsequent step of Wiener filtering on the velocity data, rigorously identifying and removing substructure is less critical than it appeared in Section 4.2. More important is the insight that, since reconstruction quality from the different mocks does not differ as much as one could expect, the procedure of Wiener filtering and subsequent RZA reconstruction is restricted more by the limitation of the scheme itself because of the underlying non-linearity of the system, and to a lesser degree by the actual data quality.

4.4.5 Radial vs. three-dimensional data

It is worth to analyse in more detail the impact of having only constraints on the radial velocity component. The d^{RZA} distribution function revealed that this fundamental limitation for RZA may be more significant than suggested by just the overall reconstruction rms error. Let us consider the "viewing angle" γ of a single peculiar velocity observation,

$$\gamma = \mathrm{acos}\left(\frac{|\boldsymbol{v}\cdot\boldsymbol{r}|}{|\boldsymbol{v}|\cdot|\boldsymbol{r}|}\right) \quad , \tag{4.14}$$

which is the angle between the full three-dimensional velocity vector of an object and its observed radial component. At $\gamma = 0°$, v_r will have the complete information on an object's peculiar

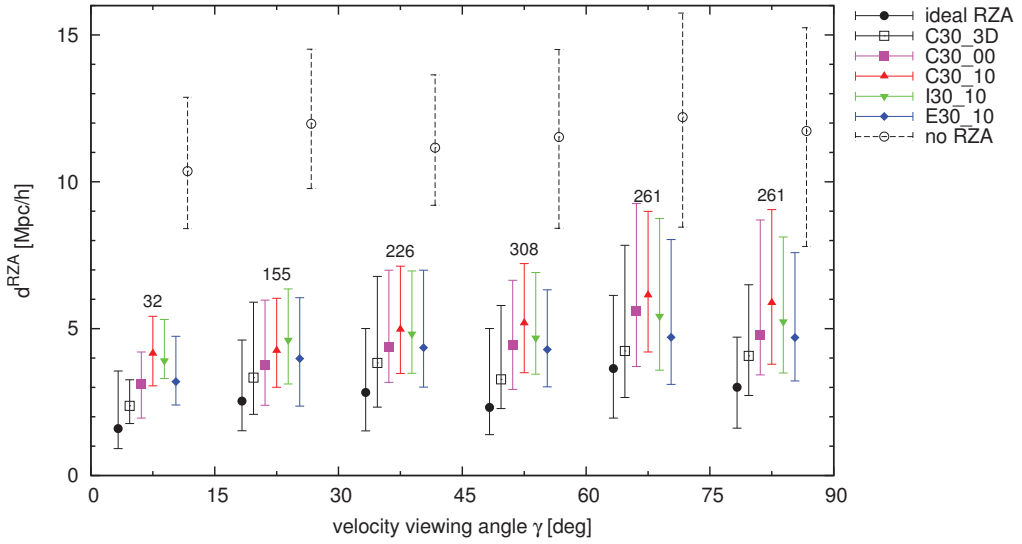

Figure 4.11: RZA displacement error d^{RZA} of haloes inside the data zone binned over the peculiar velocity viewing angle γ, i.e. the angle between the full 3D velocity of the halo $\boldsymbol{v}$ and its observed radial component. The C30_3D reconstruction utilises full three-dimensional data and thus does not depend on γ, as well as the ideal RZA and no RZA.

velocity, while at $\gamma = 90°$, the object's motion will be completely obscured. The majority of objects in the 30 Mpc/h mock volumes have, in this sense, "unfavourable" viewing angles: $\gamma > 45°$ for 67% of the haloes. We can individually compare the reconstruction quality for $\boldsymbol{\psi}^{\mathrm{WF}}$ for objects depending on their viewing angle to quantify its impact.

Figure 4.11 shows a binning of d^{RZA} over γ. Along with the models in Figure 4.10, it also includes the C30_3D mock reconstruction, which has the same datapoints as C30_00, but with all three components, and is therefore unbiased with respect to γ. In comparison with this, for the radial velocity reconstructions (coloured bars) there is a tendency of higher d^{RZA} with increasing γ. This is expected: it is more difficult for the WF to reconstruct the displacement field at a position where the data has a high γ. It is particularly instructive to compare the 30° – 45° bin with the 75° – 90° bin. In both, the unbiased ideal RZA and C30_3D have similar values, but the radial velocity reconstructions (coloured) show a higher d^{RZA} scatter in the 75° – 90° bin. This additional scatter is however only at the order of 1 – 2 Mpc/h. This means that the WF yields a reasonable reconstruction for $\boldsymbol{\psi}^{\mathrm{WF}}$ even for datapoints at unfavourable viewing angles.

We can see the net effect of radial vs. 3D data by directly comparing C30_3D to C30_00 (purple). Then, the effect of having only radial data can be decomposed into an additional local error at the most affected positions (high γ), causing some of the d^{RZA} values to be significantly higher (more skewed d^{RZA} distribution), and an error affecting the reconstruction as a whole by increasing d^{RZA} by 1 – 1.5 Mpc/h on average. These errors are remarkably small compared to how much information on the full velocity vector is obscured by the radial limitation. Therefore we can state that the WF + RZA procedure performs well on radial velocity data. Most importantly, from Figure 4.11 it is clear that the radial limitation is not the dominating error source of RZA reconstruction.

Finally, we want to discuss the supplementary E60_3D mock. This mock was constructed to contain the maximum amount of input data that can be retrieved from the AHF catalogue. It contains the full three-dimensional velocity vectors of all haloes with $\log(M/M_{\odot}) \geq 11.5$ within

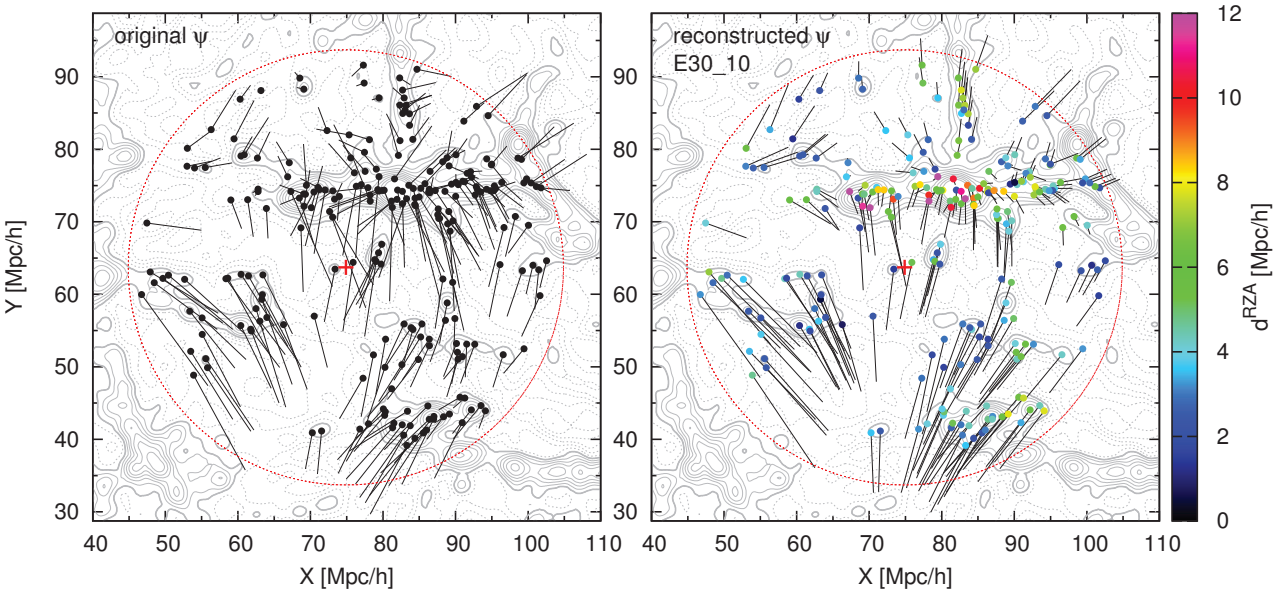

Figure 4.12: Actual BOX160 displacement field (left) vs. RZA reconstruction from radial peculiar velocities (using mock E30_10). The dots show the $z = 0$ position $\boldsymbol{r}$ of the haloes contained in the mock data; the lines show the displacement vector $\boldsymbol{\psi}$ from the initial conditions. In the reconstruction, the RZA error d^{RZA} is highlighted by colour-coding of the points. The contours show the underlying dark matter density (smoothed with a 1.5 Mpc/h Gaussian). Shown is a 15 Mpc/h thick slice containing the simulated Local Supercluster with Virgo at X=82, Y=74. The red cross marks the position of the mock observer.

60 Mpc/h of $\boldsymbol{r}^{\mathrm{MW}}$, which amounts to 22911 constraints in total. While such a mock has little physical meaning compared to realistic data, it is useful to test the performance of the WF when presented with a very high-quality sampling of the peculiar velocity field at $z = 0$.

The E60_3D yields a reconstruction that is, as expected, by far the most accurate of all mocks. The velocity field reconstruction $\boldsymbol{v}^{\mathrm{WF}}$ has a slope of 1.001, an rms error of only 66 km/s and a correlation factor of 0.984 when comparing it to the original $\boldsymbol{v}$ without smoothing. Because of the very high data quality, the filtering bias is completely removed: the slope is almost exactly 1 (using $\sigma_{\mathrm{NL}} = 158$ km/s). For the displacement field, it yields a slope of 0.998, an rms error of 3.67 Mpc/h and a correlation factor of 0.890. This error is produced exclusively by the RZA errors of the sample and the WF filtering of the velocities, since no other error sources are present, and again with no filtering bias. The computation is also exceptionally fast: the procedure of computing and inverting the 22911×22911 matrix $\langle c_i c_j \rangle$ and computing the η_i correlation vector took just 13 minutes on a consumer-grade laptop (2.4 GHz Intel Core 2 Duo) to complete. For comparison, the same procedure for 588 constraints (like in the standard C30_10 mock) takes just a few seconds. We acknowledge that the ICECORE code is not only fast in computing the correlation matrix (as we already saw in Section 3.4), but we can also observe a steady increase in reconstruction quality if the amount of data is increased. This proves that the procedure is numerically stable. During numerical tests, we successfully tested datasets with up to 50 000 constraints, and it is possible to handle more if one has enough memory to store the correlation matrix for the inversion.

4.4.6 The reconstructed displacement field

This section analyses in more detail the quality of the halo displacement reconstruction and how its error d^{RZA} varies locally. Figure 4.12 shows the original halo displacements $\boldsymbol{\psi}$ of a representative slice of the E30_10 mock (left side) alongside with the reconstructed displacements $\boldsymbol{\psi}^{\mathrm{WF}}$. The prominent feature is the "Local Supercluster" of BOX160, a very thick filament with the Virgo Cluster at its centre. The displacement field is governed by infall on this region only in its immediate vicinity, on a scale of 10 Mpc/h or so, while the flow at larger distances from it shows a more complicated pattern. This is remarkably similar to the situation in the observed Local Universe, as pointed out by Courtois et al. (2012). On the left side, the field is dominated by a large-scale flow towards the Hydra/Norma/Centaurus region, which itself lies outside of the data zone.

The reconstruction map (right side) reveals a good agreement with the actual displacement field. The large-scale features of the displacement field are successfully recovered by the WF + RZA procedure. The reconstruction also gives a reasonable estimate for the initial positions $\boldsymbol{x}_{\mathrm{init}}$ of most haloes (the non-dotted ends of the displacement lines). The RZA error d^{RZA} is highlighted by colour for each halo. The map illustrates not only the generally good agreement of $\boldsymbol{\psi}^{\mathrm{WF}}$ with $\boldsymbol{\psi}$, but also some typical situations where the reconstruction fails, resulting in a high d^{RZA}. Most striking is the already established connection of higher d^{RZA} with higher underlying density. Whereas most of the objects outside of the Local Supercluster have d^{RZA} below 3 Mpc/h (blue points), most of the objects inside have very high d^{RZA} – their assembly history can not be recovered from their observed orbits at $z = 0$.

As already established, in high-density regions the motion of objects does no longer follow the Zeldovich approximation; the actual travelled paths are curved. Even outside virialised clusters the simulation at $z = 0$ can have paths strongly deviating from the quasilinear assumption, for example haloes coming from very different directions but being very close together and with a similar $\boldsymbol{v}$ vector at $z = 0$. In all those cases, information about the initial positions x_{init} where the haloes originated are erased. The initial velocity vector, an imprint of the initial conditions, is "overwritten" with the mean flow by advection and merging processes, or with virial motion when accreted into clusters. Instead of the correct displacement the WF result will then recover the mean large-scale flow which has a smoother appearance. For this reason, smoothing of the result to remove the errors from small scales, in the way it is done for other reconstruction problems, cannot be applied here. The situation can not be resolved by the simple statement that the WF would yield a wrong result on scales below some critical scale of a few Mpc/h. If one first constructs the full $\boldsymbol{\psi}^{\mathrm{WF}}(\boldsymbol{r})$ field and then smooths it on a scale of a few Mpc/h before extracting individual reconstructed halo displacements, the result will be even more over-smoothed and the errors d^{RZA} will be even higher than before: it would be a waste of reconstructed details without adding information. It is thus best to use the $\boldsymbol{\psi}^{\mathrm{WF}}$ without any additional smoothing/filtering procedures, despite the sometimes high errors.

An interesting special case of the general curved-path problem is the situation for filaments. A good specimen is the filament in vertical (Y) direction above the Virgo Cluster in Figure 4.12. The analysis of Sousbie et al. (2008) draws a picture of the general pattern of filament evolution that consists, from the viewpoint of an individual halo, of two distinct stages. The accretion of haloes onto a filament generally happens orthogonally to the filament; once inside, the haloes change direction and move along the filament, following the 'highway' corresponding to the mean flow. This change in direction represents an example of non-linear structure formation (i.e. it deviates significantly from the straight path predicted by the Zeldovich approximation). The total displacement of these haloes from the initial conditions, displayed by the lines in Figure 4.12 (left side), is the net result of both stages, the accretion and the subsequent laminar flow.

The peculiar velocity at $z = 0$ however reflects only the latter, and the memory of the original initial position $\boldsymbol{x}_{\rm init}$ outside of the filament has been erased. The WF therefore has no chance of recovering it; the resulting reconstructed $\boldsymbol{\psi}^{\rm WF}$ reflects only the laminar flow along the filament.

This analysis demonstrates that the RZA errors introduced by the WF + RZA reconstruction procedure are not some type of statistical noise, but rather systematic errors that directly depend on the haloes' environment and its assembly history. These errors can even result in "unphysical" reconstructed displacement paths, such as objects moving across voids, which does not happen in reality. How these effects may affect the construction of constrained initial conditions and the properties of constrained simulations will be analysed in more detail in Chapter 5.

4.4.7 Filtering bias

Another systematic error source, the filtering bias, is a fundamental property of the Wiener Filter and a result of its "conservative" nature. In the case of poor data sampling, the WF tends to zero (i.e. the unconstrained mean field), always favouring the "dampening" of the field over the introduction of additional noise. This tendency can be seen in the slope β when comparing the WF mean field to the "true" underlying field (Figure 4.8). This β will always be ≤ 1 because of the signal/(signal+noise) weighting that the WF invokes. This weighting is essentially controlled by the $\sigma_{\rm NL}$ term and the errors ε_i of the constraints that are both added to the correlation matrix $\langle c_i c_j \rangle$ prior to inversion, playing the role of the "noise". The previous sections viewed the filtering bias, quantified by β, as a means to estimate the constraining power of the data. Since the RZA reconstruction of the halo displacements with WF is an intermediate step in the process of constructing constrained initial conditions, this filtering bias is an error source in itself, so one can think of compensating for this effect.

The simplest approach would be the multiplication of $\boldsymbol{\psi}^{\rm WF}$ with a fudge factor $1/\beta$ to force the slope to be 1. Visually this corresponds to tilting the distribution in Figure 4.7 until the solid line lies on the dashed line. This removes the filtering bias while keeping the scatter constant, but dramatically increases the rms error of the reconstruction (by a factor $1/\beta$). We found that in all such cases, while formally removing the bias, the increased overall error leads to a worse result for the displacement field $\boldsymbol{\psi}$. We also tried different fudge factors in the range between 1 and $1/\beta$, and different factors for the three components, leading to the same conclusion. In terms of the rms error of $\boldsymbol{\psi}^{\rm WF}$ and the distribution of $d^{\rm RZA}$, the obtained WF result is optimal without an additional fudge factor despite the filtering bias.

A less trivial optimisation can be thought of when considering the influence of the additional parameter $\sigma_{\rm NL}$ on β. Although a nonzero $\sigma_{\rm NL}$ does not necessarily lead to a filtering bias if the data quality is high enough (cf. the E60_3D mock), there is an obvious trend of increasing filtering bias, or smaller β, with higher $\sigma_{\rm NL}$ (compare Table 4.1 and Figure 4.8). This follows directly from the definition of the Wiener Filter mean field. Previously, we chose $\sigma_{\rm NL}$ for each case such that $\chi^2/{\rm dof} = 1$. This gives the maximum likelihood solution given the assumption of Gaussianity and the linear overdensity power spectrum $P(k)$. While we saw that generally Gaussianity is a reasonable assumption in case of halo displacements, it is not necessarily the case for smaller mock volumes. Further, it is by no means certain that the optimal solution for $\boldsymbol{\psi}$ is the one where $\delta^{\rm WF}$ is the most probable realisation of the given $P(k)$. We can therefore allow for $\chi^2/{\rm dof} > 1$ and in this way induce a less conservative behaviour in the WF by decreasing or even omitting $\sigma_{\rm NL}$. This would not be advisable when creating Gaussian initial conditions, but perfectly acceptable if one seeks the optimal solution for $\boldsymbol{\psi}$.

We therefore re-computed the reconstruction for some of the mocks with varying $\sigma_{\rm NL}$ values. We found that the effects on the reconstructed halo displacements $\boldsymbol{\psi}^{\rm WF}$ are small and $\sigma_{\rm NL}$ has to be decreased all the way to zero for them to become noticeable. The sparse C30_10 mock,

reconstructed using $\sigma_{\rm NL} = 0$, yields a slope of 0.73, an rms error per component of 4.11 Mpc/h, and a median $d^{\rm RZA}$ of 5.11 Mpc$h$, whereas the original reconstruction using $\sigma_{\rm NL} = 235$ km/s had a slope of 0.64, rms error per component of 4.17 Mpc/h, and a median $d^{\rm RZA}$ of 5.29 Mpc/h. With $\sigma_{\rm NL} = 0$, the filtering bias is noticeably decreased, but the average improvement of $d^{\rm RZA}$ is only minor at about 0.2 Mpc/h. For more accurate mocks like the E60_10, having had a lower $\sigma_{\rm NL}$ initially, this effect is smaller or even insignificant; but for none of the mocks $\sigma_{\rm NL} = 0$ would lead to a worse result than $\sigma_{\rm NL} > 0$ (as in the standard procedure) for RZA reconstruction of $\boldsymbol{\psi}$, so $\sigma_{\rm NL} = 0$ seems to be the optimal strategy.

Yet another approach on the filtering bias of the WF was suggested by Zaroubi (2002). This focuses on the aspect that the bias results not only from the regularisation parameter $\sigma_{\rm NL}$, but more prominently from the errors ε_i attached to each constraint c_i. These are added as a diagonal term to the correlation matrix $\langle c_i c_j \rangle$ in the same fashion (cf. equation 3.45), the difference being that ε_i varies for each constraint and thus the filtering bias of the WF will also vary locally. To overcome this feature of the WF, Zaroubi (2002) replaces it with an unbiased minimal variance (UMV) estimator. Upon closer examination, his UMV estimator is equivalent to a WF that is applied after all ε_i are set to zero and the regularisation is done by $\sigma_{\rm NL}$ alone. It can be thus directly computed with ICECORE by modifying the input constraints file accordingly. Our experiments with this approach, on all the mocks that have observational errors attached, revealed that in all cases the UMV estimator significantly increases the rms error and $d^{\rm RZA}$ distribution and leads to a significantly less accurate reconstruction of $\boldsymbol{\psi}$ than the corresponding WF reconstruction.

To summarise, the best results on RZA reconstruction are obtained with the standard Wiener Filter, no additional manipulation of $\boldsymbol{\psi}^{\rm WF}$ by means of a fudge factor, and $\sigma_{\rm NL} = 0$. Attempts to squeeze out more information from this procedure invariably enhance the scatter and degrade reconstruction quality. Since, given the mock data, there seems to be no way of further improvement at this point, one can attempt to make up for it by refining the subsequent RZA equations that determine the initial position guess $x^{\rm RZA}_{\rm init}$. In the next section such an attempt will be presented by expanding the RZA method to second-order Lagrangian perturbation theory.

4.5 Extending RZA to higher order

4.5.1 The Reverse 2LPT approximation

In the RZA reconstruction, we compute the Wiener filter mean displacement field $\boldsymbol{\psi}^{\rm WF}$ from radial peculiar velocity data and then directly use it as an estimate for the actual displacement field $\boldsymbol{\psi}$, which provides the mapping from the data positions at $z = 0$ to the initial conditions. This is done by first computing the Wiener filter mean field for the overdensity, $\delta^{\rm WF}$, as given by equation 3.27, and then to compute $\boldsymbol{\psi}^{\rm WF}$ in Fourier space by solving $-\boldsymbol{\nabla}\cdot\boldsymbol{\psi}^{\rm WF} = \delta^{\rm WF}$. Although we perform just these two steps, by first evaluating the WF operator and then performing three FFTs, this is mathematically equivalent to solving the Poisson equation for a potential ϕ,

$$\boldsymbol{\psi}^{\rm WF} = -\boldsymbol{\nabla}\phi^{\rm WF} \quad ; \quad \delta^{\rm WF} = \nabla^2\phi^{\rm WF} \quad . \tag{4.15}$$

Here, we extend this computation of the displacement field to second-order Lagrangian perturbation theory (2LPT). It is known that 2LPT performs better than first-order Lagrangian perturbation theory (= the Zeldovich approximation), if one wants to analytically estimate the evolution of the initial linear density field into the non-linear regime (Bouchet et al. 1995). This is accomplished by adding a second-order term that encodes the dynamical change of the gravitational potential. It is interesting to learn if 2LPT can, in a similar way, improve the estimation

of the initial conditions from the evolved non-linear field. As before, we are not interested to directly reconstruct the initial linear field δ_0 (as opposed to e.g. Kitaura 2012), but instead we want to reconstruct the displacement field $\boldsymbol{\psi}$ at discrete positions, which can be used to constrain the initial conditions (Chapter 5). For this, we combine the WF reconstruction from peculiar velocities with the 2LPT technique presented in Kitaura & Angulo (2012); Kitaura et al. (2012).

In 2LPT, the displacement field $\boldsymbol{\psi}$ can be approximated as (Buchert et al. 1994; Bouchet et al. 1995):

$$\boldsymbol{\psi} = -D_1 \boldsymbol{\nabla}\phi^{(1)} + D_2 \boldsymbol{\nabla}\phi^{(2)} \quad , \tag{4.16}$$

where $D_1 \equiv D_+$ and $D_2 \approx -\frac{3}{7}\Omega_{\rm m}^{-1/143} D_+^2$ (Buchert & Ehlers 1993; Bouchet et al. 1995). The potentials $\phi^{(1)}$ and $\phi^{(2)}$ are obtained by solving a pair of Poisson equations,

$$\delta^{(1)} = \nabla^2 \phi^{(1)} \quad ; \quad \delta^{(2)} = \nabla^2 \phi^{(2)} \quad . \tag{4.17}$$

Here, $\delta^{(1)}$ represents the linear component of the overdensity field[20], and $\delta^{(2)}$ is the second-order non-linear term. We are not concerned here about how to compute the total non-linear density field δ from these components. It is however important that if we have $\delta^{(1)}$, then this fully determines $\delta^{(2)}$ through the following quadratic expression (Kitaura & Angulo 2012; Kitaura et al. 2012):

$$\delta^{(2)} = \sum_{\alpha} \sum_{\beta<\alpha} \left(\frac{\partial^2 \phi^{(1)}}{\partial q_\alpha^2} \frac{\partial^2 \phi^{(1)}}{\partial q_\beta^2} - \left[\frac{\partial^2 \phi^{(1)}}{\partial q_\alpha \partial q_\beta} \right]^2 \right) \quad ; \quad \alpha, \beta \in \{x, y, z\} \quad , \tag{4.18}$$

which can be solved numerically at $z = 0$ if $\delta^{(1)}$ and therefore $\phi^{(1)}$ are known. With equation 4.17, we can then directly obtain an estimate of the displacement field $\boldsymbol{\psi}$ (4.16).

At this point we have to make an assumption that would allow us to evaluate the 2LPT equations. We saw that the Wiener filter mean field $\delta^{\rm WF}$, that we computed from the data (i.e. the mock catalogues of halo radial peculiar velocities), is not a very good estimate of the initial linear density field δ_0 at $z_{\rm init}$ (this is the reason why we introduced Lagrangian reconstruction in the first place). However, we can assume that $\delta^{\rm WF}$ is actually a good estimate of the linear component of the density field at $z = 0$, i.e. of $\delta^{(1)}$. This makes sense, since the WF is an estimator based on linear theory, and designed to produce a linear Gaussian random field from the data by filtering out both the noise and the non-linear term present in the field at $z = 0$. We therefore plug in the Wiener filter mean field obtained from the data into the 2LPT scheme by assuming $\delta^{(1)} \equiv \delta^{\rm WF}$ and compute the 2LPT estimate of the displacement field $\boldsymbol{\psi}$.

One problem remains: as we saw in Chapter 3, $\delta^{\rm WF}$ is not a statistically homogeneous field and will decay in regions not sampled by the data. On the other hand, 2LPT is a non-local scheme and assumes that we have an estimate of the whole field. We solve the problem by instead computing a large number of constrained realisations $\delta^{\rm CR}$ from the data on the whole computational volume, which yields a statistically homogeneous field. For each constrained realisation, we then set $\delta^{(1)} \equiv \delta^{\rm CR}$ and compute the 2LPT estimate of $\boldsymbol{\psi}$ (equation 4.16) for the whole computational box. We then average over all displacement fields obtained in this manner, to yield the estimate of the underlying displacement field of the input data. In the following, we will call this field $\boldsymbol{\psi}^{\rm R2LPT}$. We now construct the "reverse 2LPT approximation", or R2LPT, by

[20] The linear component $\delta^{(1)}$ of the overdensity field δ is the Gaussian component of δ and therefore a field that evolves with time alongside δ. It should not be confused with the linear overdensity field δ_0, which describes the full overdensity field in the linear regime $\boldsymbol{x} \to \boldsymbol{q}$.

assuming that the displacement field and initial position of each dark matter halo is given by the value of $\boldsymbol{\psi}^{\mathrm{R2LPT}}$ at the position of the halo (i.e. datapoint),

$$\boldsymbol{x}_{\mathrm{init}}^{\mathrm{R2LPT}} = \boldsymbol{r} - \boldsymbol{\psi}^{\mathrm{R2LPT}} \quad . \tag{4.19}$$

4.5.2 Results and discussion

We ran the R2LPT reconstruction for two mocks, the sparse C30_10 and the less sparse E60_10, which also covers a larger data volume, and computed the displacement fields $\boldsymbol{\psi}^{\mathrm{R2LPT}}$ for both. This was performed with the numerical code by Kitaura & Angulo (2012); Kitaura et al. (2012). Figure 4.13 shows the comparison between the original and the reconstructed displacement fields for the 1243 halo positions within 30 Mpc/h for both mocks, computed with the methods RZA with a non-zero σ_{NL}, RZA with $\sigma_{\mathrm{NL}} = 0$ to reduce the filtering bias, and R2LPT. The R2LPT creates a less biased estimate of the actual $\boldsymbol{\psi}$. While for the C30_10, there is still some bias comparable to the filtering bias of the WF, the R2LPT estimation from the E60_10 mock is unbiased with a slope very close to 1. We can explain the bias in R2LPT for the C30_10 mock with the random modes: in each 2LPT run, we used δ^{CR} as input, which combines the WF mean field with a random residual. Since the mock is rather sparse, even inside the 30 Mpc/h the field has a significant contribution to $\boldsymbol{\psi}$ from the random modes which will be averaged away when we compute $\boldsymbol{\psi}^{\mathrm{R2LPT}}$. This effect is slightly smaller than the WF filtering bias. For the denser and larger E60_10 mock, within the inner 30 Mpc/h region the random modes do not contribute much to the displacement field, and since R2LPT does not suffer from a filtering bias, we obtain an unbiased estimate of the actual displacement field.

On the other hand, the R2LPT estimate of the displacement field shows a somewhat larger scatter around the correct displacement field, resulting in a larger rms error compared to the RZA method. As a result, the average error on the reconstructed initial positions d^{R2LPT} (defined in the same way as d^{RZA}) is larger as well. Figure 4.14 shows d^{RZA} and d^{R2LPT} for the different reconstruction methods. The effect of the larger scatter is especially severe for the R2LPT reconstruction from the better E60_10 mock. However, in our interpretation of this result, this is not a failure of the 2LPT theory, but lies in the fact that we computed δ^{CR} using the Wiener Filter on peculiar velocities. Surprisingly, we found that the mean overdensity field produced by the WF from radial peculiar velocity data is not strictly Gaussian distributed. Despite the fact that the WF employs a Gaussian prior model, and the presence of the additional regularisation parameter σ_{NL} which enforces $\chi^2/\mathrm{dof} = 1$, the systematic non-linearities of the v_r^{pec} at $z = 0$ corrupt the Gaussianity of the result. If the input data is very sparse, such as for the C30_10 mock, the WF can compensate, but if the data density is higher, as for the E60_10 mock, the intrinsic non-linearities of the data become imprinted on the resulting WF, even though it explicitly uses a Gaussian prior model. We can see this effect in Figure 4.15, which shows the probability distribution function of δ^{CR} realisations from both mocks (using the same random seed), compared to that of the BOX160 initial conditions. The CR from the C30_10 mock is fairly Gaussian, as are the BOX160 initial conditions, because the original δ^{CR} of BOX160 was likewise computed from very sparse data (Gottlöber et al. 2010). In contrast, the distribution function of δ^{CR} from the E60_10 mock shows a significant skewness and therefore a deviation from the assumed Gaussianity. In the 2LPT formalism, the non-linear term scales with the square of the growth factor D_+ (equation 4.16). If we now use such a skewed δ^{CR} for R2LPT reconstruction, the non-linear term will be significantly amplified, resulting in a larger amount of noise.

Because of the larger rms errors and the noise enhancement, we do not use the R2LPT reconstruction in the remainder of this work. However, it may be a very promising path. We

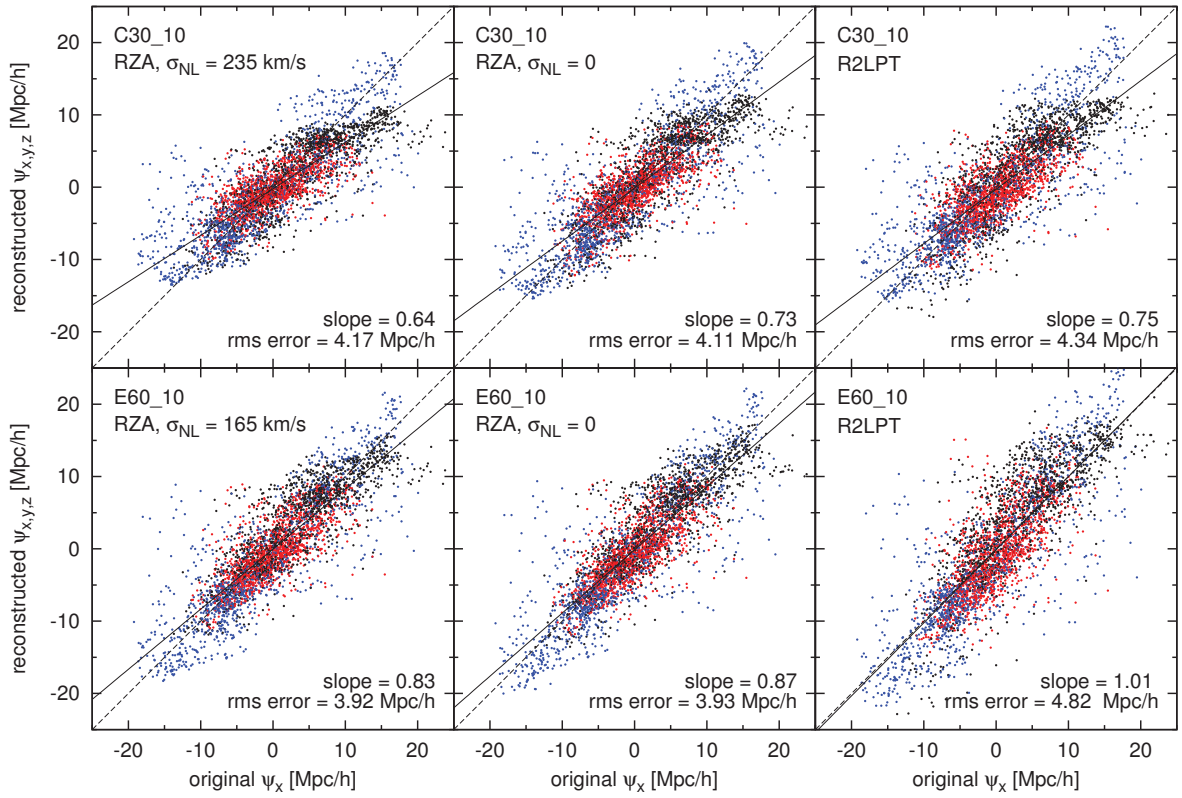

Figure 4.13: Scatter plots of WF reconstructed vs. actual displacement $\boldsymbol{\psi}$ for all haloes within 30 Mpc$/h$ of the observer, using the C30_10 (top) and E60_10 (bottom) mocks. Left panels: RZA reconstruction with $\sigma_{\rm NL}$ such that $\chi^2/{\rm dof} = 1$. Middle panels: RZA reconstruction with $\sigma_{\rm NL} = 0$. Right panels: R2LPT reconstruction. Compared to Figure 4.7, all three cartesian components of $\boldsymbol{\psi}$ have been merged into one plot, with ψ_x in red, ψ_y in black, and ψ_z in blue.

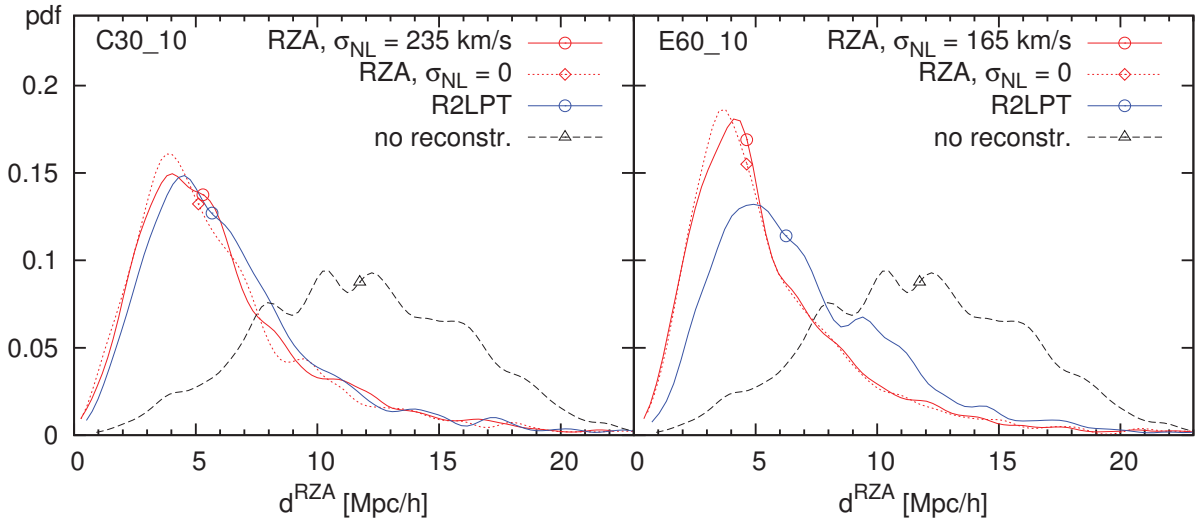

Figure 4.14: Probability distribution function (pdf) of the RZA displacement error $d^{\rm RZA}$ for different mock reconstructions, similar to Figure 4.9, but only for the C30_10 (left) and E60_10 (right) mocks, for RZA reconstruction (red lines), R2LPT reconstruction (blue line) and no reconstruction (black dashed line). The dot marks the median of each distribution.

expect that 2LPT could lead to better results than RZA, if we would manage to obtain a better estimate of the Gaussian component $\delta^{(1)}$ from the peculiar velocity data, i.e. if we could reconstruct an overdensity field from the data that would be more strictly Gaussian distributed than the standard WF. This could be done either by a better linearisation of the input data, such as compensating for the non-linear enhancement of the peculiar velocity amplitudes in overdense regions, or by a reconstruction of the underlying field that would be self-consistent with 2LPT and thus replace the first-order approximation WF. A better Gaussianisation of the field reconstructed from the peculiar velocity data could significantly improve the results presented here. Unfortunately, Gaussianisation methods so far have concentrated on the non-linear density field at $z = 0$ as an input (Weinberg 1992; Neyrinck et al. 2009, 2011; Joachimi et al. 2011; Yu et al. 2011, 2012; Zhang et al. 2011; Kitaura & Angulo 2012; Kitaura et al. 2012; Kitaura 2012) and not the non-linear peculiar velocity field at $z = 0$. This will be the subject of future work. For now, the RZA reconstruction does a fairly good job at recovering $\boldsymbol{\psi}$ from the data. While the direct application of the WF/CR algorithm to the data, δ^{WF}, is not a very good way to recover the initial density field δ_0, the corresponding $\boldsymbol{\psi}^{\mathrm{WF}}$ seems to be a good way to recover the displacement field $\boldsymbol{\psi}$. From this displacement field we obtain a fair mapping from Eulerian coordinates $\boldsymbol{x}$ at $z = 0$ to the Lagrangian coordinates $\boldsymbol{q}$, or the initial positions $\boldsymbol{x}_{\mathrm{init}}$ which we consider equivalent. This is the reason why the RZA reconstruction works reasonably well. We will use this method in the next chapter to generate cosmological initial conditions from the data that can be used for cosmological simulations.

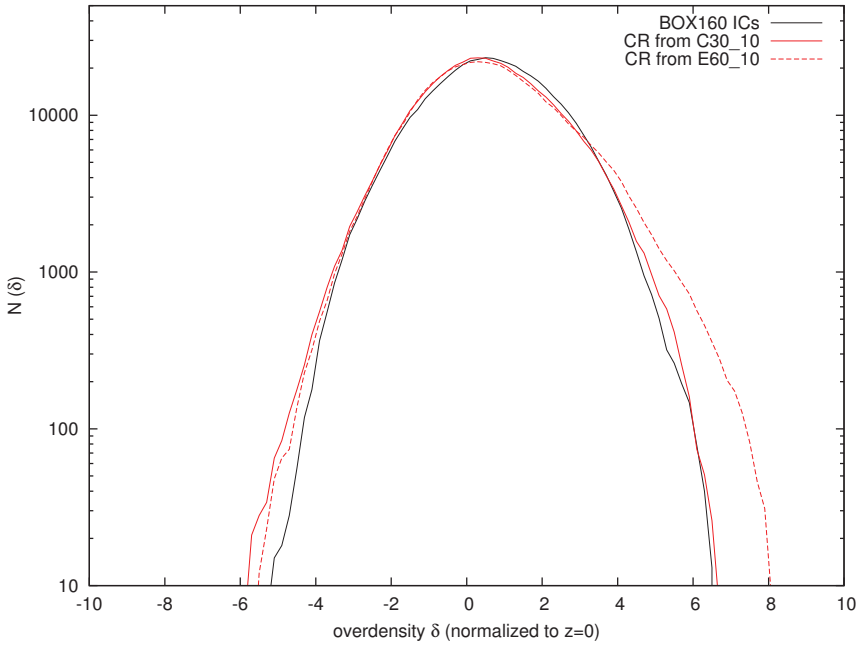

Figure 4.15: Distribution of the overdensity field δ within 30 Mpc$/h$ of the mock observer for the Wiener Filter mean field δ^{WF} from the C30_10 mock (solid red) and E60_10 mock (dashed red), compared to the distribution of the original BOX160 initial conditions (black).

Chapter 5

Constrained Simulations

In this chapter, we test the constrained simulations technique by applying it to the BOX160 simulation, which serves as the reference Universe that shall be re-created based on the mock catalogues discussed in Chapter 4. We first reconstruct the initial conditions of BOX160 with the constrained realisations technique from this mock data. We modify the method of placing constraints on the initial conditions by incorporating the previously developed RZA reconstruction of the initial displacement field, and discuss different variations of this method. We then run several realisations of the reconstructed initial conditions forward until $z = 0$ and compare the evolved constrained resimulations with the original evolved BOX160. The addition of RZA significantly improves both the reconstruction of the initial conditions and the accuracy of the obtained constrained resimulations and removes the systematic shift of resimulated structures from their actual positions due to the displacement. With a mock catalogue of sufficient data volume, the RZA method gives a reasonable reconstruction of the initial conditions down to length scales of ≈ 4 Mpc/h, and is able to reproduce the distribution of dark matter haloes in the original BOX160 simulation with an accuracy of a few Mpc/h on their position and a factor of 2 in mass. Individual haloes are robustly recovered at $z = 0$ down to a mass scale of at least $\approx 10^{14} M_{\odot}/h$. This is a significant improvement over the previously used method. The results suggest that by applying RZA reconstruction to observational data of galaxy peculiar velocities, it is possible to obtain more precise constrained simulations of the observed Local Universe.

5.1 Generating initial conditions

In this chapter, we test the technique of generating constrained initial conditions and running constrained simulations with a reconstruction from peculiar velocity catalogues. For the input data, we use the mock catalogues of BOX160 that have been described in the previous chapter. The aim of this test is to understand to what extent the "test Universe" provided by the BOX160 simulation can be reproduced in a constrained resimulation. The obvious advantage of working with a test simulation is that we can immediately check the results, both the reconstructed initial conditions and the finished resimulations at $z = 0$, against the "true" result. This is not possible with observational data, since the complete total matter density and velocity fields are not accessible observationally. With this test, we can quantify to what extent constrained realisations are able to reproduce the distribution, formation and evolution of the large-scale structure underlying the input data. This in turn allows us to make predictions about the

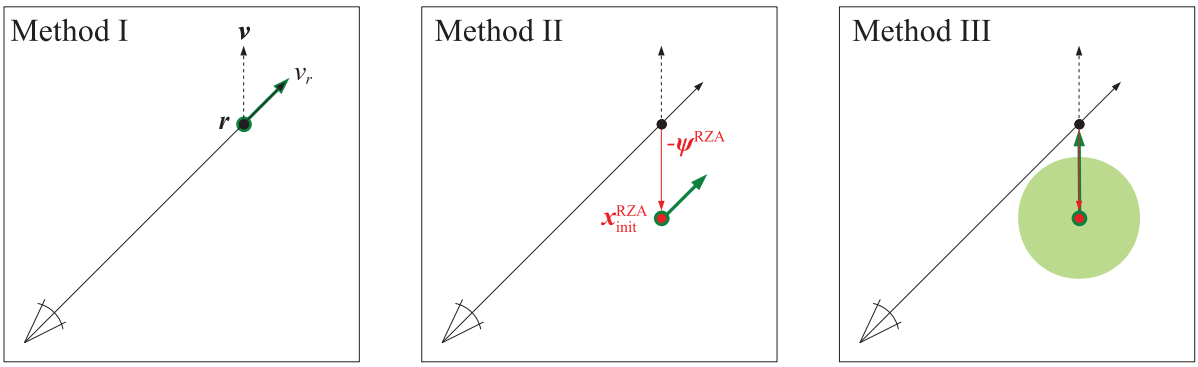

Figure 5.1: Three different methods of placing a peculiar velocity constraint (green arrow) generated from a radial peculiar velocity datapoint (solid black arrow). Method I: The constraint is made equal to the datapoint itself at its $z = 0$ position (black dot). Method II: The datapoint is shifted back to its RZA reconstructed initial position, $\boldsymbol{x}^{\mathrm{RZA}}_{\mathrm{init}} = \boldsymbol{r} - \boldsymbol{\psi}^{\mathrm{RZA}}$ (green dot); the direction and value of the constrained velocity component is not changed. Method III: The reconstructed three-dimensional displacement $\boldsymbol{\psi}^{\mathrm{RZA}}$ at the reconstructed initial position $\boldsymbol{x}^{\mathrm{RZA}}_{\mathrm{init}}$ is used as a constraint with a smoothing radius corresponding to the object's Lagrangian volume (light green sphere).

accuracy of constrained simulations of the Local Universe based on observed peculiar velocity data.

The main focus is to incorporate the RZA reconstruction technique that was described in the previous chapter in order to increase the quality of reconstructed initial conditions and the resulting constrained simulations. In this context, we want to give a quantitative estimate about how well constrained realisations resemble the actual solution, i.e. how well the peculiar velocity field is recovered and whether we can find objects in the resimulations at $z = 0$ that resemble their original counterparts in the BOX160, and down to what distance and mass scales this is reliable. This will depend on the method one chooses to construct the initial conditions and the quality of the input data. We test the latter by using different mock catalogues for the reconstruction.

In the first part of this chapter, we concentrate on generating run-capable initial conditions from a reconstruction on the input data that can be used for running constrained simulations. An analysis of the evolved resimulations follows in Section 5.2.

5.1.1 Placing constraints

The preferred method to place constraints on the initial conditions is to constrain the initial displacement field $\boldsymbol{\psi}$ (or equivalently the initial velocity field $\boldsymbol{u}$, which at redshift z_{init} equals $\dot{a} f \boldsymbol{\psi}$). From this field, both initial positions and initial velocities of the N-body particles for a cosmological simulation are determined (see Section 3.1.2). The displacement field shows much more correlation on the large-scale modes of the computational box, and thus will provide much more effective constraints than constraining other linear functionals of the overdensity. If one has a catalogue of galaxy peculiar velocities and their positions in real-space, possibly also grouped to reduce the non-linearity of the data, the basic idea is to place one constraint on the initial displacement field for each data point in the (mock) galaxy catalogue. With these constraints, a constrained realisation of a linear Gaussian random field, i.e. the cosmological initial conditions, is obtained by applying the WF/CR operator (3.25).

Figure 5.1 visualises the three different methods of placing such constraints that will be used in the remainder of this chapter. "Method I" refers to the "classical" CLUES method described in Klypin et al. (2003) and was used for all constrained simulations in the CLUES project up to now (an overview is given in Gottlöber et al. 2010). Here, the observed velocity field at $z = 0$ is assumed to be approximately equal to the initial velocity fiel. The radial velocity datapoints are directly taken as constraints for the initial conditions, which encompasses their positions, velocity amplitudes, and errors. The regularisation parameter σ_{NL} (3.45) is then used to compensate for the non-linearities of this data and create a result with the desired linear Gaussian statistics. The most apparent flaw of this method is that it does not take the cosmic displacement into account. Therefore, the results can be trusted only on scales above that of the displacement from the initial conditions at $z = 0$, which varies locally and is of the order of 10 Mpc$/h$; we already discussed the implications of this method in Section 3.3.3.

To improve on that method, we want to make use of the RZA reconstruction discussed in the previous chapter, since it allows for an estimate of the initial position and displacement of the Lagrangian regions corresponding to the dark matter haloes hosting observed galaxies. The (reverse) Zeldovich approximation assumes that at both initial time z_{init} and final time $z = 0$, the velocity and displacement fields are equal to each other (except for the appropriate scaling and unit conversion factor, which is known) and that their direction stays constant in time. Therefore, in principle we are free to choose either the present-day peculiar velocity v_r^{pec} or the reconstructed displacement $\boldsymbol{\psi}^{\mathrm{RZA}}$ of a halo as an estimate of its initial displacement.

Our first approach for such a scheme (called "Method II" here) was developed simultaneously with the RZA reconstruction and continues to make use of the present-day peculiar velocity as an estimate of the initial displacement. This approach consists of displacing each constraint in Method I from the observed position $\boldsymbol{r}$ at $z = 0$ to its estimated initial position $\boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}} = \boldsymbol{r} - \boldsymbol{\psi}^{\mathrm{RZA}}$. This "back in time" displacement is depicted in the centre panel Figure 5.1 by the red arrow. The constraint (green arrow) is then placed at $\boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}}$ instead of $\boldsymbol{r}$. All other properties that define the constraint are left unchanged: we still constrain the velocity component along the same direction $\hat{\boldsymbol{e}}_\mu$, and with the same error ε_i attached to its value c_i. Note in Figure 5.1 that the direction of this constrained component is not anymore the radial direction with respect to the observer, because the position has changed. Like in Method I, we use the σ_{NL} parameter to compensate for the non-linearity of this data to create a constrained Gaussian random field. In doing so, we observed that the value of σ_{NL} that is needed to achieve $\chi^2/\mathrm{dof} = 1$ (3.43) is always lower for Method II than for Method I, which means that the observed radial velocities are in better agreement with a linear Gaussian random field when they are displaced back in time to their reconstructed initial positions.

As an alternative approach we implement also the method described in Lavaux (2010), which we here refer to as "Method III". This method uses the reconstructed initial displacement vector (i.e. all three components), which in our case is $\boldsymbol{\psi}^{\mathrm{RZA}}$ at the reconstructed initial position $\boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}}$ as constraints. Additionally, following Lavaux (2010), the method also takes into account that the Lagrangian region around $\boldsymbol{x}_{\mathrm{init}}$ that corresponds to the protohalo of an observed object has a non-negligible volume (cf. Figure 4.1). So, instead of constraining the displacement at the point x_{init}, the method constraints the mean displacement in some volume around it, which is assumed to be spherical because we do not have any information about the possible protohalo shape. This is achieved by attaching a Gaussian smoothing radius R_{G} to each constraint,

$$R_{\mathrm{G}} = \frac{1}{\sqrt{5}} R_{\mathrm{L}} \quad , \tag{5.1}$$

Method	constraint type	position $\boldsymbol{x}$	value c_i	error ε_i	direction vector $\hat{\boldsymbol{e}}_\mu$	smoothing radius R_G
Method I	ψ_μ	$\boldsymbol{r}$	$v_r^\mathrm{pec}/\dot{a}f$	$\Delta v_r^\mathrm{pec}/\dot{a}f$	$\dfrac{\boldsymbol{r}-\boldsymbol{r}_0}{\lvert\boldsymbol{r}-\boldsymbol{r}_0\rvert}$	0
Method II	ψ_μ	$\boldsymbol{x}_\mathrm{init}^\mathrm{RZA}$	$v_r^\mathrm{pec}/\dot{a}f$	$\Delta v_r^\mathrm{pec}/\dot{a}f$	$\dfrac{\boldsymbol{r}-\boldsymbol{r}_0}{\lvert\boldsymbol{r}-\boldsymbol{r}_0\rvert}$	0
Method III	ψ_x, ψ_y, ψ_z	$\boldsymbol{x}_\mathrm{init}^\mathrm{RZA}$	$\boldsymbol{\psi}^\mathrm{RZA}$	0	$\hat{\boldsymbol{e}}_x, \hat{\boldsymbol{e}}_y, \hat{\boldsymbol{e}}_z$	$\frac{1}{\sqrt{5}}R_\mathrm{L}$

Table 5.1: Constraint types, positions, values, errors, directions, and smoothing radii that define each constraint for the three constraint placement methods. For methods I and II, we constrain only one component ψ_α of the initial displacement (the one known from the radial peculiar velocity data), while in Method III we constrain all three components of it (known from the RZA reconstructed displacement vector), i.e. we have three constraints per datapoint. The given quantities correspond to the quantities listed in 3.1 that completely define each constraint.

where R_L is the Lagrangian radius of the mass M of the constraint's underlying dark matter halo, i.e. the radius of a sphere containing the mass M at mean cosmic density $\bar{\rho}$:

$$\frac{4}{3}\pi\bar{\rho}\,(R_\mathrm{L})^3 = M \quad . \tag{5.2}$$

In other words, in the linear regime of the initial conditions, when density is nearly uniform, a protohalo with mass M will enclose the volume of a sphere with radius R_L. This volume will have an average initial displacement that is now constrained. The reason for the $1/\sqrt{5}$ factor in equation 5.1 is that a Gaussian kernel with radius R_G contains the same volume as a spherical top-hat kernel with radius R_L. The only difference between the constraint-placing technique of Lavaux (2010) and Method III is now that we use initial positions and displacements that were reconstructed using RZA from peculiar velocities, while Lavaux (2010) uses initial positions and displacements that were reconstructed from a galaxy redshift survey using the MAK reconstruction (as discussed in Section 4.1.2). For the masses M that are needed to determine R_G for each constraint from the mock catalogue, we use the known virial halo masses of BOX160 at $z = 0$ computed by the AHF halo finder. In the case of observational data, workable estimations of the objects' total masses could be obtained as well, for example from the maximum circular velocities of galaxies (Binney & Tremaine 2008) and X-ray observations of clusters (Reiprich & Böhringer 2002). However, as we will see later in this study, such constraints on the objects' masses are not necessary to obtain an accurate reconstruction of the ICs.

Table 5.1 summarises the quantities that define each constraint for the three methods. A detail of Method III is that, since each constrained position has a nonzero smoothing radius R_G, we have to take care that there is enough space between the constrained positions, so that their constrained volumes do not overlap. This case does not occur in Lavaux (2010) because of the way how MAK reconstruction is designed, but can occur here if we use the $\boldsymbol{x}_\mathrm{init}^\mathrm{RZA}$ for the constrained positions. This can be achieved by performing a pair-wise check of the constrained positions, and if two are found closer than $f_\mathrm{G}\cdot(R_{\mathrm{G},1}+R_{\mathrm{G},2})$ to each other (with a free parameter f_G) the one with the smaller R_G is removed.

5.1.2 Reconstructed initial conditions

For the following study, we selected two of the mock catalogues discussed in the previous chapter, C30_10 and E60_10. The C30_10 has 588 radial peculiar velocities inside a sphere of radius 30 Mpc/h around the mock observer; the E60_10 has 7637 radial peculiar velocities inside a bigger sphere with radius 60 Mpc/h; both use rms distance errors of 10%, which is comparable to observational distance catalogues. The data quality of C30_10 resembles the grouped galaxy distance catalogue of Tully et al. (2008) but is more sparse. The E60_10 is closer in its properties to the upcoming Cosmicflows-2 catalogue, covering a significantly larger observational volume and being complete down to smaller dark matter halo masses (in this case, $10^{11.5}\ M_\odot/h$). This way we can test how the improved data will influence the reconstruction quality.

We generated constrained initial conditions from these mock catalogues with the three methods I, II, and III with the ICeCoRe code. For the methods II and III we used the values of the reconstructed displacements $\boldsymbol{\psi}^{\mathrm{RZA}}$ and initial positions $\boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}}$ that were obtained for the mock catalogues in the previous chapter by RZA reconstruction. For each mock-method combination, we created six different realisations with different seeds for the random component, abbreviated here as A through F. We construct these initial conditions on a grid of 256^3 cells with boxsize $L = 160$ Mpc/h, just like the "source simulation" BOX160 (as discussed in the previous chapter, we resimulated BOX160 with 256^3 resolution for this study). In the end, we have 36 sets of constrained initial conditions. The naming convention followed here is to add the method and seed numbers to the end of the mock name, so that for example C30_10_II_A refers to the first out of six realisations that were constructed with constraints from the C30_10 mock using Method II. We continue to use the analytic correlator with a k_{min} cut at the fundamental frequency of BOX160 at $k_L = 2\pi/L$ with $L = 160$ Mpc/h, like for the RZA reconstruction in the previous chapter, so that the total variance and correlation functions reflect the finite-volume effect. Since the constraints are confined to a relatively small spherical volume well within the box, the influence of periodic boundary conditions and the anisotropic geometry is small. We tested also the grid correlator and found no significant differences in the reconstruction quality for this specific setup.

Figure 5.2 shows a cell-to-cell comparison between the reconstructed and the original initial conditions for the C30_10 realisations with seed A for the different methods. We found that this overall error does not change much across the differently seeded realisations. All velocity and density fields were smoothed with a Gaussian kernel of 5 Mpc/h radius.

All methods lead to reconstructed initial conditions that show a clear correlation with the original initial conditions. At the 5 Mpc/h smoothing scale, the reconstruction quality of Method I with the C30_10 mock is relatively poor, and the rms error on the velocity components of 166 km/s is about 2/3 of the total standard deviation of the actual initial velocity field (about 250 km/s per component). The RZA reconstruction used for Method II clearly improves the reconstruction quality, lowering the rms error to about 1/2 of the standard deviation. For C30_10, the best reconstruction is obtained with Method III, which has an rms error of only 88 km/s on the velocity field. If we consider the density field, there is a similar increase in reconstruction quality by adding RZA. In the lower right panel of Figure 5.2 we see that Method III seems to introduce a systematic error on the density field. It has less variance than the original field, which flattens the shape of the distribution. This will be discussed further in Section 5.1.4.

If we increase the number and extension of constraints by using the E60_10 mock, the results change (Figure 5.3). For Method I without RZA, the increased number of constraints leads only to a very small decrease in the velocity field error from 166 km/s to 154 km/s. The rms error of the density field even increases slightly. On the other hand, Methods II and III that use RZA benefit a lot from the increased number of constraints, and the errors on the velocity field are lowered

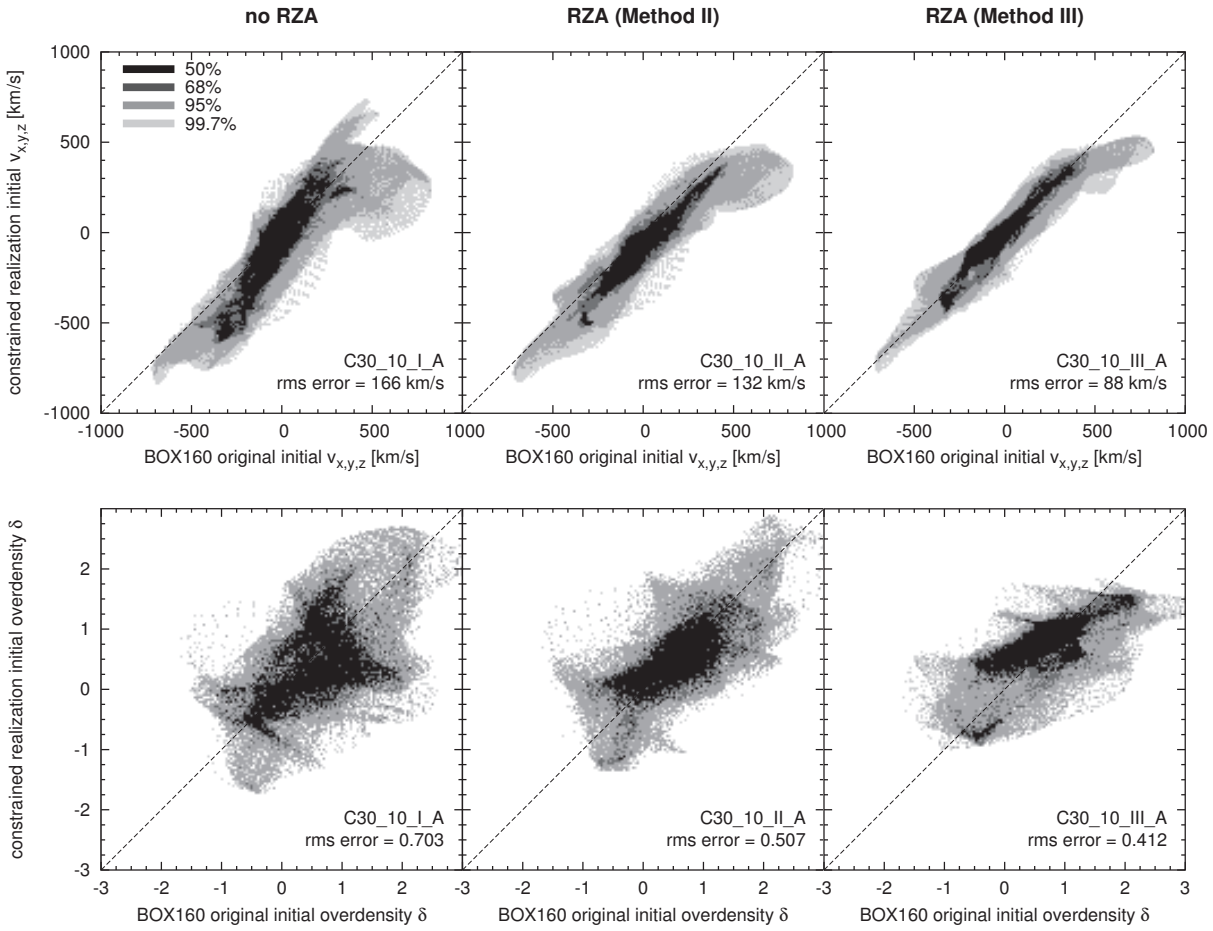

Figure 5.2: Cell-to-cell comparison between the reconstructed and original initial conditions for constrained realisations from the C30_10 mock using different constraint placement methods. Top row: velocity field (all three components were concatenated); Bottom row: density field. All fields were smoothed with a 5 Mpc/h Gaussian. Only the cells within the inner spherical region of 30 Mpc/h radius around the observer were considered; for the C30_10 mock, this is equivalent to the volume containing constraints.

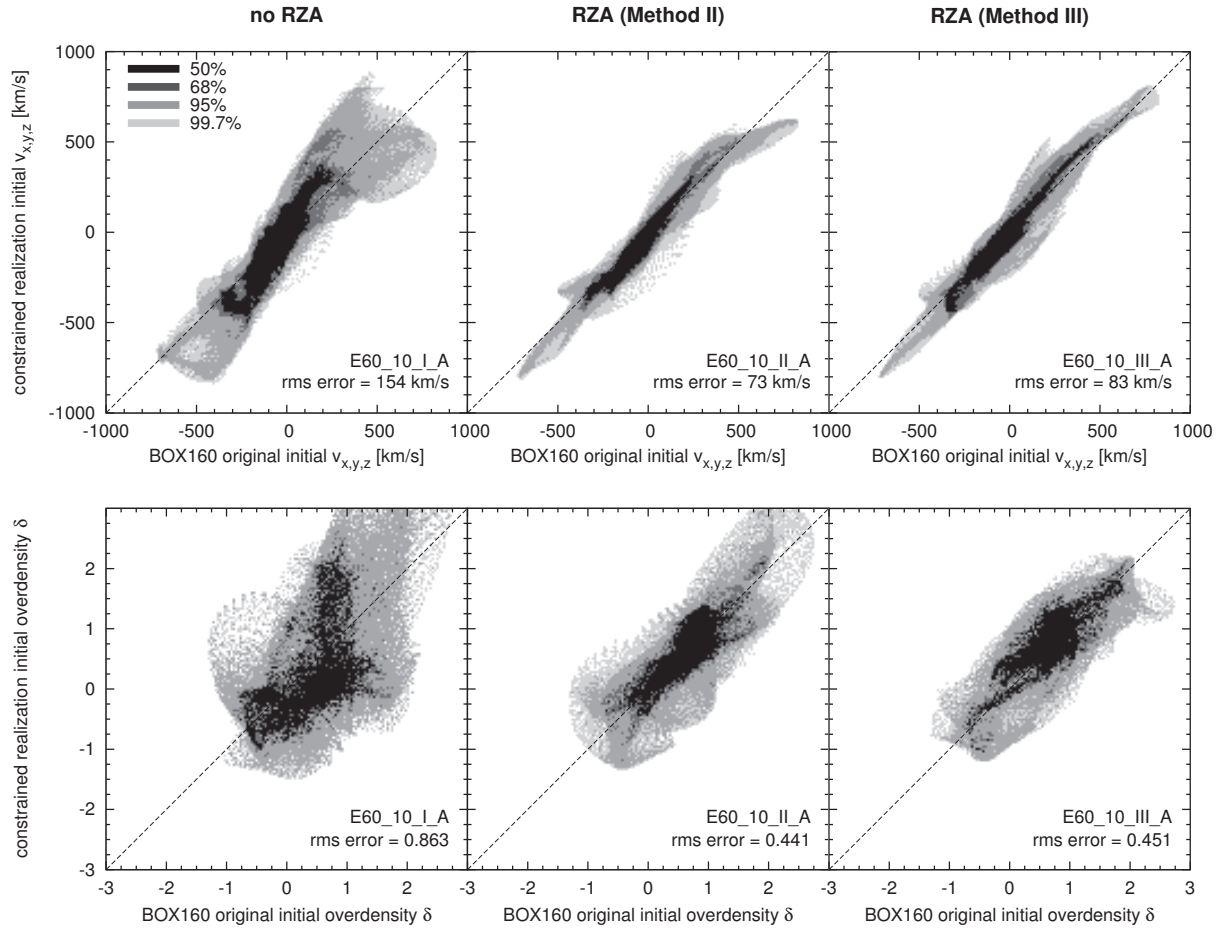

Figure 5.3: As Figure 5.2, but for constrained realisations of the initial conditions from the higher-quality E60_10 mock.

to 73 km/s and 83 km/s, respectively, which is only 1/3 of the total variance. It seems that the systematic errors introduced in Method I by neglecting the cosmic displacement are limiting the reconstruction quality, which is not the case if we include the Lagrangian reconstruction of the RZA. It is interesting that now, Method II gives a slightly better result than Method III. This is a hint that in Method III there are also systematic errors that limit the reconstruction quality (see Section 5.1.4). From this analysis of the overall error of the reconstruction we can conclude that, in the case of sufficiently good constraints, Method II seems to give the optimal result. In the following sections, this will be investigated further.

5.1.3 Constrained length scales

A crucial question in the context of reconstructing cosmological initial conditions is: down to what scales does the WF/CR reconstruction give reasonable results? In the literature describing the CLUES technique (that is, Method I), different numbers are given that usually lie between 5 and 10 Mpc/h, in the sense of a Gaussian smoothing scale (Klypin et al. 2003; Gottlöber et al. 2010). However, what was analysed there was the scatter of the obtained constrained realisations against the mean field of the same constraints. This essentially probes down to what scales the field is constrained by the data, but not down to what scales this result also matches the original initial conditions that we aim to reconstruct. The latter is precisely what we are interested in here, since the accuracy of the initial conditions determines how well the Universe can be reproduced in the constrained simulation. In our case, we know the original solution for the initial conditions that we want to reconstruct: these are the initial density and velocity fields of the BOX160 simulation. We can therefore directly compare the different constrained realisations of initial conditions to these "true" initial conditions. In terms of the general reconstruction error, this was already done in the previous section; now we want to obtain an estimate of the reconstruction quality on the initial density field as a function of length scale.

One could do this comparison in Fourier space by computing the Fourier-space cross-correlation coefficient of the original linear overdensity field $\delta_0^{\mathrm{BOX160}}$ and the reconstructed linear overdensity field δ_0^{CR} (analogous to equation 2.85), or to compare the power spectra of the original field, reconstructed field, and the difference between the two, which we here call the "residual field" $\delta_0^{\mathrm{CR}} - \delta_0^{\mathrm{BOX160}}$. All this can be easily computed with FFTs over the computational box. However, since we place the constraints only into a relatively small spherical subvolume of the whole box, it is better to analyse this specific subvolume in real space. Inside this volume, the residual field of any given realisation should have the most variance on scales that are unconstrained and therefore random, and less variance on scales where the realisation is close to the original initial conditions.

Figure 5.4 shows the relative standard deviation of the residual field, $\delta_0^{\mathrm{CR}} - \delta_0^{\mathrm{BOX160}}$, which is the same as the rms error of the reconstructed field, normalised to the standard deviation of the density field itself, for reconstructed initial conditions from both mock catalogues and with all three methods, as a function of Gaussian smoothing scale R_{G}. This is computed by smoothing both the reconstructed and the original initial conditions with a Gaussian filter of radius R_{G}, then determining the spherical subvolume we are interested in, and compute the mean rms error (i.e. the standard deviation of the residual) relative to the standard deviation of the field in this volume. With this definition, a relative rms error of 0.5 in Figure 5.4 corresponds to a "signal-to-noise ratio" of 1:1 in the reconstructed field (where the "noise" originates from the added random modes), and a relative rms error of 0.25 to a ratio of 2:1. The analysis is done both for the smaller 30 Mpc/h spherical subvolume (left panels in Figure 5.4) and the larger 60 Mpc/h subvolume that corresponds to the data volume of the E60_10 mock (right panels).

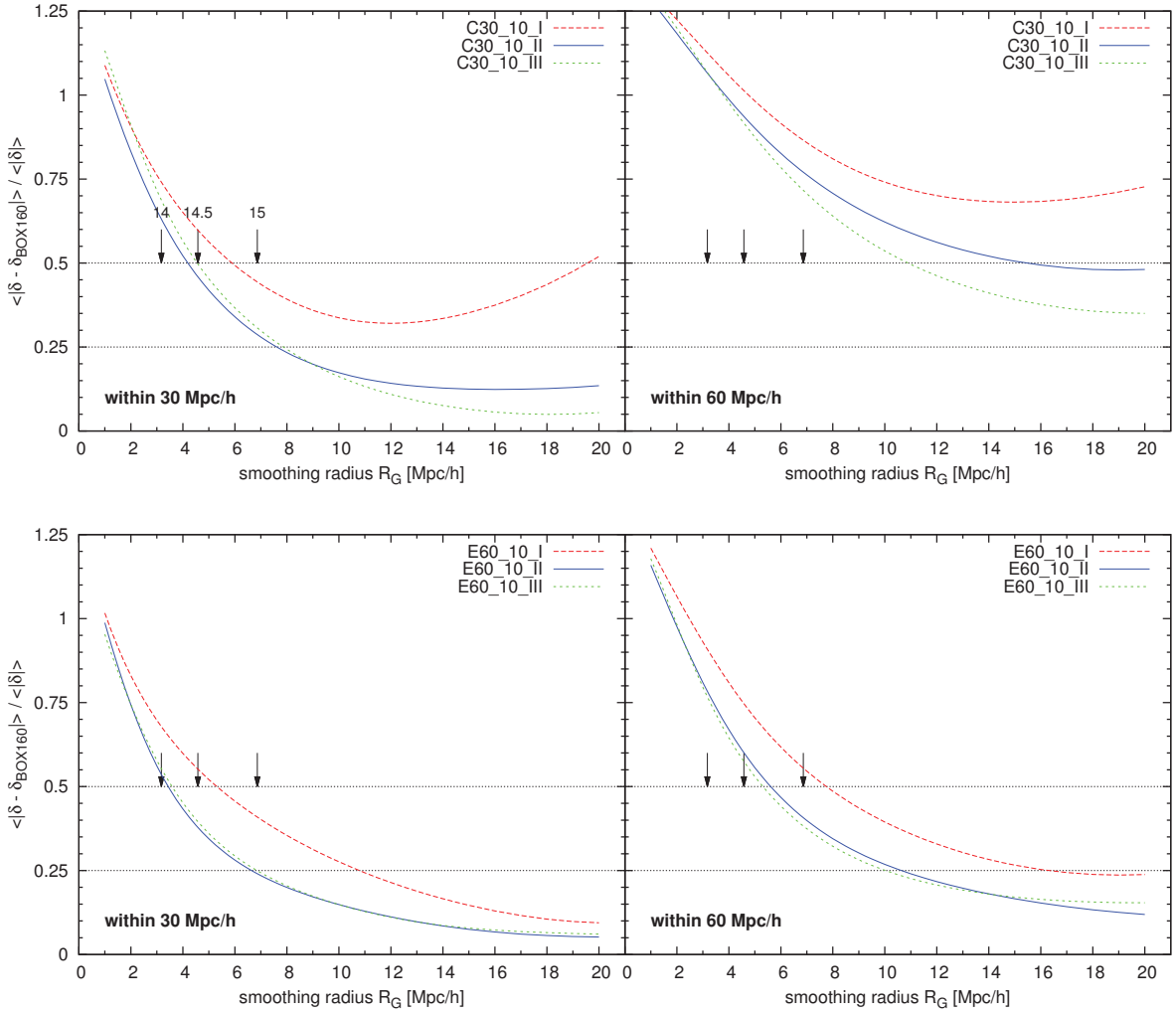

Figure 5.4: Standard deviation of the residual field (i.e. the rms error of the reconstructed initial density field) relative to the standard deviation of the original initial density field, as a function of Gaussian smoothing radius $R_{\rm G}$ for different constrained realisations with and without RZA. Top row: initial conditions from the C30_10 mock, with Methods I (dashed red, without RZA), II (solid blue, with RZA) and III (dotted green, with RZA). Bottom row: E60_10 mock. Spherical volumes within 30 Mpc/h (left panels) and 60 Mpc/h (right panels) of the mock observer were considered. The arrows mark the Gaussian smoothing scales corresponding to a mass scale of 10^{14}, $10^{14.5}$, and 10^{15} $M_\odot/h$.

The general trend is that on the smallest scales around 1 – 2 Mpc/h, the variance of the residual approaches the total variance of the field (i.e. the ratio approaches 1), which means that the field is completely dominated by the random component. This is the expected behaviour for all reconstructions – information on the small scales of the initial conditions is erased by $z = 0$ due to non-linear structure formation and there is no way to recover it from the data at $z = 0$, even if one applies RZA reconstruction. Interestingly, the ratio becomes even larger than 1 for the smallest scales. The reason for this is that the subvolume considered here is an overdense region in BOX160, and its variance is larger than the overdensity variance of the cosmic mean (i.e. the mean of the whole 160 Mpc/h box). Therefore, if the CR is dominated by random modes, the residual will have likewise a larger variance than the cosmic mean.

If one considers larger smoothing scales $R_{\rm G}$, the quality improves considerably. For the reconstructions from the C30_10 mock, a ratio of 0.5 (signal-to-noise 1:1) is achieved at a smoothing scale of around 6 Mpc/h for Method I without RZA (dashed red line in Figure 5.4), and at a scale around 4.5 Mpc/h for the methods using RZA (blue and green). This is an important result: it means that the RZA allows us to recover smaller scales in the initial conditions than Method I which does not use any Lagrangian reconstruction. The largest scales with $R_{\rm G} > 8$ Mpc/h are recovered very robustly with RZA. On the other hand, for Method I the error increases again with larger $R_{\rm G}$. The reason is that the largest-scale modes are not recovered well. In fact, the Method I reconstruction fails to reproduce the correct total overdensity inside the constrained volume, instead resulting in a field that has less overdensity and is closer to the cosmic mean. This is even more so if distances out to 60 Mpc/h from the mock observer are considered (upper right panel in Figure 5.4), which are outside of the constrained region for C30_10. There, even the largest modes of the overdensity field are not recovered very well by any of the reconstructions. This is expected behaviour. In terms of the velocity field, the WF/CR can generate an accurate estimate about the total bulk flow of the constrained volume and the tidal component of the velocity field that comes from outside of the data zone (Courtois et al. 2012). For example, the WF on the Cosmicflows-1 catalogue successfully recovers such features as the flows towards the Great Attractor or the Perseus-Pisces cluster, which are structures outside of the data zone of Cosmicflows-1. However, the obtained estimate of the density distribution outside of the data zone is not reliable: both the GA region and the Perseus-Pisces cluster cannot be recovered as clearly localised structures of the density field, but only as general large-scale overdensities towards their respective directions (Figure 3.2).

What influence on the reconstruction does the quality of the input data have? The bottom row of Figure 5.4 shows the reconstruction quality of the initial overdensity field for the CRs from the E60_10 mock, which has a larger data volume out to 60 Mpc/h and a higher density of datapoints, similar to the upcoming Cosmicflows-2 data. If we use this mock, the reconstruction quality improves. In the inner region, a ratio of 0.5 (signal-to-noise 1:1) is now reached at a Gaussian smoothing scale of about 5 Mpc/h for Method I without RZA and at about 3.5 Mpc/h for the methods using RZA reconstruction. The outer 60 Mpc/h region is also recovered much better, because now it is also covered by the input data, although the reconstruction quality is still worse because the mock observational errors on the peculiar velocities become very high in the mock catalogue (rms error of 600 km/s at a distance of 60 Mpc/h).

From the analysis of the reconstruction error as a function of Lagrangian scale we can make some predictions about the quality of the resulting CRs that will be obtained by running these reconstructed initial conditions forward. We are interested in an estimate of what kinds of features of the large-scale matter distribution can be reproduced in the simulation if the initial conditions have a sufficient accuracy down to some Lagrangian scale. Specifically, down to what

mass scale can we expect objects like dark matter haloes hosting clusters and galaxies to emerge at their original positions in the constrained resimulations?

From equations 5.1 and 5.2 we can assign a mass scale to each Gaussian smoothing scale R_{G}. For example, an object with virial mass $3 \times 10^{14}\ M_{\odot}/h$ corresponds to a Lagrangian volume in the initial conditions that is equivalent to a spherical top-hat with radius $R_{\mathrm{L}} \approx 10$ Mpc$/h$ (cf. Figure 4.1). This in turn corresponds to a Gaussian smoothing scale of $R_{\mathrm{G}} \approx 4.5$ Mpc$/h$. Therefore, there must be an overdensity at this Gaussian smoothing scale at the right position in the initial conditions for this object to form. For it to be at the right position at $z = 0$, the large-scale displacement field has also to be recovered with sufficient accuracy. We can now make a rough estimate about whether there is enough information in the reconstructed initial conditions to recover such an object. Let us assume that we need at least a signal-to-noise ratio of 1:1 to robustly recover a given structural feature on some Gaussian scale R_{G}, or mass scale M, in each CR. The three arrows in Figure 5.4 mark this point for three different mass scales $M = 10^{14}$, $10^{14.5} (\approx 3 \times 10^{14})$, and $10^{15}\ M_{\odot}/h$. We can then see that, for the sparse and limited C30_10 mock, the classical CLUES Method I shows a sufficient correlation with the original simulation for a mass scale of not much lower than $10^{15}\ M_{\odot}/h$. So we expect these resimulations to recover only the most massive clusters of the data zone, while all smaller structures will be generated from the random modes. This behaviour is consistent with the constrained simulations discussed in Klypin et al. (2003) and Gottlöber et al. (2010)[21]. On the other hand, the RZA method lowers this critical scale to just below $3 \times 10^{14}\ M_{\odot}/h$ for the C30_10 mock, and even to $1 \times 10^{14}\ M_{\odot}/h$ for the E60_10 mock. We therefore expect that with the help of RZA reconstruction, we can obtain constrained simulations that recreate more details of the matter distribution in the Local Universe and produce more objects that correspond to observed clusters, rather than being a product of the random modes. This prediction will be tested in practice on the BOX160 universe in Section 5.2.

5.1.4 Number density of constraints and constraining power

Having an estimate of the reconstruction quality of the WF/CR method for constrained initial conditions, another important question is: how many constraints are needed to obtain the optimal reconstruction? Velocities (or displacements) have a high correlation length and do not contain much information on smaller scales. Additionally, information on smaller scales of the linear field is lost at $z = 0$ due to non-linear structure formation. We therefore expect that at some point, adding constraints that probe these smaller scales in the non-linear regime does not significantly improve the reconstruction quality of the linear field, resulting in a saturation at some constraint density. It is important to understand this if one wants to estimate the constraining power of observational data.

This aspect was investigated in Lavaux (2010), where constraints from MAK-reconstructed displacements and initial positions were placed in a spherical data zone of radius 60 Mpc$/h$ using Method III. Initially they placed constraints on ψ_x, ψ_y, and ψ_z at 1314 positions inside this data zone, resulting in 3942 constraints. One result was that if the number of constraints is reduced by a factor of four, keeping only ≈ 1000 constraints, the error on the velocity field does not change, and the error on the density field changes only slightly. This would mean that the solution is saturated already at 1000 constraints distributed on a few hundred positions inside a sphere with radius 60 Mpc$/h$. This is a quite low density of constraints compared to what length scales we want to constrain with them.

[21] In their simulations, essentially only the Local Supercluster could be recovered directly from the radial peculiar velocity data. The more distant clusters were recovered with the help of additional smoothed density constraints that we do not use here.

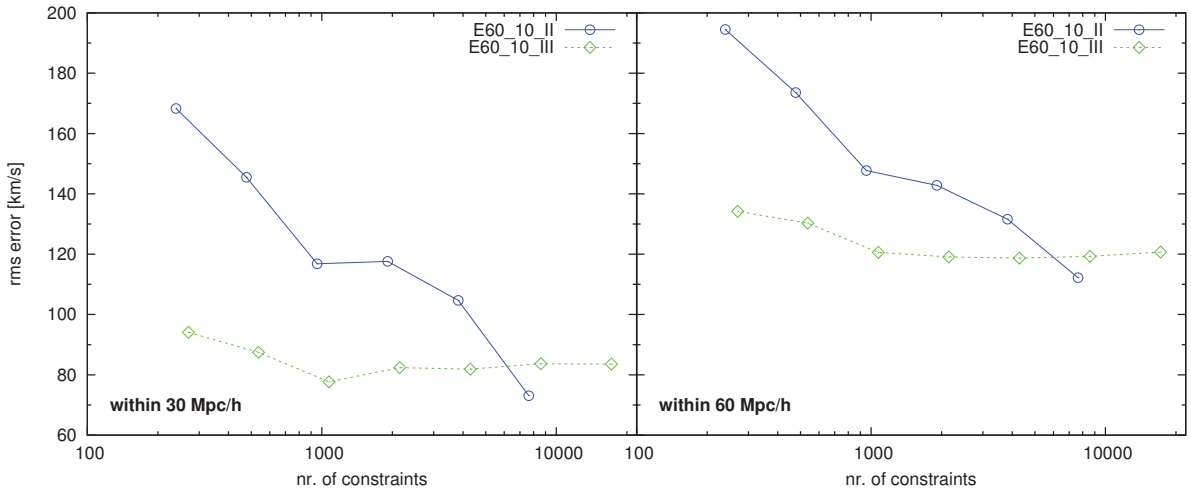

Figure 5.5: rms error on the velocity field of the reconstructed initial conditions from the E60_10 mock, as a function of the number of constraints used. The rightmost point of each curve denotes the full set of constraints for Method II (solid blue) and Method III (dotted green) RZA reconstruction; in each step to the left, the number of constraints was then reduced by a factor of two. The left panel shows the rms error within 30 Mpc/h of the mock observer; the right panel corresponds to the larger volume with 60 Mpc/h radius.

In order to repeat a similar analysis with constraints obtained from RZA reconstruction, we took the constraints obtained with Methods II and III from the E60_10 mock, which likewise covers a spherical volume with 60 Mpc/h radius. The E60_10_II set contains 7637 radial displacement constraints at 7637 positions, while the E60_10_III set contains 17181 constraints (for each displacement component) at 5727 positions (1910 positions were discarded due to overlapping, using $f_{\mathrm{G}} = 1$). We then successively reduced the number of constraints by factors of 2, 4, 8, ... , and in each step we created a realisation of reconstructed initial conditions and determined its rms error on the reconstructed initial velocity field. The result is shown in Figure 5.5. The blue curve corresponds to Method II constraints, and the green curve stands for Method III. The rightmost point always corresponds to the full set of constraints.

The plot reveals that, if one uses Method III, the reconstruction quality is indeed saturated at about 1000 constraints inside a 60 Mpc/h radius, confirming the result of Lavaux (2010). For Method II, one needs many more constraints to achieve the same reconstruction quality; if the number density of constraints is reduced, the error quickly increases. This is similar to the RZA reconstruction of the displacement field in Chapter 4, where the number of datapoints used had a significant impact on the reconstruction quality. On the other hand, with the full E60_10_II set of constraints, the error is eventually even lower than for Method III. Moreover, it does not seem to saturate at this level, and if one would use even more constraints, probably the error could be reduced further. This means that, using Method II to generate constrained initial conditions from peculiar velocity data, it is beneficial to have as many datapoints as possible.

We want to investigate in some more detail why varying the number of constraints has so different effects on the two different constraint placing methods. A useful quantity in this context is the correlation vector η_i (see equation 3.26). For each constraint c_i, the value of η_i determines its statistical weight for the WF/CR solution. The higher the absolute value of η_i, the more will the constraint c_i contribute to the complete solution (3.27) for the mean field and its constrained realisations. This kind of analysis could also be useful in other contexts, such as finding an

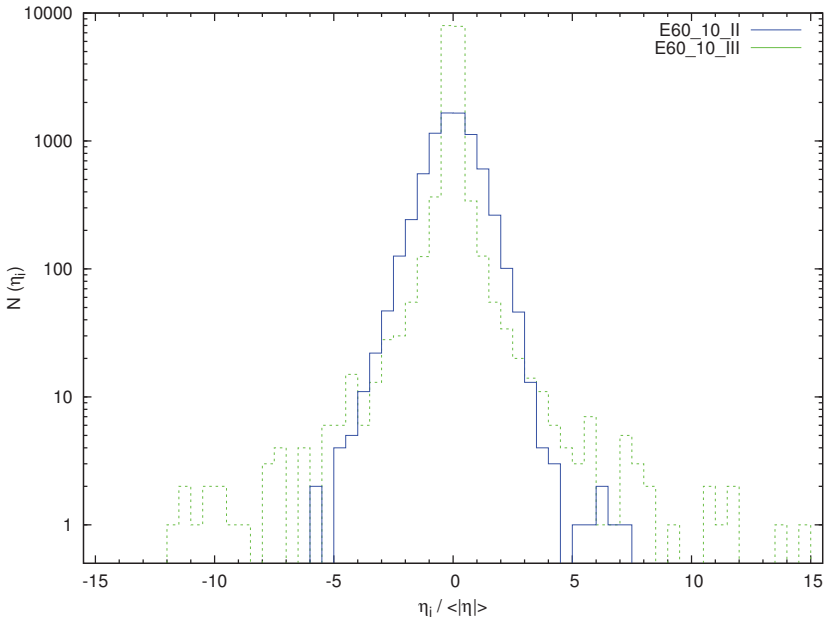

Figure 5.6: Distribution function of the correlation vector values η_i for constraints c_i used in the E60_10 reconstructed initial conditions with Method II (solid blue) and Method III (dotted green). The higher the absolute value of η_i, the more the corresponding constraint c_i will contribute to the solution for the constrained field. For the green histogram, eleven values are outside of the histogram range, the lowest being at -30 and the highest at 55.

optimal grouping or other post-processing procedure to linearise the observational data before applying WF/CR.

Figure 5.6 shows the distribution of the η_i values (normalised to their variance) for the 7637 and 17181 constraints in the E60_10_II and E60_10_III set of constraints, respectively. For Method II, the distribution is relatively close to Gaussian (blue histogram), while for Method III it has a very high kurtosis (green histogram). Thus, in Method III, a few constraints (at the outer ends of the histogram) contribute most to the solution, while the majority of constraints have η_i values close to zero and do not significantly influence the result. On the other hand, for Method II the information determining the WF/CR solution is spread out much more equally across all constraints. This is not unexpected: if we use radial peculiar velocities as constraints, each constraint determines only one component of the displacement field, with a significant observational error attached, and therefore carries less information than if all three components of the RZA reconstructed displacement field are fixed, which strongly constraints the large-scale modes of the field. On the other hand, the radial velocity approach seems to give a more precise result if the full set of constraints is used. This suggests that by combining Method III with the RZA reconstruction, we introduce some kind of systematic error.

Another peculiarity of Method III can be seen from the obtained values of χ^2 computed from the autocorrelation matrix of the constraints (equation 3.43). If the constraints are statistically compatible with a Gaussian random field that has the assumed linear power spectrum $P(k)$, the value of χ^2 should be equal to the degrees of freedom ($\chi^2/\mathrm{dof} = 1$). For valid cosmological initial conditions, the requirement of a Gaussian random field that follows $P(k)$ is absolutely necessary. Since constraints from data taken at $z = 0$ usually have more power due to the observational errors and the non-linearity of the data, we used the regularisation parameter σ_{NL} to enforce $\chi^2/\mathrm{dof} = 1$ (see equation 3.45). In the case of Method III, we actually find that even with

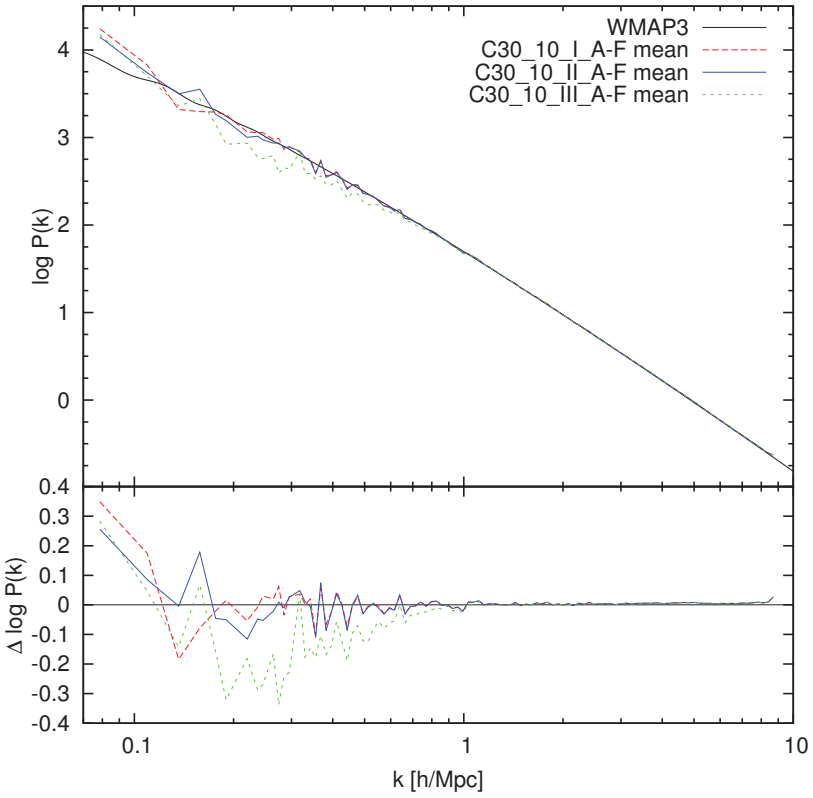

Figure 5.7: Power spectrum $P(k)$ in the inner half of the box for reconstructed initial conditions from the C30_10 mock, with methods I (dashed red), II (solid blue) and III (dotted green), averaged over all six realisations (A – F) for each method. The lower panel shows the difference between the $P(k)$ of the realisations and the WMAP3 power spectrum used for the prior model (black).

$\sigma_{\rm NL} = 0$, we have $\chi^2/{\rm dof} < 1$, meaning that the constraints impose a solution that has not enough power compared to $P(k)$. This is observed in the initial conditions created with Method III. In order to make the effect more apparent, we cropped the box to a sub-cube of half the boxlength (80 Mpc/h) centred on the mock observer. Figure 5.7 shows the power spectrum $P(k)$ of reconstructed initial conditions with all three methods I, II and III in this sub-box, averaged over all six realisations A-F for each method that uses the C30_10 mock (the results are similar for the E60_10 realisations). While the power spectrum for Methods I and II is compatible with the input power spectrum from WMAP3 (solid black line), the average power spectrum of Method III realisations shows a significant lack of power for wavenumbers between 0.2 and 1 h/Mpc (corresponding to length scales between 6 and 30 Mpc/h), rendering them unusable for the task of performing constrained simulations.

To analyse this further, we can determine the Gaussianity of the recovered initial conditions. Figure 5.8 shows the distribution function of the overdensity field within the $R_{\rm max} = 30$ Mpc/h sphere of the reconstructed initial conditions for both the C30_10 and the E60_10 mock with all three reconstruction methods for one particular seed. As we already have seen before, Method I initial conditions show a significant skewness for the E60_10 mock (this is equivalent to Figure 4.15), over-emphasising overdensities. Conversely, the reconstructions with Method III are clearly skewed towards the low-density end. Only Method II recovers initial conditions for both mock catalogues that are very closely Gaussian and therefore show the statistics required for valid initial conditions.

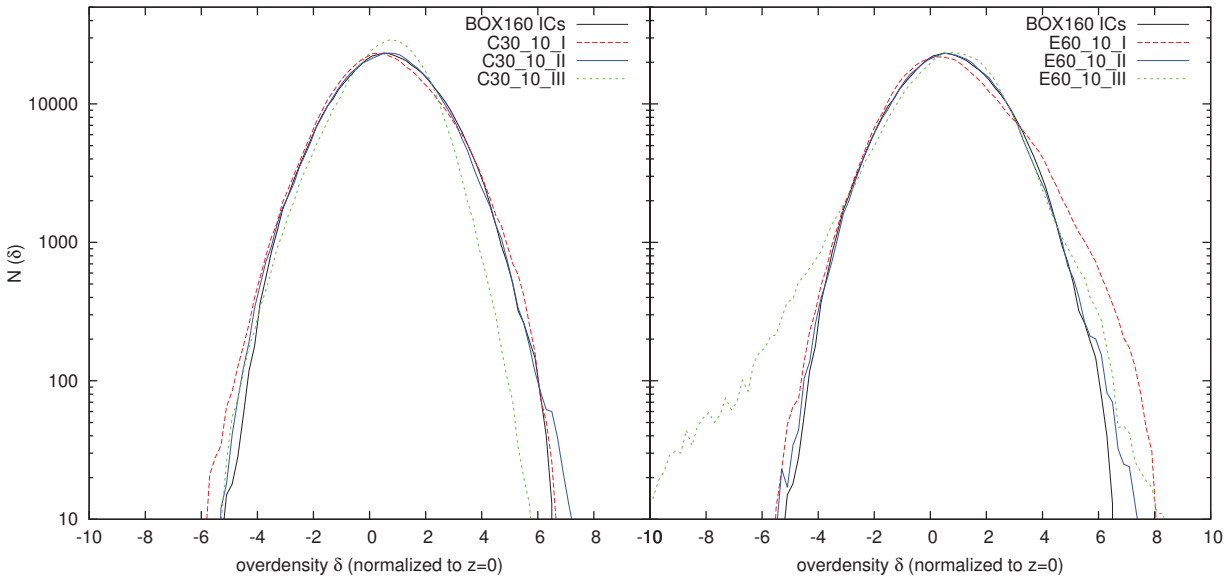

Figure 5.8: Distribution of the overdensity field δ within 30 Mpc/h of the mock observer for reconstructed initial conditions from the C30_10 mock (left panel) and E60_10 mock (right panel), with Methods I (dashed red), II (solid blue) and III (dotted green), for the A realisations. The distribution of the original BOX160 initial conditions is shown in black.

How can we explain that Method III works so well with displacements coming from MAK reconstruction (Lavaux 2010), but fails on displacements coming from RZA reconstruction? A possible explanation is the fact that the power spectrum of the non-linear velocity field at $z = 0$ actually has less power on these scales compared to the linear power spectrum, as was shown in Section 2.3 (see Figure 2.12 and the visual appearance of the velocity field at $z = 0$ in Figure 2.11). The collapse of overdense regions into haloes, the infall of surrounding matter onto these haloes, and the expansion of underdense regions leading to large voids makes the velocity flow more smooth and laminar on scales larger than those where virialised objects are formed (a few Mpc/h). The RZA reconstruction additionally smooths this field by applying a Wiener filter to the velocity data, so that the resulting RZA-reconstructed displacement field ψ^{RZA} suffers from a lack of power that is incompatible with the assumed linear $P(k)$. If we now directly take the values of $\boldsymbol{\psi}^{\mathrm{RZA}}$ as constraints for the initial conditions, we transfer this lack of power to the resulting constrained realisation, which will have an incorrect power spectrum. Of course, the changed power spectrum of the non-linear field should also affect $z = 0$ radial velocity data itself which we use as constraints in methods I and II. However, by constraining only one component of the velocity vector per point, and due to the significant observational and non-linearity errors that add a lot of noise, the constraining power of each individual point is reduced. The WF/CR operator has then more maneuvering room to find a solution that is the best compromise between the imposed constraints and the correct linear power spectrum $P(k)$.

The lack of power in Method III initial conditions can also be explained completely independent from the effect of the deviation in the actual $P(k)$ of the velocity data at $z = 0$. By combining RZA with Method III, we construct in the end a CR by using as constraints the result of a previous Wiener filtering. We are therefore actually applying the Wiener filter iteratively. This is known to fail (see Section 3.2.1). Conversely, with Method II we are also applying the

WF/CR twice, but each time on the same unfiltered data that is just displaced between the first step and the second. In this way, we do not introduce additional unnecessary filtering and do not remove power from the final solution.

Apart from the constraint-placement methods I, II, and III discussed here, we tried a number of different configurations. Among them was omitting the Lagrangian smoothing radius from Method III, adding this smoothing radius to Method II, and changing the direction $\hat{\boldsymbol{e}}_\alpha$ in Method II so that it was again the radial component with respect to the mock observer at the position $\boldsymbol{x}_{\rm init}^{\rm RZA}$ of the constraint. All of these setups lead to reconstructions of lesser quality than those obtained with Method II. Therefore, we use Method II as the tool of choice in the remainder of this chapter. We will also continue to use Method I for comparison.

To summarise it can be said that, if one wishes to create initial conditions for constrained simulations based on peculiar velocity data, the optimal method at this point is the procedure of combining these datapoints with the RZA reconstruction by shifting them to their reconstructed initial positions and subsequently using them as constraints for the initial displacement field. The method of Lavaux (2010), which instead uses the reconstructed three-dimensional displacement field itself as constraints, does not work in this case because of the systematic errors that are introduced by the RZA reconstruction. It is however, by itself, a very efficient method of constraining the initial field to a high degree with relatively few data points. It may therefore be the method of choice if the reconstructed displacement field originates from galaxy redshift surveys, such as in the case of MAK reconstruction. Other advanced methods for reconstructing the initial displacement field from galaxy redshift surveys are being developed, such as the work of Kitaura & Angulo (2012); Kitaura et al. (2012); Kitaura (2012), who utilise higher-order Lagrangian perturbation theory. These results could be used as well to constrain initial conditions, and in this case the approach of Method III to use three-dimensional constraints on the displacement field may be the optimal choice.

5.2 Constrained resimulations of BOX160

Having obtained several realisations of reconstructed initial conditions of our test universe, the BOX160 simulation, we continue the study of the constrained realisation method by running these initial conditions forward. We omit the realisations obtained with Method III for reasons discussed in Section 5.1.4, and are left with 24 sets of initial conditions: C30_10_I_A – F, without RZA; C30_10_II_A – F, with RZA; E60_10_I_A – F, without RZA but using the larger-volume E60_10 mock catalogue as the data source, and E60_10_II_A – F, with RZA using that mock. All these initial conditions were normalised to a starting redshift of $z = 30$ and then run forward until $z = 0$ using the cosmological code GADGET with a resolution of 256^3 dark matter particles. In the following, we analyse these simulations and compare them to the original BOX160 simulation at $z = 0$. The goal of this study is to investigate to what degree the observed large-scale structure of the BOX160 universe at $z = 0$ are recreated in the constrained resimulations. We can then estimate from this test how accurately constrained realisations generated from observed peculiar velocity catalogues, such as the Cosmicflows-1 catalogue and the upcoming Cosmicflows-2 catalogue, can recreate the observed Local Universe.

5.2.1 Resimulated fields at $z = 0$

Figure 5.9 shows a cell-to-cell comparison between the evolved constrained resimulations obtained from the C30_10 mock and the original field of BOX160 in the constrained volume (out to 30 Mpc/h distance from the mock observer), both for Method I (without RZA) and Method II (with

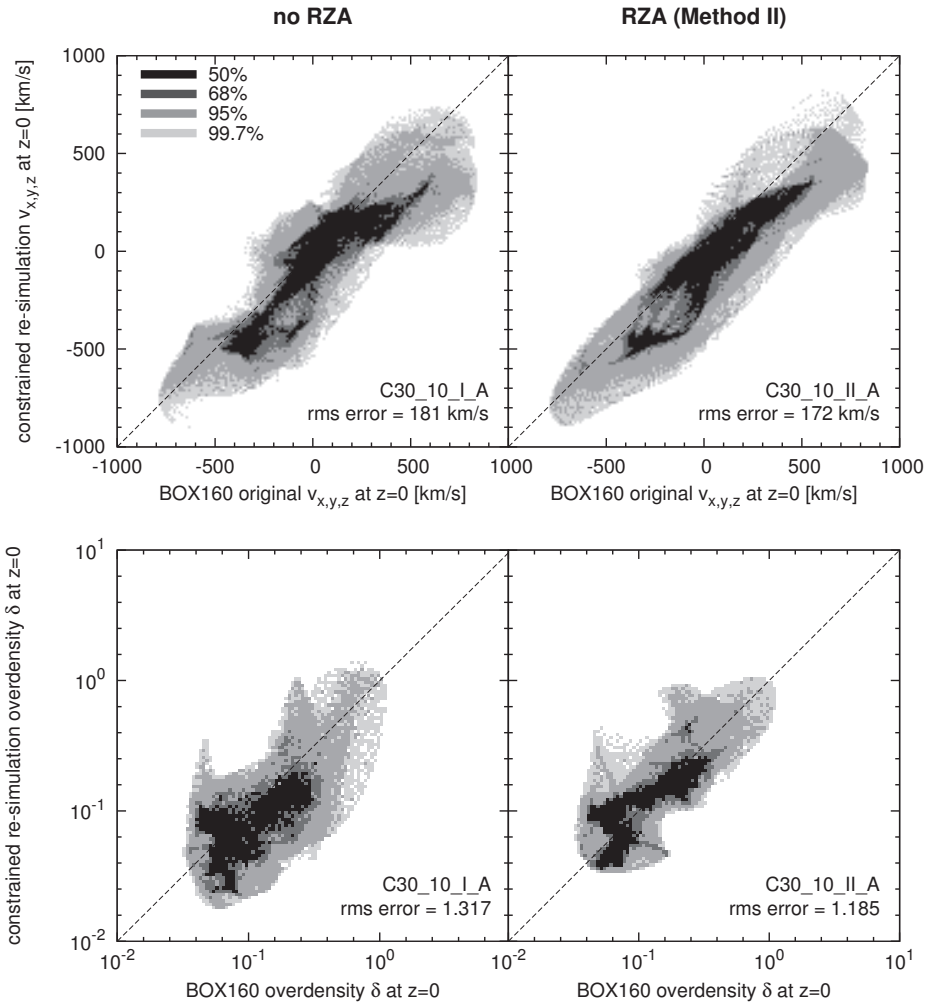

Figure 5.9: Cell-to-cell comparison between the evolved constrained resimulations at $z = 0$ for and the original BOX160 simulations. Top row: velocity field inside the mock volume (all three components were concatenated); Bottom row: density field. All fields were smoothed with a 5 Mpc/h Gaussian.

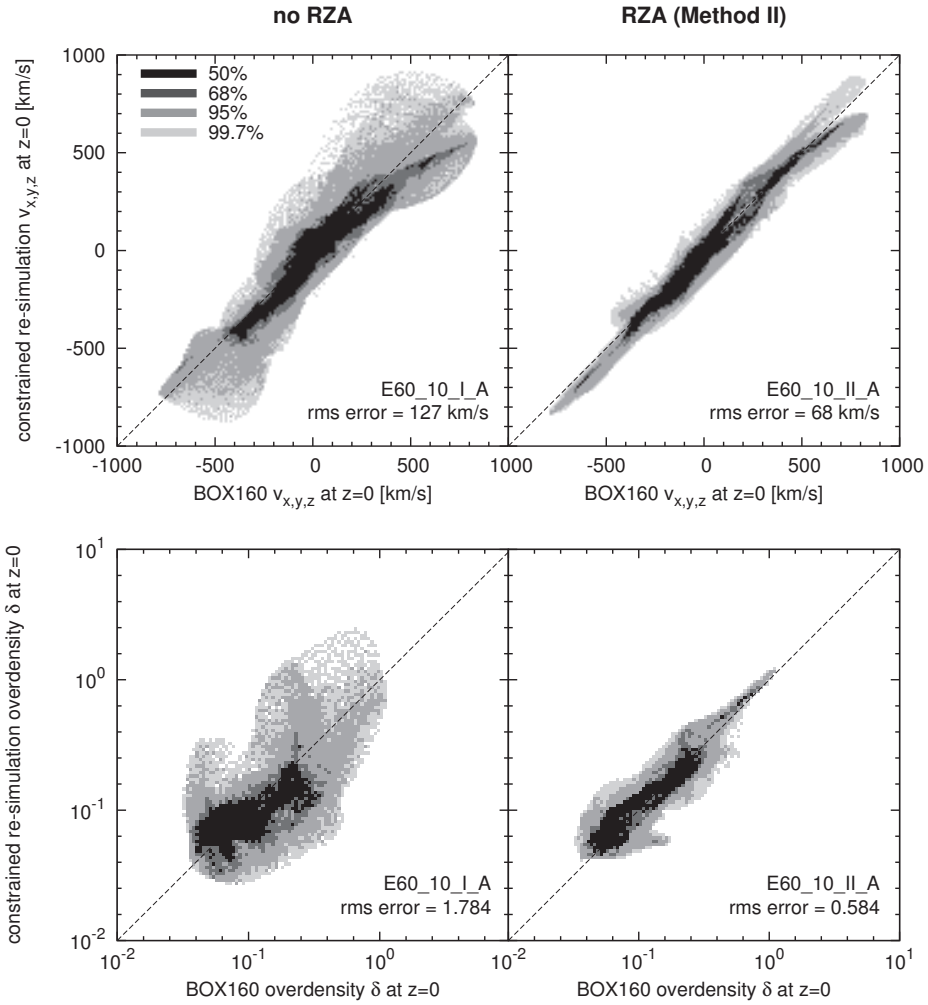

Figure 5.10: As Figure 5.9, but for constrained realisations from the E60_10 mock with twice the data zone radius.

RZA). Figure 5.10 displays the same for the resimulations from the E60_10 mock. Especially for the more accurate E60_10 mock, the added RZA reconstruction that was used to obtain the initial conditions in Method II significantly improves the correlation of the obtained non-linearly evolved simulation with the actual field of BOX160 at $z = 0$. In the latter case, the peculiar velocity field is reproduced accurately down to an accuracy of about 1/4 of its total standard deviation, as it was already the case for the linear field of the reconstructed initial conditions. This is an impressive improvement of the reconstruction quality. From the correlation of the density fields, it can be seen that Method I, which does not use RZA, significantly underestimates the total density of the constrained region, which is overdense compared to the cosmic mean. The RZA method, on the other hand, accurately reproduces this total overdensity, for both the C30_10 and E60_10 resimulations. The total overdensity as well affects the abundance of dark matter haloes. Figure 5.11 shows the binned dark matter halo mass functions of all realisations obtained from the C30_10 mock catalogue, both with and without RZA. The mass functions of the individual realisations are shown with coloured lines and points, their average is represented by the thick dashed black line, and the actual mass function of the BOX160 inside the 30 Mpc/h region is shown with the solid dashed black line. The mass function of this region lies significantly above the cosmic mean (grey lines); the volume contains about 2.5 times more haloes with masses $> 10^{13} M_\odot/h$ than a region of mean cosmic density would. The constrained resimulations without RZA only partially follow that behaviour. Their average mass function is also above the cosmic mean, but still systematically underestimates the actual halo abundance in the corresponding region of the original BOX160 simulation. On the other hand, the resimulations where RZA was used for constructing the initial conditions follow the mass function of BOX160 much more closely.

5.2.2 The resimulated BOX160 Universe

The large-scale structure of the "local universe" for our test case, i.e. the region around the mock observer in the BOX160 simulation, is shaped by several dominant structures that to some degree resemble the observed Local Universe. The "Milky Way" analogue halo that was chosen as the position of the mock observer ($X = 75$; $Y = 64$; $Z = 80$) lies in the vicinity of a "Virgo" cluster with a virial mass of $3.25 \times 10^{14} M_\odot$, which is embedded in a thick filament parallel to the X-axis. The local flow of the mock observer is directed towards this structure, resembling the observed Virgocentric infall (Tully et al. 2008). On a larger scale, the whole region is dominated by a flow towards the "Great attractor" (GA), a massive structure at about $X = 30$, $Y = 60$ that lies outside of the 30 Mpc/h data zone. This configuration can be seen in the top left map of Figure 5.12. The BOX160 Virgo cluster is however not the most massive structure within 30 Mpc/h, as there is an even more massive region at a distance of slightly less than 30 Mpc/h, lying in a direction in between Virgo and the GA, that contains two clusters with masses of $6.06 \times 10^{14} M_\odot/h$ and $5.20 \times 10^{14} M_\odot/h$, respectively. We associate them with the Centaurus and Hydra clusters. They in turn cause a significant infall flow towards them on the surrounding structure. In the other direction (towards negative Y), there is also a large void that contributes to the shape of the large-scale velocity field by creating a push outwards of it, although this particular feature in the BOX160 may not be as dominant as the observed Local Void (Tully et al. 2008). Constrained resimulations of this test universe should be able to recover all these characteristic structures. BOX160 contains this specific configuration because it is itself a constrained simulation of the Local Universe, created with what we call here Method I from peculiar velocity data drawn from the MARK III (Willick et al. 1997), SBF (Tonry et al. 2001) and Karachentsev et al. (2004) catalogues. It is not entirely accurate, for example the virial mass of the BOX160 Virgo cluster is 2 – 3 times less than the estimated virial mass of its

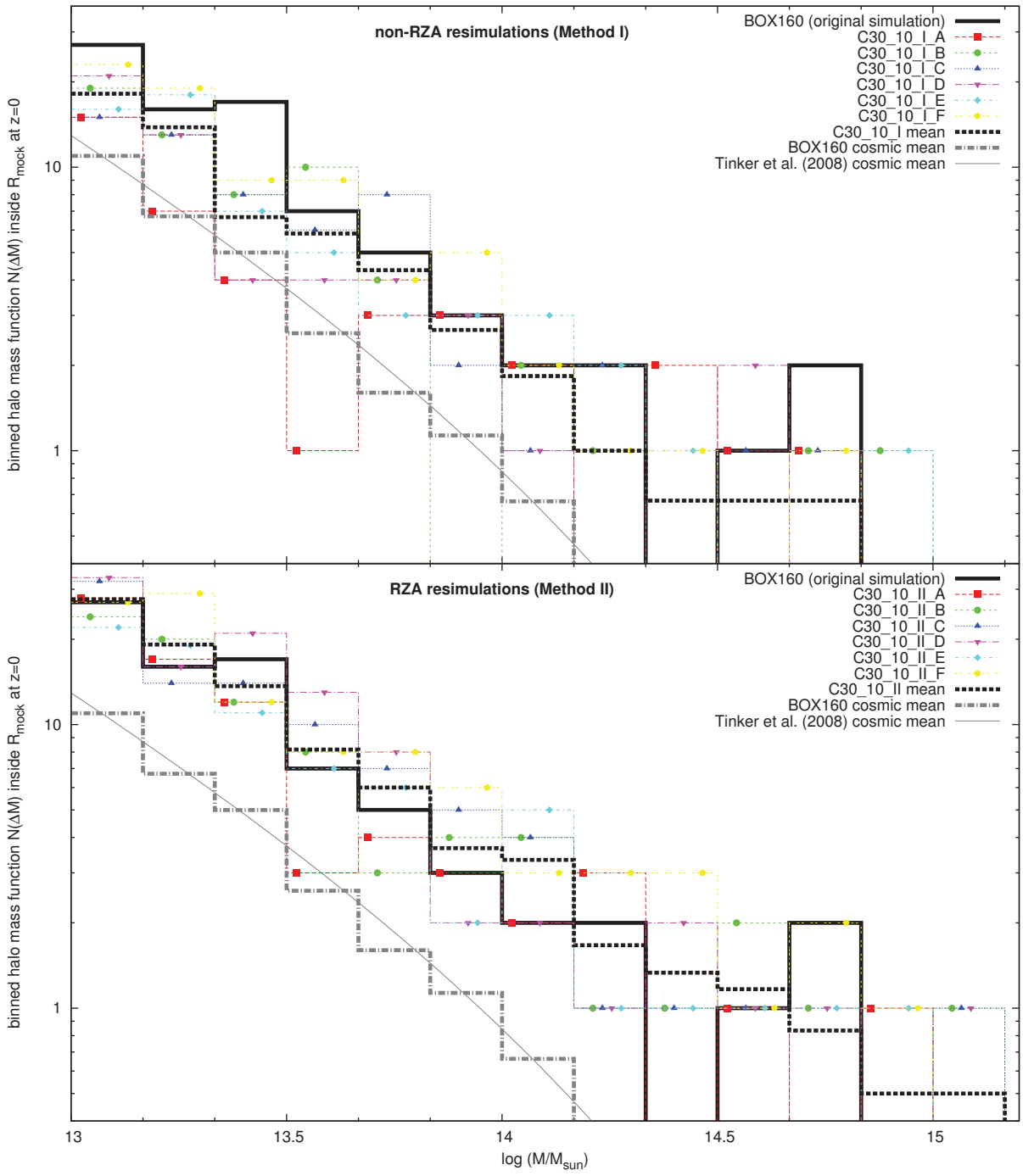

Figure 5.11: Dark matter halo mass function inside the 30 Mpc/h mock volume for the six different realisations from C30_10 without RZA (top) and with RZA (bottom). The individual realisations are shown with coloured lines/points; their average mass function is the thick black dashed line. The original mass function of the BOX160 in this volume is the thick solid black line. The mass function for an average subvolume of radius 30 Mpc/h is shown in thick grey (by averaging over the full $L = 160$ Mpc/h volume of BOX160) and thin grey (theoretical model of Tinker et al. 2008).

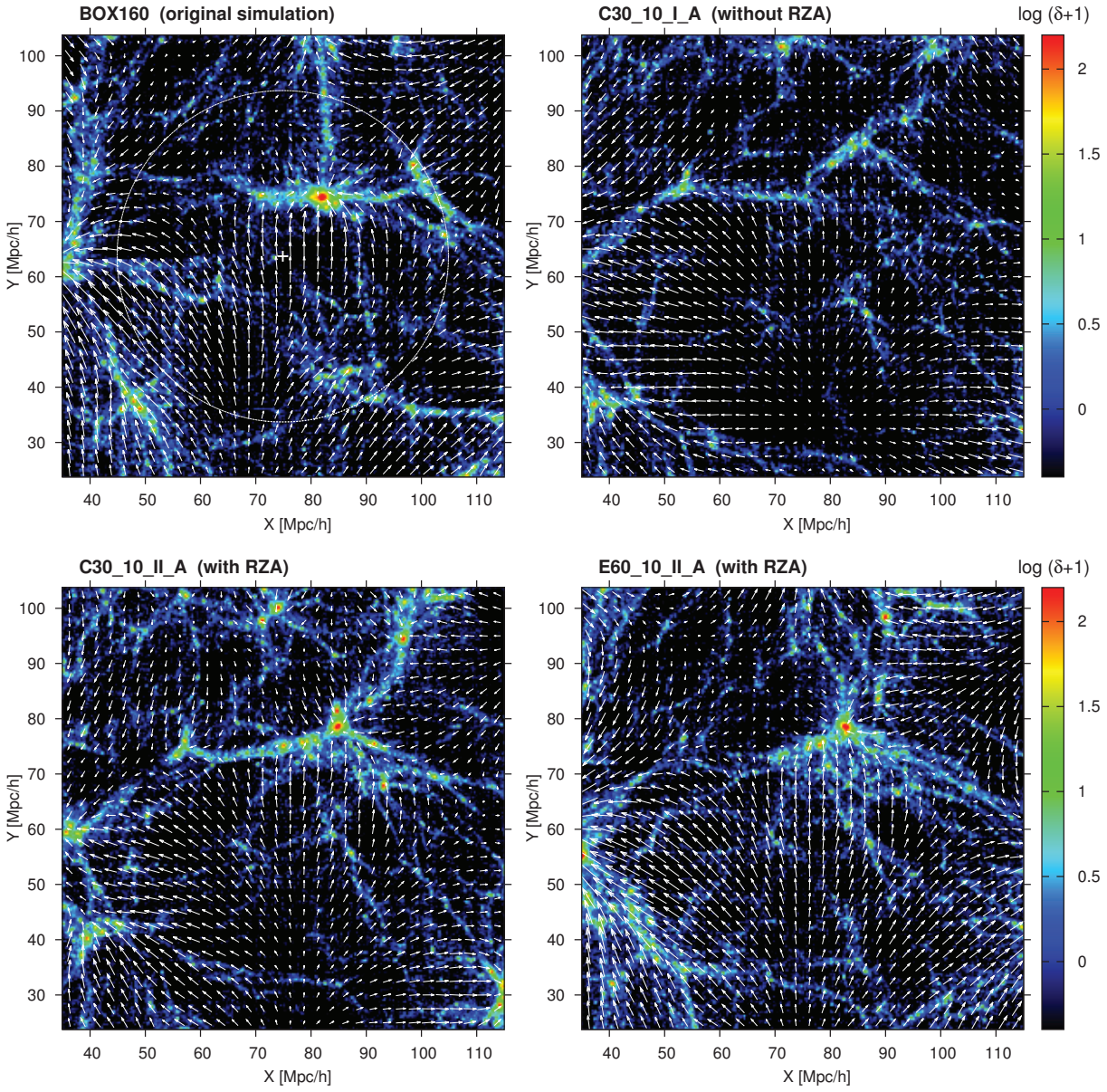

Figure 5.12: Density and velocity fields in a 10 Mpc/h thick slice at $87 < Z < 97$ Mpc/h for the original BOX160 simulation (top left) and three constrained resimulations without RZA (top right), with RZA (bottom left), and with RZA from the E60_10 mock using a larger data volume (bottom right). The white circle marks the constrained region for the C30_10 mock. The density is shown in logarithmic scale. The arrows are proportional to the amplitude of the peculiar velocity at each position. The prominent object inside the BOX160 data zone is the simulated Virgo cluster with a virial mass of $3.25 \times 10^{14} M_{\odot}/h$. In all three constrained resimulations the same random seed was used.

observed counterpart (Fouqué et al. 2001), but the described main characteristics of the large-scale structure are present. Ignoring the origin of BOX160 for this test and taking its large-scale matter distribution as "given", we try to reproduce it in the constrained resimulations, which should retain the main characteristics that BOX160 shares with the observed Local Universe in this "second pass" of the reconstruction-resimulation cycle.

The other panels of Figure 5.12 show three constrained realisations obtained from the C30_10 mock catalogue without RZA (top right), with RZA (bottom left), and from the larger-volume E60_10 mock catalogue with RZA, in the 10 Mpc/h thick slice that should contain the BOX160 Virgo cluster. One can immediately notice that, while all three simulations faithfully reproduce the large-scale flow towards the GA, the resimulation that was constructed without RZA fails to create a Virgo cluster. Apparently, the mass of the BOX160 Virgo cluster of $3.25 \times 10^{14} M_\odot/h$ lies below the scale that could be reproduced with the non-RZA Method I of generating constrained initial conditions, which is consistent with the estimate of the constrained mass scales from Section 5.1.3. Why then does the BOX160, which was created with the same method, have a Virgo cluster in the first place? The reason is that the observed Virgo cluster has a mass that is 2 – 3 times higher, which places it in the recoverable mass range. It was however reproduced with a too low mass in BOX160 and thus disappears on the second pass of the reconstruction-resimulation cycle. The non-RZA resimulations show some overdensity in that region, and there is also the tendency of a flow in that direction, but there is not enough overdensity to form a cluster of comparable mass. On the other hand, both resimulations in Figure 5.12 that include RZA reconstruction have a cluster at that position, which is similarly embedded in a thick filament, with a strong flow towards it from the position of the mock observer.

In order to understand more clearly how robustly the Virgo cluster is recovered in the constrained resimulations we searched for it in all resimulations by searching for haloes that would be within 10 Mpc/h of the original BOX160 Virgo's position and would have a mass of at least a $10^{14} M_\odot/h$. The result was that in all realisations using RZA, both from the C30_10 mock and the E60_10 mock, we could find a cluster corresponding to Virgo. These objects are listed in Table 5.2. On the other hand, we could not find such an object in any of the realisations created without RZA, which clearly displays that the RZA improves recovering the original structures in constrained simulations. The resimulated Virgo is not exactly at its BOX160 position in the resimulations; the error lies between 1.7 and 6.3 Mpc/h and varies with the random seed. The average position of all resimulated Virgos is only about 2 Mpc/h away from its original position in BOX160, so this fluctuation is probably due to the influence of the random modes rather than a systematic shift. It is interesting to note that in the RZA resimulations, all the Virgo reproductions have a lower mass than the original BOX160 Virgo, just as the latter has a lower mass than the observed Virgo cluster. This may be connected to the findings of Courtois et al. (2012), who analysed the Cosmicflows-1 distance catalogue with the Wiener filter and found that the Virgo cluster is not dominating the peculiar velocity field as much as expected. It may therefore be harder to recover accurately in a constrained simulation.

Another possibility to estimate the robustness of the reconstruction is to compute an average of the density and velocity fields over the different evolved realisations. This way, structures coming from the random component will tend to average out and be suppressed, while structures appearing consistently in every realisation will be enhanced. Figure 5.13 shows the same slice as in Figure 5.12, but averaged over all six realisations A – F for the C30_10_I (without RZA), C30_10_II (with RZA), and E60_10_II (with RZA, larger mock). The position of the original BOX160 Virgo cluster is marked with a blue cross. The resimulations done without RZA (top right) show some overdensity in this region, and a tendency of a flow towards it, but the overdensity is not high enough to create a cluster of mass comparable to Virgo in any of the

non-RZA realisations. The density peak is much more pronounced in the simulations with RZA (bottom panels), which all have a massive object near to that location. We also see that other structures with less mass, such as the overdense region below Virgo around $X = 90\ Y = 40$, are not present in the averaged fields, which means that they lie in a mass range that is too low to be recovered by the reconstruction of initial conditions for any of the methods/mock catalogues.

5.2.3 Recovered mass scales

To obtain a more precise estimate about the mass scale on which objects can be consistently recovered in the constrained resimulations, we search for other massive clusters in the data zone. Table 5.3 shows the seven most massive haloes within the 30 Mpc/h data zone in BOX160. The largest overdense region within the data zone is dominated by the Hydra and Centaurus clusters, both with masses above $5 \times 10^{14} M_\odot/h$, separated by a distance of 10 Mpc/h. This overdense region is robustly recovered in all simulations including the ones from Method I not using RZA. It therefore lies above the minimum mass scale that can be recovered without RZA. Figure 5.14 shows the corresponding slice of the density and velocity fields that contains the Hydra and Centaurus clusters, averaged over the different realisations[22]. All resimulations show

Simulation	Position X, Y, Z [Mpc/h]	pec. vel. v_x, v_y, v_z [km/s]	Mass [$10^{14} M_\odot/h$]	Pos. error [Mpc/h]	Mass rel. to BOX160 Vir
BOX160 Vir	82.0, 74.4, 91.8	+22, +142, −473	3.25	–	–
C30_10_II_A	84.8, 78.5, 92.6	−23, −61, −391	1.66	5.0	51 %
C30_10_II_B	80.5, 76.8, 96.7	−22, +6, −912	1.98	5.7	61 %
C30_10_II_C	80.3, 72.6, 94.4	+26, +135, −403	3.24	3.6	100 %
C30_10_II_D	83.6, 73.8, 92.1	+101, +231, −326	1.21	1.7	37 %
C30_10_II_E	80.2, 76.5, 94.3	+72, +247, −755	2.33	3.7	71 %
C30_10_II_F	87.1, 78.0, 92.1	−273, +75, −527	2.83	6.3	87 %
C30_10_II mean	82.8, 76.0, 93.7	−19, +106, −552	2.21	2.6	68 %
E60_10_II_A	82.8, 78.5, 90.5	−113, −64, −507	1.87	4.4	58 %
E60_10_II_B	78.6, 74.7, 93.9	−200, +149, −682	2.73	4.1	84 %
E60_10_II_C	77.1, 72.8, 91.8	+29, +135, −374	1.21	5.2	37 %
E60_10_II_D	82.0, 76.0, 89.5	−142, +201, −576	3.03	2.8	93 %
E60_10_II_E	78.5, 75.3, 92.2	+85, +289, −706	1.31	3.7	40 %
E60_10_II_F	84.0, 76.3, 88.8	+21, +259, −486	2.80	4.1	86 %
E60_10_II mean	80.5, 75.6, 91.1	−53, +161, −555	2.16	2.1	66 %

Table 5.2: Virgo candidates found in the RZA resimulations of BOX160 at $z = 0$. The first line corresponds to the original "Virgo" object in the BOX160. The following lines list the haloes found in the AHF catalogues of the respective resmulations that are within 10 Mpc/h of the BOX160 Virgo position and have a virial mass of at least $10^{14} M_\odot/h$.

[22] Note that in the observational data the Hydra/Centaurus clusters and the Virgo cluster lie within the supergalactic plane around $SGZ = 0$ (Figure 3.2). On the other hand, while BOX160 uses the same coordinate system, there the Hydra/Centaurus clusters and the Virgo cluster are not located at the same Z (=in the same X/Y plane), but actually in planes about 10 – 15 Mpc/h apart from each other in the Z dimension. This is

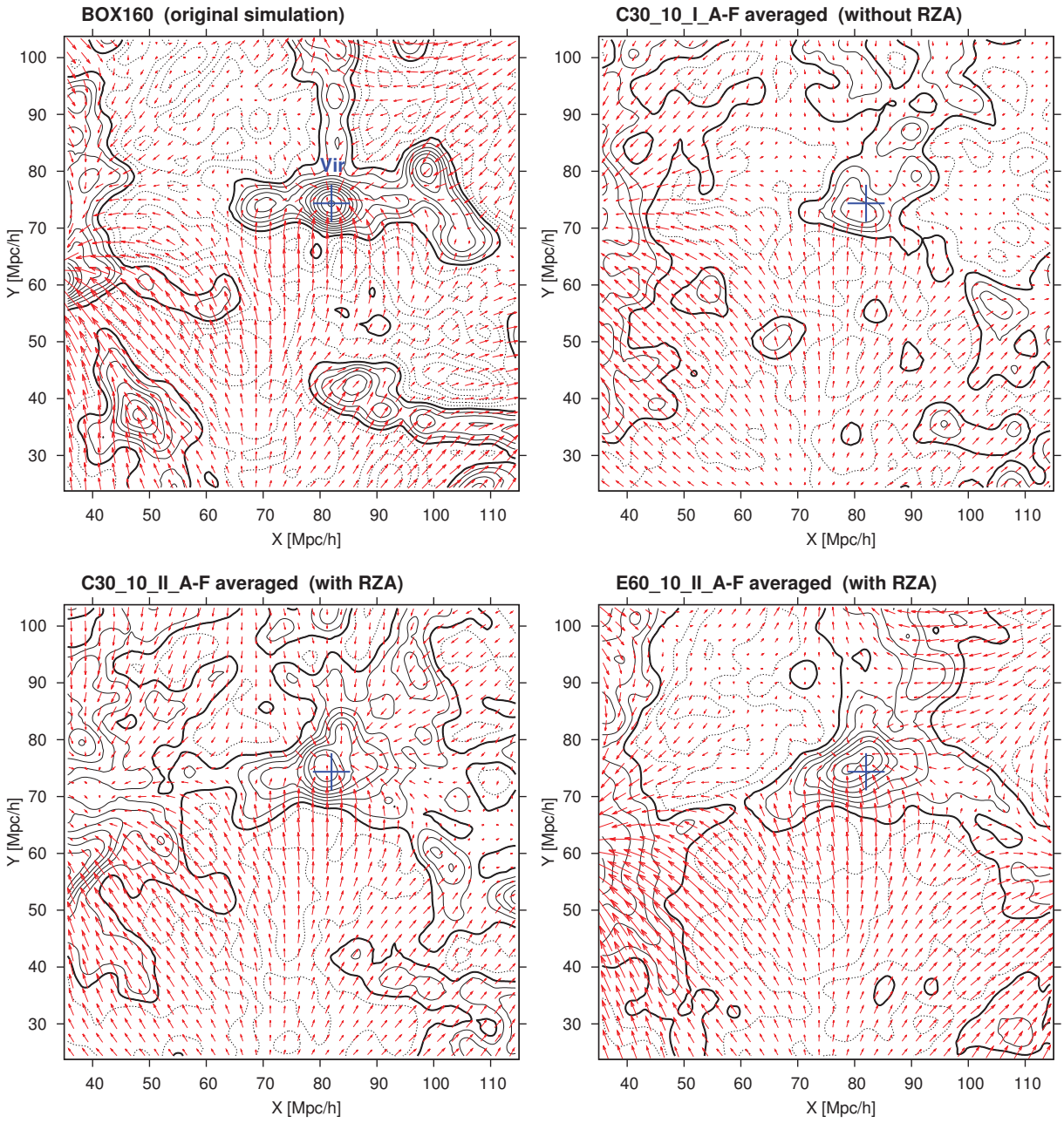

Figure 5.13: Density and velocity fields in a 10 Mpc/h thick slice at $87 < Z < 97$ Mpc/h for the original BOX160 simulation (top left) and the average of all six constrained realisations without RZA (top right), with RZA (bottom left), and with RZA from the E60_10 mock using a larger data volume (bottom right). The density fields were smoothed with a Gaussian of radius 2.5 Mpc/h. The original centre of the BOX160's Virgo cluster is marked with a blue cross in each map. The thick contour line marks the cosmic mean density; solid contour lines are drawn for overdensities and dotted contour lines for underdensities.

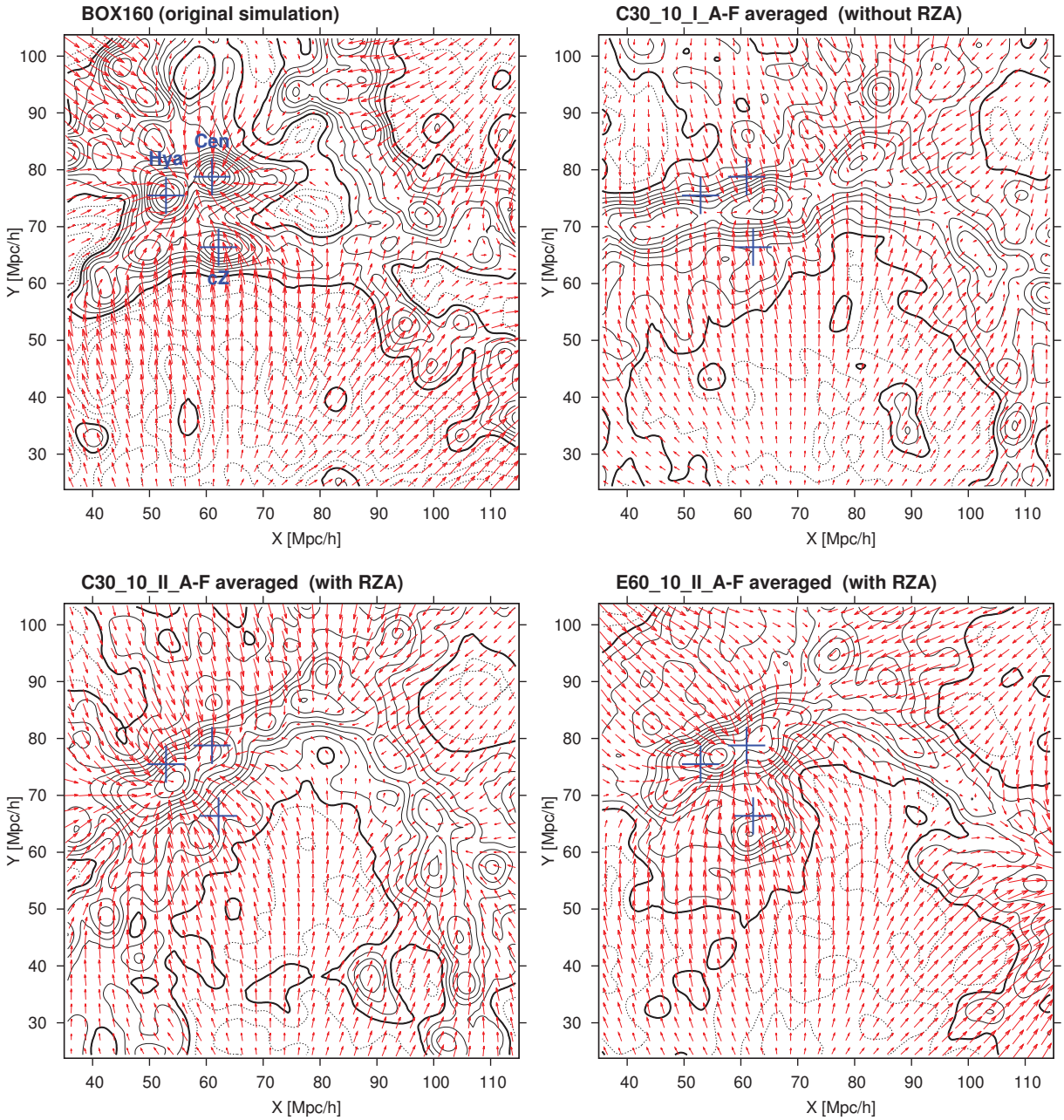

Figure 5.14: Same as Figure 5.13, but for a different slice at $69 < Z < 79$ Mpc$/h$. In this slice, the BOX160 contains three massive clusters: Centaurus (Cen) with virial mass $6.06 \times 10^{14} M_{\odot}/h$, Hydra (Hya) with $5.20 \times 10^{14} M_{\odot}/h$, and "Cluster Z" (cZ) with $0.96 \times 10^{14} M_{\odot}/h$. The original centres of these three objects in BOX160 are marked with blue crosses in each map.

Designation	Position X, Y, Z [Mpc/h]	pec. vel. v_x, v_y, v_z [km/s]	Mass [$10^{14} M_\odot$]
Centaurus (Cen)	$61.0, 78.8, 71.5$	$-1, -136, +187$	6.06
Hydra (Hya)	$52.9, 75.5, 77.0$	$+335, +4, -88$	5.20
Virgo (Vir)	$82.0, 74.4, 91.8$	$+22, +142, -473$	3.25
Fornax (For)	$97.8, 58.3, 62.6$	$+377, +215, +140$	2.41
Cluster X (cX)	$70.7, 73.7, 64.0$	$-169, +290, +353$	1.55
Cluster Y (cY)	$69.1, 68.0, 58.6$	$+225, +417, +470$	1.25
Cluster Z (cZ)	$62.2, 66.4, 74.0$	$-198, +676, -74$	0.96

Table 5.3: Seven most massive clusters in the BOX160 simulation inside the volume within 30 Mpc/h of the mock observer.

a massive overdensity in this region. However in the Method I resimulations there is a clear systematic shift in position. Further, in the averaged map one can see only one smeared out peak instead of two, which is additionally shifted towards positive X and negative Y. In two of the six C30_10_I realisations (B and E), we can find only one massive cluster around the region where the Hydra-Centaurus pair should be. In these realisations, either the two overdense regions merged during their evolution due to shifts in their position and/or displacement, or already in the reconstructed initial conditions the accuracy was not sufficient to robustly resolve them into two separate peaks. On the other hand, in the RZA resimulations, both clusters are resolved robustly and show up as separate peaks that are very close to their intended position; in every realisation using Method II, we can find appropriate objects for Hydra and Centaurus at $z = 0$ within a few Mpc/h of their original positions, just like for the Virgo cluster. Like in the case of Virgo, the masses of the resimulated Hydra/Centaurus clusters at $z = 0$ show both a scatter and a systematic deviation from the original masses of the original BOX160 clusters of $6.06 \times 10^{14} M_\odot/h$ and $5.20 \times 10^{14} M_\odot/h$, respectively. The average mass of the resimulated Centaurus is at 90% of the original mass in BOX160, with a scatter within a factor of two. Interestingly, the resimulated Hydra cluster is more massive than it should be, at 173% of the original mass on average. In three out of the six C30_10_II realisations it is the more massive of the pair, although it should be the less massive, and in four out of six it even has a mass slightly above $10^{15} M_\odot/h$. As it is in the case of the Virgo cluster, this systematic error in the cluster masses does not improve if one goes from the C30_10_II realisations to the E60_10_II realisations.

If we add more constraints and increase the data volume, the mass scale that can be reproduced with the RZA method increases noticeably. For the C30_10_II simulations, we can always find a Virgo cluster, but already the next massive cluster (which we call Fornax here, although we do not imply any connection to the actually observed object) with $2.41 \times 10^{14} M_\odot/h$ cannot be unambiguously identified in one of the six realisations, the C30_10_II_E. Namely, within a distance of 10 Mpc/h and a mass within a factor of 3 there is no matching object in this realisation. The next-massive clusters cX, cY, and cZ cannot be reliably found anymore. On the other hand, if we go to the E60_10_II simulations, we find resimulated counterparts for all seven clusters in Table 5.3 in all six realisations, all within a factor of 3 in mass and within 7

another example of the systematic shifts that occur in constrained simulations from peculiar velocity data if one does not use Lagrangian reconstruction.

Mpc/h in distance. The lightest of these, cluster cZ at $0.96 \times 10^{14} M_\odot/h$, appears as a clear peak in the averaged map of E60_10_II realisations (Figure 5.14, bottom right) close to its original position. For even lighter objects, it is not possible anymore to find unambiguous counterparts in the E60_10_II realisations. This is aggravated by the fact that below a certain mass the probability of a seemingly matching, but in fact randomly created object to appear around the right position increases.

Generally speaking, the predictions of the analysis in Section 5.1.3 can be confirmed in the evolved resimulations at $z = 0$. We find that with the previous method of generating constrained initial conditions, if we use a sparse mock limited to 30 Mpc/h and do not employ a Lagrangian reconstruction scheme, the threshold for robustly recovering structures is around roughly $5 \times 10^{14} M_\odot/h$, and the accuracy on their positions is typically at the scale of 10 Mpc/h. Adding the RZA reconstruction, we can lower this threshold to under $3 \times 10^{14} M_\odot/h$, enough to robustly recover the BOX160 Virgo cluster. If we use the more complete E60_10 dataset, which covers a larger volume, the quality of reconstruction without using RZA does not increase much; but if we use RZA, we achieve a significantly better resolution that allows us to robustly recover structure down to mass scales of about $1 \times 10^{14} M_\odot/h$. Additionally, the RZA method reduces the errors on the positions where the resimulated structures appear to a scatter within about 5 Mpc/h around the intended position, which is also the order of the RZA displacement error d^{RZA}.

The origin of the systematic errors on the reproduced objects' masses can be understood from the non-linearity of the structure formation process. The typical mass scatter within a factor of two that we observe here is consistent with the findings of Ludlow & Porciani (2011), who studied the relation between virialised haloes and their protohalo peaks in the initial conditions. They found that a protohalo peak on some fixed scale determines the mass of the resulting halo only within a factor of two. This explains the cluster mass discrepancies in our constrained resimulations. The exact virial mass of an object at redshift $z = 0$ is a product of various non-linear structure formation processes, such as at what rate it can accrete mass from the surrounding structure and how efficiently it is fed from outside by connected filaments. Such details of the structure around clusters cannot be recovered from reconstructed linear initial conditions. In the mass function of the 30 Mpc/h data zone (Figure 5.11) we see that BOX160 has a very specific distribution: two objects (Hydra and Centaurus) in the highest-mass bin, one object (Virgo) in the next bin, and no objects in the bin after that. The average mass function of the constrained realisations does not follow this peculiar shape, but instead tends towards a distribution that is statistically more likely to occur. But we can argue here that if the large-scale velocity field is recovered accurately by the constrained resimulations, the exact virial mass of a particular cluster plays a lesser role and does not substantially limit the overall meaningfulness of the constrained simulation. Indeed, the peculiar velocity field is recovered exceptionally well in the RZA resimulations, especially for the better E60_10 mock. We already saw that in this case the deviation of the resimulations from the original BOX160 peculiar velocity field is only on the level of 1/4 of the total standard deviation of the field. Comparing in Figure 5.12 the peculiar velocity field of the original BOX160 (upper left) and the E60_10_II resimulation (bottom right), the structure of these two velocity fields is practically identical. If we imagine that a Local Group-like object would be placed at the centre of the E60_10_II resimulation in Figure 5.12, then it would experience practically the same large-scale flow and would be embedded in a very similar environment compared to the original BOX160. This is precisely the main motivation of running constrained simulations, where of course the reference universe is the actually observed Local Universe. We expect that the same degree of accuracy seen here can be obtained by applying the same technique of RZA reconstruction and subsequent CR resimulation to observational data.

To obtain simulations with this degree of accuracy from current observational data is the main focus for future work on the method. These simulations will present a significant methodical improvement over the currently available CLUES simulations and will provide an ideal numerical laboratory for studying the dynamics of the Local Universe.

5.2.4 Filaments and voids

Apart from massive dark matter haloes, another characteristic of the large-scale distribution of matter is the presence of filaments and voids. An example in the selected BOX160 data zone is the filament parallel to the Y-axis, which goes upward from Virgo towards the BOX160's Coma cluster (outside of the map); it is not recovered consistently in the RZA resimulations. We saw already in the RZA reconstruction of the displacement field (cf. Figure 4.12, which contains the same slice shown in Figures 4.12 and 5.13), that the RZA reconstruction struggles to recover such filamentary features accurately. This could be due to the non-linear structure formation in such regions: as we saw in Section 2.2.7, structure formation first proceeds along one dimension, forming sheets, followed by a collapse in the next dimension, forming filaments, and finally the material in these filaments falls into haloes. The peculiar velocity field at $z = 0$ only retains information about this last stage. In the particular case of the filament that connects the Virgo cluster to the Coma cluster in BOX160, the input data contains only the infall of objects along the filament toward Virgo, but not the initial velocity distribution that led to the formation of that filament, and therefore the WF/CR cannot recover it. In the case of the C30_10_II_A resimulation, this filament is instead replaced by another filament that is imposed by the random modes and aligned differently. This behaviour is repeatedly seen in the resimulations at other locations as well. At the other end of the filament there is a flow towards the Coma cluster seen at the upper edge of the slice in Figure 5.12 (around $X = 80$; $Y > 100$). This flow is missed by all resimulations that use the C30_10 mock, since the BOX160 Coma cluster is too far away from the data zone. On the other hand, in the E60_10_II resimulations, this flow is recovered accurately because of the enlarged data zone, but this is not sufficient to also faithfully reproduce the alignment of the filament. We therefore cannot generally trust that the alignment of filamentary structure in the Local Universe can be sufficently reproduced in constrained simulations, except if their formation is strongly constrained by massive objects inside the data zone, as is the case for the thick filament hosting the Virgo cluster. The presence of the latter is generally reproduced in all resimulations with RZA, although again its exact alignment is unconstrained.

We did not compare the alignment of voids in the resimulations in detail, but we see that the presence and alignment of the most prominent voids in the data zone is generally recovered well. The largest voids develop by expansion of underdensities in the initial conditions that are above the minimum scale of initial conditions reconstruction and are therefore sufficiently constrained. The reconstruction of the displacement field using RZA helps track the expansion of voids back in time and generate appropriate initial conditions for them. Since it is believed that in the Local Universe, the outward push caused by the Local Void has an important influence on shaping the Local Flow (Tully et al. 2008), it would be interesting to study in detail to what extent this behaviour can be seen in the constrained simulations. This will be the subject of future work.

Chapter 6

Summary, conclusions and outlook

In this concluding chapter, we summarise the methodology of reconstructing initial conditions and performing constrained simulations from radial peculiar velocity data with the RZA technique that was developed in this thesis and the test study that was carried out based on mock peculiar velocity catalogues drawn from a test simulation. We present a final discussion and interpretation of the obtained results. We also give an outlook on the planned follow-up studies and possible further developments. In this context, we propose a new idea how to reproduce Local Group-like objects in constrained simulations of the Local Universe, as an attempt to overcome the limitation that the corresponding mass scales are unconstrained by the data and not recoverable by a reconstruction of the initial conditions.

6.1 Summary

6.1.1 Motivation

The observed peculiar velocity field of galaxies in the Local Universe and the peculiar velocity of our own Milky Way galaxy hold fascinating puzzles about the large-scale structure and dynamics of the Local Universe. These peculiar velocities can be derived from combining the observed galaxy redshifts with galaxy distance measurements such as the Tully-Fisher method. They provide a direct and unbiased tracer of the underlying total (i.e. dark and baryonic) matter density and the resulting large-scale gravitational potential. From observations of the cosmic microwave background dipole it is now well established that our Local Group has a large peculiar motion of 630 km/s with respect to the rest frame of the large-scale structure (Fixsen et al. 1996). The origin of this motion is however the subject of a lively debate. Several components on different scales are believed to shape this configuration (Tully et al. 2008): the interaction of the Milky Way with the Andromeda galaxy, the infall towards the nearby Virgo cluster, the push from the Local Void, the flows toward the Great Attractor and the Perseus-Pisces clusters, and the possible influence of more distant objects like the massive Shapley concentration. The discrepancy between these large-scale flows and the observed CMB dipole may pose a challenge to the widely accepted ΛCDM model (Lavaux et al. 2010). The formation and evolution of our own Galaxy seems to be influenced to a high degree by this specific configuration of its large-scale environment. It is now more enlightening than ever to study these mysteries, since both the available observations and numerical modelling such as the powerful technique of N-body simulations have tremendously increased in accuracy over the last years.

6.1.2 Constrained Local Universe simulations

In this thesis, we focussed on constrained simulations of the Local Universe in the framework of the concordance ΛCDM model of cosmic structure formation. The constrained simulations technique enables to run numerical N-body simulations of large-scale structure formation that are distinguished from the usual random-realisation cosmological simulations by reproducing the observed large-scale structure of our Local Universe. This is accomplished via the constrained realisations (CR) technique based on the algorithm of Hoffman & Ribak (1991). This method allows us to create Gaussian random fields that are subject to a cosmological prior model given by the standard ΛCDM cosmology, and at the same time adhere to a set of predefined constraints. Such random fields describe the primordial density fluctuations of the early Universe and can be used for simulation initial conditions. In order to reconstruct the initial Gaussian random field underlying our own Local Universe, the constraints have to be derived from observational data. Due to their much higher linearity and large-scale correlation compared to tracers of the density field (such as galaxy redshift surveys), peculiar velocities of galaxies provide a very powerful source of such constraints, linking together the observations and the numerical modelling.

Zaroubi et al. (1995, 1999) have extended the formalism of the CR method to incorporate such constraints on the peculiar velocity field. For the study presented in this thesis, we developed the numerical software package ICeCoRe. This code is designed around a highly optimised implementation of their algorithms and is able to handle the upcoming large observational peculiar velocity datasets from the Cosmicflows program, which will provide tens of thousands of constraints (Courtois et al. 2011a,b; Courtois & Tully 2012; Tully & Courtois 2012). To our knowledge, so far no constrained simulations have been conducted with such a high number of constraints. The ICeCoRe code can also be used for the related Wiener filter reconstruction from peculiar velocity data, which is a useful cosmographical tool to reconstruct the underlying three-dimensional density and peculiar velocity fields of the data, filtering out noise and extrapolating the data into unsampled regions.

The current technique of generating constrained initial conditions from peculiar velocities with the Hoffman-Ribak algorithm is formulated in the linear theory of Gaussian random fields. We found that the peculiar velocity field at $z = 0$ resembles much more closely the primordial velocity field of the initial conditions, both statistically and topographically, than the density field at $z = 0$ would resemble the primordial overdensity fluctuations. Even if traced only with dark matter haloes (or galaxies that inhabit them), that follow only the high-density peaks of the matter distribution, the peculiar velocity field at $z = 0$ is remarkably closer to the primordial linear Gaussian statistics than the density field. The velocity field at $z = 0$ is however not strictly Gaussian. Effects like the cosmic displacement field, which shifted the field on scales of ≈ 10 Mpc$/h$ from the initial conditions, the non-linear enhancement of velocities in high-density regions, and non-linear virial motions introduce significant systematic errors if treated with the simple linear theory model. As a result, previous constrained simulations could robustly reproduce only structure on the largest scales, such as clusters with masses not much smaller than $10^{15} M_{\odot}/h$, and they appear shifted compared to the original configuration. The simulations fail to recover structure on smaller scales, which is completely dominated by random modes. In this thesis, we investigated in what ways this approach can be improved to yield more accurate constrained simulations. This led to the development of the Reverse Zeldovich Approximation (RZA) reconstruction method.

6.1.3 RZA reconstruction

We studied the peculiar velocities of dark matter haloes in a cosmological simulation under the assumption that the observable peculiar motions of galaxies follow the velocities of the dark matter haloes in which they reside. We found that present-day ($z = 0$) peculiar velocities of galaxies are a surprisingly accurate tracer of the cosmological displacement field $\boldsymbol{\psi}$, which connects present-day positions $\boldsymbol{x}$ of observable objects to their Lagrangian coordinates $\boldsymbol{q}$ in the primordial state via $\boldsymbol{x} = \boldsymbol{q}+\boldsymbol{\psi}$ and therefore allows for a reconstruction of the cosmological initial conditions. A simple yet powerful approximation in this context is the Zeldovich approximation (Zeldovich 1970), which assumes that all tracers of the cosmic density field move on straight paths and therefore their peculiar velocity $\boldsymbol{u}$ is at all times directly proportional to the displacement field via $\boldsymbol{u} = \dot{a} f \boldsymbol{\psi}$. It is surprising how well this relatively simple approximation holds even at $z = 0$, if the Zeldovich model is directly applied to peculiar velocities rather than to the density field, as previous studies did. We find that for the majority of haloes that are not gravitationally bound to larger objects, $\boldsymbol{u}$ and $\boldsymbol{\psi}$ at $z = 0$ are almost perfectly aligned with an angular deviation of less than 10°. The approximation holds quite well throughout most of the volume and is especially accurate for isolated objects found in less dense environments. However, the approximation fails in high-density regions that are dominated by virial motions and accretion onto massive clusters. This can be alleviated by an adequate grouping or selection of data points. The most effective way of doing so for massive clusters is not to consider their internal structure at all, but reduce them to a single peculiar velocity data point. In order to reconstruct the initial conditions at some early redshift z_{init} in the linear regime, we then reverse the Zeldovich equation and estimate the Lagrangian positions $\boldsymbol{q} \approx \boldsymbol{x}_{\mathrm{init}}$ via

$$\boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}} = \boldsymbol{r} - \frac{\boldsymbol{v}}{H_0 f} \quad , \tag{6.1}$$

where $\boldsymbol{r}$ and $\boldsymbol{v}$ are the position and peculiar velocity of an object observed at $z = 0$, respectively. We quantify the error on this guess via the RZA error d^{RZA}, which is the distance between the estimated initial position $\boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}}$ and the original initial position $\boldsymbol{x}_{\mathrm{init}}$ of an object's centre-of-mass in the initial conditions of the simulation at $z = 30$,

$$d^{\mathrm{RZA}} = |\boldsymbol{x}_{\mathrm{init}} - \boldsymbol{x}_{\mathrm{init}}^{\mathrm{RZA}}| \quad . \tag{6.2}$$

We find that if we straightforwardly apply the approximation to the peculiar velocities of haloes identified in the simulation at $z = 0$, the median d^{RZA} is at only 1.36 Mpc/h and the mean d^{RZA} at 2.3 Mpc/h. We therefore obtain a highly accurate estimate of the initial conditions, especially when compared to the linear-theory approach of neglecting $\boldsymbol{\psi}$, in which case the typical error on the initial position is ≈ 10 Mpc/h. A great advantage of the RZA is that it is a completely local approximation that can be carried out on any sparse and inhomogeneous sampling of the peculiar velocities. No information about the complete underlying field or its statistical properties is needed. In contrast, Lagrangian reconstruction schemes based on information about the density field instead of the velocity field, such as galaxy redshift surveys, do not work well with sparse and inhomogeneous samplings. They always need an estimate of the complete field, together with some way to compensate for redshift distortions, galaxy bias and the strong non-linearities of the density field at $z = 0$.

6.1.4 RZA on radial peculiar velocity data

In order to apply the RZA to observational data, we need to take into account that it traces only the radial component of the galaxy peculiar velocities and is corrupted by significant observational

errors. In this case, the full three-dimensional peculiar velocity vectors can be reconstructed with the Wiener filter method. We test this procedure on mock catalogues based on the dark matter halo catalogue extracted at $z = 0$ from a test simulation. We construct an ensemble of such mock catalogues, varying the observational errors, data volume, and degree of incompleteness. We then obtain an estimate of their three-dimensional peculiar velocity $\boldsymbol{v}$ with the Wiener Filter and apply the RZA reconstruction to this data to recover the initial positions $\boldsymbol{x}_{\rm init}$ of the haloes. We find that for a typical sparse peculiar velocity catalogue, with statistical distance errors of 10% and a data volume limited to a distance of $R_{\rm max} = 30$ Mpc$/h$ within the observer, the median $d^{\rm RZA}$ is around 4 – 5 Mpc$/h$, which is still a factor of 2 – 3 better than with no Lagrangian reconstruction. The situation is much better in less dense regions where the reconstruction works exceptionally well even with the realistic mock data. The quality of the reconstruction can be increased if we reduce the observational errors, and even more if we instead increase the amount of datapoints. This is interesting in the context of the upcoming observational data in the Cosmicflows-2 catalogue, which will significantly increase the amount of data points, but only to a lesser degree decrease the observational distance errors. We also find that a better reconstruction is obtained with a more homogeneous sample of data points, providing a more complete mapping of the volume, and the reconstruction quality worsens if the data points are biased towards the most massive objects and therefore preferentially located in high-density regions. This translates to the statement that observational distance measurement methods selecting galaxies in a more random fashion and not biased towards high-density environments provide ideal input data for RZA. This is the case for peculiar velocities obtained with the Tully-Fisher method, which selects spiral galaxies on the random basis of their inclination on the sky being greater than 45 degrees, and ignores elliptical galaxies that are strongly biased towards high-density regions. While future observational data will surely allow for a significant improvement, we also acknowledge that even with very sparse and noisy data (a few hundred datapoints within 30 Mpc$/h$) it is possible to obtain a reasonable reconstruction quality.

6.1.5 Constrained simulations

We now go on and use the RZA reconstructed displacements and initial positions to construct constrained initial conditions. We use for this test two of the mock catalogues generated for the RZA reconstruction study. One mock is very sparse and limited to $R_{\rm max} = 30$ Mpc$/h$, containing only 588 radial peculiar velocity datapoints, while the second is more complete and reaches out to $R_{\rm max} = 60$ Mpc$/h$ with 7637 datapoints, similar in quality to the upcoming new observational data from the Cosmicflows program. The underlying simulation is the BOX160 simulation, which is itself a constrained realisation of the Local Universe and therefore features a large-scale cosmography that is similar to the observed one. The aim of this test is to reconstruct the initial conditions of BOX160, to integrate them forward until $z = 0$ using an N-body code, and then to compare the evolved re-simulation with the original BOX160 simulation at $z = 0$. The advantage of such a test is that the obtained results can be directly compared with the "true" solution, allowing to quantify the precision of the reconstruction and the obtained re-simulations. On the other hand, testing with observational data would make this impossible, because the actual underlying density and velocity fields are not known. For each mock and each reconstruction method that we tried, we ran six such re-simulations of BOX160 with different seeds for the random components.

We find that the optimal method to place constraints on the initial conditions is to take the radial velocity datapoints directly from the catalogue at $z = 0$ and to displace them "backwards in time" to their RZA-reconstructed initial position $\boldsymbol{x}_{\rm init}$ at initial redshift $z_{\rm init}$. This is compared against the previous method discussed in Klypin et al. (2003); Gottlöber et al. (2010), which

uses no such displacement of the data, but is identical otherwise. We find that the non-RZA method is able to generate initial conditions that are clearly correlated with the original ones, with a scatter in the initial velocity field of about 2/3 of its total σ (standard deviation) at initial redshift $z_{\rm init} = 30$. Interestingly, the non-RZA reconstructed initial conditions do not improve significantly if we use the better mock instead, because the reconstruction quality is limited by the large systematic errors of the method. By contrast, using our new method, the RZA-reconstructed initial velocity field has a scatter of about $1/2\ \sigma$ with the poor mock and improves considerably to a scatter of only $1/4\ \sigma$ with the better mock. The scatter is similar if we instead compare the evolved fields at $z = 0$ of the re-simulations with the original simulation.

We analysed the evolved re-simulations at $z = 0$ and compared them to the cosmography of the original BOX160 simulation at $z = 0$ inside the "Local Universe" subvolume that was constrained. We find that, regardless of the used mock, the non-RZA method is able to robustly recover only objects on mass scales above $\approx 5 \times 10^{14} M_\odot$, while most structure below that threshold is dominated by the random component. This scale corresponds to the couple of most massive clusters in the local subvolume. Additionally, the positions of these clusters are subject to large systematic shifts of ≈ 10 Mpc$/h$ and more compared to their original locations in BOX160. This is about the same accuracy that is known from previous attempts of running constrained simulations of the Local Universe, such as the BOX160 itself. By contrast, the RZA re-simulations perform significantly better, robustly recovering all objects above masses of $3 \times 10^{14} M_\odot$ (for the poor mock) and $1 \times 10^{14} M_\odot$ (for the better mock) in each realisation. The original cluster positions are recovered within ≈ 5 Mpc$/h$, however when averaged over the different realisations this error reduces to only ≈ 2 Mpc$/h$, indicating that the position errors are due to the random component and not due to a systematic shift. We therefore acknowledge that setting up constrained initial conditions with the RZA reconstruction method significantly improves the accuracy of constrained simulations.

In all re-simulations, the virial masses of the clusters are recovered only within a factor of 2. This is the accuracy expected from peak-patch theory: the properties of the progenitor peak seem to determine the mass of a cluster at $z = 0$ only within this range (Ludlow & Porciani 2011), with the rest being mostly determined by non-linear processes. Therefore, we do not expect that this mass accuracy could be significantly improved by any reconstruction method of initial conditions from data at $z = 0$.

6.1.6 Conclusions

As the main finding of this work we conclude that, for the task of generating constrained realisations of the Local Universe from peculiar velocity data, a Lagrangian reconstruction scheme such as the RZA reconstruction presented here provides a significant improvement over treating the peculiar velocities with linear theory, which was the previous approach. We found that directly applying the WF/CR operator to peculiar velocity data at $z = 0$ is actually not a very good estimate of the initial conditions due to the displacements, the non-linearities of the data, and the relatively large systematic errors. However, the WF reconstructed 3D peculiar velocity field seems to be an accurate estimate of the displacement field $\boldsymbol{\psi}$. This in turn allows us to generate a better estimate of the initial conditions, which can be set up in a second WF/CR operator step by using the RZA reconstructed displacements at the discrete data positions for constraints. With this setup, we have to take care not to actually apply the WF iteratively and thus filter away the information contained in the data. We found a way to place constraints on the initial conditions that avoids this problem. A very powerful strategy is to displace all input radial peculiar velocity datapoints to their estimated initial positions $\boldsymbol{x}_{\rm init}^{\rm RZA}$ in the linear phase,

while leaving their observational errors and the spatial component of the velocity vector that they trace unchanged.

It was surprising to find how well the relatively simple Zeldovich approximation performs when applied directly to peculiar velocities, instead of the density field as it is commonly done in various other contexts. Because the Zeldovich approximation performs so well, we did not see a significant overall improvement by employing higher order Lagrangian perturbation theory (2LPT) for the reconstruction. Interestingly, we noticed that 2LPT does not suffer from the filtering bias of the linear theory approach, which is a desirable property; on the other hand, 2LPT introduces higher overall errors to the reconstruction. We can explain this with the fact that any method of higher order than the Zeldovich approximation will break the locality of the approximation, and the whole field has to be considered instead of the given discrete data points. This will invariably introduce additional systematic errors. With RZA, we therefore may have found a near-optimal method of reconstructing initial conditions from peculiar velocities.

For a reconstruction of the three-dimensional galaxy peculiar velocities from the observed radial components, we used the well-established Wiener filter method. We found that the WF performs very well in compensating for the radial limitation and the observational errors, but suffers from a filtering bias due to its conservative nature. To obtain an accurate unbiased reconstruction and sufficiently constrain the cosmological initial conditions, we require as many data points as possible. Since increasing the amount of data, increasing the data volume, and decreasing the observational errors all improve the resulting constrained simulations, we expect that the quality of constrained simulations of the observed Local Universe will significantly improve if the RZA method is applied with the upcoming high-quality observational data.

With the obtained results we can make a point about what kind of peculiar velocity data is optimal for setting up constrained simulations. We think that methods focussing on spiral galaxies, such as the Tully-Fisher method, are a better choice than data covering early-type galaxies such as the fundamental plane and surface brightness fluctuations methods. Spiral galaxies provide a more homogeneous mapping of the sky and are less biased towards high-density regions where non-linear effects become stronger. These regions are exactly where the RZA reconstruction fails. It may also be important to find an optimal data grouping method for filtering out virial motions. In general, the best strategy is to reduce each group of data points that form a virialised or otherwise strongly grativationally interacting structure into a single data point, in order to remove non-linear virial motions completely.

Considering the re-simulations that we obtained from the BOX160 simulation, at first sight the overall accuracy of the constrained realisations method does not seem to be very high if one considers recovered positions and masses of clusters at $z = 0$. On the other hand, and more importantly, the peculiar velocity field – which is the focus of this work – is recovered to a very high degree. We know that the large-scale peculiar velocity field that drives the dynamics of the Universe is largely determined by the most massive observable objects, within the scales that are recovered in constrained simulations. Within the RZA re-simulations that used the better mock, the velocity field within the local subvolume is structurally almost indistinguishable from the original field. This quality makes the method of RZA constrained simulations a perfect numerical laboratory to study the large-scale velocity field of the Local Universe despite its limitations towards smaller scales.

However, we have to remember that although we performed the tests on fairly realistic mock catalogues, the considered "Universe" was itself a simulation. We could therefore not consider the effects of the finite simulation volume and the imposed periodic boundary conditions when estimating the reconstruction quality. We found that these effects unfortunately have a considerable negative impact if one wants to recreate a faithful representation of the observed Local

Universe in a constrained simulation. Therefore we can trust only the result in the innermost part of the box (approximately within a distance of $L/4$ from the box centre, where L is the simulation boxsize), and only if the box is large enough. Despite a self-consistent method of treating those effects when generating constrained initial conditions that we implemented, small boxsizes below 100 Mpc/h will always be subject to large systematic errors. This must be taken into account when constructing constrained realisations on such small boxsizes. Finite volume has a very strong effect on the variance and correlations of the peculiar velocity field in the resulting realisation. This variance is underestimated even on boxsizes of several hundreds of Mpc/h. If the goal is to model the large cosmic flows, such as the flow towards the Shapley concentration which is a subject of the current scientific debate, with constrained simulations, we even may need boxsizes as large as 1000 Mpc/h. At these scales, it will become very interesting to combine the methods presented here with constraints coming from deep galaxy redshift surveys and encompassing very large volumes. Such methods are currently in development (e.g. Kitaura 2012).

6.2 Outlook

We want to give an outlook for further developments of the methodology presented in this work. The obvious next step of the technique developed here is to apply the machinery of RZA reconstruction to produce constrained initial conditions and perform constrained simulations from the most current observational datasets of galaxy peculiar velocities. This is straightforward since all tests presented here were applied on realistic mock radial peculiar velocity catalogues with observational errors, sparseness, incompleteness, and limited data volume. From the results presented here we expect that constrained simulations constructed in this way from observational data will be a significant step forward compared to the current CLUES simulations. It will be interesting to study the cosmography of the Local Universe mapped by this new data not only by reconstructions of the underlying field, but in the framework of constrained simulations that gives access to the dynamics and evolution of the system.

Further studies have to go into improving the procedure of RZA reconstruction and subsequent constraining of the initial condions. It is important to search for optimal ways of grouping or linearisation of the peculiar velocity data. One step would be to explicitly add a theoretical or empirical model that describes the non-linear enhancement of peculiar velocities in overdense regions. We also saw that the WF reconstruction of radial peculiar velocities yields results that are not quite consistent with the assumed Gaussian statistics if the data density is high, and introduces a significant skewness. One could therefore explore methods that result in a better Gaussianisation of the field. Because the local Zeldovich approximation works so well when applied to peculiar velocities, we do not expect that higher-order schemes, which invariably become non-local, will perform significantly better on sparse data, but it still may be interesting to explore them. It may be possible to obtain better results with higher-order LPT, if the WF reconstruction from radial peculiar velocities would be replaced by a reconstruction procedure that would be self-consistent with that higher-order LPT.

One fundamental problem remains with constrained simulations. The linear initial conditions for smaller scales down to the mass scale of our Local Group, i.e. scales of the order of 10^{12} – $10^{13} M_\odot$, seem to be unrecoverable from the data. The total virial mass of the Local Group is of the order of 5×10^{12} $M_\odot$, with the virial mass of the Milky Way being of the order of 2×10^{12} $M_\odot$ (Li & White 2008). Although the tidal field of the Local Group region is constrained well, the scale of structural features in the initial conditions that would lead to the formation of such Local Group-like haloes is quite small and cannot be constrained directly from the data.

The efforts so far indicate that this may be unfeasible even with a significantly better method of reconstructing the initial conditions. If we still want to study the formation and evolution of our Local Group with constrained simulations, we can follow the approach taken before by Gottlöber et al. (2010). Using high-quality observational data and the methods presented here we can construct a large sample with hundreds of realisations of constrained simulations. From such an ensemble of constrained simulations we can then find instances where Local Group-like objects happen to appear at approximately the right location and then study those in high resolution.

However, here we want to point out another approach that we will explore in future studies. The peak-patch theory suggests that a majority of haloes form from regions in the vicinity of peaks in the initial conditions, and that the scale of such a peak determines the mass of the resulting collapsed halo within a factor of 2 or so (Ludlow & Porciani 2011). The primordial peaks that would be associated with the formation of the haloes hosting the Milky Way and Andromeda galaxies are so small that these scales are completely unconstrained by the data in our method. We can turn this fundamental problem into a virtue. By placing appropriate constraints on the initial overdensity field at relatively small scales, we can introduce overdensity peaks at manually selected positions and smoothing scales in the initial conditions. Working on Local Group-like mass scales, corresponding to approximately a 2 Mpc/h Gaussian smoothing scale, we would not disturb the larger scales that are constrained by peculiar velocities. We can then iteratively find the optimal position and smoothing scale for the additional peak constraints such that we recover in the evolved simulation Local Group-like haloes with exactly the desired positions and masses. By construction, the peculiar velocities of these emergent haloes would be determined by the large-scale tidal field that is directly constrained by the data. This would combine two opposed approaches of constraining initial conditions: running Local Universe simulations and creating tailor-made initial conditions with predefined peaks. We will explore such methods in upcoming studies. On the other hand, if the formation of the Local Group is in fact tied to its large-scale environment, it could also turn out that a better quality of constrained simulations of the Local Universe already leads to a higher rate of Local-Group like objects to form in the simulations, making it unnecessary to introduce them manually to the simulations. This could provide valuable clues about whether the Local Group and its large-scale environment are indeed an unusual case in a ΛCDM Universe.

Appendix A

ICECORE User's Guide

ICECORE (**I**nitial **C**onditions **&** **Co**nstrained **Re**alisations) is a command-line program written in C++ and developed for this work. It implements a combination of very efficient algorithms to create constrained (as well as conventional, i.e. unconstrained) initial conditions for cosmological N-body simulations and to compute Wiener filter mean fields from density and/or peculiar velocity data. The concepts and the theoretical framework behind these applications have been discussed in Chapter 3; knowledge of this material is assumed here. What sets ICECORE apart from previous implementations is that it allows us to impose very large numbers of constraints on the linear density and velocity fields (up to $M \approx 10^5$ and possibly more), while at the same time the evaluation of the WF/CR operator is still numerically efficient and feasible to perform on consumer-grade hardware without having to resort to high-performance parallel computing. Beyond this main objective, ICECORE also offers some basic tools to analyse and edit density and velocity fields that may be useful when setting up cosmological initial conditions for N-body simulations. However, ICECORE is primarily designed to *generate* these fields and not to analyse them, for example it contains no visualisation tools. It also focuses on dark-matter-only density and velocity fields given on uniform cubic grids, and contains no functionality towards multi-scale fields (such as for zoom-in initial conditions), multi-phase fields (such as initial conditions for baryonic matter), or manipulating particle data such as N-body simulation snapshots. See Section 3.5 for further information.

ICECORE features a simple command-line driven interactive user interface. This is separated from the underlying C++ API that contains the actual numerical implementation. The interface was designed for ease of use and flexibility. It is possible to make full use of the program without having to deal with the underlying C++ source code. In this User's guide, we discuss how to install and use ICECORE and provide a tutorial on the command-line interface covering most of the available functions. The focus lies on how to generate initial conditions and perform Wiener filtering in practice with ICECORE. A full documentation of the underlying C++ API is beyond the scope of this document (it is available in the form of Doxygen-generated code documentation from the author). The knowledge of the C++ implementation is however not required in order to use ICECORE, as all its functions can be triggered directly from the command-line user interface. As a supplement to this User's guide, the code reference in Appendix B contains a complete documentation of the user interface covering all functions currently available in ICECORE. This Appendix discusses ICECORE version 1.0. The functionality of the code may be changed and expanded in future versions.

ICECORE may be released as an open-source, publicly available code in the future. Until then, please contact the author to obtain your copy of the code. Suggestions for improving the code and adding more features, as well as reports of existing bugs, are also greatly appreciated.

A.1 Getting started

A.1.1 Requirements

ICECORE should install and run in any Unix/Linux environment if all required external libraries are present. It was successfully tested on several Linux distributions and on Mac OS X. ICECORE is written in C++ and therefore requires a C++ compiler to build. We successfully tested the code with both the GNU C++ compiler (`g++`) and the Intel C++ compiler (`icpc`); the latter tends to produce significantly faster code, so we do not recommend to use `g++`. ICECORE makes extensive use of the C++ standard library and the C++ standard template library (STL), which usually come with the compiler. It also includes a parallelisation with OpenMP. Compiling with OpenMP is optional. In addition, ICECORE requires the following non-standard external libraries:

- GNU scientific library (GSL), available at `www.gnu.org/gsl`. GSL functions are used, among other things, for generating Gaussian-distributed random numbers, performing spline interpolation on tabulated functions, numerically evaluating integrals, and computing spherical Bessel functions.
- LAPACK (Linear Algebra PACKage), available at `www.netlib.org/lapack`. This library is used for the inversion of the data autocorrelation matrix via Cholesky decomposition (see Section 3.4.3). The original LAPACK library is written in FORTRAN 90 and somewhat intricate to link against from a C/C++ compiler; CLAPACK could be used instead. On Linux, we obtained best results with the LAPACK implementation contained in the MKL (Intel Math Kernel Library). Of all implementations that we tested, it is the easiest to link against and by far the fastest (it also includes parallelisation), therefore we strongly recommend to use MKL for compiling and running ICECORE. On Mac OS X (as of version 10.7), a LAPACK implementation comes preinstalled with Apple's Accelerate framework, which is an integral part of the operating system.
- FFTW3, available at `www.fftw.org`. This is used for performing discrete Fourier transforms on the density and velocity grids, which are required for many functions of ICECORE. FFTW3 must be compiled in double precision, i.e. without the `--enable-float` option.
- GNU readline library. This is optional but highly recommended if you use ICECORE in interactive mode. It enables several convenient features on the ICECORE command line, such as command history and editing with the arrow keys and tab completion on file names, mimicking the behaviour of other Unix-compatible command-line interpreters such as `bash` or `python`. In most Linux distributions, the GNU readline library is either preinstalled or available through the default package manager. It can also be installed from source, which is available at `www.gnu.org/software/readline`.

A.1.2 Install and run

The ICECORE source directory contains a `Makefile` compatible with GNU make. In order to compile ICECORE, most probably you will have to modify the `Makefile`. Currently it contains predefined link lines for Linux, with or without OpenMP (assuming that the MKL is available), and for Mac OS X (without OpenMP). In order to use them, uncomment the line `SYSTEM = ...` that suits your configuration. In the Linux/MKL case, you will also have to specify the locations of the libraries by setting the `FFTW3PATH` and `MKLPATH` correctly, if they are not in one of your standard `PATH` directories. For Mac OS X, everything should work out of the box if

you specify `SYSTEM = "OSX"`. If you use another configuration, especially in the case of another LAPACK implementation, you will have to write your own link line. The `Makefile` also offers some additional compilation options. They can be triggered by (un)commenting and/or editing the lines starting with `CXXFLAGS += ...` setting the compiler flags. The predefined options are optimal in most cases, so change them only if you need to. Details are given in the `Makefile`.

In order to compile ICeCoRe with GNU make, go into the ICeCoRe source directory and type `make`. This will generate an executable binary called `icecore` in the same directory. In order to run ICeCoRe, simply copy this executable to your working directory and there type `./icecore`. A copy of the `icecore` executable should be kept in every directory where you intend to run it; ICeCoRe currently cannot access files outside of the directory where it is running. If you use OpenMP, you may also want to set the `OMP_THREADS` environment variable before running ICeCoRe to specify the number of threads that should be used. If you change the compiler options, run `make clean` in the source directory before re-compiling to remove the pre-compiled object files and the old executable.

A.1.3 The user interface

There are two ways to run ICeCoRe, interactive mode and script mode. Launching the ICeCoRe executable with `./icecore` will start the interactive mode. This offers a command-line interface where the user can enter commands which are then executed. All functions of ICeCoRe are accessed by entering such commands. Appendix B includes a list all available commands in alphabetic order.

The script mode is started by launching ICeCoRe with the filename of a script file as an argument, for example `./icecore my_scriptfile.ic` . Then, ICeCoRe will go through the script line by line, execute the commands specified there, and then quit when the end of the script file is reached. Also, in script mode ICeCoRe will immediately quit if it encounters an error, while in interactive mode it will merely display an error message but the session can be continued. In interactive mode, the session is terminated by entering the command `quit`.

The command language of ICeCoRe is case sensitive, with all command names written in lowercase. Generally, there must be always one command per line, i.e. commands must be separated by a newline. Redundant newline, whitespace, and tab characters will be ignored. After a hash character (`#`), the rest of the line will be ignored, which can be used to place comments into script files. If a line starts with an exclamation mark (`!`), the following command will be passed to the system. In this way, it is possible to execute external Unix commands from within ICeCoRe. For example, with the command `!ls` one can display all files in the current working directory without leaving an interactive ICeCoRe session. One could also move, copy or delete files in this way. This can also be used in ICeCoRe scripts.

The `help` command gives access to the built-in help and code reference. If entered with no arguments, `help` will display some general info about ICeCoRe and provide references to other help topics. `help commands` will display a list of all available commands. Giving the name of a command as an argument, `help` will display a short description of the command along with an explanation of its options, arguments, and their syntax.

A.2 Basic usage

A.2.1 Density and velocity fields

The data inside ICeCoRe is organised into *objects*, which are responsible to manage the different fields, vectors, matrices and tables in computer memory that are needed for calculations. Most

of the ICECORE commands operate on one of these objects. The central objects are `dgrid` and `vgrid`, which hold the overdensity field $\delta(\boldsymbol{x})$ and peculiar velocity field $\boldsymbol{u}(\boldsymbol{x})$, respectively. The `dgrid` object holds the field $\delta(\boldsymbol{x})$ sampled on a three-dimensional, uniform cubic grid with boxsize L, given in Mpc/h, and resolution n, which is the number of grid cells per dimension. The total number of grid cells is $N = n^3$. The `vgrid` object holds three such grids, one for each cartesian component of $\boldsymbol{u}(\boldsymbol{x})$, i.e. $u_x(\boldsymbol{x})$, $u_y(\boldsymbol{x})$, and $u_z(\boldsymbol{x})$. A new `dgrid` object is allocated with the command `new dgrid` *`resolution boxsize`*. We use the convention to write arguments in *`italic typewriter`* font which are tokens and have to be replaced by some actual name or value. For example, a new density grid with boxsize $L = 160$ Mpc/h and resolution $N = 256^3$ is allocated with the command

```
new dgrid 256 160
```

which will not only allocate the grid, but also initialise the values of all its cells to zero. In the same way, `new vgrid 256 160` allocates and initialises to zero a peculiar velocity field with matching boxsize and resolution. These grids can be manipulated with several different ICECORE commands, and written to or read from files. ICECORE supports several different file formats for these grids. The simplest is `binary`, which simply dumps the grid (for `dgrid`) or the three grids (for `vgrid`) as arrays of N^3 binary 4-byte floats; the produced files will have a size of $4 \cdot N^3$ bytes. A more advanced format is `bov`, which corresponds to the BOV (Brick Of Values) format also used by the visualisation software VISIT[23]. In BOV format, each grid consists of a data file (`*.bov.data`) in the same binary format and an accompanying human-readable header file (`*.bov`), which lists several parameters like the boxsize and the resolution. Another format is the GRAFIC format, which is a binary float array file similar to `binary`, but containing FORTRAN-style record markers and a header record with parameters. This format comes in two flavours: `grafic` for cosmological fields and `graficwn` for white noise fields. Using the `ascii` format, the grids can also be dumped as a list of N^3 human-readable values in a text file, although such files are rather inefficient, because they are slower to read and write and can become very large compared to binary files. Appendix B.5 provides an overview over all currently supported formats. Grids can be written to files with the `write` command, where the second argument is the object that will be written, the third argument is the format, and the fourth is the filename:

```
write dgrid bov my_file.bov
write dgrid binary my_file.dat
write dgrid grafic my_file.dat
write dgrid graficwn my_file.dat
write dgrid ascii my_file.txt
...
```

Loading a grid from a file is done in the same way with the `load` command:

```
load dgrid bov my_file.bov
...
```

The velocity grid `vgrid` is stored in three such files, one for each component. Therefore, writing and loading them requires to supply three files:

```
write vgrid bov vx.bov vy.bov vz.bov
...
```

[23] available at visit.llnl.gov .

```
load vgrid bov vx.bov vy.bov vz.bov
...
```

The ICeCoRe code assumes that for all operations on the density and velocity fields the linear theory of Gaussian random fields is valid. In this framework, the linear displacement field $\boldsymbol{\psi}(\boldsymbol{x})$ and velocity field $\boldsymbol{u}(\boldsymbol{x})$ are proportional to each other, $\boldsymbol{u} = \dot{a}f\boldsymbol{\psi}$. For this reason, ICeCoRe consistently uses internal units of Mpc/h. The peculiar velocity field in ICeCoRe is always given in terms of the displacement field, i.e. in units of Mpc/h. These internal units have the advantage that no operation explicitly depends on the cosmology, i.e. on the factor $\dot{a}f$. Before using a peculiar velocity field given in km/s in ICeCoRe, it has to be converted to units of Mpc/h by multiplying all values by the factor $1/\dot{a}f$. The numerical value of this factor can be accessed by the predefined token `kmstompch`. These predefined tokens can be used wherever a number is expected. The conversion of a velocity grid from km/s to Mpc/h can be accomplished with

```
multiply vgrid kmstompch
```

where `multiply vgrid` *`factor`* is a command that can be used to multiply the `vgrid` with any number *`factor`*. Velocity fields that are computed within ICeCoRe will always be in units of Mpc/h. The conversion to km/s can be used in the same way with

```
multiply vgrid mpchtokms
```

where `mpchtokms` is a predefined token for the numerical value of $\dot{a}f$.

A.2.2 Cosmology and power spectrum

Besides the density and velocity fields, several other objects exists. The `cosmology` object holds the cosmological parameters Ω_m and Ω_Λ, the Hubble constant H_0, the current redshift z, and all parameters that are derived from them and needed in the calculations: the scale factor a, the expansion rate $\dot{a}$, the linear growth factor D_+ and the linear growth rate f. The cosmology object is initialised with the command

```
new cosmology omega_m omega_L H_0 z
```

where numbers have to be given for Ω_m, Ω_Λ, H_0, and z. For generating initial conditions, z will be the starting redshift of the simulation z_init, while for Wiener filter reconstruction of the large-scale structure from observational data one takes $z = 0$. For example, a WMAP7 cosmology for generating initial conditions at $z_\mathrm{init} = 30$ is initialised with

```
new cosmology 0.272 0.728 70.2 30
```

The value of H_0 is not actually used in any calculations, since the internal units of Mpc/h do not depend on it. However, the value of H_0 is needed if one wants to write cosmological initial conditions in the Grafic and Gadget-2 formats: both include a header with the cosmological parameters. It is therefore also possible to initialise the `cosmology` object by directly reading the parameters from such files, instead of manually entering them. This is done with `load cosmology grafic` *`file`* and `load cosmology gadget` *`file`* , respectively.

Another object required for most calculations in ICeCoRe is the `pk` object. It holds the tabulated power spectrum $P(k)$ that describes the assumed cosmological prior model, and computes values of $P(k)$ by interpolating over this table. ICeCoRe does not contain code to compute $P(k)$ from cosmological parameters; this has to be done with other software such as Cmbfast, Camb or iCosmo (see Section 2.2.4). The tabulated $P(k)$ can then be loaded with the command

```
load pk file
```

The input format consists of a whitespace or tab-separated ASCII table with two columns, k (in units of h/Mpc) and $P(k)$, *with no additional lines in the file*. This is the same format that the CAMB code uses for output.

Before use, the $P(k)$ has to be normalised to a normalisation parameter σ_8. This is done with the `normalise` command and the desired σ_8 as an argument, for example

```
normalise 0.807
```

for a WMAP7 cosmology. This command will normalise the tabulated power spectrum in the `pk` object such that the integral in equation 2.58 has the given σ_8 value at $z = 0$. Additionally, if the redshift specified in `cosmology` is $z > 0$, then according to equation 2.57 $P(k)$ is scaled with the square of the linear growth factor. The integral in equation 2.58 is by default evaluated over the whole range of wavenumbers k where $P(k)$ is tabulated. As discussed in Section 3.1.3, there are other methods of normalising $P(k)$ which take into account the finite-volume effect and use equation 3.9 instead. This can be performed in ICECORE with additional arguments to `normalise`:

```
normalise 0.807 kl kny    # use CLUES convention (e.g.\ BOX160)
normalise 0.807 kl2 kny   # use Klypin & Holtzman (1997) convention
```

The two additional arguments specify the interval $[\,k_{\min}, k_{\max}\,]$ that defines the integration range in equation 2.58. `normalise 0.807 kl kny` causes the integral to be evaluated from $k_{\min} = k_L$, the fundamental frequency of the box, to $k_{\max} = k_{\mathrm{Ny}}$, the Nyquist frequency. `normalise 0.807 kl2 kny` option instead uses $k_{\min} = k_L/\sqrt{2}$, where `kl2` is a token for the value $k_L/\sqrt{2}$. The involved frequencies k_L and k_{Ny} are evaluated on the basis of the boxsize and resolution of the currently loaded `dgrid`.

Since the `cosmology` and `pk` objects and the correct normalisation of the latter are required for most computations in ICECORE, especially everything that has to do with generating initial conditions, these objects should always be set up at the beginning of an ICECORE script/session.

A.2.3 Generating initial conditions for cosmological N-body simulations

With the objects introduced so far, generating a random realisation of cosmological initial conditions can be done with just a few commands, following the algorithm laid out in Section 3.1.1. The cosmology, boxsize L, resolution n, and starting redshift z_{init} are all determined by initialising the `cosmology`, `pk` and `dgrid` objects. Then, the first step is to create a Gaussian white noise field $w(\boldsymbol{x})$ (equation 3.2) on `dgrid`. This is done with the command

```
seed number
```

where *`number`* is an integer used as the seed. With this seed, ICECORE will initialise a random number generator and then fill `dgrid` with zero-mean, unity-variance Gaussian-distributed random numbers. The initial overdensity field $\delta(\boldsymbol{x})$ is then generated with the command

```
colour
```

This command first performs a forward FFT, then evaluates equation 3.3, using the `pk` object to determine $P(k)$ for every $\boldsymbol{k}$ on the grid, and then performs a backward FFT. The resulting $\delta(\boldsymbol{x})$ is stored in `dgrid`, overwriting the white noise field. ICECORE will also check the variance of $\delta(\boldsymbol{x})$, which for cosmological initial conditions should lie approximately in the range $0.1 < \langle|\delta|\rangle < 0.2$. If this is not the case, a warning will be displayed recommending another choice for z_{init}.

The linear displacement field $\boldsymbol{\psi}(\boldsymbol{x})$ that corresponds to $\delta(\boldsymbol{x})$ can be obtained with

```
vsolve
```

which will evaluate equation 3.5 for `dgrid` and store the resulting comoving linear displacement field in `vgrid` in units of Mpc/h. At this point, the initial field of the random realisation is defined and the further proceeding depends on what you want to do: directly start an N-body simulation or just write the grids to files and/or do something else with them. The grids can be written to files at any time in any supported format using the `write dgrid` and `write vgrid` commands. To start a simulation from the created random realisation with the RAMSES code, it is sufficient to write them in GRAFIC format with specific predefined filenames:

```
multiply vgrid mpchtokms    # convert displacement to velocity for RAMSES!
write dgrid grafic ic_deltab
write vgrid grafic ic_velcx ic_velcy ic_velcz
```

and then to supply these four files as input. RAMSES will then itself set up particle initial conditions. Section B.5.3 provides more information about the GRAFIC format.

To start a simulation with GADGET, you have to set up those particle initial conditions yourself and supply them in a GADGET-format snapshot file. This can be done within ICECORE with the method discussed in Section 3.1.2. The whole procedure is encapsulated in the single command

```
writeics gadget file
```

which will set up the N-body particles and write the initial conditions to *`file`*. It is important that for this command the displacement field must be kept in units of Mpc/h and *not* converted to velocity in km/s. The code will print out warnings if something goes wrong. For example, if the unit of `vgrid` is wrong or the starting redshift $z_{\rm init}$ was chosen too low, shell crossing can occur on the initial particle distribution, and no valid particle initial conditions can be generated.

Below is a complete ICECORE example script that generates random initial conditions for a WMAP7 cosmology with a resolution of $N = 256^3$, a boxsize of $L = 100$ Mpc/h, and a starting redshift of $z_{\rm init} = 30$, and writes them into a file in GADGET format that can be directly used to start a simulation.

```
# Example 1 - generating random cosmological ICs for GADGET-2

new cosmology 0.272 0.728 70.2 30  # compute cosmology from parameters
load pk WMAP7.dat                  # load tabulated power spectrum
normalise pk 0.807                 # normalise to sigma8
new dgrid 256 100                  # alloc grid with N=256^3, L=100 Mpc/h
seed 12345                         # generate white noise field
colour                             # compute deltaRR
vsolve                             # compute linear displacement field
writeics gadget ics.gadget         # write Gadget-2 initial conditions
quit
```

For the overdensity and displacement/velocity grids stored in the `dgrid` and `vgrid` objects, ICECORE provides a few basic analysis tools. With the commands `stat dgrid` and `stat vgrid`, it is possible to check the most important statistical properties of the grids, such as the minimum and maximum values and the first four statistical moments. The command `binpk` computes a periodogram estimate of the power spectrum $P(k)$ of the current `dgrid`, and the command `histogram` computes the distribution function of `dgrid` and `vgrid`. The `xcorr` command computes the Fourier-space cross-correlation (equation 2.85) between the currently loaded `dgrid` and

a reference overdensity field that will be read from a file. See the code reference in Appendix B for details on these commands. For more advanced analysis and especially for visualisation, other tools have to be used, for example the VISIT software, which can directly open grids saved in BOV format, or IDL, which can handle both binary and ASCII files. Most of the plots in this thesis were generated with GNUPLOT[24], which understands binary data as well.

A.2.4 Computing correlators

For constrained realisations and Wiener filtering it is first necessary to initialise the `correlator` object. It holds the precomputed correlation functions ξ, ψ_R, ψ_T and ζ needed to compute the data autocorrelation matrix $\langle c_i c_j \rangle$ and to evaluate the WF/CR operator. This object is responsible for computing all necessary correlation values when needed, which it does by interpolation on the precomputed functions.

There are two types of correlators: `analytic` and `grid`. They have been discussed in Sections 3.4.1 and 3.4.2. The `analytic` correlator computes and stores one-dimensional correlation functions as tables over the comoving distance x and the smoothing radius R_G, and computes correlation values with GSL cubic spline interpolation. The `grid` correlator instead computes three-dimensional correlation functions with FFTs on grids, stores them, and computes correlation values with trilinear grid interpolation. Initialising the correlator, i.e. precomputing these correlation functions, can be performed with the command `new correlator analytic`, or `new correlator grid`, respectively. This requires the `pk` object to be present and correctly normalised to the assumed σ_8 and z. It also requires a `dgrid` object, since the correlator may depend on the boxsize L and resolution n. In the case of the analytic correlator, the `new correlator` command accepts some additional options. The general syntax is:

```
new correlator analytic [ k_min k_max [ R_G_max ] ]
```

where square brackets denote optional arguments. By default, the integration range for the integrals 3.46 – 3.49 goes over the whole k range of the tabulated $P(k)$. With ***k_min*** and ***k_max*** one can specify other upper and lower bounds by giving numerical values in units of h/Mpc, which of course have to be within the tabulated range. Instead of numerical values, it is also possible to specify `kl` for ***k_min***, which corresponds to the fundamental mode $k_L = 2\pi/L$ of the current `dgrid`, and `kny` for ***k_max***, which is the Nyquist frequency[25] $k_\mathrm{Ny} = \pi/\Delta x$. The `kl` option can be used in order to explicitly take into account the large-scale cutoff by the finite-volume effect of the computational box on the correlation functions. The tokens `kminpk` and `kmaxpk` stand for the default option, the tabulated range of $P(k)$. The optional ***R_G_max*** argument specifies up to what smoothing radius R_G (in Mpc/h) the tables should be computed. By default this will be the maximum value that can occur for the loaded data/constraints, or $R_\mathrm{G} = 0$ if no constraints were loaded. For example,

```
new correlator analytic kminpk kmaxpk 10
```

will compute analytic correlation functions using the full tabulated range of $P(k)$ for the integration interval, and tabulate them for smoothing radii R_G between 0 and 10 Mpc/h. When you

[24] available at www.gnuplot.info .

[25] It is however not recommended to limit the analytic correlator integration to frequencies below k_Ny if you want to take into account the finite grid resolution. The cut at some k_max will invariably result in Gibbs ringing artefacts on the small-x end of the correlation functions, which should be kept well below the grid resolution to avoid numerical errors (i.e. k_max should be larger than at least $\approx 5\,k_\mathrm{Ny}$). If you want to accurately describe the effects of the grid discretisation on the level of k_Ny with correlation functions, use the grid correlator instead.

determine the required R_G, note that the correlation function of two constraints c_1 and c_2 will be evaluated at $R_\mathrm{G} = \sqrt{R_{\mathrm{G},1}^2 + R_{\mathrm{G},2}^2}$.

Both types of correlators can also be backed up to files and then loaded from these files later. The command `write correlator analytic` will write four files to the current working directory, containing the precomputed tables: `xi.dat`, `psiR.dat`, `psiT.dat`, and `zeta.dat`. The command `write correlator grid` will instead write four grids with similar names in BOV format. The corresponding commands `load correlator analytic` and `load correlator grid` will then search for files with exactly these names and initialise the `correlator` object from them. While the grid correlator can be computed very fast, with only four FFTs, the procedure of backing up the correlator makes more sense for the analytic correlator. Especially if the tables go over a wide range of smoothing radii R_G, computing all integrals can take considerable time. It would be a waste of time to repeat that same computation over and over again in each ICeCoRe session. On the other hand, if you use only data with $R_\mathrm{G} = 0$, computing the analytic correlator is in general also very fast, and backing it up may not be worth the effort. Using a backed up correlator also carries the risk that it may not be compatible with the loaded data/constraints and boxsize L. This is not checked by the code; it is the user's responsibility to keep all objects consistent with each other.

A.2.5 Placing constraints

In order to compute a Wiener filter reconstruction or a constrained realisation from a set of data, or *constraints*, they have to be prepared in the correct format and loaded into ICeCoRe. Internally they are managed via the `constraints` object, which is simply a list of all constraints. This is initialised by loading a data file:

```
load constraints file
```

We use the convention to add the `.co` extension to data files in the ICeCoRe constraints format, but this is not required. The file must consist of an ASCII table, one line per constraint, with ten whitespace or tab-separated columns. The value of each column corresponds to the quantities listed in Table 3.1: the constraint type Y_i, the constraint position (x_i, y_i, z_i), the constraint value c_i, the estimated constraint error ε_i, the direction vector $(\hat{e}_{\mu,x}, \hat{e}_{\mu,y}, \hat{e}_{\mu,z})$ and the Gaussian smoothing radius R_G. The constraint type is encoded with an integer ($\delta \equiv 1$, $\psi_\mu \equiv 2$). The other fields contain their numerical values as real numbers. If the position of the constraints is given in terms of distance, latitude and longitude, it first has to be converted to cartesian coordinates (equations 3.31) to yield the constraint positions $\boldsymbol{x}_i = (x_i, y_i, z_i)$ for columns 2 – 4. For type 2 constraints, the value c_i and error ε_i must be in units of Mpc/h (displacement). If the original data file uses units of km/s (velocity), the loaded constraints can be converted using the command

```
multiply constraints kmstompch
```

which will multiply all c_i and ε_i with $1/\dot{a}f$. If the data uses mixed constraint types (density and velocity), one should apply the above multiplication to the type 2 constraints only with

```
multiply constraints2 kmstompch
```

If using radial peculiar velocity data, care must be taken that ε_i (column 6) actually contains the estimated *absolute* error on the velocity/displacement, and *not* the relative error or the error on the distance. Also, with radial peculiar velocity data, the values of columns 7 – 9 (the $\hat{\boldsymbol{e}}_\mu$ vector) must be set to the cartesian components of the distance vector from the observer to the

datapoint in units of Mpc/h. If the observer is at $\boldsymbol{r}_0 = (0,0,0)$ in the original data, then columns 7-9 will simply contain the same values as columns 2 – 4. The $\hat{\boldsymbol{e}}_\mu$ vector must be normalised to $|\hat{\boldsymbol{e}}_\mu| = 1$; this is required to evaluate equations 3.29 and 3.30. However this does not have to be done for the datafile by the user, because columns 7 – 9 will be normalised anyway by ICECORE automatically by the `load constraints` command.

Another limitation with the `constraints` object is that all constraints must lie within the computational volume, i.e. $0 \leq x_i, y_i, z_i \leq L$. In the case of radial peculiar velocity data, if the observer is at $\boldsymbol{r}_0 = (0,0,0)$ initially, all data positions have to be shifted such that they lie fully inside the box, for example to $\boldsymbol{r}_0 = (\frac{L}{2}, \frac{L}{2}, \frac{L}{2})$ in order to be aligned around the centre of the box. This can be done with the `shift constraints` command. For example, to shift such data to the centre of a box with $L = 120$ Mpc/h, the command is

```
shift constraints vector 60 60 60
```

which will add the vector $\boldsymbol{r}_0 = (60, 60, 60)$ to the positions of constraints $\boldsymbol{x}_i = (x_i, y_i, z_i)$. Note that the vector $\hat{\boldsymbol{e}}_\mu$ in columns 7 – 9 is not shifted and now points along the radial direction with respect to $\boldsymbol{r}_0 = (60, 60, 60)$. An example line in a constraint file could then be like this:

```
2     63     64     60    400     50      3      4      0      0
```

In a box with $L = 120$ Mpc/h with the observer at $\boldsymbol{r}_0 = (60, 60, 60)$, this places a radial velocity constraint (type 2) at a distance of $\boldsymbol{r} - \boldsymbol{r}_0 = (3, 4, 0)$, absolute distance $r = |\boldsymbol{r}| = 5$ Mpc/h with a value of $c_i = 400$ km/s. The absolute velocity error is $\varepsilon_i = 50$ km/s, corresponding to an estimated relative distance error of $\delta r = 10\%$. There is no smoothing attached to the constraint, $R_\mathrm{G} = 0$. It will then be necessary to convert this file from km/s to Mpc/h before running `cr` or `crv`. As another example,

```
1     60     60     90      9      0      0      0      0      7
```

places an overdensity constraint on the position $\boldsymbol{r} = (60, 60, 90)$, or $\boldsymbol{r} - \boldsymbol{r}_0 = (0, 0, 30)$ with respect to the box centre, constraining the Gaussian smoothed density with $R_\mathrm{G} = 7$ Mpc/h to a value of 9. In this case, the values in columns 7 – 9 are irrelevant. Note that the overdensity field scales with the linear growth factor, $\delta \propto D_+$, so the numerical value of such a constraint must be chosen carefully.

Besides shifting the constraint positions by a constant position vector with `shift constraints vector`, there is also the possibility to shift them by a vector field with `shift constraints vgrid`; this will become interesting for RZA in Section A.3.1. Other commands manipulating the constraints are: `addvector`, which adds a constant vector to velocity/displacement-type constraint values and is useful for frame-of-reference conversions; `deoverlap`, which detects and removes constraints with $R_\mathrm{G} > 0$ and overlapping smoothing volumes; and `sparse`, which detects and removes constraints that are closer to each other than some critical distance. See the code reference in Appendix B for details. An edited set of constraints can be saved to a new file with the command `write constraints` *`file`*.

A.2.6 Wiener filter and constrained realisations

Using the `constraints` and `correlator` objects, it is possible to compute Wiener filter (WF) reconstructions and constrained realisations (CR). The command

```
cr sigma_nl
```

computes $\delta^{\mathrm{CR}}(\boldsymbol{x})$, the overdensity field of a CR defined by equation 3.38, with the non-linearity parameter σ_{NL} (equation 3.45) given as an argument. The `cr` command assumes that the required random realisation is currently loaded, i.e. `dgrid` contains $\delta^{\mathrm{RR}}(\boldsymbol{x})$ and `vgrid` contains $\boldsymbol{u}^{\mathrm{RR}}(\boldsymbol{x})$. The latter must be in units of Mpc/h, so it actually will contain $\boldsymbol{\psi}^{\mathrm{RR}}(\boldsymbol{x})$. The `cr` command will then trigger a series of computations. First, the data autocorrelation matrix $\langle c_i c_j \rangle$ will be computed. Then, the chosen σ_{NL} will be added to its diagonal and the matrix will be inverted with Cholesky decomposition to yield $\langle c_i c_j \rangle^{-1}$. Then, using lookups on `dgrid` and `vgrid`, the set of mock constraints $\tilde{c}_i$ is computed. The next step consists of computing the correlation vector η_i (equation 3.39). Finally, the WF/CR operator (equation 3.40) is applied to every cell of `dgrid`. As a result, `dgrid` will be overwritten with $\delta^{\mathrm{CR}}(\boldsymbol{x})$.

The `cr` command is also used to compute $\delta^{\mathrm{WF}}(\boldsymbol{x})$. The only difference is that to compute the WF instead of a CR, `dgrid` and `vgrid` should contain only zeroes. This can be achieved by calling `new dgrid` and `new vgrid` immediately before `cr`. Then, both the mock constraints $\tilde{c}_i$ and the δ^{RR} term will be zero, and the equations 3.39 and 3.40 will reduce to 3.26 and 3.27, respectively. The result saved in `dgrid` at the end of the computation will then be $\delta^{\mathrm{WF}}(\boldsymbol{x})$ instead of $\delta^{\mathrm{CR}}(\boldsymbol{x})$.

Explicitly giving a σ_{NL} parameter is optional; one could also just type `cr` without additional arguments. Then the code will try to find a value for σ_{NL} such that $\chi^2/\mathrm{dof} \approx 1$, where $\mathrm{dof} = M$ is the number of constraints. This involves several iterative matrix inversions and can take a very long time if the matrix is large. It is therefore recommended to always provide a *`sigma_nl`* parameter. In order not to repeat this step in each run that uses the same data, it is also possible to pre-compute the inverted data autocorrelation matrix with a defined σ_{NL}; this will be explained in Section A.3.3.

The following example script illustrates the complete procedure of computing a $\delta^{\mathrm{WF}}(\boldsymbol{x})$ field with ICECORE. It uses a WMAP7 cosmology and a box of $L = 100$ Mpc/h with resolution $N = 256^3$. By using the analytic correlator it assumes an infinite, homogeneous and isotropic Universe. It loads a datafile `data.co` containing constraints, carries out the computation of $\delta^{\mathrm{WF}}(\boldsymbol{x})$, and writes it to a file in BOV format.

```
# Example 2 - computing a WF mean field

new cosmology 0.272 0.728 70.2 0   # compute cosmology from parameters
load pk WMAP7.dat                  # load tabulated power spectrum
normalise pk 0.807                 # normalise to sigma8
new dgrid 256 100                  # alloc grids with N=256^3, L=100 Mpc/h
new vgrid 256 100
load constraints data.co           # load constraints (all of type 2)
multiply constraints kmstompch     # convert from km/s to Mpc/h
shift constraints vector 50 50 50  # shift to box centre
new correlator analytic            # generate correlator object
cr 3                               # compute deltaWF with sigma_NL=3 Mpc/h
write d bov delta_wf.bov           # write deltaWF to file
quit
```

If you want to obtain the WF mean field of the peculiar velocity, $\boldsymbol{u}^{\mathrm{WF}}(\boldsymbol{x})$, there are several possibilities depending on what exactly you want to do. In principle there are two possibilities: either to obtain $\boldsymbol{u}^{\mathrm{WF}}(\boldsymbol{x})$ by FFTs from $\delta^{\mathrm{WF}}(\boldsymbol{x})$ using equation 3.5, or directly computing it with the WF/CR operator using equation 3.41. The same options exists for a CR, where $\boldsymbol{u}^{\mathrm{CR}}(\boldsymbol{x})$ is determined either through $\delta^{\mathrm{CR}}(\boldsymbol{x})$ and equation 3.5, or directly through equation 3.41. Although

these two options seem to be equivalent in Gaussian linear theory, they are not. The FFT approach, which can be easily performed by running the `vsolve` command on $\delta^{\mathrm{WF}}(\boldsymbol{x})$ or $\delta^{\mathrm{CR}}(\boldsymbol{x})$, assumes periodic boundary conditions (PBCs) on the box, and the resulting $\boldsymbol{u}^{\mathrm{WF}}(\boldsymbol{x})$ or $\boldsymbol{u}^{\mathrm{CR}}(\boldsymbol{x})$ will be periodic. The direct evaluation of the WF/CR operator for the velocity field, on the other hand, depends on the correlator. The analytic correlator assumes an infinite, homogeneous and isotropic Universe instead of PBCs, and the resulting $\boldsymbol{u}^{\mathrm{WF}}(\boldsymbol{x})$ or $\boldsymbol{u}^{\mathrm{CR}}(\boldsymbol{x})$ will not be periodic. The ramifications of this difference have been discussed in Sections 3.3.3 and 3.4.2.

Solving for $\boldsymbol{u}^{\mathrm{CR}}(\boldsymbol{x})$ by explicitly evaluating the WF/CR operator in equation 3.41 can be accomplished with the command

```
crv sigma_nl
```

This command behaves in the same way as `cr`, but additionally it also evaluates equation 3.41 for each grid cell of `vgrid`. This means that at the end not only `dgrid` will contain $\delta^{\mathrm{WF}}(\boldsymbol{x})$ or $\delta^{\mathrm{CR}}(\boldsymbol{x})$, but also `vgrid` will contain $\boldsymbol{u}^{\mathrm{WF}}(\boldsymbol{x})$ or $\boldsymbol{u}^{\mathrm{CR}}(\boldsymbol{x})$. The latter will be again in units of Mpc/h, so it is actually the linear displacement field $\boldsymbol{\psi}^{\mathrm{WF}}(\boldsymbol{x})$ or $\boldsymbol{\psi}^{\mathrm{CR}}(\boldsymbol{x})$. Whether it is WF or CR again depends only on the state of `dgrid` and `vgrid` before calling `crv`. Running `crv` instead of `cr` will be many times slower, because for equation 3.41 many more correlation function values have to be evaluated. To carry out the procedure, the script example 2 will have to be changed to

```
...
crv 3                                          # compute deltaWF and uWF
multiply vgrid mpchtokms                       # convert uWF to km/s
write d bov delta_wf.bov                       # write deltaWF to file
write v bov vx_wf.bov vy_wf.bov  vz_wf.bov     # write uWF to files
quit                                           # - this will have no PBCs!
```

Typically the `crv` method will be used if one wants to obtain a WF reconstruction of the LSS for cosmographic studies, where an infinite, homogeneous and isotropic Universe is assumed and no PBCs are desired on the fields.

On the other hand, the FFT approach of obtaining $\boldsymbol{u}^{\mathrm{WF}}(\boldsymbol{x})$ will use the `cr` and `vsolve` commands:

```
...
cr 3                                           # compute deltaWF
vsolve                                         # solve uWF assuming PBCs
multiply vgrid mpchtokms                       # convert uWF to km/s
write d bov delta_wf.bov                       # write deltaWF to file
write v bov vx_wf.bov vy_wf.bov  vz_wf.bov     # write uWF to files
quit
```

This will be typically used if one wants to obtain constrained initial conditions, which must have a displacement field with PBCs. The complete procedure of first generating a random realisation, then load the constraints, and then generate a CR from that, is illustrated in the following script:

```
# Example 3 - generating a CR

new cosmology 0.272 0.728 70.2 0   # compute cosmology from parameters
load pk WMAP7.dat                  # load tabulated power spectrum
normalise pk 0.807                 # normalise to sigma8
```

```
new dgrid 256 100                  # alloc grid with N=256^3, L=100 Mpc/h
seed 12345                         # generate white noise field
colour                             # compute deltaRR
vsolve                             # solve uRR needed for mock constraints
new correlator grid                # generate correlator object
load constraints data.co           # load constraints
multiply constraints kmstompch     # convert from km/s to Mpc/h
shift constraints vector 50 50 50  # shift to box centre
cr 3                               # compute deltaCR with sigma_NL=3 Mpc/h
vsolve                             # solve uCR assuming PBCs
multiply vgrid mpchtokms           # convert uCR to km/s (if you want)
write dgrid bov delta_cr.bov                      # write deltaCR to file
write vgrid bov vx_cr.bov vy_cr.bov  vz_cr.bov    # write uCR to files
quit
```

The computations in this script are all carried out at $z = 0$. This is a useful normalisation to analyse the resulting CR. However, cosmological initial conditions have to be generated at the starting redshift $z_{\rm init}$ of the simulation, which must be chosen such that the overdensity perturbations are sufficiently close to the linear regime at all scales represented in the box. There are two possibilities: either the CR is rescaled to $z_{\rm init}$ after it has been generated, and before writing ICs for a simulation, or the constrained ICs are generated at $z_{\rm init}$ in the first place. In the case of peculiar velocity data, the first option is more useful for several reasons: the constraints are usually normalised to $z = 0$, as is the $\sigma_{\rm NL}$ parameter, and the $\boldsymbol{\psi}^{\rm CR}(\boldsymbol{x})$ field necessary for RZA has to be normalised to $z = 0$ as well. We followed this approach in this work, and for this reason all the scatter plots in Chapter 5 illustrating the generated constrained ICs use the $z = 0$ normalisation. Rescaling an overdensity field $\delta(\boldsymbol{x})$ from $z = 0$ to some other z can be done later with the command

```
multiply dgrid growthd
```

where `growthd` is the numerical value of the linear growth factor D_+ that corresponds to the currently loaded values of the `cosmology` object. By definition, `growthd` has the value 1 at $z = 0$. The following script illustrates the process of loading a $\delta^{\rm CR}(\boldsymbol{x})$ field from a file, rescaling it from $z = 0$ to $z_{\rm init} = 30$, and then generating GADGET ICs from it:

```
# Example 4 - rescale CR to z_init and then generate ICs

new cosmology 0.272 0.728 70.2 30     # this time we use z=30 for ICs!
load dgrid bov delta_cr.bov           # read deltaCR (at z=0) from file
multiply dgrid growthd                # renormalise deltaCR to z=30
vsolve                                # solve for uCR at z=30
writeics gadget ics_cr.gadget         # write constrained ICs for GADGET
quit
```

The alternative approach of generating $\delta^{\rm CR}(\boldsymbol{x})$ directly at $z_{\rm init} = 30$ can be done by loading a `cosmology` with $z = 30$ at the beginning and then to rescale the constraints themselves to $z_{\rm init} = 30$ before calling `cr`:

```
new cosmology 0.272 0.728 70.2 30   # working at z=30
...
load constraints data.co            # load constraints
```

```
multiply constraints2 kmstompch     # convert from km/s to Mpc/h
multiply constraints growthd        # scale from z=0 to to z=30
shift constraints vector 50 50 50
cr                                  # now sigma_nl will have changed!
...
```

Since in linear theory, $\boldsymbol{\psi}(\boldsymbol{x})$ scales with D_+ just like $\delta(\boldsymbol{x})$, `multiply constraints growthd` will rescale both type 1 and type 2 constraints from $z = 0$ to $z = 30$ (remember that type 2 constraints have to be in units of the displacement $\boldsymbol{\psi}$, i.e. Mpc/h). However, at least for velocity-type constraints drawn from $z = 0$ data, we do not recommend to run `cr` at redshifts other than $z = 0$ for the reasons mentioned above. Always having the CR normalised to $z = 0$ is also more flexible: it can be later re-normalised to any other redshift without ambiguity. All manipulations on the CR, such as changing the boxsize and/or resolution (see Section A.4), can be done at $z = 0$, and rescaling to $z_{\rm init}$ can be performed as the last step immediately before writing the actual simulation ICs.

A.3 Advanced usage

A.3.1 Reverse Zeldovich approximation

In this section, we discuss how setting up ICs from radial peculiar velocity data with the RZA reconstruction (Chapter 4) can be performed in ICECORE. To set up ICs, we use Method II discussed in Section 5.1. The first step is to generate an ordinary WF mean field with the `cr` command from the data normalised at $z = 0$, i.e. a $\delta^{\rm WF}$ field, with the technique illustrated in Example 2. The data should be well within the box, ideally within $\frac{L}{4}$ of the box centre, to avoid problems due to the inconsistency of the data with the PBCs. The box should be also large enough to reduce finite-volume effects. We can then obtain $\boldsymbol{\psi}^{\rm WF}$ from $\delta^{\rm WF}$ by running `vsolve`. The RZA method rests on the assumption that $\boldsymbol{\psi}^{\rm WF} \equiv \boldsymbol{\psi}^{\rm RZA}$ is an estimate of the cosmic displacement field $\boldsymbol{\psi}$ from early redshift $z_{\rm init}$ in the linear regime to $z = 0$. We saw in Section 4.4.7 that in this case one should use $\sigma_{\rm NL} = 0$ for the *`sigma_nl`* parameter of `cr` to reduce the WF filtering bias. To generate ICs with RZA, we now have to displace all constraints from their positions $\boldsymbol{r}_i$ at $z = 0$ to their estimated comoving positions $\boldsymbol{x}_i^{\rm init}$ at $z_{\rm init}$,

$$\boldsymbol{x}_i^{\rm init} = \boldsymbol{r}_i - \boldsymbol{\psi}^{\rm WF}(\boldsymbol{r}_i) \quad . \tag{A.1}$$

In order to shift the positions of the constraints with a vector field, ICECORE provides the command

```
shift constraints vgrid
```

where the field is assumed to be loaded into the `vgrid` object. This command will go through the list of constraints in the `constraints` object. For each constraint, it will take its position $\boldsymbol{x}_i$ (equivalent to $\boldsymbol{r}_i$ at $z = 0$), lookup/interpolate the value of `vgrid` at this position, and then add this vector to $\boldsymbol{x}_i$. The other quantities defining the constraint are left unchanged. Since we want to subtract $\boldsymbol{\psi}^{\rm WF}(\boldsymbol{r}_i)$ from $\boldsymbol{r}_i$, the procedure consists of first loading the previously computed $\delta^{\rm WF}(\boldsymbol{x})$ into `vgrid`, then compute $\boldsymbol{\psi}^{\rm WF}(\boldsymbol{x})$, then multiply it with -1 to reverse the sign, and then run `shift constraints vgrid`. If you already ran the WF-generating script in Example 2, this whole procedure will be performed like this:

```
# Example 5 - generating RZA constraints

new cosmology 0.272 0.728 70.2 0    # RZA must be done at z=0 !
load constraints data.co            # load the original constraints again
multiply constraints kmstompch      # convert from km/s to Mpc/h
shift constraints vector 50 50 50   # shift to box centre
load dgrid bov delta_wf.bov         # load deltaWF with L=100 Mpc/h
vsolve                              # compute psiWF == psiRZA
multiply vgrid -1                   # psiRZA -> -psiRZA
shift constraints vgrid             # perform RZA: xRZA_i = x_i - psi_RZA
write constraints rza.co            # write RZA constraints to file
quit
```

This must be done with everything normalised to $z = 0$. The RZA shifted set of constraints in `rza.co` can then be used to run the WF/CR operator again to produce a CR (like in Example 2), this time using $\sigma_{\rm NL}$ such that $\chi^2/{\rm dof} \approx 1$. The new $\delta^{\rm CR}(\boldsymbol{x})$ can then be scaled to $z_{\rm init}$ to generate ICs, or processed further if required, such as whitening it and using the resulting white noise field to set up high-resolution ICs with another code. As we saw in Chapter 5, this is the ideal method to set up ICs for constrained simulations from radial peculiar velocity data.

A.3.2 Display and delete objects

As already illustrated, the ICECORE user interface is organised around a set of objects. These can be written to files, initialised from files, or created within an ICECORE session. The current state of a session is completely defined by the state of its objects. For an overview, the `stat` command displays information for all objects, whether they are currently initialised or not, and some basic parameters for each one. It is possible to display more detailed information about each object by giving its name as an argument. For example, `stat cosmology` will display the loaded cosmological parameters, the current redshift z, and the derived quantities a, $\dot{a}$, D_+, f, and t. `stat pk` will display the tabulated range and current σ_8 normalisation of the loaded power spectrum $P(k)$. `stat dgrid` will display some statistical properties of the current overdensity field, such as the minimum and maximum values and the first four statistical moments, and `stat vgrid` will do the same for the velocity/displacement field. With the `reset` command, it is also possible to manually delete objects and free their previously allocated memory; for example `reset dgrid` will delete and deallocate the overdensity grid. If called without arguments, `reset` will delete all currently loaded objects. This "cleans up" the session and leaves it in the same state as if it would be terminated and a new one started.

A.3.3 Splitting the WF/CR procedure

There are three ICECORE objects not yet discussed here: `corrmatrix`, `invmatrix`, and `eta`. If you only work with small sets of constraints, you may never need them, but they come in handy if running many WF/CR computations with large sets of constraints ($M > 10^4$). The `cr` command performs a sequence of several computational steps in order to compute the WF/CR from the constraints and the correlator. If working many times with the same set of constraints, it may be inefficient to repeat all those steps for each run of `cr`. The additional objects help to reduce some redundancy.

If called like in the previous examples, the `cr` command first computes the data autocorrelation matrix $\langle c_i c_j \rangle$, then inverts it to obtain $\langle c_i c_j \rangle^{-1}$, then computes η_i, and finally evaluates

the WF/CR operator. The `crv` command behaves in the same way. Let us consider the first step of computing $\langle c_i c_j \rangle$. Each time this happens, the result is stored in the auxiliary object `corrmatrix`. It is also possible to perform just this first step with the command `new corrmatrix`. The numerical values of the matrix can be written to a file with `write corrmatrix` and analysed separately. It is then possible to re-load this matrix in another ICECORE session with `load corrmatrix`. Then, calling `cr` will not re-compute it, but instead directly continue with the pre-computed `corrmatrix`, picking up where you left off.

In the second step, `cr` inverts $\langle c_i c_j \rangle$. When this step is complete, the result is saved in the `invmatrix` object. Again, it is possible to perform just this step with `new invmatrix`, back up the inverted matrix with `write invmatrix`, and re-load it with `load invmatrix`. If an `invmatrix` object was loaded, `cr` will pick up at that point and continue directly with computing the correlation vector η_i and evaluating the WF/CR operator. Thus, calling

```
cr 3
```

will do exactly the same thing as

```
new corrmatrix
new invmatrix 3
cr
```

but the latter can be split and computed in separate sessions:

```
...
new corrmatrix                      # you have to do that only
write corrmatrix binary corrm.dat   # once per set of constraints!
...
load corrmatrix binary corrm.dat    # you have to do that only
new invmatrix 3                     # once per set of constraints
write invmatrix binary invm.dat     # if you do not change sigma_nl!
...
load invmatrix binary invm.dat      # load that each time you need it
cr                                  # ... and this will now be faster!
```

The `binary` argument is one of the possible storage formats for matrices (the other one is `ascii`). The *`sigma_nl`* parameter (here the value of 3) is added to the diagonal of `corrmatrix` before inversion. Therefore, it is not stored with `corrmatrix`, but considered a part of the inversion process. If you give a σ_{NL} explicitly (recommended), and you compute the WF/CR step-wise as above, the *`sigma_nl`* argument goes to the `new invmatrix` command. With no *`sigma_nl`* parameter, `new invmatrix` will try to iteratively find one such that $\chi^2/\mathrm{dof} \approx 1$.

Finally the η_i vector computed by `cr` is stored in the `eta` object and can be exported to a file with `write eta`. However it makes no sense to split that step as well, since η_i also depends on the RR via the mock constraints $\tilde{c}_i$ and therefore has to be recomputed for each individual CR anyway. For this reason, there is no `load eta` command.

A.3.4 Scripting

The possibility to run ICECORE in script mode makes it very easy to run repetitive tasks from the command line. As an example, we show one possible way to generate a large number of constrained initial conditions with RZA from the same data, varying just the seed of the random component. We assume here that you already performed the RZA shift of the data (Example 4)

and then precomputed the inverted correlation matrix $\langle c_i c_j \rangle^{-1}$ of the RZA shifted constraints with a suitable $\sigma_{\rm NL}$ parameter. The basic script to generate ICs with RZA could then be as follows:

```
# Example 6 - basic_rza_script.ic

new cosmology 0.272 0.728 70.2 0    # compute cosmology from parameters
load pk WMAP7.dat                   # load tabulated power spectrum
normalise pk 0.807                  # normalise to sigma8
new dgrid 256 100                   # alloc grid with N=256^3, L=100 Mpc/h
seed 12345                          # generate white noise field
colour                              # compute deltaRR
vsolve                              # solve uRR needed for mock constraints
load correlator grid                # load pre-computed correlator
load constraints rza.co             # load RZA-shifted constraints
multiply constraints2 kmstompch     # convert from km/s to Mpc/h
load invmatrix binary rza_invm.dat  # load precomputed <ci cj>^-1
cr                                  # compute deltaCR
vsolve                              # solve uCR assuming PBCs
writeics gadget ics_cr_12345.gadget # write gadget ICs
quit
```

Running the same IceCoRe script one hundred times varying the seed integer can now be easily accomplished, for example with a short `bash` script:

```
#!/bin/bash
for i in {1..100}
do
   sed "s/12345/$i/g" basic_rza_script.ic > temp.ic
   ./icecore temp.ic
done
```

In each loop iteration, this will execute IceCoRe with a script where the original seed integer has been replaced with another number from 1 to 100. In the end, one hundred different Gadget IC files `ics_cr_1.gadget`, ..., `ics_cr_100.gadget` will be written, each with a different random seed. This is just an example how scripting can be used with IceCoRe, of course much more is possible.

Just like other command-line interpreters on Unix/Linux-style systems, instead of launching IceCoRe with a script file as the argument (`./icecore my_script.ic`), you can just execute the script itself (`./my_script.ic`) if you add a hash-bang sequence as the first line of the script file, i.e. the characters '`#!`' followed by the absolute or relative path of the `icecore` executable.

A.4 Editing overdensity and velocity grids

Besides the essential functionality of generating random and constrained cosmological ICs, computing Wiener filter mean fields, and performing RZA reconstruction, the IceCoRe code provides several additional utilities. Most of them are tools to analyse and modify the `dgrid` and `vgrid` objects, i.e. the grid-discretised overdensity and velocity fields. The analysis commands `binpk`, `histogram`, `stat`, and `xcorr` have already been mentioned. IceCoRe further has some commands that *modify* the grids, providing some functionality that could be useful for setting up or analysing cosmological ICs.

A.4.1 Whitening

In Section 3.5 we discussed how the white noise field $w(\boldsymbol{x})$ of an overdensity field realisation can be used as an input for other IC-generating codes such as GRAFIC-2, MPGRAFIC, MUSIC and GINNUNGAGAP. This allows us to generate high-resolution, multi-scale and baryonic ICs with these codes from constrained realisations generated by ICECORE. The white noise field $w(\boldsymbol{x})$ corresponding to a realisation $\delta(\boldsymbol{x})$ can be computed in Fourier space through equation 3.3. This is performed with the command `whiten`. The overdensity field $\delta(\boldsymbol{x})$ will be read from `dgrid`, and the resulting $w(\boldsymbol{x})$ will be also saved in `dgrid`, overwriting the original field. This command is the exact opposite of `colour`. Both commands require the correct power spectrum to be loaded in the `pk` object. For whitening with the `whiten` command, it is absolutely necessary that the power spectrum is exactly the same that was used for `colour`, normalised to the same σ_8 at the same redshift with the same normalisation method (cf. Sections 3.1.3 and A.2.2). After running `whiten`, you should check with `stat dgrid` that the result satisfies the white noise condition (equation 3.2).

For compatibility with other IC-generating codes, the GRAFIC white noise (`graficwn`) format is the most suitable option to write whitened fields. Technical details about this format are given in Section B.5.4. Below is an example script to whiten a previously generated $\delta^{\mathrm{CR}}(\boldsymbol{x})$ field and write the resulting white noise field $w(\boldsymbol{x})$:

```
# Example 7 - converting a CR to white noise

load dgrid bov delta_cr.bov   # load deltaCR (normalised to z=0)
load pk WMAP7.dat             # load P(k) used to generate that deltaCR
normalise pk 0.807            # normalise to z=0 exactly as before
whiten                        # compute white noise field
write dgrid graficwn wn.dat   # write output in grafic white noise format
quit
```

A.4.2 Smoothing filters

The `filter` command can be run on either `dgrid` or `vgrid` to apply a linear smoothing filter. The syntax is

`filter { dgrid | vgrid }` *`filter_type radius`*

where the notation `{ dgrid | vgrid }` means that `dgrid` and `vgrid` are mutually exclusive options. The *`filter_type`* argument can be either `gauss`, applying a Gaussian filter (equation 2.54), or `sth`, applying a spherical top-hat filter (equation 2.55) on the chosen grid. The *`radius`* parameter specifies the smoothing radius R in units of Mpc/h. The computation is carried out by forward FFT, multiplying with the filter kernel, and backward FFT. The Gaussian filter was used in many places of this work, for example the fields plotted with contour lines in the bottom row of Figure 2.8 were created with the command `filter dgrid gauss 2.5`. The spherical top-hat filter can be used, for example, to compute the value of σ_8 of a given overdensity field:

```
# Example 8 - compute actual sigma8 on grid

load dgrid bov delta_cr.bov   # load an overdensity field at z=0
filter dgrid sth 8            # apply spherical tophat filter, R=8 Mpc/h
stat dgrid                    # show statistical moments
quit                          # (sigma8 will be the standard deviation)
```

A.4.3 Downsampling and cropping

The commands `degrade`, `kdegrade`, and `crop` can be used to reduce either the resolution of `dgrid` or `vgrid` (by downsampling), or the boxsize (by cropping). Both procedures share the Fourier uncertainty principle: they can be performed either exact in real space, which will introduce errors in Fourier space, or exact in Fourier space, which preserves the correct $P(k)$ but leads to artefacts in the real-space distribution. A discussion of these effects can be found in Bertschinger (2001). The command `degrade` performs a downsampling in real space. The syntax is

```
degrade { dgrid | vgrid } new_resolution
```

which will overwrite the original grids. The integer argument *`new_resolution`* specifies the new resolution n_{new} of the downsampled field and has to be smaller than the original resolution n, but n does not have to be a multiple of *`new_resolution`*, and neither have to be powers of two. The downsampling is performed by trilinear interpolation. If *`new_resolution`* is exactly $n/2$, then `degrade` is equivalent to replacing each group of $2 \times 2 \times 2$ cells with one cell that will be assigned the average value of the original eight cells. This procedure will be exact in real-space in the sense of preserving mass (`dgrid`) and momentum (`vgrid`), but introduce a power aliasing error on the $P(k)$ of the downsampled grid which significantly underestimates the input $P(k)$ for wavenumbers above $\approx 0.2\,k_{\mathrm{Ny}}$. An alternative is the command `kdegrade`, which performs the downsampling in Fourier space (similar to the `degraf` utility of MPGRAFIC, see Prunet et al. 2008). This will preserve the correct $P(k)$ for all wavelengths below the Nyquist frequency of the downsampled grid, but on the other hand introduce significant non-local errors in the real-space distribution of the field ("Gibbs ringing").

The `crop` command can be used to crop either the `dgrid` or the `vgrid` to a smaller boxsize while keeping the same grid cell spacing. The cropped box will be again a cubic box. The syntax is

```
crop { dgrid | vgrid } new_resolution x_offset y_offset z_offset
```

The argument *`new_resolution`* specifies the new resolution n_{new} of the cropped box, which must be smaller than the original resolution n. The arguments *`x_offset`*, *`y_offset`*, and *`z_offset`* specify the position of the resulting sub-box relative to the 0,0,0 cell of the original box, i.e. how many grid cells will be cut off at the left/lower edge in each dimension. They must be non-negative integers and specify the cropping positions in cells, i.e. in units of Δx, and *not* in units of Mpc/h. For example, the following command takes a `dgrid` with resolution $n = 256$ and arbitrary boxsize L and crops it to a sub-box with boxsize $L/2$ centred on the original box centre:

```
crop dgrid 128 64 64 64
```

As with `degrade` and `kdegrade`, n does not have to be a multiple of n_{new}, and neither have to be powers of two. After the cropping procedure, the remaining grids will not any longer have periodic boundary conditions (PBCs), regardless of whether they had PBCs before. Grids without PBCs cannot be used to generate cosmological ICs. Furthermore, cropping in real space introduces artefacts on the power spectrum $P(k)$. A more advanced application of the `crop` command is the method introduced in Bertschinger (2001), which leads to a cropping that is correct in Fourier space and preserves PBCs. The algorithm consists of first whitening the grid, then cropping it, and then colouring it again:

```
whiten
crop dgrid 128 64 64 64
colour
```

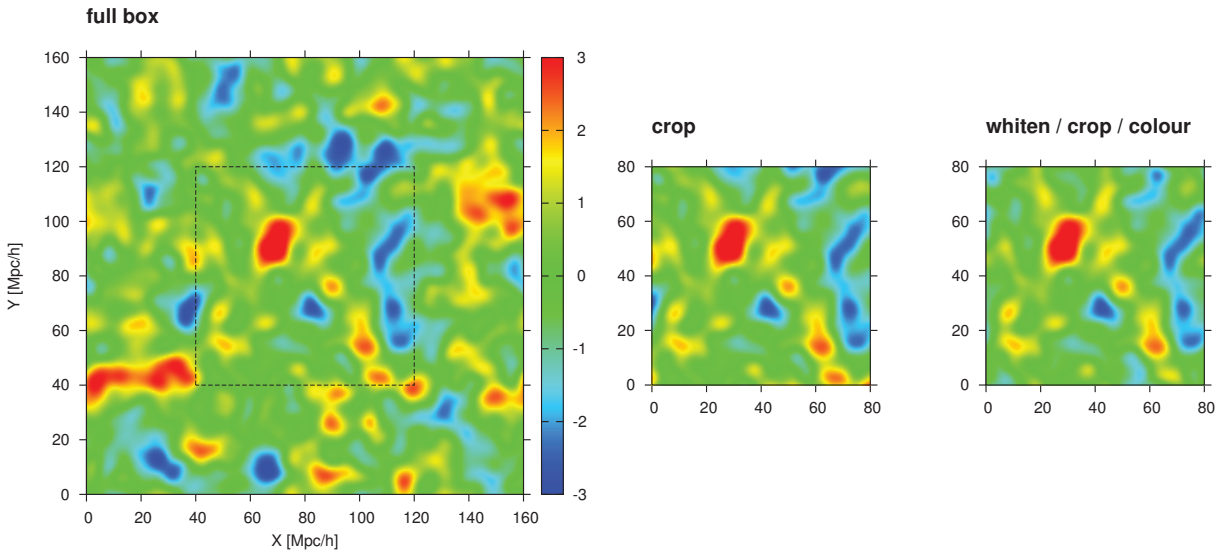

Figure A.1: Left: original periodic box with $n = 256$, $L = 160$ Mpc/h. Middle: cropped box with $n = 128$, $L = 80$ Mpc/h. Right: box cropped while preserving periodic boundary conditions (PBCs). Notice how the density distribution at the edge of the cropped box changed to reflect the PBCs, while everywhere else the field stays the same. Shown is the overdensity field smoothed with a 2.5 Mpc/h Gaussian kernel.

This can be used to obtain sub-boxes with smaller boxsize from a larger original box, while retaining PBCs and zero mean, so that they have the required statistical properties to be used for setting up cosmological ICs. The result is depicted in Figure A.1. This simple algorithm can be useful if one wants to simulate the same cosmic structures with different simulation boxsizes.

A.4.4 Snapshot binning with mass assignment schemes

ICeCoRe performs all calculations with fields discretised on uniform cubic grids and generates N-body particle data only when it writes ICs in Gadget format. However, as an additional tool, ICeCoRe also offers the possibility to convert a cosmological simulation snapshot, containing particle data, to density and velocity fields on a uniform cubic grid. This can be done with the `bin` command. It can be useful to quickly analyse the evolved fields of cosmological simulations. Currently, Gadget and Art-format snapshots are supported. Snapshots from the Ramses code can be used as well after conversion to Gadget format; please contact the author for an appropriate conversion tool. ICeCoRe was specifically developed for this work and for this reason currently supports only single-file dark matter snapshots. Large simulations ($N > 1024^3$), zoom-in simulations containing multiple particle masses, and snapshots containing baryonic physics, such as hydrodynamical data, star particles, etc. are *not* supported by ICeCoRe; appropriate analysis tools have been developed by other authors.

The basic syntax of the `bin` command is:

```
bin { dgrid | vgrid } format input_file resolution mas_type
```

This will open the snapshot contained in the file *`input_file`*, assuming a snapshot file format *`format`*, read all contained particle positions $\boldsymbol{x}_i$ (`bin dgrid`) and velocities $\boldsymbol{u}_i$ (`bin vgrid`) and

bin them to a grid with resolution n, specified with the ***resolution*** argument, using a mass assignment scheme specified with the argument ***mas_type***. The `bin dgrid` command will first evaluate in each cell the number density $n(\boldsymbol{x})$ of particles convolved with $W(\boldsymbol{x})$,

$$n(\boldsymbol{x}) = \sum_i W(\boldsymbol{x} - \boldsymbol{x}_i) \quad , \tag{A.2}$$

where $W(\boldsymbol{x})$ is the kernel of the mass assignment scheme, and then convert this to density ρ in units of the mean density $\bar{\rho}$:

$$\frac{\rho(\boldsymbol{x})}{\bar{\rho}} = \frac{N_{\mathrm{cells}}}{N_{\mathrm{particles}}}\, n(\boldsymbol{x}) \quad . \tag{A.3}$$

The field $\rho(\boldsymbol{x})/\bar{\rho}$ will then be saved in the `dgrid` object. The `bin vgrid` command will instead evaluate

$$\boldsymbol{u}(\boldsymbol{x}) = \frac{\sum_i \boldsymbol{u}_i W(\boldsymbol{x} - \boldsymbol{x}_i)}{\sum_i W(\boldsymbol{x} - \boldsymbol{x}_i)} \tag{A.4}$$

and save the result in the `vgrid` object. In both cases, the boxsize of the grid L in Mpc/h will be read automatically from the snapshot file. Note that `bin dgrid` produces a *density* field in units of the mean density $\bar{\rho}$, and *not* an overdensity field $\delta = \rho/\bar{\rho} - 1$, which would otherwise be the quantity represented with `dgrid`. If you need to convert this density field to the overdensity $\delta(\boldsymbol{x})$, it is sufficient to subtract 1 from the field, which can be done with

```
addconst dgrid -1
```

The unit of the result produced in `vgrid` by `bin vgrid` will be whatever unit was used in the snapshot to store the particle velocities. It is the responsibility of the user to pay attention to the units when using the `bin` command.

The three most commonly used mass assignment schemes are available (Hockney & Eastwood 1992): nearest grid cell (NGC), cloud-in-cell (CIC), and triangular-shaped cloud (TSC). To select one of these, type `ngc`, `cic` or `tsc` for the ***mas_type*** argument. This will determine the window function $W(\boldsymbol{x})$ that will be used for the binning. These window functions are not isotropic, as for example the Gaussian or top-hat kernels, but instead follow the cubic-symmetrical geometry

$$W(\boldsymbol{x}) = W(x_x) \cdot W(x_y) \cdot W(x_z) \quad , \tag{A.5}$$

where the one-dimensional kernels corresponding to the three schemes write

$$W_{\mathrm{NGC}}(x) = \begin{cases} 1 & \text{if } \dfrac{x}{\Delta x} < \frac{1}{2} \quad , \\ 0 & \text{else} \quad ; \end{cases} \tag{A.6}$$

$$W_{\mathrm{CIC}}(x) = \begin{cases} 1 - \left|\dfrac{x}{\Delta x}\right| & \text{if } \dfrac{x}{\Delta x} < 1 \quad , \\ 0 & \text{else} \quad ; \end{cases} \tag{A.7}$$

$$W_{\mathrm{TSC}}(x) = \begin{cases} \frac{3}{4} - \left(\frac{x}{\Delta x}\right)^2 & \text{if } \frac{x}{\Delta x} < \frac{1}{2} \quad , \\ \frac{1}{2}\left(\frac{3}{2} - \frac{x}{\Delta x}\right)^2 & \text{if } \frac{1}{2} \leq \frac{x}{\Delta x} < \frac{3}{2} \quad , \\ 0 & \text{else} \quad ; \end{cases} \tag{A.8}$$

with $\Delta x = L/n$ being the chosen grid cell spacing. Choosing a mass assignment scheme is always a tradeoff between a large smoothing of the field and a large amount of shot noise from the inhomogeneous particle distribution. The NGC scheme features the least smoothing, while the TSC scheme features the least amount of shot noise. The CIC scheme is a popular compromise. For example, the density field slices shown in Figure 5.12 are plotted from grids generated with the command

```
bin dgrid gadget snapshot_006 256 cic
```

where in this case `snapshot_006` is the GADGET-format snapshot file at $z = 0$.

The `binpk` command, which computes periodogram estimates of the power spectrum $P(k)$ of `dgrid`, offers the option to compensate for the smoothing introduced by the mass assignment schemes by adding `ngc`, `cic` or `tsc` as an additional option. Combined with `bin`, this can be used to quickly generate an estimate of the non-linear $P(k)$ from a simulation snapshot. For example, the black curve in Figure 2.12 was generated from an overdensity grid with

```
load dgrid bov box160_ics_delta.bov  # read IC overdensity grid (n=256)
binpk box160_ics_pk.dat 256          # write P(k) to file
```

while the red curve was generated from a GADGET-2 snapshot at $z = 0$ with

```
bin dgrid gadget box160_snapshot_006 256 cic   # bin z=0 Gadget snapshot
binpk box160_z0_pk.dat 256 cic                 # write P(k) to file
```

Please note that the `binpk` command corrects for the effect of mass assignment smoothing, but does not correct for aliasing effects due to the particle distribution, as this requires a prior knowledge of the power spectrum (Jing 2005). This is still very well usable because the mass assignment smoothing is usually much stronger than the other effects. A more precise correction method may be included in a future ICECORE version.

Care must be taken when binning the particle velocities. In a snapshot of an evolved cosmological simulation, neighbouring particles will often be separated by distances equivalent to several grid cell spacings in underdense regions. In this case, if a cell does not touch the window kernel of any particle, the value of equation A.4 will be undefined. The `bin` command prints a message about whether this case occured and writes NaN (not a number) values in affected cells. For a continuous velocity field binning from a snapshot at $z = 0$, the grid resolution should be always chosen at least a factor of two lower than the number of particles per dimension, together with the TSC mass assignment scheme, which has the widest window function. Another possibility to overcome the inhomogeneous sampling would be the use of an adaptive mass assignment scheme; this feature will possibly be included in future ICECORE versions.

Because of the significant shot noise, smoothing from the mass assignment, and alias errors of fields binned from particle data, use density or velocity fields that were generated with `bin` only for analysis. Never use them to set up cosmological initial conditions or do anything else that would normally require fields generated on a grid.

Appendix B

ICECORE reference

B.1 Objects

The data inside ICECORE is organised into *objects* which are responsible to manage the different fields, vectors, matrices and tables in computer memory that are needed for calculations. Most of the ICECORE commands operate on one of these objects. If a command can be applied to different objects, it will feature the name of the desired object as its first argument, such as for example the `load`, `write`, and `new` commands. This section provides a description of all objects available in ICECORE version 1.0 in alphabetic order.

`constraints`

The `constraints` object is a list of M constraints (or data points) $c_1, \ldots c_M$. Each of these constraints has ten numbers attached to it (see Table 3.1): the constraint type Y_i, the constraint position (x_i, y_i, z_i), the constraint value c, the estimated constraint error ε_i, the direction vector $(\hat{e}_{\mu,x}, \hat{e}_{\mu,y}, \hat{e}_{\mu,z})$ and the Gaussian smoothing radius R_G. The constraint type is encoded with an integer ($\delta \equiv 1$, $\psi_\mu \equiv 2$). The other fields contain their numerical values as real numbers. The input data format for `constraints` consists of an ASCII table with ten whitespace or tab-separated columns corresponding to these ten quantities, one line per constraint. Such a data file can be loaded into the `constraints` object with the `load constraints` command. Several commands exist to manipulate the constraints: `addvector`, `deoverlap`, `multiply`, `shift`, and `sparse`. Using these commands, the RZA method discussed in Chapter 5 can be performed within ICECORE. An edited set of constraints can be written to a file in the same ten-column format with `write constraints`.

`correlator`

The `correlator` object holds the precomputed correlation functions ξ, ψ_R, ψ_T and ζ needed to compute the data autocorrelation matrix $\langle c_i c_j \rangle$ and to evaluate the WF/CR operator. It is also responsible to compute all necessary correlation values when needed by interpolation on the precomputed functions. The `correlator` object needs to be initialised with the `new correlator` command and requires the `pk` object (the power spectrum $P(k)$) and the `dgrid` object (in order to know the box parameters: resolution and boxsize).

There are two types of correlators: `analytic` and `grid`. The analytic correlator is a very efficient method for Wiener filtering from peculiar velocity data and for constructing CRs if

the finite-volume effect and the periodic boundary conditions are not expected to lead to large systematic inconsistencies (i.e. if the constraints are well inside the central part of the box and the boxsize is not smaller than at least 100 Mpc/h). The grid correlator provides an alternative by computing all correlations in a way that is self-consistent with the finite volume and periodic boundary conditions. See Sections 3.4.1 and 3.4.2 for details.

The analytic correlator computes the isotropic one-dimensional correlation functions ξ, ψ_{R}, ψ_{T} and ζ over a wide range of distances x and smoothing radii R_{G} and stores them in two-dimensional lookup tables. Values for given x and R_{G} are computed by interpolation over these tables using the GSL cubic spline interpolation. The grid correlator instead stores the three-dimensional correlation grids ξ, ψ_{xx}, ψ_{xy} and ζ_x. Values for given x are computed using trilinear interpolation on these grids. For nonzero smoothing radii R_{G}, an additional Gaussian smoothing is performed on the grids, either locally in real space or with FFT, depending on what is faster for the given R_{G}.

Especially if you use the analytic correlator and a wide range of smoothing radii R_{G}, which requires a large array of precomputed interpolation tables, computing the correlator can take some time, because many numerical integrals with highly oscillatory integrand terms have to be evaluated in equations 3.46 – 3.49. In order to use the same correlator for many ICECORE runs, it can make sense to precompute and store the correlator and then to quickly reload it in each run, instead of recomputing it. This can be accomplished with `write correlator` and `load correlator`. On the other hand, computing the grid correlator requires only four FFTs and will not take much time in most cases.

---

corrmatrix

The `corrmatrix` object holds the data autocorrelation matrix $\langle c_i c_j \rangle$ that was last computed, using the definition 3.21 and 3.24, *without* the σ_{NL} parameter. The σ_{NL} is a constant term added to the diagonal of the matrix before it is inverted, which is performed by the commands `cr`, `crv`, or `new invmatrix`. It is therefore not stored in `corrmatrix` to avoid unnecessary computations and added only immediately before inversion. After the inverted data autocorrelation matrix $\langle c_i c_j \rangle^{-1}$ has been computed, the result will be stored in the `invmatrix` object, and the `corrmatrix` object will be deleted to save space in memory. It therefore normally only exists in between computing and inverting the matrix (although the deletion of `corrmatrix` can be prevented with the `retain` option).

If computing the matrix takes a long time and you want to re-use the same matrix in several ICECORE runs, it is possible to write `corrmatrix` to a file with `write corrmatrix` and then to load it later with `load corrmatrix` before the inversion. If you decide to use this possibility, be extremely careful that the pre-computed `corrmatrix` is fully consistent with the currently loaded constraints and correlator, as `load corrmatrix` does not check this. This is only recommended if you will try different σ_{NL} parameters on the same matrix. Otherwise, it is more efficient to pre-compute and store the *inverted* matrix using `write invmatrix` and `load invmatrix`.

---

cosmology

The `cosmology` object holds all necessary quantities that directly depend on the cosmological parameters. It is initialised with the `new cosmology` command, which takes as arguments the parameters Ω_{m} (total matter density), Ω_Λ (dark energy density), H_0 (the numerical value of the Hubble constant in units of km s^{-1} Mpc^{-1}, i.e. the value of $100h$), and the redshift z. For Wiener filter reconstructions of the contemporary large-scale structure, the redshift will be

$z = 0$, while for constrained and unconstrained cosmological initial conditions it will be the starting redshift z_{init} of the simulation. During initialisation, the `cosmology` object computes and stores the parameters a (scale factor), $\dot{a}$ (expansion rate), D_+ (linear growth factor), f (linear growth rate), and t (age of the Universe) at the given redshift z. These values can be retrieved at any time with the command `stat cosmology`. In this sense, ICECORE can be used as a "cosmological calculator" to quickly compute these quantities.

---

dgrid

The `dgrid` object contains the three-dimensional overdensity grid $\delta(\boldsymbol{x})$ with resolution $N = n^3$, where n is the number of grid cells per dimension, and boxsize L given in Mpc/h. The overdensity is assumed to be a real-valued scalar field with no imaginary part. Therefore, its Fourier transform is Hermitian, $\delta(\boldsymbol{k}) = \delta^*(-\boldsymbol{k})$. The data is stored internally as a `std::complex<double>` array with n^3 entries. Assuming an 8-byte `double`, the object takes $16n^3$ bytes of memory, plus a small overhead for the box parameters. This format in memory is directly compatible with the `fftw_complex` array type of FFTW3 that is used by ICECORE for all discrete Fourier transforms.

The `dgrid` can be read and written to files in various formats (see `load dgrid` and `write dgrid`) as well as created within ICECORE (see `new dgrid`). Many of the ICECORE commands directly manipulate the `dgrid` object, such as `add`, `cr`, `crop`, `degrade`, `filter`, `seed`, and `whiten`. Others allow to analyse the overdensity grid, such as computing its power spectrum (`binpk`), probability distribution (`histogram`), its statistical moments (`stat dgrid`), or its cross-correlation with another overdensity grid (`xcorr`). A `dgrid` can also be created by binning (with a mass assignment scheme) from the dark matter particles of a cosmological simulation snapshot (`bin dgrid`). In this case, the resulting `dgrid` will be a windowed density field instead of an overdensity field.

---

eta

The `eta` object holds the correlation vector η_i (equation 3.39) that was last computed in a WF/CR run. It can be written to a file using the `write eta` command, if one wants to examine the values of η_i. The `eta` object is accessible only for this diagnostic purpose and will be automatically recomputed in each WF/CR run.

---

invmatrix

The `invmatrix` object holds the inverted data autocorrelation matrix $\langle c_i c_j \rangle^{-1}$ that was last computed (using the definition 3.45). This object is needed to execute the WF/CR operator (commands `cr` and `crv`). If the WF/CR does not find an `invmatrix` object, it will automatically compute and store it. The inverted data autocorrelation matrix depends on the constraints, the correlator, and the non-linearity parameter σ_{NL}. Therefore, the `invmatrix` object will be invalidated and deleted each time one of these objects/parameters changes and automatically recomputed if the WF/CR operator is run afterwards. It can also be pre-computed explicitly using the `new invmatrix` command. To re-use a precomputed inverted data autocorrelation matrix, the commands `write invmatrix` and `load invmatrix` can be used. This is especially useful if one wants to generate many CRs in different ICECORE runs using the same constraints and σ_{NL} and save the time that would be needed to compute `invmatrix` each time. If you

decide to use this possibility, be extremely careful that the pre-computed `invmatrix` is fully consistent with the currently loaded constraints, correlator, and σ_{NL}, as `load invmatrix` does not check this. If you want to use different σ_{NL} parameters with the same pre-computed data autocorrelation matrix, you can instead pre-compute and store the non-inverted matrix using `write corrmatrix` and `load corrmatrix`.

---

pk

The `pk` object holds the currently loaded tabulated power spectrum $P(k)$ that describes the assumed cosmological prior model. It computes values of $P(k)$ by interpolating over this table. Currently this is implemented with cubic spline interpolation using the GSL. The `pk` object is required for generating cosmological random realisations from white noise fields (`colour` command) and the reverse procedure of whitening (`whiten` command). It is further needed to load a `correlator` object, i.e. to compute the correlation functions needed for WF/CR. ICECORE does not contain code to compute $P(k)$ from cosmological parameters; this has to be done with other software such as CMBFAST, CAMB or ICOSMO (see Section 2.2.4). The tabulated $P(k)$ can then be loaded with the `load pk` command. The input format consists of a whitespace or tab-separated ASCII table with two columns, k (in units of h/Mpc) and $P(k)$, *with no additional lines in the file*. This is the same format that the CAMB code uses for output. Before use, the $P(k)$ has to be normalised to a normalisation parameter σ_8 and a redshift z. This is done with the `normalise` command.

---

vgrid

The `vgrid` object holds the three grids ψ_x, ψ_y and ψ_z that descibe the three cartesian components of the linear displacement field $\boldsymbol{\psi}$. This is equivalent to the linear peculiar velocity field $\boldsymbol{u}$ divided by the factor $\dot{a}f$. Numerically the object behaves very similar to the `dgrid` object, except that it contains three such grids instead of one. Input and output is done with `load vgrid` and `write vgrid`, in the same way as with `dgrid`, except that three files have to be provided instead of one, each file holding one grid. Many of the grid operations possible with `dgrid` can be applied to `vgrid` and will do the same except with all three grids, such as `crop`, `degrade`, and `filter`. It is also possible to compute the probability distribution (`histogram`) and the statistical moments (`stat vgrid`) of the three displacement grids.

The `vgrid` can be converted from displacement in Mpc/h to peculiar velocity in km/s and vice versa using the commands `multiply vgrid mpchtokms` and `multiply vgrid kmstompch`, respectively. This should be done only for handling input and output, since ICECORE always assumes internal units of Mpc/h for all computations. The linear displacement field corresponding to the overdensity field stored in `dgrid` can be computed with the command `vsolve`, which evaluates equation (3.5), and the reverse operation is performed with `dsolve`. Both assume periodic boundary conditions.

A `vgrid` object can also be created by binning (with a mass assignment scheme) from the peculiar velocities of the dark matter particles of a cosmological simulation snapshot (`bin vgrid`). In this case, the unit of `vgrid` will correspond to whatever unit the particle peculiar velocities have in the snapshot. It is up to the user to handle these units correctly. Special care must be taken to select a grid resolution and mass assignment scheme that will create a continuous velocity field from the discrete particle data; see Section A.4.4 reference for details.

B.2 Commands

This section provides a reference for all available commands in ICECORE version 1.0 in alphabetic order. For each command it explains the syntax, the required arguments/parameters, and what action the command performs.

For the syntax explanation we use the following conventions. The names of commands, objects, and everything else that has to be typed as it is written, appear in `typewriter` font. Arguments which are tokens and have to be replaced by some actual name or value appear in *`italic typewriter`* font. In the example scripts shown for some commands, we will insert some example names and values for them. Arguments that are mutually exclusive are separated by a pipe symbol " `|` ". Arguments that are optional are surrounded by square brackets " `[ ]` ". Arguments that are required are instead surrounded by curly brackets " `{ }` " where necessary. Comments in the example scripts are preceded by a hash symbol " `#` ".

---

add

Load grids and add them together.

Syntax:
`add dgrid` *`format file`* `[` *`options`* `]`
`add vgrid` *`format vx_file vy_file vz_file`* `[` *`options`* `]`

This command uses the same syntax, formats and options as `load dgrid` and `load vgrid`, respectively. The difference between `add` and `load` is that after reading from file, the grids are *added* to the existing ones, instead of overwriting them. This can be used, for example, to quickly average over several grids:

```
add dgrid bov dgrid_1.bov             # add up four overdensity grids
add dgrid bov dgrid_2.bov
add dgrid bov dgrid_3.bov
add dgrid bov dgrid_4.bov
mult dgrid 0.25                       # average: divide by nr. of grids
write dgrid bov dgrid_averaged.bov    # write result
```

If there is no grid loaded initially, `add` will simply load the grid, i.e. do the same as the `load` command.

---

addconst

Add a constant number to the overdensity field.

Syntax:
`addconst` *`number`*

This command adds a constant real number to the field stored in `dgrid`. This can be used, for example, to convert from normalised density $\rho/\bar{\rho}$ to overdensity $\delta = \rho/\bar{\rho} - 1$:

```
bin dgrid gadget snapshot.gadget 256 cic   # particle mass assignment produces
addconst dgrid -1                          #   a normalised density field!
write dgrid bov snapshot_delta.bov
```

addvector

Add a constant vector to displacement field or constraints.

Syntax:
`addvector { constraints2 | vgrid } vx vy vz`

This commands adds a vector with three cartesian components to either the currently loaded `vgrid`, or to the values c_i of the currently loaded vector-type constraints (type 2; displacement type). The three cartesian components of the vector that will be added must be given as arguments *vx*, *vy*, and *vz*.

`addvector vgrid` is the vector analogue to the `addconst` command and can be used, for example, to add or subtract an overall bulk motion to/from the currently loaded velocity/displacement field $\boldsymbol{\psi}$. `addvector constraints2` can be used to do the same for the currently loaded set of constraints. Overdensity-type constraints (type 1) will be ignored in this case. Each type 2 constraint constrains only one component of $\boldsymbol{\psi}$ parallel to some unit vector $\hat{\boldsymbol{e}}_\mu$. So what `addvector` actually does is computing for each type 2 constraint the dot product of $\hat{\boldsymbol{e}}_\mu$ and the vector given in the argument, and then adding this value to the constraint value c_i. This can be used to convert velocity-type data from one rest frame to another (cf. Section 3.2.2). For example, let us assume that we want to compute a Wiener filter reconstruction from observational radial peculiar velocities $v_r^\odot$ given in the solar standard of rest in units of km/s, but the reconstruction shall be performed from velocities in the Galactic standard of rest (GSR). The conversion from the solar frame to the GSR frame is given by (Tully et al. 2008):

$$v_r^{\mathrm{GSR}} = v_r^\odot + 9.3\,\hat{e}_x - 218\,\hat{e}_y + 7.6\,\hat{e}_z \quad . \tag{B.1}$$

This conversion and the subsequent WF reconstruction can be performed with ICECORE like this:

```
load constraints data.co                # load data (in the correct format!)
addvector constraints2 9.3 -218 7.6     # perform conversion to GSR frame
multiply constraints kmstompch          # convert to internal units of Mpc/h
cr                                      # compute Wiener filter mean field
```

bin

Compute density and velocity fields by binning the dark matter particles from a simulation snapshot to a grid.

Syntax:
`bin { dgrid | vgrid } format file resolution mas_type`

This function can be used to map the density and velocity fields represented as a set of N-body particles in a simulation snapshot to a uniform cubic grid. It will open the snapshot contained in *file*, assuming a snapshot file format *format*, read the contained particle positions $\boldsymbol{x}_i$ and velocities $\boldsymbol{u}_i$, and then bin them to a grid. `bin dgrid` will produce the normalised density field $\rho/\bar{\rho}$ (equation A.2 and A.3) and store the result in `dgrid`, while `bin vgrid` will produce the binned velocity field (equation A.4) and store the result in `vgrid` (in whatever units $\boldsymbol{u}_i$ was stored

in the snapshot). The resolution n (i.e. number of cells per dimension) of the grid is specified with the ***resolution*** argument. The boxsize of the grid L in Mpc/h will be read automatically from the snapshot file. The mass assignment scheme is specified with the argument ***mas_type***, which can be either `ngc` (nearest grid cell), `cic` (cloud-in-cell), or `tsc` (triangular-shaped cloud). See Section A.4.4 a more detailed discussion.

Currently supported snapshot file formats are: `gadget` – single-file GADGET format with additional identifier blocks; `gadget1` – single-file GADGET format without these identifier blocks; and `art` – single-file ART snapshot format. For GADGET snapshots, the length unit in the file is assumed to be kpc/h (the GADGET standard unit); see GADGET User's guide (available at www.mpa-garching.mpg.de/gadget) for more information. Snapshots from the RAMSES code can be used as well after conversion to GADGET format; please contact the author for an appropriate conversion tool. Multiple-file snapshots, large simulations ($N > 1024^3$), zoom-in simulations containing multiple particle masses, and snapshots containing baryonic physics, such as hydrodynamical data, star particles, etc. are currently not supported.

---

binpk

Compute a periodogram estimate for the power spectrum $P(k)$ of an overdensity grid.

Syntax:
`binpk` ***file*** `[` ***n_bins*** `[` ***correction_type*** `] ]`

This command computes a periodogram estimate of the power spectrum $P(k)$ on the currently loaded `dgrid` and writes the result into ***file***. The output format is the same as the input format for tabulated power spectra: a whitespace or tab-separated two-column ASCII table with k in units of h/Mpc in the first column and $P(k)$ in the second. The computation is carried out with a forward FFT on `dgrid` and a binning of $\delta(\boldsymbol{k})$ over the wavenumbers $k = |\boldsymbol{k}|$ that correspond to the grid cells in Fourier space. The bins are placed equidistant in log space between the smallest wavenumber $k_L = 2\pi/L$ (fundamental wave) and the highest wavenumber $\sqrt{3}\,k_{\mathrm{Ny}} = |(k_{\mathrm{Ny}}, k_{\mathrm{Ny}}, k_{\mathrm{Ny}})|$ that occurs in the box (where $k_{\mathrm{Ny}} = \pi/\Delta x$ is the Nyquist frequency of the grid). The DC mode at $k = 0$ (total mean overdensity of the box) will be ignored; its value can be determined with `stat dgrid` instead.

Two optional arguments can be specified. The first, ***n_bins***, is the number of bins which will be used to sample the grid $P(k)$. If not specified, the number of bins will be set to the value of n (the resolution of the grid). The second, ***correction_type***, is used if the grid was generated by a particle binning with a mass assignment scheme (`bin dgrid` command) to perform a correction on $P(k)$ for the mass assignment smoothing effect; see Jing (2005). ***correction_type*** can be either `ngc`, `cic`, or `tsc`, following the names of the corresponding mass assignment schemes. This option should be used *only* for density grids created from particle data with one of these three mass assignment schemes.

---

chisquare

Compute the value of χ^2/dof.

This command evaluates χ^2/dof (equation 3.43) from the constraints c_i and their inverse auto-correlation matrix, where dof (degrees of freedom) is the number of constraints. This command

requires the `constraints` and `invmatrix` objects. χ^2/dof is also computed each time the autocorrelation matrix is inverted by `new invmatrix`.

`colour`

Generate a Gaussian random field with power spectrum $P(k)$ from a Gaussian white noise field.

This command assumes that `dgrid` contains a Gaussian white noise field $w(\boldsymbol{x})$ as defined by equation 3.2 (such a field can be created with `seed`). First, a forward FFT is performed. Then, the currently loaded power spectrum $P(k)$ (`pk` object) is used to evaluate equation 3.3 for each grid cell. The final step is a backward FFT. The result is an overdensity field $\delta(\boldsymbol{x})$ on the grid, a random realisation of a Gaussian random field with periodic boundary conditions and power spectrum $P(k)$. This grid is stored in `dgrid`, overwriting the white noise field, and can be used to set up ICs. The procedure can be reversed with `whiten`.

`cr`

Compute a constrained realisation or a Wiener filter mean field.

Syntax:
`cr [ sigma_nl ]`

The `cr` command computes the linear overdensity field $\delta^{\mathrm{CR}}(\boldsymbol{x})$ of a constrained realisation by computing the mock constraints $\tilde{c}_i$, the correlation vector η_i (equation 3.39), and then evaluating equation 3.40 for all grid cells of the currently loaded `dgrid`. To compute the correlations, the currently loaded `correlator` and `constraints` objects are used. The optional argument *`sigma_nl`* enables to specify the value of the non-linearity parameter σ_{NL} that will be used for the inversion of the data autocorrelation matrix (3.45).

If a *`sigma_nl`* is specified, `cr` first checks if the data autocorrelation matrix $\langle c_i c_j \rangle$ was computed yet, i.e. if there is a `corrmatrix` object. If yes, this matrix is inverted using the specified σ_{NL} by automatically calling `new invmatrix`, and `cr` proceeds with computing $\tilde{c}_i$, η_i and $\delta^{\mathrm{CR}}(\boldsymbol{x})$. If no, `corrmatrix` is computed first by automatically calling `new corrmatrix`.

If no *`sigma_nl`* is specified, `cr` first checks if the *inverted* data autocorrelation matrix $\langle c_i c_j \rangle^{-1}$ was computed yet, i.e. if there is an `invmatrix` object. If yes, `cr` proceeds directly with computing $\tilde{c}_i$, η_i and $\delta^{\mathrm{CR}}(\boldsymbol{x})$. If no, `invmatrix` is computed first by automatically calling `new invmatrix`. In this case, σ_{NL} will be automatically chosen such that $0.995 < \chi^2/\mathrm{dof} < 1.005$. This involves several iterative matrix inversions and can take a very long time if the matrix is large. It is therefore recommended to always provide a *`sigma_nl`* parameter if no `invmatrix` is loaded, except if there are only a few thousand constraints or less.

In order to compute the mock constraints $\tilde{c}_i$ and the RR term in equation 3.40, the currently loaded `dgrid` and `vgrid` are used, assuming that they contain $\delta^{\mathrm{RR}}(\boldsymbol{x})$ and $\boldsymbol{u}^{\mathrm{RR}}(\boldsymbol{x})$, respectively. The latter must be in units of displacement (Mpc/h). The contents of `dgrid` will be overwritten with $\delta^{\mathrm{CR}}(\boldsymbol{x})$.

The command `cr` is also used to compute the Wiener filter mean field, $\delta^{\mathrm{WF}}(\boldsymbol{x})$. If the currently loaded `dgrid` and `vgrid` are empty and contain only zeroes, both the mock constraints $\tilde{c}_i$ and the RR term will be zero as well, and the equations will reduce to 3.26 and 3.27, respectively. Thus, if you want to compute $\delta^{\mathrm{WF}}(\boldsymbol{x})$ instead of $\delta^{\mathrm{CR}}(\boldsymbol{x})$, call `new dgrid` and `new vgrid` immediately before `cr`.

The velocity field of the WF or CR is not computed by `cr`. It can be obtained instead from $\delta^{\mathrm{CR}}(\boldsymbol{x})$ by running `vsolve`, which assumes periodic boundary conditions (PBCs). If you want to compute a WF velocity field from the data *without* PBCs, e.g. for an analysis of the cosmography of the Local Universe, run `crv` instead, using the analytic correlator.

`crop`

Crop density and velocity fields to smaller boxsizes.

Syntax:
`crop { dgrid | vgrid }` *`new_resolution x_offset y_offset z_offset`*

This command can be used to crop either the `dgrid` or the `vgrid` to a smaller boxsize while keeping the same grid cell spacing. The cropped box will be again a cubic box. The argument *`new_resolution`* specifies the new resolution n_{new} of the cropped box, which must be smaller than the original resolution n. The arguments *`x_offset`*, *`y_offset`*, and *`z_offset`* specify the position of the resulting sub-box relative to the 0,0,0 cell of the original box, i.e. how many grid cells will be cut off at the left/lower edge in each dimension. They must be non-negative integers and specify the cropping positions in cells, *not* in units of Mpc/h. For example, the following command takes a `dgrid` with resolution $n = 256$ and boxsize L and crops it to a sub-box with boxsize $L/2$ centred on the original box centre:

```
crop dgrid 128 64 64 64
```

The old resolution does not have to be a multiple of the new resolution, and neither resolutions have to be powers of two. After the cropping procedure, the remaining grid will not any longer have periodic boundary conditions (PBCs), regardless of whether it had PBCs before, and cannot be used to generate cosmological initial conditions. To preserve PBCs while cropping, the method of Bertschinger (2001) can be used; see Section A.4.3.

`crv`

Compute a constrained realisation or a Wiener filter mean field for both the overdensity and the velocity field.

Syntax:
`crv [` *`sigma_nl`* `]`

This command behaves in the same way as `cr`, but additionally it also evaluates equation 3.41 for each grid cell of `vgrid`. After this process, `vgrid` will contain $\boldsymbol{\psi}^{\mathrm{CR}}$ (or $\boldsymbol{\psi}^{\mathrm{WF}}$) in units of Mpc/$h$; the previous values of `vgrid` will be overwritten. If the analytic correlator is used, this is the only method to compute $\boldsymbol{\psi}^{\mathrm{CR}}$, and equivalently $\boldsymbol{u}^{\mathrm{CR}}$, without assuming periodic boundary conditions on this field. It generally makes no sense to run `crv` with the grid correlator.

---

degrade

Resample grids to lower resolution in real space.

Syntax:
`degrade { dgrid | vgrid } new_resolution`

This command resamples either the `dgrid` or the `vgrid` from the original resolution n (number of cells per dimension) to a lower resolution n_{new}, specified with the *new_resolution* argument, using trilinear interpolation in real space. n does not have to be an integer multiple of n_{new}. If it is, the resampling will be equivalent to an averaging over cells. This procedure preserves mass (for `dgrid`) and momentum (for `vgrid`) in real space, but the power spectrum $P(k)$ of the downsampled grid will be underestimated at high frequencies due to power aliasing errors. If you need to preserve $P(k)$, use `kdegrade` instead.

---

deoverlap

Remove constraints with overlapping smoothing volumes.

Syntax:
`deoverlap [ factor ]`

This command goes through the list of constraints in the `constraints` object and performs a pair-wise check of the constrained positions $\boldsymbol{x}_i$ and the corresponding smoothing radii R_{G}. If two constraints are found closer than $f_{\mathrm{G}} \cdot (R_{\mathrm{G},1} + R_{\mathrm{G},2})$ to each other, the constraint with the smaller R_{G} is removed. If $R_{\mathrm{G},1} = R_{\mathrm{G},2}$ the constraint that is further down in the list will be removed. The value of the parameter f_{G} can be specified with the optional argument *factor*. If no argument is given, the default value $f_{\mathrm{G}} = 1$ will be used.

---

dsolve

Compute the overdensity field corresponding to a displacement field in linear theory.

This command solves the inverse of equation 3.5 to obtain the overdensity field $\delta(\boldsymbol{x})$ corresponding to a linear displacement field $\boldsymbol{\psi}(\boldsymbol{x})$ stored in `vgrid`. It performs a forward FFT on each component $\psi_x(\boldsymbol{x})$, $\psi_y(\boldsymbol{x})$, and $\psi_z(\boldsymbol{x})$, followed by the multiplication

$$\delta(\boldsymbol{k}) = -i\boldsymbol{k} \cdot \boldsymbol{\psi}(\boldsymbol{k}) \tag{B.2}$$

and finally a backward FFT to obtain $\delta(\boldsymbol{x})$. This result is then stored in `dgrid`. The inverse procedure can be carried out with `vsolve`.

---

filter

Apply a linear smoothing filter to the density or velocity field.

Syntax:
`filter { dgrid | vgrid } filter_type radius`

This command smooths either `dgrid` or `vgrid` with a linear filter. The type of the filter is specified with *`filter_type`*. This can be either `gauss` for the Gaussian filter (equation 2.54), or `sth` for the spherical top-hat filter (equation 2.55). The filter radius R in Mpc/h is specified by *`radius`*. The smoothing is performed by forward FFT, multiplying with the smoothing kernel $W(kR)$, and backward FFT. This method assumes periodic boundary conditions on all grids.

---

hann

Apply a Hann filter to the power spectrum.

This command applies a Hann filter to the tabulated power spectrum $P(k)$ currently stored in the `pk` object. The filter kernel is defined by

$$W_{\mathrm{H}}(k) = \begin{cases} \cos\left(\dfrac{\pi k}{2k_{\mathrm{Ny}}}\right) & \text{if } k \leq k_{\mathrm{Ny}} \quad , \\ 0 & \text{else} \quad , \end{cases} \tag{B.3}$$

where $k_{\mathrm{Ny}} = \pi/\Delta x$ is the Nyquist frequency of the currently loaded `dgrid`. This filtering damps high-frequency modes and completely removes the frequencies between k_{Ny} and $\sqrt{3}\,k_{\mathrm{Ny}}$, which are anisotropically sampled on the grids because of Fourier-space discretisation. As opposed to `filter`, the `hann` command does *not* modify the grids, but instead the `pk` object by replacing the $P(k)$ stored there by the modified power spectrum $P_{\mathrm{H}}(k) = W_{\mathrm{H}}(k) \cdot P(k)$. This modified `pk` object can then be used to set up Hann-filtered ICs sometimes used for cosmological simulations, for example the MareNostrum simulation (see Prunet et al. 2008). The benefit of this method is that small-scale anisotropies in the correlation functions will be suppressed. The downside is that a lot of useful small-scale resolution is lost. The Hann-filtered $P(k)$ can also be used to set up the grid correlator, which otherwise features small-scale anisotropies as numerical artefacts along the x, y, z axes in the grid correlation functions (Figure 3.5; see also Bertschinger 2001). This should be done consistently: use the Hann filter on the grid correlator for a CR if and only if you use it to generate the random component as well. However, we did not use Hann filtering for the simulations presented in this work, arguing that the small-scale anisotropies do not significantly affect the outcome of a simulation at $z = 0$.

---

help

Display built-in help and code reference.

Syntax:
`help [ topic ]`

If called with no arguments, `help` will display some general info about ICeCoRe and an overview over the most important help topics. If given an ICeCoRe command name as an argument, `help` will display a short description of the command along with an explanation of the syntax and required arguments. Some commands are divided into subcommands, for example `help correlator` will contain some general information about the `correlator` object, while `help correlator analytic` and `help correlator grid` provide more details about how to use the different correlator types. If given the name of an ICeCoRe object, it displays information about the object and its purpose. If called with `commands`, `objects`, or `tokens` as the *`topic`*

argument, `help` will display a list of all available commands, objects, and predefined tokens, respectively.

---

histogram

Compute histograms of the overdensity or velocity/displacement field distribution functions.

Syntax:
`histogram { dgrid | vgrid }` *file min max n_bins*

The `histogram` command computes a histogram of all grid cell values of `dgrid` and `vgrid`, respectively, i.e. the distribution functions on the grid, and write them into *file*. The binning range and number of bins has to be specified explicitly with the arguments *min*, *max*, and *n_bins*. For example, Figure 2.9 was created using this function. The output file is an ASCII table containing the centre of each bin in the first column and the corresponding cell count of $\delta(\boldsymbol{x})$ in the second column (for `dgrid`) or the cell counts of $\psi_x(\boldsymbol{x})$, $\psi_y(\boldsymbol{x})$, and $\psi_z(\boldsymbol{x})$ in columns two, three, and four, respectively (for `vgrid`).

---

kdegrade

Resample grids to lower resolution in Fourier space.

Syntax:
`kdegrade { dgrid | vgrid }` *new_resolution*

The `kdegrade` command is similar to `degrade`, but the downsampling is performed in Fourier space. `kdegrade` will perform a forward FFT, then execute the downsampling in the same way as `degrade` (but in the correct coordinates of Fourier space), and finally a backward FFT. The downsampled grid will preserve the power spectrum $P(k)$ of the original grid for scales above the resolution of the new grid. The disadvantage is that the real-space distribution of the field will be corrupted with numerical artefacts (Gibbs ringing). On the other hand, the `degrade` method preserves the correct real-space distribution, but features power aliasing errors in Fourier space.

---

load

Load objects from files.

Syntax:
```
load constraints file
load correlator { analytic | grid }
load corrmatrix format file [ retain ]
load cosmology format file
load dgrid format file [ options ]
⟶ load dgrid ascii file n boxsize
    load dgrid binary file n boxsize [ swap ]
    load dgrid bov file
    load dgrid grafic file
    load dgrid graficwn file boxsize
```

```
load invmatrix format file [ retain ]
load pk file
load vgrid format vx_file vy_file vz_file [ options ]
⟶ load vgrid ascii vx_file vy_file vz_file n boxsize
    load vgrid binary vx_file vy_file vz_file n boxsize [ swap ]
    load vgrid bov vx_file vy_file vz_file
    load vgrid grafic vx_file vy_file vz_file
    load vgrid graficwn vx_file vy_file vz_file boxsize
```

The `load` command is responsible for initialising ICeCoRe objects from data stored in files. All objects except `eta` can be initialised in this way. Generally, if an object was written to a file using `write`, the `load` command can restore the object from that file in the exact same state. However, `load` can also be used to read data generated by other software. In the current version 1.0, all input/output files must always be located in the same directory as the ICeCoRe executable.

`load constraints` will read a set of constraints from *`file`* in the ten-column ASCII format described in Section A.2.5.

`load correlator` will read a precomputed `correlator` from files. In this case, the filenames are predefined. `load correlator analytic` will search for tabulated correlation functions in files named `xi.dat`, `psiR.dat`, `psiT.dat`, and `zeta.dat`. The command will also figure out from those files over what range in distance x and smoothing radius R_G the tables were computed. Make sure this is compatible with your constraints; this is not checked by the code. `load correlator grid` will instead search for four correlation grids in BOV files named `xi.bov`, `psiXX.bov`, `psiXY.bov`, and `zetaX.bov`, together with the corresponding binary data files (`*.bov.data`). Make sure that the boxsize L and resolution N of these grids matches the `dgrid` and `vgrid` objects you are currently working with; this is not checked by the code.

`load corrmatrix` will read the correlation matrix $\langle c_i c_j \rangle$ from *`file`*. Make sure that it is compatible with the loaded correlator and constraints; this is not checked. *`format`* can be either `ascii`, which stores the matrix in human-readable text format, or `binary`, which is analogous to the `binary` format used by the grids but in 2D. If an `invmatrix` object is loaded, it will be deleted before initialising `corrmatrix` to save space in memory, except if the option `retain` is specified.

`load cosmology` will read the cosmological parameters Ω_m and Ω_Λ, the Hubble constant H_0, and the redshift z from the header of cosmological IC files and initialise the `cosmology` object with the values found. *`format`* can be: `gadget` – Gadget snapshot file with additional identifier blocks; `gadget1` – Gadget snapshot file without these identifier blocks; `art` – Art snapshot format; and `grafic` – any cosmological field in Grafic format.

`load dgrid` will load an overdensity field $\delta(\boldsymbol{x})$ into `dgrid` from the data in *`file`*. The supported formats are discussed in detail in Section B.5; currently, they are `ascii`, `binary`, `bov`, `grafic`, and `graficwn`. For `bov` and `grafic`, no further arguments are required. For `ascii` and `binary`, the boxsize L in Mpc/h and resolution n have to be given as *`options`*. For `graficwn`, only the boxsize L in Mpc/h has to be given. For `binary`, there is also the additional option `swap` to reverse the endianness.

`load invmatrix` will read the inverted correlation matrix $\langle c_i c_j \rangle^{-1}$ from *`file`*. The options are the same as for `load corrmatrix`. If an `corrmatrix` object is loaded, it will be deleted before initialising `invmatrix` to save space in memory, except if the option `retain` is specified.

`load pk` will read a tabulated power spectrum $P(k)$ from *`file`*. This has to be a whitespace- or tab-separated ASCII table with two columns, k in h/Mpc and $P(k)$, with no additional lines in the file. The same format is used by the CAMB code, so its output can be directly read by ICeCoRe.

`load vgrid` will load a velocity or displacement field $\boldsymbol{\psi}(\boldsymbol{x})$ into `vgrid` from the data in the three files *`vx_file`*, *`vy_file`*, and *`vz_file`*, containing the fields corresponding to the three cartesian components $\psi_x(\boldsymbol{x})$, $\psi_y(\boldsymbol{x})$, $\psi_z(\boldsymbol{x})$, respectively. The formats and options are the same as for `load dgrid`, except that three filenames have to be given instead of one. If the files contain a peculiar velocity field $\boldsymbol{u}(\boldsymbol{x})$ in km/s instead of a displacement field $\boldsymbol{\psi}(\boldsymbol{x})$ in Mpc/h, it has to be converted to Mpc/h with `multiply vgrid kmstompch` before using it for cosmological computations.

---

multiply

Multiply the values stored in objects with a constant factor.

Syntax:
`multiply { constraints | constraints2 | dgrid | pk | vgrid }` *`factor`*

`multiply constraints` will multiply both the constraint values c_i and their given absolute errors ε_i in the `constraints` object with *`factor`*. `multiply constraints2` will do the same but only for type 2 (displacement-type) constraints. `multiply dgrid` and `multiply vgrid` multiply the values of all cells of those fields with *`factor`*. `multiply pk` will multiply the values of the tabulated power spectrum $P(k)$ in the `pk` object with *`factor`*.

This command is useful to perform normalisations of the above quantities by hand, and for unit conversions. For example, the conversion from velocities in km/s to displacements in Mpc/h (the standard unit of ICeCoRe) can be executed with `multiply vgrid kmstompch` and `multiply constraints2 kmstompch`, respectively. The reverse unit conversion can be done using the token `mpchtokms` instead. With `multiply dgrid growthd`, it is possible to renormalise the overdensity field by hand from $z = 0$ to another redshift, where `growthd` is a token for the linear growth factor D_+ of the currently loaded `cosmology`. The same renormalisation can be done for `pk` by executing `multiply dgrid growthd` *twice* (since $P(k) \propto D_+^2$). Of course, it is possible to give any other number for *`factor`* instead of the predefined tokens.

---

new

Initialise objects.

Syntax:
`new correlator { analytic | grid } [` *`k_min k_max`* `[` *`R_G_max`* `] ]`
`new corrmatrix [ retain ]`
`new cosmology` *`omega_m omega_L H_0 z`*
`new dgrid` *`resolution boxsize`*
`new invmatrix [` *`sigma_nl`* `] [ retain ]`
`new vgrid` *`resolution boxsize`*

The `new` command initialises objects based on some default values, or computes them from other objects if possible. The objects `constraints` and `pk` cannot be initialised that way: they always have to be loaded from external data files. It is also not useful to initialise an `eta` object, since it is used only for diagnostic purposes.

`new correlator` will compute the correlator from the power spectrum $P(k)$ in the currently loaded `pk` object. A `dgrid` object has to be present as well, otherwise `new correlator` would not know what boxsize L and resolution n you want to work with when computing Wiener filter mean fields and CRs. `new correlator analytic` provides some options. With ***k_min*** and ***k_max***, the integration range $[\,k_{\min}, k_{\max}\,]$ for evaluating the integrals in equations 3.46 – 3.49 can be specified in units of h/Mpc; they default to `kminpk` and `kmaxpk`, respectively. The ***R_G_max*** option allows one to specify up to what smoothing radius R_{G} in Mpc/h the correlation functions should be tabulated; this defaults to the maximum R_{G} that can occur for the current `constraints`, or to $R_{\mathrm{G}} = 0$ if no `constraints` are present. The command `new correlator grid` will ignore these additional options.

`new corrmatrix` computes the data autocorrelation matrix $\langle c_i c_j \rangle$ based on the currently loaded `constraints` and `correlator` and stores the result in the `corrmatrix` object. If an `invmatrix` object is loaded, it will be deleted before initialising `corrmatrix` to save space in memory, except if the option `retain` is specified.

`new cosmology` initalises the `cosmology` object from the cosmological parameters Ω_{m} and Ω_Λ, the Hubble constant H_0, and the redshift z, which have to be given as arguments. This command can also be used as a "cosmological calculator": it immediately computes the scale factor a, expansion rate $\dot{a}$, linear growth factor D_+, linear growth rate f, and the age of the Universe t for the given cosmological parameters and redshift.

`new dgrid` takes a resolution n (number of cells per dimension) and boxsize L in Mpc/h as arguments and allocates a `dgrid` object with these parameters. All values of the field will be initialised to zero.

`new invmatrix` computes the inverted data autocorrelation matrix $\langle c_i c_j \rangle^{-1}$ and stores the result in the `invmatrix` object. If a `corrmatrix` object containing the non-inverted matrix $\langle c_i c_j \rangle$ is present, it will add the σ_{NL} value given by the ***sigma_nl*** parameter to its diagonal, invert the matrix using Cholesky decomposition (see Section 3.4.3) and then delete the `corrmatrix` object, except if the option `retain` is specified. If no `corrmatrix` object is present, it will first call `new corrmatrix` to compute it. If no σ_{NL} parameter is provided, a value will be automatically chosen such that $0.995 < \chi^2/\mathrm{dof} < 1.005$. This involves several iterative matrix inversions and can take a very long time if the matrix is large. It is therefore recommended to always provide a ***sigma_nl*** parameter to `new invmatrix` if there are more than a few thousand constraints.

`new vgrid` takes a resolution n (number of cells per dimension) and boxsize L in Mpc/h as arguments and allocates a `vgrid` object with these parameters. All values of the field will be initialised to zero.

normalise

Normalise $P(k)$ to a given σ_8 and redshift z.

Syntax:
`normalise sigma_8 [ k_min k_max ]`

The `normalise` command first computes the value of σ_8 of the power spectrum $P(k)$ in the currently loaded `pk` object using equation 2.58. It then normalises the power spectrum such that this value equals the given *`sigma_8`* parameter. Finally, if the current redshift in the `cosmology` object is $z \neq 0$, it also normalises $P(k)$ with the redshift using $P(k) \propto D_+^2$. By default, the integration range $[\,k_{\min}, k_{\max}\,]$ in equation 2.58 is given by the tabulated range of $P(k)$. This can be overridden with the optional *`k_min`* and *`k_max`* options. You can either give wavenumbers in units of h/Mpc, or use the predefined tokens. `normalise sigma_8 kl kny` will use the integration range between the fundamental mode k_L and the Nyquist frequency $k_{\rm Ny}$ of the current `dgrid` (CLUES convention); `normalise sigma_8 kl2 kny` will instead use the range between $k_L/\sqrt{2}$ and $k_{\rm Ny}$ (method of Klypin & Holtzman 1997). See Section 3.1.3 for details.

---

peakfind

Find peaks in the overdensity field.

Syntax:
`peakfind file`

`peakfind` is a simple peak finding utility. This command will search for peaks in the overdensity field $\delta(\boldsymbol{x})$ stored in `dgrid`. A peak is defined as a grid cell with a higher value than all of its six direct neighbours (assuming periodic boundary conditions). The coordinates x, y, z and overdensity values δ of all found peaks will be written to *`file`* in ASCII table format.

---

quit

Delete all objects and quit the current ICECORE session.

The `quit` command is the default method to deallocate and delete all objects and cleanly quit ICECORE from within an interactive session. In script mode, it is not necessary to use it, since the code will quit anyway as soon as it reaches the end of the script file, or if an error occurs.

---

reorder

Switch row-major order and column-major order on a grid.

Syntax:
`reorder { dgrid | vgrid }`

The `reorder` command rearranges the way values are stored on the grid in `dgrid`, or on the three grids in `vgrid`, in such a way that row-major order will be transformed to column-major order and vice versa. More specifically, `reorder dgrid` changes $\delta(x, y, z)$ to $\delta(z, y, x)$, and `reorder vgrid` changes $\boldsymbol{\psi}(x, y, z)$ to $\boldsymbol{\psi}(z, y, x)$.

`reset`

Delete and deallocate objects.

Syntax:
`reset [ object ]`

If called without arguments, `reset` deletes all currently loaded objects and frees all memory that was previously allocated in the current ICECORE session. This leaves the session in the same state as if it would be terminated and a new one started. The `reset` command can also be called with the name of one of the ICECORE objects as an argument, for example `reset dgrid`. In this case, only the specified object will be deleted and its memory freed.

`seed`

Generate a Gaussian white noise field.

Syntax:
`seed integer`

This command will initialise a random number generator and then fill all cells of `dgrid` with zero-mean, unity-variance Gaussian distributed random numbers. This command uses GSL functions. By default, the random number generator is the Mersenne Twister MT19937 (Matsumoto & Nishimura 1998), and the Gaussian distribution is generated with the fast Ziggurat algorithm (Marsaglia & Tsang 2000). The random number generator has to be seeded with an integer, which must be given as an argument.

`shift`

Shift the positions of constraints by a constant vector or a vector field.

Syntax:
`shift constraints { vector x y z | vgrid }`

`shift constraints vector x y z` will shift the positions $\boldsymbol{x}_i$ of all constraints in the currently loaded `constraints` object by adding the constant vector $\boldsymbol{x} = (x, y, z)$. The components of this vector have to be given as arguments in Mpc/h. All other quantities defining the constraints will be left unchanged. This can be used, for example, to change the position of the observer relative to the box when working with radial peculiar velocity constraints. This command will *not* check whether the constraints are inside the box before or after shifting and will *not* assume periodic boundary conditions.

`shift constraints vgrid` will instead shift the positions $\boldsymbol{x}_i$ of all constraints with the values of the vector field stored in `vgrid`. For each constraint at $\boldsymbol{x}_i$, it will look up the value of $\boldsymbol{\psi}(x_i)$ by interpolating on `vgrid` and then add this value to the position vector of the constraint. This will check if the constraints are inside the box defined by `vgrid`, and result in an error if they are not. If a constraint was inside the box before, but would be shifted outside of the box by this operation, the code will assume periodic boundary conditions and take the components of the

new $\boldsymbol{x}_i$ modulo the boxsize L such that $\boldsymbol{x}_i$ will be again inside the box. This command is useful to perform the RZA shift on peculiar velocity data; see Chapter 5 and Section A.3.1 for details.

---

sparse

Remove constraints that are too close to each other.

Syntax:
`sparse r_min`

This command goes through the list of constraints in the `constraints` object and performs a pair-wise check of the constrained positions $\boldsymbol{x}_i$. If any two constraints are found closer than `r_min` to each other, the constraint that is further down the list will be removed. However, if two constraints are found at *exactly* the same position $\boldsymbol{x}_i$ (within floating point precision), the code will leave both in the list, assuming that the intention was to constrain two different quantities at the same position (such as different cartesian components of the displacement field).

This command is useful to reduce the density of constraints if there are too many constraints to efficiently evaluate the WF/CR operator and/or if the WF/CR result features a lot of noise because of very noisy constraints that are close together. This will typically also have the side effect that χ^2/dof is comparatively large; then, `sparse` can be an alternative to increasing the σ_{NL} parameter to reduce it, if the latter is not producing satisfactory WF/CR fields. If the constraints feature non-zero smoothing radii R_{G}, the `deoverlap` command should be run instead of `sparse`.

---

stat

Display information about currently loaded objects.

Syntax:
`stat [ object ]`

This diagnostic command can be used to retrieve the basic parameters of the currently loaded objects. If called with no arguments, `stat` will display a list of all ICeCoRe objects and show which ones are currently loaded. If called with the argument `object`, which has to be the name of an ICeCoRe object, it displays more detailed information about this object: the number of constraints and their types for `stat constraints`, all relevant cosmological quantities for `stat cosmology`, the tabulated range and σ_8 value of $P(k)$ for `pk`, and so forth. `stat corrmatrix`, `stat invmatrix` and `stat eta` will display their rank. `stat dgrid` and `stat vgrid` will find the minimum and maximum values of the density and velocity/displacement fields and compute the fields' first four statistical moments (mean, standard deviation, skewness, and kurtosis).

---

vsolve

Compute the displacement field corresponding to an overdensity field in linear theory.

This command solves equation 3.5 to obtain the linear displacement field $\boldsymbol{\psi}(\boldsymbol{x})$ corresponding to an overdensity field $\delta(\boldsymbol{x})$ stored in `dgrid`. It performs a forward FFT, followed by the multipli-

cations

$$\psi_\alpha(\boldsymbol{k}) = \frac{ik_\alpha}{k^2}\delta(\boldsymbol{k}) \quad , \quad \alpha \in \{x, y, z\} \quad , \tag{B.4}$$

and then performs three backward FFTs to obtain $\psi_x(\boldsymbol{x})$, $\psi_y(\boldsymbol{x})$, and $\psi_z(\boldsymbol{x})$, respectively. This result is then stored in `vgrid`. The inverse procedure can be carried out with `dsolve`.

`whiten`

Reduce a cosmological Gaussian random field to white noise.

This command assumes that `dgrid` contains a cosmological overdensity field $\delta(\boldsymbol{x})$ and the `pk` object contains the exact same power spectrum $P(k)$ that was used to generate it (including its normalisation with σ_8 and z). It then computes the white noise field $w(\boldsymbol{x})$ corresponding to $\delta(\boldsymbol{x})$ by solving equation 3.2 and stores the result in `dgrid`, overwriting the original field. This is done by performing a forward FFT, multiplying the values in all grid cells with $1/\sqrt{P(k)}$, and then performing a backward FFT. This method assumes periodic boundary conditions. The procedure can be reversed with `colour`.

`write`

Write data from objects to files.

Syntax:
`write constraints` *`file`*
`write corrmatrix` *`format file`*
`write correlator`
`write dgrid` *`format file`* `[ swap ]`
`write eta` *`format file`*
`write invmatrix` *`format file`*
`write pk` *`file`*
`write vgrid` *`format vx_file vy_file vz_file`* `[ swap ]`

The `write` command is responsible for writing the data inside ICECORE objects to files. All objects except `cosmology` can be dumped to files in this way; however, `cosmology` can instead be written into the headers of data files using `grafic` or `gadget` formats. Generally, if an object is written to a file using `write`, the `load` command can later restore the object from that file in the exact same state.

`write constraints` will write the set of constraints in the `constraints` object to *`file`* in the ten-column ASCII format described in Section A.2.5.

`write correlator` will dump the data inside the `correlator` object to files. In this case, the filenames are predefined. If the current `correlator` type is `analytic`, then `write correlator` will produce files named `xi.dat`, `psiR.dat`, `psiT.dat`, and `zeta.dat` containing the tabulated correlation functions. If the type is `grid`, the command will instead write four correlation grids in BOV format using the file names `xi.bov`, `psiXX.bov`, `psiXY.bov`, and `zetaX.bov`.

`write corrmatrix` will dump the correlation matrix $\langle c_i c_j \rangle$ that was last computed in the current session to *`file`*. *`format`* can be either `ascii`, which stores the matrix in human-readable text format, or `binary`, which is analogous to the `binary` format used by the grids but in 2D.

`write dgrid` will write the overdensity field $\delta(\boldsymbol{x})$ currently loaded in `dgrid` to *`file`*. The supported formats are discussed in detail in Section B.5; currently, they are `ascii`, `binary`, `bov`, `grafic`, and `graficwn`. The `binary` format offers the option `swap`, which will swap the endianness of the output file.

`write eta` will dump the correlation vector η_i that was last computed in the current session to *`file`*. *`format`* can be either `ascii` or `binary`.

`write invmatrix` will dump the inverted correlation matrix $\langle c_i c_j \rangle^{-1}$ that was last computed in the current session to *`file`*. The options are the same as for `write corrmatrix`.

`write pk` will write the currently loaded tabulated power spectrum $P(k)$ to *`file`*. The format will be a whitespace-separated ASCII table with two columns, k in h/Mpc and $P(k)$.

`write vgrid` will write the overdensity/displacement field $\boldsymbol{\psi}(\boldsymbol{x})$ currently loaded in `vgrid` to the three files *`vx_file`*, *`vy_file`*, and *`vz_file`*, which will contain the fields corresponding to the three cartesian components $\psi_x(\boldsymbol{x})$, $\psi_y(\boldsymbol{x})$, $\psi_z(\boldsymbol{x})$, respectively. The formats and options are the same as for `load dgrid`, except that three filenames have to be given instead of one.

---

writeics

Generate particle initial conditions and write them to a file.

Syntax:
`writeics` *`format file`*

This command generates N-body particle initial conditions from the current displacement field $\boldsymbol{\psi}(\boldsymbol{x})$ in `vgrid` using the standard method described in Section 3.1.2. The displacement field has to be in units of Mpc/h and normalised to the desired starting redshift $z_{\rm init}$ of the simulation. After generating the initial particle positions and velocities, the command then writes the particle data to *`file`* in the snapshot format specified by *`format`*. The only *`format`* currently supported by this command is `gadget`, which is the single-file GADGET-2 snapshot format with additional identifier blocks. See GADGET User's guide (available at www.mpa-garching.mpg.de/gadget) for more information. This also requires a `cosmology` object, since the cosmological parameters have to be written to the header of the GADGET file. In order to set up ICs for the RAMSES code, you should write GRAFIC-formatted fields instead of particle data.

Note that if you generate a particle file with `writeics`, and then map it on a grid again using the `bin` command, you will *not* recover the original density and velocity fields that were used to generate the particles, because of the significant smoothing introduced by the mass assignment scheme and the aliasing and shot noise caused by the particle discretisation. There is currently no method in ICeCoRe to recover the original field from a snapshot file produced by `writeics`.

---

xcorr

Compute the Fourier-space cross-correlation between two grids.

Syntax:
`xcorr` *`out_file n_bins format in_file`* `[` *`options`* `]`

The `xcorr` utility computes the Fourier-space cross-correlation $g(k)$ (equation 2.85) between the current overdensity field $\delta(\boldsymbol{x})$ in `dgrid` and a reference grid $\delta_0(\boldsymbol{x})$ that will be read from *`in_file`*. The *`format`* argument and the *`options`* following *`infile`* behave in the same way as the *`format`* and *`file`* arguments of the `load dgrid` command, i.e. it is possible to load the reference grid in all formats that ICECORE supports. However, the reference grid loaded in this way will not be stored in memory, but only used for evaluating equation 2.85 in Fourier space and discarded afterwards. $g(k)$ will be computed as a histogram over k and written to *`out_file`* as an ASCII table with two columns, k in h/Mpc and $g(k)$. The *`n_bins`* argument specifies the number of bins used to compute the $g(k)$ histogram, which will be equidistantly spaced in log k-space in the interval between k_L and $\sqrt{3}\,k_{\mathrm{Ny}}$. The curves in Figure 2.10 were computed with this utility. Below is an example script to compute $g(k)$ for two density grids stored in BOV format:

```
load dgrid bov dgrid_1.bov            # load grid 1
xcorr gk.dat 100 bov dgrid_2.bov      # load grid 2, compute and write g(k)
quit
```

B.3 Tokens for numerical values

In the ICeCoRe interface, several numerical values can be accessed directly with predefined tokens. This includes the fundamental and Nyquist frequency of the currently loaded `dgrid`, which are useful to define an integration interval for computing the $P(k)$ normalisation or the grid correlator, the growth factor D_+, which can be used to rescale overdensity fields from $z = 0$ to another redshift z, and the unit conversion factors from km/s (velocity) to Mpc/h (displacement) and vice versa. These tokens can be used interchangeably with literal numbers wherever a number is expected as a command argument. In ICeCoRe version 1.0, the following tokens are defined:

token	numerical value
`growthd`	Linear growth factor D_+ of `cosmology` object
`growthf`	Linear growth rate f of `cosmology` object
`kl`	Wavenumber of fundamental mode $k_L = 2\pi/L$ of `dgrid` object
`kl2`	$k_L/\sqrt{2}$
`kl3`	$k_L/\sqrt{3}$
`kmaxpk`	Highest wavenumber $k_{\max}$ for which there is a tabulated $P(k)$ value in `pk`
`kminpk`	Lowest wavenumber $k_{\min}$ for which there is a tabulated $P(k)$ value in `pk`
`kmstompch`	Conversion factor $1/\dot{a}f$ from comoving velocity $\boldsymbol{u}$ in km/s to comoving displacement $\boldsymbol{\psi}$ in Mpc/h at redshift z of `cosmology` object
`kny`	Wavenumber of Nyquist frequency $k_{\mathrm{Ny}} = \pi/\Delta x$ of `dgrid` object
`kny2`	$\sqrt{2}\,k_{\mathrm{Ny}}$
`kny3`	$\sqrt{3}\,k_{\mathrm{Ny}}$
`mpchtokms`	Conversion factor $\dot{a}f$ from comoving displacement $\boldsymbol{\psi}$ in Mpc/h to comoving velocity $\boldsymbol{u}$ in km/s at redshift z of `cosmology` object

B.4 Keyword aliases

The ICECORE interface features abbreviations and alternative spellings for the most commonly typed object names, commands, and tokens. They can be used interchangeably with the original variant. In ICECORE version 1.0, the following keyword aliases are defined:

original	alternative
`colour`	`color`
`constraints`	`co`
`constraints2`	`co2`
`correlator`	`corr`
`corrmatrix`	`corrm`
`cosmology`	`cosmo`, `c`
`dgrid`	`d`
`filter`	`f`
`help`	`h`
`histogram`	`hist`
`invmatrix`	`invm`
`kmstompch`	`kms2mpch, k2m, km`
`load`	`l`
`mpchtokms`	`mpch2kms, m2k, mk`
`multiply`	`mult`, `m`
`new`	`n`
`normalise`	`normalize`, `norm`
`quit`	`q`
`reset`	`r`
`stat`	`s`
`vgrid`	`v`
`write`	`w`
`writeics`	`wi`

B.5 Supported file formats for density and velocity fields

This section describes the file formats currently supported by ICeCoRe for input and output of overdensity and velocity field data. Please contact the author if you need compatibility with other file formats; additional reading and writing functions for custom formats can be easily added to the ICeCoRe code.

In ICeCoRe, all three-dimensional fields are regularly sampled on a uniform cubic grid with boxsize L and resolution (number of cells per dimension) n. The total number of cells is $N = n^3$. Each cell contains a single number representing the value of the field at this position (more precisely, the value of the field averaged over the cell volume). Reading such fields from files is invoked by the commands `load dgrid` *`format`* and `load vgrid` *`format`*, respectively; writing to files is done with `write dgrid` *`format`* and `write vgrid` *`format`*, respectively. For each `dgrid` object, i.e. overdensity grid $\delta(\boldsymbol{x})$, one file is read/written, with the file name as an argument. For each `vgrid` object, i.e. velocity/displacement field $\boldsymbol{\psi}(\boldsymbol{x})$, three files are read/written in the exact same format, each containing one cartesian component of the field $\psi_x(\boldsymbol{x})$, $\psi_y(\boldsymbol{x})$, and $\psi_z(\boldsymbol{x})$, with the three filenames as arguments. The internal code unit of ICeCoRe is Mpc/h, i.e. velocity fields $\boldsymbol{u}(\boldsymbol{x})$ are always given in terms of the corresponding linear displacement field $\boldsymbol{\psi}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x})/\dot{a}f$ in units of Mpc/h (and not in km/s). All quantities are assumed to be in the comoving frame; at $z = 0$, comoving and physical frames coincide.

A grid containing a scalar field, here $\delta(\boldsymbol{x})$, $\psi_x(\boldsymbol{x})$, $\psi_y(\boldsymbol{x})$, or $\psi_z(\boldsymbol{x})$, is stored in ICeCoRe as an array of N double-precision floating point complex numbers (see Section B.1). However, the imaginary part is only needed for operations in Fourier space. In real space, the imaginary part is always zero, since the fields represent real-valued physical quantities. The files therefore only contain N real-valued floating point numbers. There are in principle two ways of formatting such files: either as human-readable ASCII files (or "formatted" files, as they are called in Fortran), or as binary files ("unformatted" files in Fortran). Binary files generally require much less disk space and are much faster to read or write; we therefore do not recommend using ASCII files.

There are several different ways to format a binary file; there are notable differences between the C/C++ way and the Fortran way. In C/C++, usually a variable or array is written into/read from a binary file in the same way as it would be represented in computer memory. In Fortran however, each 'write' statement will produce a *record*, which contains not only the binary data/array itself, but also leading and trailing *record markers* indicating its size. Conversely, each Fortran 'read' statement will expect these record markers. On most modern Fortran compilers, these markers will consist of 4-byte integers, whose value equals the size of the enclosed record (without the markers) in bytes. Another difference between C/C++ and Fortran is the order in which the elements of multidimensional arrays are stored. The fields can be represented as three-dimensional arrays of the form $\delta(x, y, z)$. The C/C++ convention uses *row-major* order, where the fastest-running index is the last one, whereas Fortran uses *column-major* order, where the fastest-running index is the first one. Throughout the ICeCoRe code and all supported formats, we ignore these conventions and consistently use column-major order, where x will always be the fastest-running index of the array $\delta(x, y, z)$. The ordering of a grid can be changed manually with the `reorder` command.

B.5.1 Binary format

The basic binary format is used if `binary` is given as the *`format`* option in ICeCoRe's `load` and `write` commands. This format represents the data on a grid as a simple C/C++ binary array of N 4-byte floating point numbers in column-major order. Each such file will be exactly $4N$ bytes long. Although internally, ICeCoRe carries out computations in double precision

(8 bytes per number), the single precision storage format (4 bytes per number) is completely sufficient, since numerical errors acquired during any ICeCoRe computation (for example due to interpolation on the tabulated $P(k)$ when generating ICs) will always be higher than the floating point machine precision, although still negligible for practical purposes. Below is a code example in C for producing a file in the `binary` format in order to get data into ICeCoRe:

```
/* Writing binary format in C */

float data[nz][ny][nx];   /* fill this array with data! */
FILE *file = fopen("data.dat", "wb");
fwrite((void*)data, sizeof(float), nx*ny*nz, file);
fclose(file);
```

Note the reverse order of the array indices; C and C++ use row-major order by default, but x has to be the fastest-running index for ICeCoRe. Reading of a file in this format could be done with

```
/* Reading binary format in C */

float data[nz][ny][nx];
FILE *file = fopen("data.dat", "rb");
fread((void*)data, sizeof(float), nx*ny*nz, file);
fclose(file);
```

In C++, the same would be performed with

```
// Writing binary format in C++

float data[nz][ny][nx];   // fill this array with data!
ofstream file("data.dat", ios::out | ios::binary);
file.write(data, sizeof(float)*nx*ny*nz);
file.close();

// Reading binary format in C++

float data[nz][ny][nx];
ifstream file("data.dat", ios::in | ios::binary);
file.read(data, sizeof(float)*nx*ny*nz);
file.close();
```

In Fortran, this cannot be done directly with read and write statements for unformatted files, because it will add record markers that are incompatible with the `binary` format. Although this could be circumvented by using direct access files, we recommend to use BOV or Grafic formats instead if you want to get data into or out of ICeCoRe with your Fortran code.

Note that since the `binary` format only dumps the data inside the grid, no information about the boxsize or any other parameters is stored. For this reason, when reading `binary` files with ICeCoRe, you have to manually supply the boxsize L in Mpc/h and the resolution n as arguments to the `load` command in order to create a valid `dgrid` or `vgrid` object. Also, when reading `binary` files, ICeCoRe does not check for endianness, so you must make sure that the endianness of the file matches the one of the machine where you are running ICeCoRe. The `load` and `write` commands also offer the `swap` option in order to manually swap the endianness of a `binary` file.

B.5.2 BOV format

The BOV format is used if `bov` is given as the *`format`* option in ICeCoRe's `load` and `write` commands. This format is similar to `binary`, but more flexible. This format is also directly supported by the visualisation software VisIt (available at visit.llnl.gov), which can be used to analyse the fields generated by ICeCoRe.

A grid in BOV format consists of a binary file, in the same format as `binary`, and an auxiliary header text file containing the box parameters. We use the convention to add the file extension `.bov` to the header file and `.bov.data` to the corresponding binary data file. Below is an example BOV header file `dens.bov`:

```
TIME: 0
DATA_FILE: dens.bov.data
DATA_SIZE: 256 256 256
DATA_FORMAT: FLOAT
VARIABLE: Density
DATA_ENDIAN: LITTLE
CENTERING: zonal
BRICK_ORIGIN: 0. 0. 0.
BRICK_SIZE: 160. 160. 160.
BYTE_OFFSET: 0
```

The header file contains the path of the data file `DATA_FILE`, the resolution of the grid `DATA_SIZE`, the format of the stored numbers `DATA_FORMAT` (ICeCoRe will always use `FLOAT`), the endianness `DATA_ENDIAN`, and the boxsize in Mpc/h `BRICK_SIZE`. Another interesting option is `BYTE_OFFSET`, which lets you specify some number of bytes to skip at the beginning of the data file. This can be useful if you are writing binary files with Fortran. For example, the code

```
! write binary file in Fortran 90

real(kind=4) :: data(nx, ny, nz)    ! fill this array with data!
open (unit=100, file='dens.bov.data', status='replace', form='unformatted')
write(100) values
close(100)
```

will write a binary file very similar to the `binary` format, but with leading and trailing 4-byte record markers. If you then set `BYTE_OFFSET: 4` in the corresponding header file `dens.bov`, ICeCoRe can skip the leading record marker and read the binary file correctly.

There are additional options for the BOV header file, which will be ignored by ICeCoRe; see the document 'Getting Data into VisIt' (available at visit.llnl.gov) for details. Since all grids in ICeCoRe are always cubic, the three numbers in `DATA_SIZE` and `BRICK_SIZE`, respectively, must always be equal. The filename argument for ICeCoRe's `load` and `write` commands will always be the header file, not the data file; the data file will be generated automatically by `write`. In the current ICeCoRe version 1.0, both the header file and the data file must be in the same directory as the ICeCoRe executable in order to work.

B.5.3 Grafic format

The Grafic format is used for grids describing cosmological density and velocity fields if `bov` is given as the *`format`* option in ICeCoRe's `load` and `write` commands. This format is useful to store cosmological ICs and can be directly used to start a simulation with the Ramses code

if the data was written correctly. The description reproduced here can be found in the on-line GRAFIC code documentation at www.projet-horizon.fr .

A GRAFIC file is a little-endian unformatted FORTRAN file, i.e. a binary file with FORTRAN record markers. The file begins with a 44-byte header record with the following structure:

- `np1`, `np2`, `np3`: three 4-byte integers describing the resolution n of the grid per dimension. For ICECORE, all three must be equal.
- `dx`: 4-byte float describing the size Δx of a grid cell in Mpc (*not* in Mpc/h !), where $\Delta x = L/n$.
- `x1o, x2o, x3o`: three 4-byte floats describing the coordinates of the grid origin. For ICECORE, this will be (0,0,0).
- `astart`: 4-byte float describing the cosmological scale factor a.
- `omegam`: 4-byte float describing the cosmological parameter $\Omega_{\rm m}$.
- `omegal`: 4-byte float describing the cosmological parameters Ω_Λ.
- `h0`: 4-byte float describing the Hubble constant in units of km s^{-1} Mpc^{-1}.

The data is stored after the header record. It consists of `np3` records describing data planes perpendicular to the z axis, each such record containing an array of `np1*np2` 4-byte floats that corresponds to an xy-slice. Such files can be read and written easily in FORTRAN:

```
! read grafic file in Fortran 90

integer(kind=4) :: np1, np2, np3
real(kind=4)    :: dx, x1o, x2o, x3o, astart, omegam, omegal, h0
open(unit=100, file='ic_deltab', status='read', form='unformatted')
! read header
read (10) np1, np2, np3, dx, x1o, x2o, x3o, astart, omegam, omegal, h0
allocate(f(np1, np2, np3))
! read data
do i3=1, np3
  read(10) ((f(i1, i2, i3), i1=1, np1), i2=1, np2)
end do
close(10)
```

In C/C++, reading and writing such FORTRAN-style unformatted files is a little bit more difficult, because we have to explicitly take the record markers into account. There will be 4-byte record markers around the header record and around each of the n arrays which contain the xy-slices of the grid.

In order to start a hydrodynamical RAMSES simulation with uniform ICs, seven input files are required:

- `ic_deltab` : baryon density, used to set up the initial gas density field
- `ic_velbx`, `ic_velby`, `ic_velbz` : baryon velocity field, used to set up the initial gas velocity field
- `ic_velcx`, `ic_velcy`, `ic_velcz` : dark matter velocity field, used to set up the initial dark matter particle positions and velocities

In order to start a dark matter-only simulation, the files `ic_velbx`, `ic_velby`, `ic_velbz` are not required. The file `ic_deltab` is required to start the code, but the values of its data field are not going to be used. Only the fields in the files `ic_velcx`, `ic_velcy`, `ic_velcz` will be used to set up the ICs. Such ICs can be directly written from ICeCoRe. Below is an example script for random ICs at $z_{\rm init} = 30$:

```
# Generate random ICs for a RAMSES dark matter-only simulation

new cosmology 0.272 0.728 70.2 30  # compute cosmology from parameters
load pk WMAP7.dat                  # load tabulated power spectrum
normalise pk 0.807                 # normalise to sigma8
new dgrid 256 100                  # alloc grid with N=256^3, L=100 Mpc/h
seed 12345                         # generate white noise field
colour                             # compute deltaRR
vsolve                             # compute linear displacement field
multiply vgrid mpchtokms           # convert displacement to velocity !!!
write dgrid grafic ic_deltab       # this is needed but will not be used
write vgrid grafic ic_velcx ic_velcy ic_velcz  # these are the actual ICs
quit
```

The Ramses code expects that the velocity field in `ic_velcx`, `ic_velcy`, `ic_velcz` is in units of comoving km/s, so it has to be converted from Mpc/h to km/s in ICeCoRe before writing. Because `write dgrid grafic` and `write vgrid grafic` must include the header with the cosmological parameters in the output files, the `cosmology` object has to be there and initialised with the correct parameters and $z_{\rm init}$, otherwise writing to `grafic` will fail.

B.5.4 Grafic white noise format

The Grafic white noise format is a modification of the standard Grafic format which is useful for white noise files. It will be used if `graficwn` is given as the *`format`* option in ICeCoRe's `load` and `write` commands. It is the default method to get a white noise field into other IC-generating codes (see Section 3.5 and Example 6 in Section A.4.1). Besides the Grafic code, for which this format was designed, and ICeCoRe, most other IC-generating codes understand the `graficwn` format as well, such as Grafic-2, Mpgrafic, Music, and Ginnungagap. The only difference between `grafic` and `graficwn` formats is that `graficwn` uses a different, smaller header:

- `np1`, `np2`, `np3`: three 4-byte integers describing the resolution n of the grid per dimension. For ICeCoRe, all three must be equal.
- `iseed`: 4-byte integer describing the seed of the random number generator.

This header does not contain any cosmological parameters, therefore the `cosmology` object is not needed to write `graficwn`, as opposed to `grafic`. Another detail is that ICeCoRe does *not* store the seed integer in the header, but instead writes a zero in the `iseed` variable. The reason is that the same seed integer will produce different fields on different codes and platforms anyway and therefore has little meaning by itself.

B.5.5 ASCII format

The ASCII format makes it possible to use text files for storing grids as an alternative if working with binary files is not possible for some reason. Reading and writing grids in ASCII format is

done by typing `ascii` as the *`format`* option in ICECORE's `load` and `write` commands. Reading a grid in ASCII format assumes that the numbers corresponding to the grid cell values are stored as human-readable floating point numbers in a text file. The numbers must be separated by whitespace, tab or newline characters; no other delimiters, such as commas or semicolons, are allowed. However, whether one uses whitespaces, tabs, newlines, or any combination of those, does not make any difference for reading grids, as long as there are exactly $n^3 = N$ numbers in the file. The ASCII files written by ICECORE feature a combination of whitespaces and newlines for "historical" reasons (compatibility with an older code that was used by collaborators). However, reading and writing such files should be no problem in virtually any programming language or analysis software. When reading grids in ASCII format with ICECORE, the boxsize L in Mpc/h and resolution n have to be provided as arguments to the commands `load dgrid ascii` and `load vgrid ascii`, respectively.

Reading and writing ASCII files will be considerably slower than for binary files; also, they will use much more disk space. We therefore do not recommend to use them.

B.6 List of keywords and command syntax

Objects

constraints
correlator
corrmatrix
cosmology
dgrid
eta
invmatrix
pk
vgrid

Command syntax

add dgrid *format file* [*options*]
add vgrid *format vx_file vy_file vz_file* [*options*]
addconst *number*
addvector { constraints2 | vgrid } *vx vy vz*
bin { dgrid | vgrid } *format file resolution mas_type*
binpk *file* [*n_bins* [*correction_type*]]
chisquare
colour
cr [*sigma_nl*]
crop { dgrid | vgrid } *new_resolution x_offset y_offset z_offset*
crv [*sigma_nl*]
degrade { dgrid | vgrid } *new_resolution*
deoverlap [*factor*]
dsolve
filter { dgrid | vgrid } *filter_type radius*
hann
help [*topic*]
histogram { dgrid | vgrid } *file min max n_bins*
kdegrade { dgrid | vgrid } *new_resolution*
load constraints *file*
load correlator { analytic | grid }
load corrmatrix *format file* [retain]
load cosmology *format file*
load dgrid *format file* [*options*]
load invmatrix *format file* [retain]
load pk *file*
load vgrid *format vx_file vy_file vz_file* [*options*]
multiply { constraints | constraints2 | dgrid | pk | vgrid } *factor*
new correlator { analytic | grid } [*k_min k_max* [*R_G_max*]]
new corrmatrix [retain]
new cosmology *omega_m omega_L H_0 z*
new dgrid *resolution boxsize*
new invmatrix [*sigma_nl*] [retain]

```
new vgrid resolution boxsize
normalise sigma_8 [ k_min k_max ]
peakfind file
quit
reorder { dgrid | vgrid }
reset [ object ]
seed integer
shift constraints { vector x y z  | vgrid }
sparse r_min
stat [ object ]
vsolve
whiten
write constraints file
write corrmatrix format file
write correlator
write dgrid format file [ swap ]
write eta format file
write invmatrix format file
write pk file
write vgrid format vx_file vy_file vz_file [ swap ]
writeics format file
xcorr out_file n_bins format in_file [ options ]
```

Tokens for numerical values

```
growthd
growthf
kl
kl2
kl3
kmaxpk
kminpk
kmstompch
kny
kny2
kny3
mpchtokms
```

Keyword aliases

```
colour = color
constraints = co
constraints2 = co2
correlator = corr
corrmatrix = corrm
cosmology = cosmo = c
dgrid = d
filter = f
help = h
histogram = hist
invmatrix = invm
kmstompch = kms2mpch = k2m = km
load = l
mpchtokms = mpch2kms = m2k = mk
multiply = mult = m
new = n
normalise = normalize = norm
quit = q
reset = r
stat = s
vgrid = v
write = w
writeics = wi
```

Appendix C

Abbreviations and acronyms

2dF	Two-degree-Field Galaxy Redshift Survey
2LPT	second-order Lagrangian perturbation theory
2MASS	Two Micron All Sky Survey
AHF	AMIGA's Halo Finder
Amiga	Adaptive Mesh Investigations of Galaxy Assembly
AMR	adaptive mesh refinement
Art	Adaptive Refinement Tree
ASCII	American Standard Code for Information Interchange
BAO	baryon acoustic oscillations
BBN	Big Bang nucleosynthesis
BDM	Bound Density Maxima
BOV	Brick Of Values
CAMB	Code for Anisotropies in the Microwave Background
CDM	cold dark matter
Cen	Centaurus
CLUES	Constrained Local UniversE Simulations
CMB	cosmic microwave background
COBE	Cosmic Background Explorer
CR	constrained realisation
DFT	discrete Fourier Transform
EDD	Extragalactic Distance Database
FFT	Fast Fourier Transform
FFTW	Fastest Fourier Transform in the West
FLRW	Firedmann-Lemaître-Robertson-Walker
FoF	Friends-of-Friends
For	Fornax
FT	Fourier Transform
GA	Great Attractor
Gadget	GAlaxies with Dark matter and Gas intEracT
Grafic	Gaussian Random Field Initial Conditions
GSL	GNU scientific library
GSR	Galactic standard of rest
HDM	hot dark matter
Hya	Hydra

ICs	initial conditions
ICeCoRe	Initial Conditions & Constrained Realisations
IDL	Interactive Data Language
LAPACK	Linear Algebra PACKage
LG	Local Group
LPT	Lagrangian perturbation theory
LS	Local Sheet
LSC	Local Supercluster
LV	Local Void
M31	Andromeda galaxy
M33	Triangulum galaxy
MAK	Monge-Ampère-Kantorovich
MKL	Math Kernel Library
MPI	Message Passing Interface
Music	Multi-Scale Initial Conditions
MV	minimal variance
MW	Milky Way galaxy
NAM	Numerical Action Method
OpenMP	Open Multi-Processing
P^3M	particle-particle/particle-mesh
PBCs	periodic boundary conditions
PDF	probability density function
Per-Psc	Perseus-Pisces
PIZA	Path-Interchange Zeldovich Approximation
PM	particle-mesh
R2LPT	Reverse 2LPT Approximation
RR	random realisation
RZA	Reverse Zeldovich Approximation
SDSS	Sloan Digital Sky Survey
SFB	surface brightness fluctuation
SPH	smooth particle hydrodynamics
SG	supergalactic
TRGB	tip of the red giant branch
UMV	unbiased minimal variance
Vir	Virgo
VV	Virgo Void
WDM	warm dark matter
WF	Wiener filter
WMAP	Wilkinson Microwave Anisotropy Probe
ZA	Zeldovich Approximation
ZoA	Zone of Avoidance
ΛCDM	Lambda cold dark matter

Appendix D

Constants and units

Constant	Symbol	Value in SI units
Vacuum light speed*	c	2.99792458×10^8 m s^{-1}
Gravitational constant*	G	$6.67384 \pm 0.00080 \times 10^{-11}$ m^3 kg^{-1} s^{-2}

Unit name	Symbol	Value in SI units
Megaparsec†	Mpc	1 Mpc $= 10^6$ pc $= 3.08568 \times 10^{19}$ m
Solar mass‡	$M_\odot$	1 $M_\odot = 1.9884 \pm 0.0002 \times 10^{30}$ kg

*from Mohr et al. (2011)
†from Guidry (2010)
‡from Luzum et al. (2011)

Bibliography

Adler, R. J. 1981. *The Geometry of Random Fields.* Chichester: Wiley. 21

Alpher, R. A., Bethe, H., Gamow, G. 1948. *The origin of chemical elements.* Phys. Rev., 73, 803–804. 1

Ampère, A.-M. 1820. *Mémoire concernant . . . l'intégration des équations aux différentielles partielles du prèmier et du second ordre.* Journal de l'École Royale Polytechnique, 11, 1. 79

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D. 1999. *LAPACK Users' Guide.* Third edn. Society for Industrial and Applied Mathematics. 72

Antonuccio-Delogu, V., Becciani, U., van Kampen, E., Pagliaro, A., Romeo, A. B., Colafrancesco, S., Germaná, A., Gambera, M. 2002. *Properties of galaxy haloes in clusters and voids.* MNRAS, 332, 7–20. 60

Appel, A. W. 1985. *An Efficient Program for Many-Body Simulation.* SIAM J. Scientific and Statistical Computing, 6, 85. 32

Bardeen, J. M., Bond, J. R., Kaiser, N., Szalay, A. S. 1986. *The statistics of peaks of Gaussian random fields.* ApJ, 304, 15–61. 21, 22, 23, 41, 57

Barnes, J., Hut, P. 1986. *A hierarchical O(N log N) force-calculation algorithm.* Nature, 324, 446–449. 32

Begeman, K. G. 1989. *H I rotation curves of spiral galaxies. I - NGC 3198.* A&A, 223, 47–60. 17

Berger, M. J., Colella, P. 1989. *Local adaptive mesh refinement for shock hydrodynamics.* J. Comput. Phys., 82, 64âĂŞ84. 32

Bernardeau, F., Colombi, S., Gaztañaga, E., Scoccimarro, R. 2002. *Large-scale structure of the Universe and cosmological perturbation theory.* Physics Reports, 367, 1–248. 30

Bertone, G. 2010. *Particle dark matter: Observations, models and searches.* Cambridge University Press, 1 edition. 31

Bertschinger, E. 1987. *Path integral methods for primordial density perturbations - Sampling of constrained Gaussian random fields.* ApJL, 323, L103–L106. 57

Bertschinger, E. 2001. *Multiscale Gaussian Random Fields and Their Application to Cosmological Simulations.* ApJS, 137, 1–20. 45, 47, 49, 73, 165, 177, 179

Bertschinger, E., Dekel, A., Faber, S. M., Dressler, A., Burstein, D. 1990. *Potential, velocity, and density fields from redshift-distance samples: Application - Cosmography within 6000 kilometers per second.* ApJ, 364, 370–395. 80

Bertsekas, D. P. 1991. *An Auction Algorithm for Shortest Paths.* SIAM J. Optim., 1, 425–447. 79

Binney, J., Quinn, T. 1991. *Gaussian random fields in spherical coordinates.* MNRAS, 249, 678–683. 57

Binney, J., Tremaine, S. 2008. *Galactic Dynamics: Second Edition.* Princeton University Press. 2, 33, 35, 113

Bistolas, V., Hoffman, Y. 1998. *Nonlinear Constrained Realizations of the Large-Scale Structure.* ApJ, 492, 439–+. 61

Bode, P., Ostriker, J. P., Turok, N. 2001. *Halo Formation in Warm Dark Matter Models.* ApJ, 556, 93–107. 30

Bonnor, W. B. 1957. *Jeans' formula for gravitational instability.* MNRAS, 117, 104. 20

Bouchet, F. R., Colombi, S., Hivon, E., Juszkiewicz, R. 1995. *Perturbative Lagrangian approach to gravitational instability.* A&A, 296, 575. 30, 84, 105, 106

Branchini, E., Eldar, A., Nusser, A. 2002. *Peculiar velocity reconstruction with the fast action method: tests on mock redshift surveys.* MNRAS, 335, 53–72. 79

Brenier, Y., Frisch, U., Hénon, M., Loeper, G., Matarrese, S., Mohayaee, R., Sobolevskiĭ, A. 2003. *Reconstruction of the early Universe as a convex optimization problem.* MNRAS, 346, 501–524. 79

Bryan, G. L., Norman, M. L. 1997. *A Hybrid AMR Application for Cosmology and Astrophysics.* ArXiv e-prints. 33

Buchert, T., Ehlers, J. 1993. *Lagrangian theory of gravitational instability of Friedman-Lemaitre cosmologies – second-order approach: an improved model for non-linear clustering.* MNRAS, 264, 375–387. 106

Buchert, T., Melott, A. L., Weiss, A. G. 1994. *Testing higher-order Lagrangian perturbation theory against numerical simulations I. Pancake models.* A&A, 288, 349–364. 30, 106

Burgers, J.M. 1973. *The nonlinear diffusion equation: asymptotic solutions and statistical problems.* D. Reidel Pub. Co. 30

Cen, R., Ostriker, J. P. 1999. *Where Are the Baryons?* ApJ, 514, 1–6. 2

Cen, R., Ostriker, J. P. 2006. *Where Are the Baryons? II. Feedback Effects.* ApJ, 650, 560–572. 2

Chisari, N. E., Zaldarriaga, M. 2011. *Connection between Newtonian simulations and general relativity.* PhRD, 83(12), 123505. 20

Clowe, D., Bradač, M., Gonzalez, A. H., Markevitch, M., Randall, S. W., Jones, C., Zaritsky, D. 2006. *A Direct Empirical Proof of the Existence of Dark Matter.* ApJL, 648, L109–L113. 1, 17

Cole, S., Percival, W. J., Peacock, J. A., Norberg, P., Baugh, C. M., Frenk, C. S., Baldry, I., Bland-Hawthorn, J., Bridges, T., Cannon, R., Colless, M., Collins, C., Couch, W., Cross, N. J. G., Dalton, G., Eke, V. R., De Propris, R., Driver, S. P., Efstathiou, G., Ellis, R. S., Glazebrook, K., Jackson, C., Jenkins, A., Lahav, O., Lewis, I., Lumsden, S., Maddox, S., Madgwick, D., Peterson, B. A., Sutherland, W., Taylor, K. 2005. *The 2dF Galaxy Redshift Survey: power-spectrum analysis of the final data set and cosmological implications.* MNRAS, 362, 505–534. 1

Coles, P., Jones, B. 1991. *A lognormal model for the cosmological mass distribution.* MNRAS, 248, 1–13. 36

Colless, M., Saglia, R. P., Burstein, D., Davies, R. L., McMahan, R. K., Wegner, G. 2001. *The peculiar motions of early-type galaxies in two distant regions - VII. Peculiar velocities and bulk motions.* MNRAS, 321, 277–305. 11

Cooley, J. W., Tukey, J. W. 1965. *An algorithm for the machine calculation of complex Fourier series.* Math. Comp., 19, 297=301. 45

Courtois, H. M., Tully, R. B. 2012. *Cosmicflows-2: Type Ia Supernova Calibration and* H_0. ApJ, 749, 174. 6, 11, 55, 139

Courtois, H. M., Tully, R. B., Makarov, D. I., Mitronova, S., Koribalski, B., Karachentsev, I. D., Fisher, J. R. 2011a. *Cosmic Flows: Green Bank Telescope and Parkes H I observations.* MNRAS, 414, 2005–2016. 6, 11, 55, 139

Courtois, H. M., Tully, R. B., Héraudeau, P. 2011b. *Cosmic flows: University of Hawaii 2.2-m I-band photometry.* MNRAS, 415, 1935–1942. 6, 11, 55, 139

Courtois, H. M., Hoffman, Y., Tully, R. B., Gottlober, S. 2012. *Three-dimensional velocity and density reconstructions of the local universe with cosmicflows-1.* ApJ, 744(1), 43. 6, 12, 13, 14, 53, 55, 57, 87, 88, 93, 97, 103, 119, 131

Crain, R. A., Theuns, T., Dalla Vecchia, C., Eke, V. R., Frenk, C. S., Jenkins, A., Kay, S. T., Peacock, J. A., Pearce, F. R., Schaye, J., Springel, V., Thomas, P. A., White, S. D. M., Wiersma, R. P. C. 2009. *Galaxies-intergalactic medium interaction calculation - I. Galaxy formation as a function of large-scale environment.* MNRAS, 399, 1773–1794. 2, 73

Crocce, M., Scoccimarro, R. 2006. *Memory of initial conditions in gravitational clustering.* PhRD, 73(6), 063520. 36

Crocce, M., Pueblas, S., Scoccimarro, R. 2006. *Transients from initial conditions in cosmological simulations.* MNRAS, 373, 369–381. 47

Croft, R. A. C., Gaztañaga, E. 1997. *Reconstruction of cosmological density and velocity fields in the Lagrangian Zel'dovich approximation.* MNRAS, 285, 793–805. 79

Dehnen, W., Read, J. I. 2011. *N-body simulations of gravitational dynamics.* European Physical Journal Plus, 126, 55. 32

Djorgovski, S., Davis, M. 1987. *Fundamental properties of elliptical galaxies.* ApJ, 313, 59–68. 11

Dolag, K., Stasyszyn, F. 2009. *An MHD GADGET for cosmological simulations.* MNRAS, 398, 1678–1697. 2

Doumler, T., Knebe, A. 2010. *Investigating the influence of magnetic fields upon structure formation with AMIGA - a C code for cosmological magnetohydrodynamics.* MNRAS, 403, 453–473. 2, 29

Doumler, T., Hoffman, Y., Courtois, H., Gottlöber, S. 2012a. *Reconstructing cosmological initial conditions from galaxy peculiar velocities. I. Reverse Zeldovich Approximation.* MNRAS, submitted. 223

Doumler, T., Courtois, H., Gottlöber, S., Hoffman, Y. 2012b. *Reconstructing cosmological initial conditions from galaxy peculiar velocities. II. The effect of observational errors.* MNRAS, submitted. 223

Doumler, T., Gottlöber, S., Hoffman, Y., Courtois, H. 2012c. *Reconstructing cosmological initial conditions from galaxy peculiar velocities. III. Constrained simulations.* MNRAS, submitted. 223

Dubois, Y., Teyssier, R. 2008. *Supernova Feedback in Galaxy Formation. Page 388 of:* J. H. Knapen, T. J. Mahoney, & A. Vazdekis (ed), *Pathways through an eclectic universe.* Astronomical Society of the Pacific Conference Series, vol. 390. 2

Efstathiou, G., Davis, M., White, S. D. M., Frenk, C. S. 1985. *Numerical techniques for large cosmological N-body simulations.* ApJS, 57, 241–260. 2, 32, 46

Einstein, A. 1914. *Die formale Grundlage der allgemeinen Relativitätstheorie.* Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin), 1030–1085. 16

Einstein, A. 1915a. *Die Feldgleichungen der Gravitation.* Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin), 844–847. 16

Einstein, A. 1915b. *Zur allgemeinen Relativitätstheorie.* Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin), 778–786. 16

Einstein, A. 1916. *Die Grundlage der allgemeinen Relativitätstheorie.* Annalen der Physik, 354, 769–822. 16

Eisenstein, D. J., Hu, W. 1998. *Baryonic Features in the Matter Transfer Function.* ApJ, 496, 605–+. 23

Eisenstein, D. J., Seo, H.-J., Sirko, E., Spergel, D. N. 2007. *Improving Cosmological Distance Measurements by Reconstruction of the Baryon Acoustic Peak.* ApJ, 664, 675–679. 78, 79

Erdoğdu, P., Lahav, O., Huchra, J. P., Colless, M., Cutri, R. M., Falco, E., George, T., Jarrett, T., Jones, D. H., Macri, L. M., Mader, J., Martimbeau, N., Pahre, M. A., Parker, Q. A., Rassat, A., Saunders, W. 2006. *Reconstructed density and velocity fields from the 2MASS Redshift Survey.* MNRAS, 373, 45–64. 14, 53

Faber, S. M., Jackson, R. E. 1976. *Velocity dispersions and mass-to-light ratios for elliptical galaxies.* ApJ, 204, 668–683. 11

Falck, B. L., Neyrinck, M. C., Aragon-Calvo, M. A., Lavaux, G., Szalay, A. S. 2012. *Straightening the Density-Displacement Relation with a Logarithmic Transform.* ApJ, 745, 17. 79

Feldman, H. A., Watkins, R., Hudson, M. J. 2010. *Cosmic flows on 100 h^{-1} Mpc scales: standardized minimum variance bulk flow, shear and octupole moments.* MNRAS, 407, 2328–2338. 13

Fisher, K. B., Lahav, O., Hoffman, Y., Lynden-Bell, D., Zaroubi, S. 1995. *Wiener reconstruction of density, velocity and potential fields from all-sky galaxy redshift surveys.* MNRAS, 272, 885–908. 53

Fixsen, D. J., Kashlinsky, A. 2011. *Probing the Universe's Tilt with the Cosmic Infrared Background Dipole.* ApJ, 734, 61. 13

Fixsen, D. J., Cheng, E. S., Gales, J. M., Mather, J. C., Shafer, R. A., Wright, E. L. 1996. *The Cosmic Microwave Background Spectrum from the Full COBE FIRAS Data Set.* ApJ, 473, 576. 2, 11, 138

Forero-Romero, J. E., Hoffman, Y., Yepes, G., Gottlöber, S., Piontek, R., Klypin, A., Steinmetz, M. 2011. *The dark matter assembly of the Local Group in constrained cosmological simulations of a Λ cold dark matter universe.* MNRAS, 417, 1434–1443. 4, 14

Fouqué, P., Solanes, J. M., Sanchis, T., Balkowski, C. 2001. *Structure, mass and distance of the Virgo cluster from a Tolman-Bondi model.* A&A, 375, 770–780. 12, 87, 131

Freedman, W. L., Madore, B. F. 1990. *An empirical test for the metallicity sensitivity of the Cepheid period-luminosity relation.* ApJ, 365, 186–194. 11

Freedman, W. L., Madore, B. F., Gibson, B. K., Ferrarese, L., Kelson, D. D., Sakai, S., Mould, J. R., Kennicutt, Jr., R. C., Ford, H. C., Graham, J. A., Huchra, J. P., Hughes, S. M. G., Illingworth, G. D., Macri, L. M., Stetson, P. B. 2001. *Final Results from the Hubble Space Telescope Key Project to Measure the Hubble Constant.* ApJ, 553, 47–72. 10, 11

Freeman, K. C. 1970. *On the Disks of Spiral and so Galaxies.* ApJ, 160, 811. 1, 17

Friedmann, A. 1922. *Über die Krümmung des Raumes.* Zeitschrift für Physik, 10(1), 377–386. 16

Friedmann, A. 1924. *Über die Möglichkeit einer Welt mit konstanter negativer Krümmung des Raumes.* Zeitschrift für Physik, 21(1), 326–332. 16

Frigo, M., Johnson, S. G. 2005. *The Design and Implementation of FFTW3.* Proceedings of the IEEE, 93(2), 216–231. 45

Frisch, U., Matarrese, S., Mohayaee, R., Sobolevski, A. 2002. *A reconstruction of the initial conditions of the Universe by optimal mass transportation.* Nature, 417, 260–262. 79

Fromang, S., Hennebelle, P., Teyssier, R. 2006. *A high order Godunov scheme with constrained transport and adaptive mesh refinement for astrophysical magnetohydrodynamics.* A&A, 457, 371–384. 2

Gnedin, N. Y., Kravtsov, A. V., Rudd, D. H. 2011. *Implementing the DC Mode in Cosmological Simulations with Supercomoving Variables.* ApJS, 194, 46. 20, 49

Golub, G.H., Loan, C.F.V. 1996. *Matrix computations.* Johns Hopkins studies in the mathematical sciences. Johns Hopkins University Press. 72

Gorski, K. 1988. *On the pattern of perturbations of the Hubble flow.* ApJL, 332, L7–L11. 26

Gottlöber, S., Hoffman, Y., Yepes, G. 2010. *Constrained Local UniversE Simulations (CLUES).* ArXiv e-prints. 4, 61, 62, 63, 73, 74, 107, 112, 117, 120, 141, 145

Guidry, M. 2010. *Astronomical Distance Scales.* University of Tennessee, Knoxville. 202

Gurbatov, S. N., Saichev, A. I. 1984. *Probability distributions and spectra of potential hydrodynamic turbulence.* Radiofizika, 27, 456–468. 30

Hahn, O., Abel, T. 2011. *Multi-scale initial conditions for cosmological simulations.* MNRAS, 415, 2101–2121. 47, 49, 65, 74, 75

Hamana, T., Kayo, I., Yoshida, N., Suto, Y., Jing, Y. P. 2003. *Modelling peculiar velocities of dark matter haloes.* MNRAS, 343, 1312–1318. 40, 41, 61

Hamana, T., Kayo, I., Yoshida, N., Suto, Y., Jing, Y. P. 2005. *Erratum: Modelling peculiar velocities of dark matter haloes.* MNRAS, 357, 1407–1408. 40, 41

Hamilton, A. J. S. 2000. *Uncorrelated modes of the non-linear power spectrum.* MNRAS, 312, 257–284. 65

Hamilton, A. J. S. 2001. *Formulae for growth factors in expanding universes containing matter and a cosmological constant.* MNRAS, 322, 419–425. 25

Hansen, S. H., Agertz, O., Joyce, M., Stadel, J., Moore, B., Potter, D. 2007. *An Alternative to Grids and Glasses: Quaquaversal Pre-Initial Conditions for N-Body Simulations.* ApJ, 656, 631–635. 47

Harrison, E. R. 1970. *Fluctuations at the threshold of classical cosmology.* PhRD, 1, 2726–2730. 22

Heath, D. J. 1977. *The growth of density perturbations in zero pressure Friedmann-Lemaitre universes.* MNRAS, 179, 351–358. 20

Heitmann, K., Ricker, P. M., Warren, M. S., Habib, S. 2005. *Robustness of Cosmological Simulations. I. Large-Scale Structure.* ApJS, 160, 28–58. 33

Heitmann, K., Lukić, Z., Fasel, P., Habib, S., Warren, M. S., White, M., Ahrens, J., Ankeny, L., Armstrong, R., O'Shea, B., Ricker, P. M., Springel, V., Stadel, J., Trac, H. 2008. *The cosmic code comparison project.* Computational science and discovery, 1(1), 015003. 33

Hernquist, L., Katz, N. 1989. *TREESPH - A unification of SPH with the hierarchical tree method.* ApJS, 70, 419–446. 75

Hockney, R.W., Eastwood, J.W. 1992. *Computer simulation using particles.* Institute of Physics. 32, 47, 167

Hoffman, Y. 2009. *Gaussian Fields and Constrained Simulations of the Large-Scale Structure. Pages 565–583 of:* V. J. Martínez, E. Saar, E. Martínez-González, & M.-J. Pons-Bordería (ed), *Data analysis in cosmology.* Lecture Notes in Physics, Berlin Springer Verlag, vol. 665. 50

Hoffman, Y., Ribak, E. 1991. *Constrained realizations of Gaussian fields - A simple algorithm.* ApJL, 380, L5–L8. 4, 58, 139

Hoffman, Y., Ribak, E. 1992. *Primordial Gaussian perturbation fields - Constrained realizations.* ApJ, 384, 448–452. 57, 58

Hoffman, Y., Lahav, O., Yepes, G., Dover, Y. 2007. *The future of the local large scale structure: the roles of dark matter and dark energy.* JCAP, 10, 16. 12

Hubble, E. 1929. *A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae.* Proceedings of the National Academy of Science, 15, 168–173. 9

Iliev, I. T., Moore, B., Gottlöber, S., Yepes, G., Hoffman, Y., Mellema, G. 2011. *Reionization of the Local Group of galaxies.* MNRAS, 413, 2093–2102. 2, 4, 31

Jackson, J. C. 1972. *A critique of Rees's theory of primordial gravitational radiation.* MNRAS, 156, 1P. 42

Jha, S., Riess, A. G., Kirshner, R. P. 2007. *Improved Distances to Type Ia Supernovae with Multicolor Light-Curve Shapes: MLCS2k2.* ApJ, 659, 122–148. 11

Jing, Y. P. 2005. *Correcting for the Alias Effect When Measuring the Power Spectrum Using a Fast Fourier Transform.* ApJ, 620, 559–563. 39, 168, 175

Joachimi, B., Taylor, A. N., Kiessling, A. 2011. *Cosmological information in Gaussianized weak lensing signals.* MNRAS, 418, 145–169. 109

Kaiser, N. 1987. *Clustering in real space and in redshift space.* MNRAS, 227, 1–21. 42

Kantorovich, L. 1942. *On the translocation of masses.* Doklady Akademii Nauk SSSR, 37, 199. 79

Karachentsev, I. D., Makarov, D. I., Sharina, M. E., Dolphin, A. E., Grebel, E. K., Geisler, D., Guhathakurta, P., Hodge, P. W., Karachentseva, V. E., Sarajedini, A., Seitzer, P. 2003. *Local galaxy flows within 5 Mpc.* A&A, 398, 479–491. 12

Karachentsev, I. D., Karachentseva, V. E., Huchtmeier, W. K., Makarov, D. I. 2004. *A Catalog of Neighboring Galaxies.* AJ, 127, 2031–2068. 2, 11, 128

Kashlinsky, A., Atrio-Barandela, F., Ebeling, H., Edge, A., Kocevski, D. 2010. *A New Measurement of the Bulk Flow of X-Ray Luminous Clusters of Galaxies.* ApJL, 712, L81–L85. 13

Katz, N., Quinn, T., Bertschinger, E., Gelb, J. M. 1994. *Formation of Quasars at High Redshift.* MNRAS, 270, L71. 73

Khinchin, A. 1934. *Korrelationstheorie der stationÃďren stochastischen Prozesse.* Mathematische Annalen, 109. 22

Kitaura, F.-S. 2012. *The Initial Conditions of the Universe from Constrained Simulations.* ArXiv e-prints. 79, 106, 109, 125, 144

Kitaura, F.-S., Angulo, R. E. 2012. *Linearization with cosmological perturbation theory.* MNRAS, 425, 2443–2454. 106, 107, 109, 125

Kitaura, F. S., Enßlin, T. A. 2008. *Bayesian reconstruction of the cosmological large-scale structure: methodology, inverse algorithms and numerical optimization.* MNRAS, 389, 497–544. 43

Kitaura, F. S., Jasche, J., Li, C., Enßlin, T. A., Metcalf, R. B., Wandelt, B. D., Lemson, G., White, S. D. M. 2009. *Cosmic cartography of the large-scale structure with Sloan Digital Sky Survey data release 6.* MNRAS, 400, 183–203. 53

Kitaura, F.-S., Jasche, J., Metcalf, R. B. 2010. *Recovering the non-linear density field from the galaxy distribution with a Poisson-lognormal filter.* MNRAS, 403, 589–604. 43

Kitaura, F.-S., Angulo, R. E., Hoffman, Y., Gottlöber, S. 2012. *Estimating cosmic velocity fields from density fields and tidal tensors.* MNRAS, 425, 2422–2435. 37, 106, 107, 109, 125

Klar, J. S., Mücket, J. P. 2012. *Filaments and sheets of the warm-hot intergalactic medium.* MNRAS, 423, 304–319. 2

Klypin, A., Holtzman, J. 1997. *Particle-Mesh code for cosmological simulations.* ArXiv e-prints. 48, 184

Klypin, A., Gottlöber, S., Kravtsov, A. V., Khokhlov, A. M. 1999a. *Galaxies in N-Body Simulations: Overcoming the Overmerging Problem.* ApJ, 516, 530–551. 41

Klypin, A., Kravtsov, A. V., Valenzuela, O., Prada, F. 1999b. *Where Are the Missing Galactic Satellites?* ApJ, 522, 82–92. 33

Klypin, A., Hoffman, Y., Kravtsov, A. V., Gottlöber, S. 2003. *Constrained Simulations of the Real Universe: The Local Supercluster.* ApJ, 596, 19–33. 4, 61, 112, 117, 120, 141

Klypin, A. A., Trujillo-Gomez, S., Primack, J. 2011. *Dark Matter Halos in the Standard Cosmological Model: Results from the Bolshoi Simulation.* ApJ, 740, 102. 73

Knebe, A. 2007. *Computational Cosmology.* Lecture notes, University of Potsdam. 45

Knebe, A., Green, A., Binney, J. 2001. *Multi-level adaptive particle mesh (MLAPM): a c code for cosmological simulations.* MNRAS, 325, 845–864. 47

Knebe, A., Wagner, C., Knollmann, S., Diekershoff, T., Krause, F. 2009. *On the Starting Redshift Cosmological Simulations: Focusing on Halo Properties.* ApJ, 698, 266–274. 47

Knebe, A., Knollmann, S. R., Muldrew, S. I., Pearce, F. R., Aragon-Calvo, M. A., Ascasibar, Y., Behroozi, P. S., Ceverino, D., Colombi, S., Diemand, J., Dolag, K., Falck, B. L., Fasel, P., Gardner, J., Gottlöber, S., Hsu, C.-H., Iannuzzi, F., Klypin, A., Lukić, Z., Maciejewski, M., McBride, C., Neyrinck, M. C., Planelles, S., Potter, D., Quilis, V., Rasera, Y., Read, J. I., Ricker, P. M., Roy, F., Springel, V., Stadel, J., Stinson, G., Sutter, P. M., Turchaninov, V., Tweed, D., Yepes, G., Zemp, M. 2011a. *Haloes gone MAD: The Halo-Finder Comparison Project.* MNRAS, 415, 2293–2318. 35, 41

Knebe, A., Libeskind, N. I., Doumler, T., Yepes, G., Gottlöber, S., Hoffman, Y. 2011b. *Renegade subhaloes in the Local Group.* MNRAS, 417, L56–L60. 4

Knebe, A., Libeskind, N. I., Knollmann, S. R., Martinez-Vaquero, L. A., Yepes, G., Gottlöber, S., Hoffman, Y. 2011c. *The luminosities of backsplash galaxies in constrained simulations of the Local Group.* MNRAS, 412, 529–536. 4

Knollmann, S. R., Knebe, A. 2009. *AHF: Amiga's Halo Finder.* ApJS, 182, 608–624. 35, 83

Kocevski, D. D., Ebeling, H. 2006. *On the Origin of the Local Group's Peculiar Velocity.* ApJ, 645, 1043–1053. 13

Kocevski, D. D., Ebeling, H., Mullis, C. R., Tully, R. B. 2007. *A Systematic X-Ray Search for Clusters of Galaxies behind the Milky Way. II. The Second CIZA Subsample.* ApJ, 662, 224–235. 13

Kolatt, T., Dekel, A., Ganon, G., Willick, J. A. 1996. *Simulating Our Cosmological Neighborhood: Mock Catalogs for Velocity Analysis.* ApJ, 458, 419. 61

Kolb, E. W., Salopek, D. S., Turner, M. S. 1990. *Origin of density fluctuations in extended inflation.* PhRD, 42, 3925–3935. 21

Komatsu, E., Dunkley, J., Nolta, M. R., Bennett, C. L., Gold, B., Hinshaw, G., Jarosik, N., Larson, D., Limon, M., Page, L., Spergel, D. N., Halpern, M., Hill, R. S., Kogut, A., Meyer, S. S., Tucker, G. S., Weiland, J. L., Wollack, E., Wright, E. L. 2009. *Five-Year Wilkinson Microwave Anisotropy Probe Observations: Cosmological Interpretation.* ApJS, 180, 330–376. 18

Komatsu, E., Smith, K. M., Dunkley, J., Bennett, C. L., Gold, B., Hinshaw, G., Jarosik, N., Larson, D., Nolta, M. R., Page, L., Spergel, D. N., Halpern, M., Hill, R. S., Kogut, A., Limon, M., Meyer, S. S., Odegard, N., Tucker, G. S., Weiland, J. L., Wollack, E., Wright, E. L. 2011. *Seven-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation.* ApJS, 192, 18. 1, 10, 17, 18, 22, 23

Kravtsov, A. V., Klypin, A. A., Khokhlov, A. M. 1997. *Adaptive Refinement Tree: A New High-Resolution N-Body Code for Cosmological Simulations.* ApJS, 111, 73. 2, 33

Kreyszig, E. 2006. *Advanced engineering mathematics.* John Wiley. 72

Lahav, O., Lilje, P. B., Primack, J. R., Rees, M. J. 1991. *Dynamical effects of the cosmological constant.* MNRAS, 251, 128–136. 25

Lahav, O., Santiago, B. X., Webster, A. M., Strauss, M. A., Davis, M., Dressler, A., Huchra, J. P. 2000. *The supergalactic plane revisited with the Optical Redshift Survey.* MNRAS, 312, 166–176. 56

Landau, L. D., Lifshitzïÿą, E. M., Pitaevskiĭ, L. P. 1980. *Statistical physics, Volume 2.* Butterworth-Heinemann. 22

Lavaux, G. 2010. *Precision constrained simulation of the local Universe.* MNRAS, 406, 1007–1013. 63, 64, 68, 70, 72, 79, 80, 112, 113, 120, 121, 124, 125

Lavaux, G., Mohayaee, R., Colombi, S., Tully, R. B., Bernardeau, F., Silk, J. 2008. *Observational biases in Lagrangian reconstructions of cosmic velocity fields.* MNRAS, 383, 1292–1318. 43, 79

Lavaux, G., Tully, R. B., Mohayaee, R., Colombi, S. 2010. *Cosmic Flow From Two Micron All-Sky Redshift Survey: the Origin of Cosmic Microwave Background Dipole and Implications for ΛCDM Cosmology.* ApJ, 709, 483–498. 13, 14, 43, 79, 138

Leeuwin, F., Combes, F., Binney, J. 1993. *N-body simulations with perturbation particles. I - Method and tests.* MNRAS, 262, 1013–1022. 32

Lemaître, G. 1927. *Un univers de masse constante et de rayon croissant, rendant compte de la vitesse radial de nebuleuses extragalactiques.* Ann. Soc. Sci. Bruxelles A, 47, 47–59. 16

Lemaître, G. 1931. *Expansion of the universe, The expanding universe.* MNRAS, 91, 490–501. 16

Lewis, A., Challinor, A., Lasenby, A. 2000. *Efficient Computation of Cosmic Microwave Background Anisotropies in Closed Friedmann-Robertson-Walker Models.* ApJ, 538, 473–476. 23

Li, Y.-S., White, S. D. M. 2008. *Masses for the Local Group and the Milky Way.* MNRAS, 384, 1459–1468. 144

Libeskind, N. I., Yepes, G., Knebe, A., Gottlöber, S., Hoffman, Y., Knollmann, S. R. 2010. *Constrained simulations of the Local Group: on the radial distribution of substructures.* MNRAS, 401, 1889–1897. 4

Lightman, A. P., Schechter, P. L. 1990. *The Omega dependence of peculiar velocities induced by spherical density perturbations.* ApJS, 74, 831. 25

Lilje, P. B., Yahil, A., Jones, B. J. T. 1986. *The tidal velocity field in the Local Supercluster.* ApJ, 307, 91–96. 13

Ludlow, A. D., Porciani, C. 2011. *The peaks formalism and the formation of cold dark matter haloes.* MNRAS, 413, 1961–1972. 36, 82, 136, 142, 145

Luzum, B., Capitaine, N., Fienga, A., Folkner, W., Fukushima, T., Hilton, J., Hohenkerk, C., Krasinsky, G., Petit, G., Pitjeva, E., Soffel, M., Wallace, P. 2011. *The IAU 2009 system of astronomical constants: the report of the IAU Working Group on Numerical Standards for Fundamental Astronomy.* Celest. Mech. Dyn. Astr., 110(10), 293–304. 202

Lynden-Bell, D., Faber, S. M., Burstein, D., Davies, R. L., Dressler, A., Terlevich, R. J., Wegner, G. 1988. *Spectroscopy and photometry of elliptical galaxies. V - Galaxy streaming toward the new supergalactic center.* ApJ, 326, 19–49. 13

Ma, Y.-Z., Gordon, C., Feldman, H. A. 2011. *Peculiar velocity field: Constraining the tilt of the Universe.* PhRD, 83(10), 103002. 13

Ma, Z. 2007. *The Nonlinear Matter Power Spectrum.* ApJ, 665, 887–898. 37

Macciò, A. V., Governato, F., Horellou, C. 2005. *The signature of dark energy on the local Hubble flow.* MNRAS, 359, 941–948. 12

Marsaglia, G., Tsang, W. W. 2000. *The Ziggurat Method for Generating Random Variables.* Journal of Statistical Software, 5, 8–+. 45, 185

Martínez-Vaquero, L. A., Yepes, G., Hoffman, Y., Gottlöber, S., Sivan, M. 2009. *Constrained simulations of the local universe - II. The nature of the local Hubble flow.* MNRAS, 397, 2070–2080. 4

Matsumoto, M., Nishimura, T. 1998. *Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator.* ACM Transactions on Modeling and Computer Simulation, 8, 3–30. 185

McBride, C. K., Connolly, A. J., Gardner, J. P., Scranton, R., Scoccimarro, R., Berlind, A. A., Marín, F., Schneider, D. P. 2011. *Three-point Correlation Functions of SDSS Galaxies: Constraining Galaxy-mass Bias.* ApJ, 739, 85. 43

Mohayaee, R., Frisch, U., Matarrese, S., Sobolevskii, A. 2003. *Back to the primordial Universe by a Monge-Ampère-Kantorovich optimization scheme.* A&A, 406, 393–401. 79

Mohr, P. J., Taylor, B. N., Newell, D. B. 2011. *The 2010 CODATA Recommended Values of the Fundamental Physical Constants.* National Institute of Standards and Technology, Gaithersburg, MD 20899. 202

Monge, G. 1781. *Mémoire sur la théorie des déblais et remblais.* Hist. Acad. R. Sci. Paris, 666. 79

Monin, A. S., Yaglom, A. M. 1965. *Statistical Fluid Mechanics.* Moscow: Nauka. 26

Moore, B., Ghigna, S., Governato, F., Lake, G., Quinn, T., Stadel, J., Tozzi, P. 1999. *Dark Matter Substructure within Galactic Halos.* ApJL, 524, L19–L22. 33

Muñoz-Cuartas, J. C., Müller, V., Forero-Romero, J. E. 2011. *Halo-based reconstruction of the cosmic mass density field.* MNRAS, 417, 1303–1317. 43

Navarro, J. F., Frenk, C. S., White, S. D. M. 1997. *A Universal Density Profile from Hierarchical Clustering.* ApJ, 490, 493–+. 35

Neyrinck, M. C., Szapudi, I., Szalay, A. S. 2009. *Rejuvenating the Matter Power Spectrum: Restoring Information with a Logarithmic Density Mapping.* ApJL, 698, L90–L93. 109

Neyrinck, M. C., Szapudi, I., Szalay, A. S. 2011. *Rejuvenating Power Spectra. II. The Gaussianized Galaxy Density Field.* ApJ, 731, 116. 109

Noh, Y., White, M., Padmanabhan, N. 2009. *Reconstructing baryon oscillations.* PhRD, 80(12), 123501. 79

Nusser, A., Davis, M. 2011. *The Cosmological Bulk Flow: Consistency with ΛCDM and $z \approx 0$ Constraints on σ_8 and γ.* ApJ, 736, 93. 14

Nusser, A., Dekel, A. 1992. *Tracing large-scale fluctuations back in time.* ApJ, 391, 443–452. 80

Nusser, A., Dekel, A., Bertschinger, E., Blumenthal, G. R. 1991. *Cosmological velocity-density relation in the quasi-linear regime.* ApJ, 379, 6–18. 28

Nusser, A., Branchini, E., Davis, M. 2012. *Gaia: A Window to Large-scale Motions.* ApJ, 755, 58. 55, 91

Oort, J. H. 1930a. *Note on the velocities of extragalactic nebulae.* Bulletin of the Astronomical Institutes of the Netherlands, 5, 239. 1, 17

Oort, J. H. 1930b. *The estimated number of high velocities among stars selected according to proper motion.* Bulletin of the Astronomical Institutes of the Netherlands, 5, 189. 1, 17

Oort, J. H. 1930c. *The motion of the Sun with respect to the interstellar gas.* Bulletin of the Astronomical Institutes of the Netherlands, 5, 192. 1, 17

Partl, A. M., Dall'Aglio, A., Müller, V., Hensler, G. 2010. *Cosmological radiative transfer for the line-of-sight proximity effect.* A&A, 524, A85. 2

Pauls, J. L., Melott, A. L. 1995. *Hierarchical pancaking: why the Zel'dovich approximation describes coherent large-scale structure in N-body simulations of gravitational clustering.* MNRAS, 274, 99–109. 30

Peacock, J. A. 2007. *Cosmological Physics*. Cambridge University Press, 8th ed. 15, 21

Peacock, J. A., Dodds, S. J. 1994. *Reconstructing the Linear Power Spectrum of Cosmological Mass Fluctuations*. MNRAS, 267, 1020. 37

Peacock, J. A., Dodds, S. J. 1996. *Non-linear evolution of cosmological power spectra*. MNRAS, 280, L19–L26. 37

Peebles, P. J. E. 1980. *The large-scale structure of the universe*. Princeton University Press, 1980. 435 p. 1, 15, 20, 25

Peebles, P. J. E. 1989. *Tracing galaxy orbits back in time*. ApJL, 344, L53–L56. 79

Peebles, P.J.E. 1993. *Principles of physical cosmology*. Princeton series in physics. Princeton University Press. 1, 15

Pen, U.-L. 1997. *Generating Cosmological Gaussian Random Fields*. ApJL, 490, L127+. 47, 48, 72, 74

Penzias, A. A., Wilson, R. W. 1965. *A Measurement of Excess Antenna Temperature at 4080 Mc/s*. ApJ, 142, 419–421. 1

Perlmutter, S., Aldering, G., Goldhaber, G., Knop, R. A., Nugent, P., Castro, P. G., Deustua, S., Fabbro, S., Goobar, A., Groom, D. E., Hook, I. M., Kim, A. G., Kim, M. Y., Lee, J. C., Nunes, N. J., Pain, R., Pennypacker, C. R., Quimby, R., Lidman, C., Ellis, R. S., Irwin, M., McMahon, R. G., Ruiz-Lapuente, P., Walton, N., Schaefer, B., Boyle, B. J., Filippenko, A. V., Matheson, T., Fruchter, A. S., Panagia, N., Newberg, H. J. M., Couch, W. J., The Supernova Cosmology Project. 1999. *Measurements of Omega and Lambda from 42 High-Redshift Supernovae*. ApJ, 517, 565–586. 1, 17

Phelps, S. D., Desjacques, V., Nusser, A., Shaya, E. J. 2006. *Numerical action reconstruction of the dynamical history of dark matter haloes in N-body simulations*. MNRAS, 370, 1361–1371. 79

Power, C., Knebe, A. 2006. *The impact of box size on the properties of dark matter haloes in cosmological simulations*. MNRAS, 370, 691–701. 47

Press, W. H., Schechter, P. 1974. *Formation of Galaxies and Clusters of Galaxies by Self-Similar Gravitational Condensation*. ApJ, 187, 425–438. 34

Prunet, S., Pichon, C., Aubert, D., Pogosyan, D., Teyssier, R., Gottlöber, S. 2008. *Initial Conditions For Large Cosmological Simulations*. ApJS, 178, 179–188. 45, 47, 73, 165, 179

Rasera, Y., Teyssier, R. 2006. *The history of the baryon budget. Cosmic logistics in a hierarchical universe*. A&A, 445, 1–27. 2

Refregier, A., Amara, A., Kitching, T. D., Rassat, A. 2011. *iCosmo: an interactive cosmology package*. A&A, 528, A33. 23

Reiprich, T. H., Böhringer, H. 2002. *The Mass Function of an X-Ray Flux-limited Sample of Galaxy Clusters*. ApJ, 567, 716–740. 62, 113

Ribas, I., Jordi, C., Vilardell, F., Fitzpatrick, E. L., Hilditch, R. W., Guinan, E. F. 2005. *First Determination of the Distance and Fundamental Properties of an Eclipsing Binary in the Andromeda Galaxy*. ApJL, 635, L37–L40. 12

Riebe, K., Partl, A. M., Enke, H., Forero-Romero, J., Gottlöber, S., Klypin, A., Lemson, G., Prada, F., Primack, J. R., Steinmetz, M., Turchaninov, V. 2011. *The MultiDark Database: Release of the Bolshoi and MultiDark Cosmological Simulations.* ArXiv e-prints. 7

Riess, A. G., Filippenko, A. V., Challis, P., Clocchiatti, A., Diercks, A., Garnavich, P. M., Gilliland, R. L., Hogan, C. J., Jha, S., Kirshner, R. P., Leibundgut, B., Phillips, M. M., Reiss, D., Schmidt, B. P., Schommer, R. A., Smith, R. C., Spyromilio, J., Stubbs, C., Suntzeff, N. B., Tonry, J. 1998. *Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant.* AJ, 116, 1009–1038. 1, 17

Rizzi, L., Tully, R. B., Makarov, D., Makarova, L., Dolphin, A. E., Sakai, S., Shaya, E. J. 2007. *Tip of the Red Giant Branch Distances. II. Zero-Point Calibration.* ApJ, 661, 815–829. 11

Robertson, H. P. 1933. *Relativistic Cosmology.* Reviews of Modern Physics, 5(1), 62–90. 15

Robertson, H. P. 1935. *Kinematics and World-Structure.* ApJ, 82, 284–+. 15

Robertson, H. P. 1936a. *Kinematics and World-Structure II.* ApJ, 83, 187–+. 15

Robertson, H. P. 1936b. *Kinematics and World-Structure III.* ApJ, 83, 257–+. 15

Robin, A. C., Luri, X., Reylé, C., Isasi, Y., Grux, E., Blanco-Cuaresma, S., Arenou, F., Babusiaux, C., Belcheva, M., Drimmel, R., Jordi, C., Krone-Martins, A., Masana, E., Mauduit, J. C., Mignard, F., Mowlavi, N., Rocca-Volmerange, B., Sartoretti, P., Slezak, E., Sozzetti, A. 2012. *Gaia Universe model snapshot. A statistical analysis of the expected contents of the Gaia catalogue.* A&A, 543, A100. 55

Romano-Diaz, E., Faltenbacher, A., Jones, D., Heller, C., Hoffman, Y., Shlosman, I. 2006. *Constrained Cosmological Simulations of Dark Matter Halos.* ApJL, 637, L93–L96. 60

Romano-Diaz, E., Shlosman, I., Trenti, M., Hoffman, Y. 2011. *The Dark Side of QSO Formation at High Redshifts.* ApJ, 736, 66. 31, 60

Rybicki, G. B., Press, W. H. 1992. *Interpolation, realization, and reconstruction of noisy, irregularly sampled data.* ApJ, 398, 169–176. 50

Scannapieco, C., Tissera, P. B., White, S. D. M., Springel, V. 2005. *Feedback and metal enrichment in cosmological smoothed particle hydrodynamics simulations - I. A model for chemical enrichment.* MNRAS, 364, 552–564. 2

Scannapieco, C., Tissera, P. B., White, S. D. M., Springel, V. 2006. *Feedback and metal enrichment in cosmological SPH simulations - II. A multiphase model with supernova energy feedback.* MNRAS, 371, 1125–1139. 2

Scannapieco, C., White, S. D. M., Springel, V., Tissera, P. B. 2011. *Formation history, structure and dynamics of discs and spheroids in simulated Milky Way mass galaxies.* MNRAS, 417, 154–171. 2, 4

Schneider, P. 2006. *Extragalactic Astronomy and Cosmology: An Introduction.* Springer, 1st ed. 15

Seljak, U., Zaldarriaga, M. 1996. *A Line-of-Sight Integration Approach to Cosmic Microwave Background Anisotropies.* ApJ, 469, 437–+. 23

Shandarin, S. F., Zeldovich, Y. B. 1989. *The large-scale structure of the universe: Turbulence, intermittency, structures in a self-gravitating medium.* Reviews of Modern Physics, 61, 185–220. 28

Shaya, E. J., Peebles, P. J. E., Tully, R. B. 1995. *Action Principle Solutions for Galaxy Motions within 3000 Kilometers per Second.* ApJ, 454, 15. 79

Sheth, R. K., Diaferio, A. 2001. *Peculiar velocities of galaxies and clusters.* MNRAS, 322, 901–917. 40, 41, 42, 84

Sheth, R. K., Mo, H. J., Tormen, G. 2001. *Ellipsoidal collapse and an improved model for the number and spatial distribution of dark matter haloes.* MNRAS, 323, 1–12. 34, 61

Simon, J. D., Geha, M. 2007. *The Kinematics of the Ultra-faint Milky Way Satellites: Solving the Missing Satellite Problem.* ApJ, 670, 313–331. 33

Sirko, E. 2005. *Initial Conditions to Cosmological N-Body Simulations, or, How to Run an Ensemble of Simulations.* ApJ, 634, 728–743. 49

Smith, R. E., Peacock, J. A., Jenkins, A., White, S. D. M., Frenk, C. S., Pearce, F. R., Thomas, P. A., Efstathiou, G., Couchman, H. M. P. 2003. *Stable clustering, the halo model and non-linear cosmological power spectra.* MNRAS, 341, 1311–1332. 37

Smoot, G. F., Bennett, C. L., Kogut, A., Aymon, J., Backus, C., de Amici, G., Galuk, K., Jackson, P. D., Keegstra, P., Rokke, L., Tenorio, L., Torres, S., Gulkis, S., Hauser, M. G., Janssen, M. A., Mather, J. C., Weiss, R., Wilkinson, D. T., Wright, E. L., Boggess, N. W., Cheng, E. S., Kelsall, T., Lubin, P., Meyer, S., Moseley, S. H., Murdock, T. L., Shafer, R. A., Silverberg, R. F. 1991. *Preliminary results from the COBE differential microwave radiometers - Large angular scale isotropy of the cosmic microwave background.* ApJL, 371, L1–L5. 1

Sousbie, T., Pichon, C., Colombi, S., Novikov, D., Pogosyan, D. 2008. *The 3D skeleton: tracing the filamentary structure of the Universe.* MNRAS, 383, 1655–1670. 103

Spergel, D. N., Bean, R., Doré, O., Nolta, M. R., Bennett, C. L., Dunkley, J., Hinshaw, G., Jarosik, N., Komatsu, E., Page, L., Peiris, H. V., Verde, L., Halpern, M., Hill, R. S., Kogut, A., Limon, M., Meyer, S. S., Odegard, N., Tucker, G. S., Weiland, J. L., Wollack, E., Wright, E. L. 2007. *Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Implications for Cosmology.* ApJS, 170, 377–408. 1, 18

Springel, V. 2005. *The cosmological simulation code GADGET-2.* MNRAS, 364, 1105–1134. 2, 33

Springel, V. 2010. *E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh.* MNRAS, 401, 791–851. 2

Springel, V., Hernquist, L. 2003a. *Cosmological smoothed particle hydrodynamics simulations: a hybrid multiphase model for star formation.* MNRAS, 339, 289–311. 2

Springel, V., Hernquist, L. 2003b. *The history of star formation in a Λ cold dark matter universe.* MNRAS, 339, 312–334. 2

Springel, V., Yoshida, N., White, S. D. M. 2001. *GADGET: a code for collisionless and gasdynamical cosmological simulations.* New Astronomy, 6, 79–117. 2, 32, 33

Springel, V., White, S. D. M., Jenkins, A., Frenk, C. S., Yoshida, N., Gao, L., Navarro, J., Thacker, R., Croton, D., Helly, J., Peacock, J. A., Cole, S., Thomas, P., Couchman, H., Evrard, A., Colberg, J., Pearce, F. 2005. *Simulations of the formation, evolution and clustering of galaxies and quasars.* Nature, 435, 629–636. 1, 73

Springel, V., Wang, J., Vogelsberger, M., Ludlow, A., Jenkins, A., Helmi, A., Navarro, J. F., Frenk, C. S., White, S. D. M. 2008. *The Aquarius Project: the subhaloes of galactic haloes.* MNRAS, 391, 1685–1711. 73

Springob, C. M., Masters, K. L., Haynes, M. P., Giovanelli, R., Marinoni, C. 2007. *SFI++. II. A New I-Band Tully-Fisher Catalog, Derivation of Peculiar Velocities, and Data Set Properties.* ApJS, 172, 599–614. 14

Steinmetz, M. 1999. *Numerical Simulations of Galaxy Formation.* ApSS, 269, 513–532. 2

Suhhonenko, I., Gramann, M. 2003. *The rms peculiar velocity of galaxy clusters for different cluster masses and radii.* MNRAS, 339, 271–279. 41

Talman, J. D. 1978. *Numerical Fourier and Bessel Transforms in Logarithmic Variables.* Journal of computational physics, 29, 35. 65

Tasker, E. J., Brunino, R., Mitchell, N. L., Michielsen, D., Hopton, S., Pearce, F. R., Bryan, G. L., Theuns, T. 2008. *A test suite for quantitative comparison of hydrodynamic codes in astrophysics.* MNRAS, 390, 1267–1281. 33

Tegmark, M., Zaldarriaga, M. 2002. *Separating the early universe from the late universe: Cosmological parameter estimation beyond the black box.* PhRD, 66(10), 103508. 23

Tegmark, M., Blanton, M. R., Strauss, M. A., Hoyle, F., Schlegel, D., Scoccimarro, R., Vogeley, M. S., Weinberg, D. H., Zehavi, I., Berlind, A., Budavari, T., Connolly, A., Eisenstein, D. J., Finkbeiner, D., Frieman, J. A., Gunn, J. E., Hamilton, A. J. S., Hui, L., Jain, B., Johnston, D., Kent, S., Lin, H., Nakajima, R., Nichol, R. C., Ostriker, J. P., Pope, A., Scranton, R., Seljak, U., Sheth, R. K., Stebbins, A., Szalay, A. S., Szapudi, I., Verde, L., Xu, Y., Annis, J., Bahcall, N. A., Brinkmann, J., Burles, S., Castander, F. J., Csabai, I., Loveday, J., Doi, M., Fukugita, M., Gott, III, J. R., Hennessy, G., Hogg, D. W., Ivezić, Ž., Knapp, G. R., Lamb, D. Q., Lee, B. C., Lupton, R. H., McKay, T. A., Kunszt, P., Munn, J. A., O'Connell, L., Peoples, J., Pier, J. R., Richmond, M., Rockosi, C., Schneider, D. P., Stoughton, C., Tucker, D. L., Vanden Berk, D. E., Yanny, B., York, D. G., SDSS Collaboration. 2004. *The Three-Dimensional Power Spectrum of Galaxies from the Sloan Digital Sky Survey.* ApJ, 606, 702–740. 1

Teyssier, R. 2002. *Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES.* A&A, 385, 337–364. 2, 33

Teyssier, R., Pires, S., Prunet, S., Aubert, D., Pichon, C., Amara, A., Benabed, K., Colombi, S., Refregier, A., Starck, J.-L. 2009. *Full-sky weak-lensing simulation with 70 billion particles.* A&A, 497, 335–341. 73

Thomas, B. C., Melott, A. L., Feldman, H. A., Shandarin, S. F. 2004. *Quantifying the Bull's-Eye Effect.* ApJ, 601, 28–36. 42

Tikhonov, A. V., Klypin, A. 2009. *The emptiness of voids: yet another overabundance problem for the Λ cold dark matter model.* MNRAS, 395, 1915–1924. 2

Tinker, J., Kravtsov, A. V., Klypin, A., Abazajian, K., Warren, M., Yepes, G., Gottlöber, S., Holz, D. E. 2008. *Toward a Halo Mass Function for Precision Cosmology: The Limits of Universality.* ApJ, 688, 709–728. 35, 129

Tonry, J. L., Dressler, A., Blakeslee, J. P., Ajhar, E. A., Fletcher, A. B., Luppino, G. A., Metzger, M. R., Moore, C. B. 2001. *The SBF Survey of Galaxy Distances. IV. SBF Magnitudes, Colors, and Distances.* ApJ, 546, 681–693. 2, 11, 128

Toro, E. F. 1999. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Berlin: Springer Verlag. 75

Tully, R. B. 1988. *Nearby galaxies catalog.* Cambridge University Press. 2, 11

Tully, R. B., Courtois, H. M. 2012. *Cosmicflows-2: I-band Luminosity-H I Linewidth Calibration.* ApJ, 749, 78. 6, 11, 55, 139

Tully, R. B., Fisher, J. R. 1977. *A new method of determining distances to galaxies.* A&A, 54, 661–673. 10

Tully, R. B., Shaya, E. J., Karachentsev, I. D., Courtois, H. M., Kocevski, D. D., Rizzi, L., Peel, A. 2008. *Our Peculiar Motion Away from the Local Void.* ApJ, 676, 184–205. 2, 10, 11, 12, 56, 88, 114, 128, 137, 138, 174

Tully, R. B., Rizzi, L., Shaya, E. J., Courtois, H. M., Makarov, D. I., Jacobs, B. A. 2009. *The Extragalactic Distance Database.* AJ, 138, 323–331. 6, 11, 87

van de Weygaert, R., Bertschinger, E. 1996. *Peak and gravity constraints in Gaussian primordial density fields: An application of the Hoffman-Ribak method.* MNRAS, 281, 84. 59, 60, 67, 68, 70

van de Weygaert, R., Hoffman, Y. 1999. *Cold flows and large scale tides. Page 178 of:* A. J. Banday, R. K. Sheth, & L. N. da Costa (ed), *Evolution of Large Scale Structure: From Recombination to Garching.* 13, 61

van de Weygaert, R., Hoffman, Y. 2000. *The Structure of the Local Universe and the Coldness of the Cosmic Flow. Page 169 of:* S. Courteau & J. Willick (ed), *Cosmic flows workshop.* Astronomical Society of the Pacific Conference Series, vol. 201. 61

van der Wel, A., Bell, E. F., Holden, B. P., Skibba, R. A., Rix, H.-W. 2010. *The Physical Origins of the Morphology-Density Relation: Evidence for Gas Stripping from the Sloan Digital Sky Survey.* ApJ, 714, 1779–1788. 11, 42

Wagner, C. 2009. *Probes of Dark Energy Using Cosmological Simulations.* Ph.D. thesis, Universität Potsdam. 36, 79

Walker, A. G. 1937. *On Milne's Theory of World-Structure.* Proc. London Math. Soc., s2-42(1), 90–127. 15

Wang, J., White, S. D. M. 2007. *Discreteness effects in simulations of hot/warm dark matter.* MNRAS, 380, 93–103. 47

Wang, J., Frenk, C. S., Navarro, J. F., Gao, L., Sawala, T. 2012. *The missing massive satellites of the Milky Way.* MNRAS, 424, 2715–2721. 33

Weinberg, D. H. 1992. *Reconstructing primordial density fluctuations. I - Method.* MNRAS, 254, 315–342. 78, 109

Weinberg, S. 2008. *Cosmology.* Oxford University Press. 1, 15, 17, 20

White, S. D. M. 1994. *Formation and Evolution of Galaxies: Les Houches Lectures.* Arxiv e-prints. 47

White, S. D. M., Frenk, C. S. 1991. *Galaxy formation through hierarchical clustering.* ApJ, 379, 52–79. 1, 30, 33

White, S. D. M., Rees, M. J. 1978. *Core condensation in heavy halos - A two-stage theory for galaxy formation and clustering.* MNRAS, 183, 341–358. 1, 33

Wiener, N. 1930. *Generalized harmonic analysis.* Acta Mathematica, 55. 22

Wiener, N. 1949. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series.* New York: Wiley. 50

Willick, J. A., Courteau, S., Faber, S. M., Burstein, D., Dekel, A., Strauss, M. A. 1997. *Homogeneous Velocity-Distance Data for Peculiar Velocity Analysis. III. The Mark III Catalog of Galaxy Peculiar Velocities.* ApJS, 109, 333. 2, 61, 128

Wise, J. H., Abel, T. 2011. *ENZO+MORAY: radiation hydrodynamics adaptive mesh refinement simulations with adaptive ray tracing.* MNRAS, 414, 3458–3491. 2

Wise, J. H., Turk, M. J., Norman, M. L., Abel, T. 2012. *The Birth of a Galaxy: Primordial Metal Enrichment and Stellar Populations.* ApJ, 745, 50. 2

Yamamoto, K., Sugiyama, N., Sato, H. 1998. *Evolution of Small-Scale Cosmological Baryon Perturbations and Matter Transfer Functions.* ApJ, 501, 442. 75

Yoshida, N., Sugiyama, N., Hernquist, L. 2003. *The evolution of baryon density fluctuations in multicomponent cosmological simulations.* MNRAS, 344, 481–491. 75

Yu, Y., Zhang, P., Lin, W., Cui, W., Fry, J. N. 2011. *Gaussianizing the non-Gaussian lensing convergence field: The performance of the Gaussianization.* PhRD, 84(2), 023523. 109

Yu, Y., Zhang, P., Lin, W., Cui, W., Fry, J. N. 2012. *Gaussianizing the non-Gaussian lensing convergence field II. The applicability to noisy data.* PhRD, 86(2), 023515. 109

Zaroubi, S. 2002. *Unbiased reconstruction of the large-scale structure.* MNRAS, 331, 901–908. 105

Zaroubi, S., Hoffman, Y., Fisher, K. B., Lahav, O. 1995. *Wiener Reconstruction of the Large-Scale Structure.* ApJ, 449, 446–+. 50, 53, 139

Zaroubi, S., Hoffman, Y., Dekel, A. 1999. *Wiener Reconstruction of Large-Scale Structure from Peculiar Velocities.* ApJ, 520, 413–425. 26, 50, 53, 55, 139

Zeldovich, Y. B. 1970. *Gravitational instability: An approximate theory for large density perturbations.* A&A, 5, 84–89. 27, 28, 140

Zemp, M., Stadel, J., Moore, B., Carollo, C. M. 2007. *An optimum time-stepping scheme for N-body simulations.* MNRAS, 376, 273–286. 32

Zhang, T.-J., Yu, H.-R., Harnois-Déraps, J., MacDonald, I., Pen, U.-L. 2011. *Increasing the Fisher Information Content in the Matter Power Spectrum by Nonlinear Wavelet Wiener Filtering.* ApJ, 728, 35. 109

Zwicky, F. 1933. *Die Rotverschiebung von extragalaktischen Nebeln.* Helvetica Physica Acta, 6, 110–127. 1, 17

Zwicky, F. 1937. *On the Masses of Nebulae and of Clusters of Nebulae.* ApJ, 86, 217–+. 1, 17

Acknowledgements

The process of completing this PhD thesis has not been an easy road, and often it was very bumpy. It would not have been possible without the support of many people. First of all I would like to thank the three people who most directly supervised my scientific work. I am very grateful to Hélène Courtois, for always guiding me in the right direction and providing her support when I most needed it; to Stefan Gottlöber, for offering me the great opportunity to work at the AIP in Potsdam, giving me access to all tools and facilities that I needed, and constantly reminding me about our scientific goal; and to Yehuda Hoffman, for patiently sharing with me his immense knowledge about physics, math, and the world of constrained simulations.

Dealing with the multinational nature of this work and the associated administrative hurdles was a difficult task. I would like to thank the responsible people at the respective institutes and universities who helped me a lot in the process, especially Matthias Steinmetz and Volker Müller at the AIP, Britta van Kempen at the University of Potsdam, and Christophe Dujardin in Lyon. Many thanks go to R. Brent Tully for inviting me to the Institute for Astronomy in Hawaii and the wonderful time I spent there, and to Joel Primack, for the great opportunity to participate in the UC-HIPACC summer school in Santa Cruz, California.

It is a pleasure to thank the people whom I was happy to meet and collaborate with, for the many fruitful scientific discussions, valuable insights, and inspiring ideas: Francisco-Shu Kitaura, Steffen Knollmann, Noam Libeskind, Steffen Hess, Gustavo Yepes, Anatoly Klypin, Guilhem Lavaux, Saleem Zaroubi, Romain Teyssier, Stephane Colombi, Christian Wagner, Philipp Richter, and many others. I greatly enjoyed working in the CLUES project, and I would like to extend my gratitude to all people in this collaboration. Lots of thanks go to everybody at the AIP in Potsdam, where I spent several wonderful years, for offering me such a pleasant and fruitful atmosphere. Special thanks to Jochen Klar and Adrian Partl, for supplying me with unlimited amounts of high-quality coffee, numerous interesting discussions, and (not only) computer related help on many occasions.

The computations and simulations performed for this thesis were conducted at the computing facilities of the AIP. I want to thank all people involved in maintaining and administrating them and providing computer support, in particular Karl-Heinz Böning and Harry Enke. Furthermore, for almost every detail of this work I was relying on free and open-source software, and I want to express my deepest gratitude for all the developers who put a lot of effort into making these numerous tools available. I also want to thank Jenny Sorce, Marija Vlajić, and Uğur Ural for proofreading of my manuscript, and Jean-Christophe Olaya for his help with the French language.

I am very thankful to Alexander Knebe for being such a great advisor and teacher. The wonderful time I spent working with him and his research group showed me how fascinating science can be and motivated me to continue working in cosmology and to pursue a PhD in this field.

Most of all I would like to thank Zhenja, for her love and support, for always being there for me despite of all the trouble and doubt, and for bringing light and joy into my life.

Declaration of independent work

I hereby declare that this thesis is the product of my own independent work and that I have listed all the literature and resources that have been used.

Déclaration sur l'honneur

Je certifie que cette thèse est le fruit de mon propre travail et qu'elle a été rédigée sans autre aide que les sources et références dûment citées.

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Timur Doumler
Potsdam, 26 September 2012

The scientific results presented in this thesis have been submitted to publication in MNRAS in the form of a series of three articles (Doumler et al. 2012a,b,c).