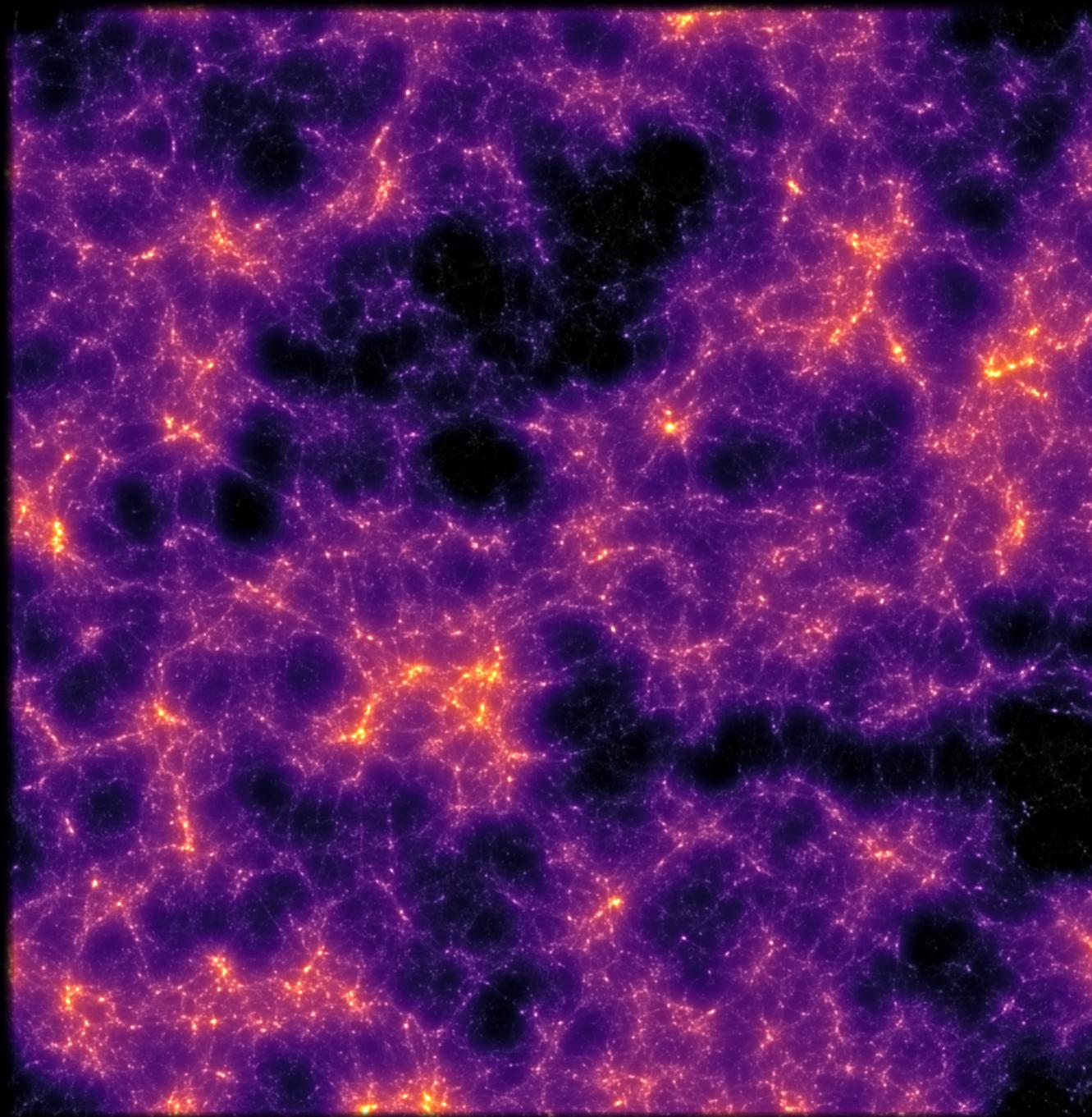


LINEAR AND NON-LINEAR
MASSIVE NEUTRINOS IN COSMOLOGICAL
N-BODY SIMULATIONS



Jeppe Mosgaard Dakin
PhD Thesis 2018

LINEAR AND NON-LINEAR
MASSIVE NEUTRINOS
IN COSMOLOGICAL
N-BODY SIMULATIONS

PHD THESIS

JEPPE MOSGAARD DAKIN

NOVEMBER 2018

DEPARTMENT OF PHYSICS AND ASTRONOMY
AARHUS UNIVERSITY

This thesis is submitted to the Faculty of Science at Aarhus University, Denmark, in order to fulfil the requirements for obtaining the PhD degree in Physics.

The studies have been carried out under the supervision of Prof. Steen Hannestad in the Cosmology Group at the Department of Physics and Astronomy at Aarhus University from January 2016 to November 2018.

Cover illustrations: Projected density fields in Λ CDM cosmology from simulations using linear (back) and non-linear (front) neutrinos with a mass of 0.1 eV, at the present epoch. The coloured splotches and the white filaments are the neutrino and matter fields, respectively. The two components are harder to distinguish in the non-linear case as the neutrinos here follow the evolution of the matter more closely. The renders span an area of 1024 Mpc \times 1024 Mpc and the projected dimension has a thickness of 102.4 Mpc. The renders and their underlying simulations have been produced using the CONCEPT code.

Preface and Acknowledgements

This thesis presents the research in which I have been engaged during my PhD studies from January 2016 to November 2018. The majority of this time was spent on developing a new self-consistent method for implementing non-linear massive neutrinos into cosmological simulations, in the group of Steen Hannestad at Aarhus University, Denmark. I was fortunate enough to be allowed to use my own brainchild ‘CONCEPT’ as the platform for this development, CONCEPT being a matter-only cosmological simulation code I had previously written.

This new method for dealing with non-linear neutrinos was developed in collaboration between Steen Hannestad, Jacob Brandbyge, Thomas Tram, Troels Haugbølle and myself, drawing upon the expertise of Steen and Jacob regarding cosmological neutrino simulations, Thomas’ understanding of the details of linear perturbation theory and his CLASS code and Troels’ knowledge of computational fluid dynamics. After two years we presented a working implementation of our new method in a paper on arXiv [1]. The paper is also included in this thesis.

I spent May and June of 2018 at the Institute for Computational Science, University of Zürich, collaborating with the cosmology group of Romain Teyssier and Joachim Stadel. The goal became to include neutrinos into the PKDGRAV code, a state-of-the-art cosmological simulation code of which Joachim is the main author. Unlike my previous work, the aim here was to treat the neutrinos linearly, vastly simplifying the problem at hand. During my stay and in the following months, we managed to carry out this linear implementation of not only neutrinos, but also of photons and general relativistic corrections, into both PKDGRAV and CONCEPT. Thomas Trams, Jacob Brandbyge, Steen Hannestad and I wrote a paper about this inclusion of linear species into cosmological simulations, which appears on arXiv [2] and is also included in this thesis.

I would like to thank Steen Hannestad for his guidance and willingness to discuss problems and solution strategies of my own, and for creating a great research environment in general. I am also grateful to Thomas Tram and Jacob Brandbyge for sharing their knowledge and for being great collaborators throughout my PhD. I would like to thank Romain

Tessier and Joachim Stadel for their kindness and enthusiasm during my stay in Zürich, and Joachim Stadel and Douglas Potter for allowing me to fundamentally change their PKDGRAV code under their skilled supervision and close collaboration. I am happy that my stay in Zürich overlapped with a corresponding stay of Hugues de Laroussilhe, giving me a partner in crime for adventures in both PKDGRAV and the Swiss Alps. A big thank you to Kathrine Bundgaard Henriksen and Katrine Alice Glasscock for daring to base their Bachelor’s projects on CONCEPT, serving as valuable testers of both corner cases and overall usability of the code. Thanks to Mischa Knabenhans for thorough testing of the CONCEPT CLASS utility. Thanks to Niels Carl Hansen for technical support and to Sofie Koksbang, Rasmus Sloth, Io Oddershov, Jacob Brandbyge, Thomas Tram and Steen Hannestad for company during trips, conferences, dinners and the innumerable number of coffee breaks.

Jeppe Dakin, November 2018

- [1] J. Dakin, J. Brandbyge, S. Hannestad, T. Haugbølle and T. Tram, “ ν CONCEPT: Cosmological neutrino simulations from the non-linear Boltzmann hierarchy,” arXiv:1712.03944 [astro-ph.CO].
- [2] T. Tram, J. Brandbyge, J. Dakin and S. Hannestad, “Fully relativistic treatment of light neutrinos in N -body simulations,” arXiv:1811.00904 [astro-ph.CO].

Summary

Modern cosmology, the study of our Universe as a whole, progresses by comparing predictions of cosmological models to real world observations. One such observation is the present day large-scale structure of matter, i.e. how the luminous and dark matter within the Universe is structured at scales beyond that of individual galaxies. The precise layout of this structure is intricately dependent on various cosmological parameters, such as the neutrino mass, or rather the three neutrino masses if we consider the full neutrino hierarchy. Depending on the values of these unknown but constrained masses, the existence of neutrinos affect the large-scale matter structure at the percent level, directly through gravitational interaction and indirectly by modifying the Hubble expansion.

In the near future, extremely precise observations of the large-scale structure in our Universe will be available, thanks to surveys such as LSST and Euclid. At this level of precision, the effects of massive neutrinos cannot be neglected, and thus must be included in our models from which we produce predictions of the large-scale structure. If we succeed in modelling the cosmic effects from massive neutrinos, we will then be able to extract the neutrino mass from the observations of the large-scale structure.

The standard way of producing predictions for the large-scale structure is through large computer simulations, evolving virtual universes from a few million years after the Big Bang through cosmic history up until the present. While the treatment of matter in such cosmological simulations is a relatively settled issue, the best way of including massive neutrinos remains an open problem.

My PhD work and this thesis focus on a new method for dealing with massive neutrinos in cosmological simulations. At a glance, this method works by treating the neutrinos as a fluid and solving the non-linear fluid equations on a grid in real space. The fluid variables considered are constructed from velocity moments of the distribution function, forming a hierarchy of variables known as the Boltzmann hierarchy. As this hierarchy is much too large to be solved in full in 3D, we keep only the

lowest few moments non-linear. The full linear Boltzmann hierarchy is then used to close the truncated non-linear Boltzmann hierarchy.

Besides the development of this non-linear neutrino method, I have also been concerned with a method for implementing massive neutrinos into cosmological simulations at the purely linear level, also described in this thesis. I have implemented both the linear and non-linear method into my own code by the name of CONCEPT. In addition, the linear method has also been implemented into the state-of-the-art PKDGRAV code. An extremely large simulation using this code is planned, which will serve as the flagship simulation for the Euclid mission. Furthermore, this simulation is planned to be run with the inclusion of massive linear neutrinos using my implementation.

This thesis contains two research papers of which I am a co-author, written during my PhD. The papers describe and demonstrate the linear and non-linear method, respectively. With the linear implementation we are able to achieve agreement with linear theory at the sub-per-mille level. However, linear theory is not applicable throughout time for massive neutrinos, and so a non-linear description is needed for more realistic modelling of the neutrinos. For the non-linear implementation we generally find results consistent with the literature. Unlike the linear method which is rather unique in the sense that there is nothing much one can change about it, the non-linear method is really a collection of methods, with no a priori means of knowing which yields the best results. Though the entire scope of this collection is discussed in this thesis, only a limited subset has been implemented in practice.

Dansk Resumé (Danish Summary)

Fremskridt indenfor moderne kosmologi, studiet af Universet som helhed, foregår ved at sammenligne forudsigelser fra kosmologiske modeller med observationer fra den virkelige verden. Blandt de vigtigste observerbare fænomener har vi storskalastrukturen af stof, dvs. hvordan det lysende og mørke stof i Universet er fordelt på størrelsesskalaer større end enkelte galakser. Det detaljerede udseende af denne struktur afhænger på komplicerede måder af diverse kosmologiske parametre, såsom neutrinomassen, eller rettere de tre neutrinomasser hvis vi betragter det fulde neutrinoehierarki. Eksistensen af neutrinoer påvirker stoffets storskalastruktur på procent-niveau, med en større effekt for større værdier af de ukendte men afgrænsede neutrinomasser. Påvirkningen er både direkte via gravitationel interaktion, men også indirekte idet neutrinoerne modifierer Hubble-ekspansionen.

Vi vil i den nære fremtid have adgang til meget præcise observationer af Universets storskalastruktur, takket være surveys som LSST og Euclid. På dette præcisionsniveau kan vi ikke længere ignorere de massive neutrinoers indflydelse, hvorfor vi er nødt til at inkludere dem i vores modeller hvorfra vi producerer forudsigelser for storskalastrukturen. Hvis det lykkes for os at modellere de kosmiske effekter fra massive neutrinoer, vil vi være i stand til at udtrække neutrinomassen fra observationerne af storskalastrukturen.

Den typiske måde hvorpå vi producerer forudsigelser af storskalastrukturen er gennem store computersimuleringer, som udvikler virtuelle universer fra få millioner år efter Big Bang og frem til nutiden. Mens man har standardiserede teknikker til at behandle stof i disse simuleringer, er den bedste måde til at inkludere massive neutrinoer stadig et åbent spørgsmål.

Mit PhD-arbejde og denne afhandling fokuserer på en ny måde hvorpå massive neutrinoer kan inkluderes i kosmologiske simuleringer. Groft sagt virker metoden ved at behandle neutrinoerne som et fluid og løse de ikke-lineære hydrodynamiske ligninger på et gitter i reelt

rum. De betragtede fluid-variable konstrueres fra hastighedsmomenter af fordelingsfunktionen, som tilsammen danner et hierarki af variable kaldet Boltzmann-hierarkiet. Da dette hierarki er alt for stort til at blive fuldstændigt løst i 3D, beholder vi kun de få laveste momenter som ikke-lineære. Det komplette lineære Boltzmann-hierarki bruges da til at lukke det trunkerede ikke-lineære Boltzmann-hierarki.

Udover udviklingen af denne ikke-lineære neutrino-metode har jeg også beskæftiget mig med en metode til at implementere massive neutrinoer i kosmologiske simuleringer på et rent lineært niveau. Denne metode er også indeholdt i denne afhandling. Jeg har implementeret både den lineære og den ikke-lineære metode i min egen kode ved navn CONCEPT. Derudover har jeg også implementeret den lineære metode i den sofistikerede og moderne PKDGRAV-kode. Denne kode vil blive brugt til at køre en ekstremt detaljeret simulering, som skal tjene som flagskibs-simuleringen for Euclid-missionen. Det er ydermere planen at denne simulering skal inkludere massive neutrinoer via min implementering.

Denne afhandling indeholder to forskningsartikler som jeg er medfatter på, og som blev skrevet i løbet af min PhD. Artiklerne beskriver og demonstrerer henholdsvis den lineære og den ikke-lineære metode. Med den lineære metode er vi i stand til at opnå overensstemmelse med lineær teori på bedre end promille-niveau. Lineær teori er dog ikke anvendelig for massive neutrinoer gennem hele Universets udvikling, hvorfor vi er nødt til at overgå til en ikke-lineær beskrivelse for at opnå en mere realistisk modellering af neutrinoerne. Resultaterne fra den ikke-lineære implementering stemmer generelt overens med litteraturen. Modsat den lineære metode som er stort set unik i den forstand at den næsten ikke kan modificeres, er den ikke-lineære metode nærmere en hel samling af ikke-lineære metoder, uden nogen a priori måde hvorpå disse kan rangordnes. Hele denne samling af metoder vil blive præsenteret i denne afhandling, men kun en begrænset delmængde af dem er blevet implementeret i praksis.

Contents

Preface and Acknowledgements	i
Summary	iii
Dansk Resumé (Danish Summary)	v
Contents	vii
1 Introduction	1
1.1 Cosmology	1
1.2 The Importance of Neutrinos	2
1.3 My PhD Work	4
1.3.1 Non-linear Neutrinos	4
1.3.2 Linear Neutrinos	5
1.4 The CONCEPT Code	7
1.5 This Thesis	8
I Theory	11
2 The Homogeneous Universe	13
2.1 The Hubble Expansion	13
2.1.1 Cosmic Inventory	16
2.2 Neutrinos	18
3 The Perturbed Universe	27
3.1 Newtonian Matter Perturbations	27
3.1.1 Eulerian Equations of Motion in Comoving Space .	27
3.1.2 Lagrangian Equations of Motion in Comoving Space	30
3.1.3 Linear Perturbations	32
3.2 General Relativistic Fluid Dynamics	34
3.2.1 Fluid Variables and the Boltzmann Hierarchy . .	34
3.2.2 Fluid Equations	36

3.3	Linear Perturbation Theory	42
3.3.1	Continuity and Euler equation	43
3.3.2	Beyond the Euler Equation	45
3.3.3	Transfer Functions	48
4	Power Spectra	51
4.1	Power Spectra as Measures of Correlation	51
4.2	The Primordial Power Spectrum	52
4.2.1	The Variance as Amplitude	54
4.3	Realisations	55
II	Numerics	59
5	Overview of Standard N-body Techniques	61
5.1	Basic Set-up	61
5.2	Hamiltonian Dynamics	63
5.2.1	The Leapfrog Integrator	63
5.2.2	Adaptive Time Stepping	65
5.3	Newtonian Gravitation	66
5.3.1	The Particle-Particle Method	67
5.3.2	The Particle-Mesh Method	71
5.3.3	Advanced Methods	78
6	Fluid Dynamics	83
6.1	Introduction	83
6.1.1	Numerical Fluid Equations	84
6.1.2	General Considerations	85
6.2	The MacCormack Method	89
6.2.1	Source Terms	92
6.2.2	Vacuum Corrections	93
6.3	The Kurganov-Tadmor Method	96
6.4	Gravity	102
7	Realisations	107
7.1	Numerical Details	107
7.1.1	The Complex Conjugacy Symmetry	109
7.2	Particle Realisation	114
7.2.1	The Zel'dovich Approximation	114
7.2.2	Particle Momenta	115
7.3	Late-time Normalisation	117

7.3.1	Back-scaling	117
7.3.2	Relativistic Perturbations and the N -body Gauge .	119
7.4	Non-linear Realisations	124
7.4.1	Forward Fourier Transforms	124
7.4.2	Dynamic “Random” Field	126
7.4.3	Injecting Non-linearity	127
8	The CONCEPT Code	133
8.1	Software Aspects	134
8.1.1	Cython	134
8.1.2	Libraries and Installation	134
8.2	Numerical Methods	135
8.2.1	Gravity	136
8.2.2	Time Integration	138
8.2.3	Realisations	140
8.2.4	Parallelisation	141
8.3	Integration with CLASS	143
8.3.1	The CONCEPT CLASS Utility	144
III	Linear and Non-linear Neutrinos	147
9	Linear Neutrinos	149
9.1	Introduction	152
9.2	Method and Implementation	153
9.3	Numerical Set-up and Results	156
9.3.1	Main Results	157
9.3.2	Comparison with PKDGRAV	160
9.4	Discussion	160
9.A	Computing γ	162
10	Non-linear Neutrinos	167
10.1	Introduction	170
10.2	Theory	172
10.2.1	The Non-linear Boltzmann Equation	172
10.2.2	The Moment Equations	174
10.2.3	Closing the Hierarchy	175
10.2.4	The Continuity and Euler Equations in Conservation Form	175
10.3	The Linear Computation	176
10.3.1	$\delta P/\delta\rho$ and σ in Linear Theory	178

10.4 Implementation Details	180
10.4.1 Dynamics	181
10.4.2 Gravity	184
10.5 Results	189
10.5.1 The Neutrino Power Spectra	189
10.5.2 Neutrino Suppression of the Relative Total Matter Power Spectra	192
10.5.3 Comparison with the Hybrid Code	192
10.5.4 The Effect of Anisotropic Stress	193
10.5.5 Convergence	194
10.6 Conclusions	196
10.A N-body Realisations of the Boltzmann Hierarchy Variables	198
10.A.1 The Zel'dovich Approximation	199
10.A.2 Density and Velocity Fields	199
10.A.3 Anisotropic Stress	200
10.A.4 Non-linear Realisations	201
10.B Comparison between CLASS and CAMB	203
11 Conclusions and Outlook	209
Bibliography	213

1 Introduction

1.1 Cosmology

Cosmology is the study of the *Cosmos* (or Universe) as a whole. It strives to explain why the Universe is as it is, how it has evolved and what the future will bring. Throughout human history, different cultures have invented thousands of imaginative cosmologies as answers to this self-imposing riddle. Only in recent times, starting with the advent of the theory of general relativity a century ago, have we developed a scientific, *physical cosmology*.

General relativity promotes space (and time) from mere static and eternal labels to a dynamical scene in which the cosmic story can unfold. Though the heavens appear static on the time scales of humans, the observation that distant galaxies uniformly are moving away from us made it clear that the Universe is expanding, and hence was smaller in the past. This is the beginning of the well-known Big Bang theory, which states that our universe began its life 13.8 billion years ago in a very hot and dense state, from which it has expanded and cooled ever since. Most of contemporary cosmology aims to explain how we get from this hot and dense state to the present day universe. Ideally we would like to also understand the Big Bang itself and go beyond it. For this, a plethora of ideas do exist, but the honest answer is that our current understanding of Nature predicts its own downfall at the Big Bang, and so these ideas are highly speculative.

Exactly how the fabric of space expands is described by the Einstein field equations, which state that this expansion depends exclusively on what kind of stuff the universe contains. The scenery (spacetime) and actors (content within the universe) of the cosmic story are thus intimately linked. As we have no good theory describing the very birth of a universe, we have no a priori idea of what these actors may be. From everyday experience we know that our universe contains atoms and light, from which we ourselves are build and interact with the world. More generally, we know of the existence of all of the particles in the standard

model of particle physics, which additionally include more exotic particles like the neutrinos.

Through detailed observations of how the luminous atomic stuff is distributed throughout the Universe (the cosmic large-scale structure), as well as of the distribution of light from the Big Bang itself (the cosmic microwave background) and of the late expansion history of the Universe (via supernovae), it is clear that the content of our universe mostly consist of forms of energies well outside our standard model of particle physics. The standard model of Big Bang cosmology, Λ CDM, gets its name from these additional, exotic species. The Λ refers to a cosmological constant; the simplest type of *dark energy*, a term reserved for whatever substance makes the universal expansion accelerate over time. At the present epoch, $\sim 70\%$ of the total energy content of our universe is contributed by this dark energy. The CDM stands for *cold dark matter*, meaning stuff that gravitate and cluster like luminous matter (atoms), but does not interact electromagnetically. Currently $\sim 25\%$ of the total energy of our universe is contained in cold dark matter. This leaves us with a mere $\sim 5\%$ for “ordinary” stuff, including atoms, photons and neutrinos. As most of the energy in atoms is bound in the nucleus, which in turn consists of baryons, atomic matter is commonly referred to as *baryonic* matter.

Commonly, the Λ CDM model is extended with *inflation*, which amounts to the addition of yet another exotic species (the inflaton), the purpose of which is to create exponential expansion in the very early universe. This idea solves the problem of why distant regions seem to have once been in thermal equilibrium, and also explains why the Universe appears ‘flat’ (as in Euclidean). In addition, this rapid early expansion serves to magnify quantum fluctuations to macroscopic scales, providing an explanation as to how primordial fluctuations may arise from a classically completely homogeneous background.

1.2 The Importance of Neutrinos

Progress in cosmology is achieved through testing models of the Universe against observations, a key example of which is the large-scale (matter) structure of our universe. Given a cosmological model with precisely specified parameters, large computer simulations allow us to compute predictions of physical observables such as the large-scale structure.

Though the Λ CDM model is firmly established, it leaves quite some room for specification of various details, one of which is the precise nature of neutrinos. This topic really belongs to particle physics, from which

we have learned a lot about how neutrinos interact weakly and undergo flavour oscillations, though their masses remain undetermined. From a cosmological point of view, the most important neutrino parameters are the masses of the three mass states, since this directly relate to the amount of energy that is tied up in neutrinos and how strongly they interact gravitationally with other species.

The existence of neutrinos affect the large-scale structure in several principle ways: The flatness of our universe demands a specific total energy density, and so raising/lowering the energy density put into neutrinos alters the remaining energy available for the other species. To a first approximation then, the sum of neutrino masses, written $\sum m_\nu$, is then all we need to know in order to compute the energy fraction taken up by neutrinos and hence how much is “left over” for matter. Introducing neutrinos into the cosmology also has an effect on the expansion history, which affect the clustering of matter. More directly, neutrinos also affect the large-scale structure by interacting gravitationally with matter.

Upcoming large-scale structure surveys such as LSST [1] and Euclid [2] will map the large-scale structure extremely precisely. At this level of precision we will need to take the effects of neutrinos into account when producing predictions via the simulations. As such, my PhD work and this thesis is about my attempt to incorporate massive neutrinos into cosmological simulations.

Massive Neutrino Schemes in the Literature

Before introducing the scheme(s) developed during my PhD for incorporating massive neutrinos into cosmological simulations in the next section, let us briefly outline existing methods in the literature.

- Treat neutrinos as particles and evolve them using more or less standard N -body techniques [3–10]. Due to the large thermal velocities of neutrinos, the particles need to sample the full 6-dimensional phase space, which should be contrasted with the usual cold dark matter particles where only the three spatial dimensions are sampled. In practice then, this technique requires extremely many neutrino particles in order to obtain good resolution.
- Avoid the problem of non-linearly evolving the neutrinos by just using the linear evolution. One way of doing this is through repeated realisation of the linear neutrino density field throughout the simulation, from which the gravitational forces on the particles are computed [11]. This is a simple and computationally cheap

strategy which works well in practice. As it does come with the assumption that the neutrinos behave linear throughout the simulation, accurate results may only be obtained for small neutrino masses.

This strategy is identical to what this thesis refer to as ‘the linear method’, though here we embed the neutrino realisation within the framework of the N -body gauge similar to what is done for massless neutrinos in [12].

Some non-linearity of the neutrino evolution can be achieved by feeding the non-linear gravitational potential from the simulation back into the linear computation of the neutrino evolution [13, 14].

- One can marry the two approaches above, with the neutrinos initially evolved linearly and then at some time converted into particles in the simulation [15]. This particle dump should then occur at a time where the thermal velocities of the neutrinos are not much greater than the gravitationally induced streaming velocities, so that the depth of the momentum dimensions in the neutrino distribution which require sampling is small, leading to a manageable number of neutrino particles.

1.3 My PhD Work

1.3.1 Non-linear Neutrinos

The first ~ 2.5 years of my PhD work was focused on developing a method which could handle neutrinos in complete generality, i.e. regardless of the value of $\sum m_\nu$. Here, the neutrinos are introduced in an N -body code on equal footing with matter. The neutrinos are evolved in time according to their internal dynamics, while interacting gravitationally with the matter and amongst themselves. This general method, which I shall refer to simply as ‘the non-linear neutrino method’, was implemented into the CONCEPT N -body code [16], a code I originally wrote for my master’s thesis in 2015. A large part of the numerical work consisted in fully integrating the Einstein-Boltzmann code CLASS [17] into CONCEPT, essentially enabling CONCEPT to query CLASS for various cosmological background quantities and linear perturbations.

A paper titled ‘vCONCEPT: Cosmological neutrino simulations from the non-linear Boltzmann hierarchy’ which outlines the method and its implementation in CONCEPT, and also discusses the results of test

simulations, can be found on arXiv [18]. It was submitted to JCAP in December 2017, though the referee has requested two major revisions. I plan to re-submit it this December. The paper is included in this thesis in chapter 10, which also explains the reason for these major revisions.

1.3.2 Linear Neutrinos

I had the pleasure of spending May and June of 2018 at the Institute for Computational Science, University of Zürich, under the supervision of Romain Teyssier, who is the main author of the RAMSES code [19], one of the main cosmological/galactical simulation codes. However, I ended up working closely together with Joachim Stadel on his PKDGRAV code [20], which again is one of the main cosmological N -body codes. In particular, an extremely large simulation using four trillion particles is planned to be carried out with this code, which shall then serve as the flagship simulation for the Euclid mission. As mentioned, for precise comparisons to observations, neutrinos need to be taken into account. As far as physics is concerned, however, PKDGRAV is very limited, capable only of simulating matter particles in an expanding universe. As the best way of implementing fully non-linear neutrinos is still unclear, a linear approach was opted for PKDGRAV. Here, ‘linear’ means that the neutrinos themselves are computed exclusively using linear perturbation theory, but that we include their gravitational interaction on the N -body particles (and not vice versa).

Adding neutrinos to PKDGRAV meant that it now had to interface closely with an Einstein-Boltzmann code, just as CONCEPT had been interfaced with CLASS. To save (a lot of) time, I decided to use CONCEPT as a middleman between PKDGRAV and CLASS. I added a functionality to CONCEPT (the so-called CLASS utility) enabling it to dump the processed background quantities and perturbations from CLASS to a single HDF5 file, which would then be read in from PKDGRAV.

For a linear treatment of neutrinos to be accurate, the neutrino mass $\sum m_\nu$ has to be small enough so that the neutrinos remain relativistic throughout at least most of cosmic history, as they will otherwise undergo non-linear clustering just like matter. As shown in [11], treating neutrinos linearly as opposed to non-linearly leads to a 1 % error* in the present day matter power spectrum for $\sum m_\nu \sim 0.5$ eV. For the flagship simulation,

*The error on the late-time neutrino distribution (and hence the neutrino power spectrum) itself is much larger. As we have no hope to observationally map the perturbations in the cosmic neutrino background within the near future, this is not problematic.

a value of $\sum m_\nu = 0.06$ eV has been decided. With such nearly massless neutrinos, their effect on the matter can no longer be expected to be vastly bigger than the effect from photons. A very nice feature of the linear neutrino method now implemented in PKDGRAV is that it is completely independent of the actual physical nature of the species, as the complicated physical evolution is handled by the Einstein-Boltzmann code. We thus decided to include both neutrino and photon perturbations in PKDGRAV.

When working at the level of photon perturbations, perturbations in spacetime itself (the metric) ought to be taken into account as well, as demonstrated in [12]. From the point of view of the linear implementation in PKDGRAV, this is just another species, on equal footing with photons and neutrinos. As all communication from CLASS to PKDGRAV goes through CONCEPT, the linear neutrinos (and photons etc.) essentially had to be implemented in CONCEPT first, and so CONCEPT has both the linear and non-linear implementations built in. This also keeps the interface clean on the PKDGRAV side, as CLASS does not readily provide e.g. the metric perturbations in the same form as photon perturbations.

I worked on the implementation of the linear species into CONCEPT and PKDGRAV for the last half of a year of my PhD. A paper titled ‘Fully relativistic treatment of light neutrinos in N -body simulations’, describing the method (but not the details of its implementation) and demonstrating results from it, can be found on arXiv [21]. It was submitted to JCAP in early November of 2018, from which we await response. The paper is included in this thesis in chapter 9.

In the remainder of this thesis, I will try not to describe the linear and non-linear neutrino methods as being separate. In fact the non-linear method has an associated order of linearity which can be tuned depending on how non-linear the neutrinos are expected to be. At the lowest order, the “non-linear” method reduces precisely to the linear method. As described for the linear case above, the method extends generally to other species (e.g. photons) as well, and this remains true in the non-linear case. Thus, though we have neutrinos in mind, the method presented in this work can straightforwardly be applied to other species. In fact, due to the excellent integration of CLASS into CONCEPT, any species implemented in CLASS can readily be used by CONCEPT in this manner. As an example, Katrine Alice Glasscock used CONCEPT to study the effects on matter from linear dark energy perturbations for her Bachelor’s project.

1.4 The CONCEPT Code

The CONCEPT code ('COsmological N -body CodE in PyThon') is the main product of my PhD work. It is free and open-source and available on GitHub*. It was originally written for my Master's thesis [22] in 2015. At the time the code was able to evolve dark matter particles under their own gravity in an expanding background, the evolution of which was solved internally and could contain matter, Λ and radiation. Gravity was (and is to this day) implemented using three different methods; the PP, PM and P³M methods. For the neutrino implementation developed during my PhD, only the PM method (or rather a generalisation thereof) is suitable, and so we will primarily be concerned with this method in this thesis.

Though CONCEPT is written in Python, it achieves C-like performance thanks to extensive use of Cython [23]. Furthermore it makes use of several high-performance libraries such as GSL, FFTW and HDF5. In order for CONCEPT to be able to scale to large problems it uses MPI parallelisation. Though much of the practical work put into the development of CONCEPT comes down to software technical details, often vastly complicated by the parallelisation aspect, these are not the focus of this thesis.

In order for neutrinos to be incorporated into CONCEPT, it has been extended with the following (non-exhaustive) list of features during the course of my PhD:

- Fluid dynamics.
- Integration with CLASS.
- Particle and fluid realisation.

The neutrinos are treated as a fluid rather than as particles. The non-linear fluid equations are then solved to integrate the neutrino fluid forward in time. Our strategy for doing this rely on repeated realisation (i.e. 3D real-space instantiation) of linear theory input, and so having CLASS fully integrated into CONCEPT is a must. On top of this, CLASS also provides the background evolution and the linear theory transfer functions needed for initial condition generation for both the matter particles and the neutrino fluid.

Though CONCEPT is fully usable in its current state, it lacks behind contemporary N -body codes in one aspect in particular: To achieve

*<https://github.com/jmd-dk/concept>

high spatial resolution of the particle distribution over a wide range of scales, the PM method is not suitable. Instead, the P³M (or its fancier cousin, the treePM) method could be used. Though the current P³M implementation in CONCEPT does provide the missing resolution, it is implemented non-optimally and increases the simulation time beyond acceptance. I will stay in the cosmology group of Steen Hannestad for the next six months as a postdoc, during which I will work on reimplementing the P³M method, after which I would consider CONCEPT to be highly competitive with other state-of-the-art cosmological simulation codes.

1.5 This Thesis

The remainder of this thesis is structured into three parts. Part I deals with the theory of cosmological perturbations on top of a homogeneous background, with special attention given to massive neutrinos. Part II then goes into the numerical methods developed for simulating the physics described in part I. Finally part III contains the two research papers I have co-authored during my PhD, along with conclusions and outlook.

Part I starts out with chapter 2 in which the basic cosmology of a homogeneous universe including massive neutrinos is laid out. Chapter 3 adds perturbations in all species, though the focus is on matter and neutrino perturbations in particular. These perturbations are discussed in both linear and non-linear theory, as well as both general relativistically and in the Newtonian limit. Chapter 4 introduces the idea of the power spectrum, which shall be our primary measure of the amount of structure generated within the simulations, which we can use to compare simulations with each other and with linear theory, and ultimately with real world observations.

Part II starts out with a general review of cosmological *N*-body methods in chapter 5. These are augmented with less standard methods for fluid dynamics in chapter 6, needed to evolve the non-linear neutrinos. These include an extension of my own invention to an existing method, as well as a generalisation of the PM method for gravity in the case of a simulation involving both matter particles and a neutrino fluid. In chapter 7 various techniques for realising both particles and fluid variables are developed. In particular, the linear realisation of the metric perturbations is explained and new non-linear realisation schemes are proposed, capable of performing realisations using information from both a linear transfer function and the current non-linear state of a simulation. Part II then finishes off with chapter 8 which review the CONCEPT code

and makes it clear exactly which of the many methods discussed in this part is actually implemented, and to what extent.

Part III starts out with chapter 9, which contains the paper entitled ‘Fully relativistic treatment of light neutrinos in N -body simulations’ along with a short introduction. Similarly, chapter 10 contains the paper ‘ ν CONCEPT: Cosmological neutrino simulations from the non-linear Boltzmann hierarchy’, along with a short introduction. The thesis is then concluded and future prospects considered in chapter 11.

Part I

Theory

2 The Homogeneous Universe

The key to cosmology is the fact that the Universe is statistically homogeneous on large enough scales. This in effect provides a “largest scale” of interest, beyond which nothing new appears. Together with a smallest scale provided by quantum mechanics, this constrain our possibly infinite universe to a finite range of scales, and so the amount of information needed to statistically describe it is finite as well. Throughout this chapter we will make the assumption that the Universe is homogeneous *on all scales*, which will produce equations correctly describing the Cosmos at these largest scales of interest (typically quoted as $\gtrsim 100$ Mpc). Later chapters will then go on to add inhomogeneities in order to describe smaller scales as well. Our homogeneous universe is allowed to contain quite a general assembly of different species, though we do not discuss the possibility of these to interact by any means other than gravity.

We shall work in units in which the speed of light c , Planck’s reduced constant \hbar and Boltzmann’s constant k_B are all set equal to unity. We will make use of implied summation through the Einstein index notation. All equations will be written with respect to cosmic (as opposed to conformal) time t , with an overdot $\dot{\cdot} \equiv \partial_t$. We shall often leave out explicit time dependencies when writing functions.

2.1 The Hubble Expansion

The homogeneity of the universe does not allow it to do much. If we also impose complete isotropy (consistent with large-scale observations) the only possible dynamics is that of universal expansion/contraction in all directions. That is, the symmetry of space is not shared by time, in which case the squared spacetime line element ds^2 decompose as

$$ds^2 = -dt^2 + (a(t)d\Sigma)^2, \quad (2.1)$$

with $d\Sigma$ being the line element on space, the expansion of which is encoded in the scale factor $a(t)$. Observations [24] tell us that our

Universe is flat, and so the spatial metric is just that of Euclidean space:

$$ds^2 = -dt^2 + a^2(t) d\mathbf{x}^2, \quad (2.2)$$

with $\mathbf{x} = (x, y, z)$ a Cartesian coordinate. The metric (2.2) is known as the flat Friedmann-Lemaître-Robertson-Walker (FLRW) metric. The dynamics of this FLRW metric, i.e. the evolution of the scale factor $a(t)$, depends upon the content within the universe, as described by GR (general relativity), the central equations of which are the Einstein field equations*

$$R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = 8\pi G\bar{T}_{\mu\nu}, \quad (2.3)$$

with G being Newton's gravitational constant and we have a metric $g_{\mu\nu} = \text{diag}(-1, a^2, a^2, a^2)$ for our homogeneous spacetime (2.2) in Cartesian coordinates $x^\mu = (t, \mathbf{x})$. In fact, the entire left-hand-side of (2.3) is just a particular way of writing the curvature of spacetime, and so the Ricci curvature tensor $R_{\mu\nu}$ and scalar $R \equiv R_\alpha{}^\alpha$ may be given in terms of the metric alone;

$$R_{\mu\nu} = \Gamma_{\mu\nu,\alpha}^\alpha - \Gamma_{\mu\alpha,\nu}^\alpha + \Gamma_{\mu\nu}^\alpha \Gamma_{\alpha\beta}^\beta - \Gamma_{\mu\beta}^\alpha \Gamma_{\nu\alpha}^\beta, \quad (2.4)$$

$$\Gamma_{\nu\rho}^\mu = \frac{1}{2}g^{\alpha\mu}(g_{\alpha\nu,\rho} + g_{\alpha\rho,\nu} - g_{\nu\rho,\alpha}), \quad (2.5)$$

with $\Gamma_{\nu\rho}^\mu$ the Cristoffel symbols and a comma index implies partial differentiation with respect to the following indices. The metric $g^{\mu\nu}$ with upper indices is the matrix inverse of its lower index counterpart, which for our diagonal metric reduces to elementwise reciprocals.

The right-hand-side of (2.3) contains the stress-energy tensor $\bar{T}_{\mu\nu}$, which hold within it the total energy density, momentum density, pressure and shear stress from all species, as function of time and space. The overbar is there to remind us that it is the stress-energy tensor of a homogeneous universe. For a general set of non-interacting (gravity aside) species, all of the quantities stored in the stress-energy tensor are additive and so we are allowed to combine all species into one. Since non-zero off-diagonal elements in $\bar{T}_{\mu\nu}$ (momentum or shear) would break isotropy, the state of this combined species has to be fully specified by its homogeneous energy density $\bar{\rho}$ and pressure \bar{P} , i.e. it is a perfect fluid, $\bar{T}^{\mu\nu} = \text{diag}(\bar{\rho}, \bar{P}, \bar{P}, \bar{P})$. With both sides of the Einstein field equations (2.3) fully specified, it is just a matter of working out all of the Cristoffel

*An explicit cosmological constant Λ is left out, as we shall think of this as a component of the energy density and pressure contained within $\bar{T}_{\mu\nu}$.

symbols and entries in the Ricci tensor and see what it all boils down to. This is done in e.g. [22], with the final results being the two Friedmann equations

$$\frac{\dot{a}^2}{a^2} = \frac{8\pi G}{3}\bar{\rho}, \quad (2.6)$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3}(\bar{\rho} + 3\bar{P}), \quad (2.7)$$

where the first equation results from the time/density component of the Einstein field equations, whereas the second equation results from the space/pressure component of the Einstein field equations together with the first equation.

The two Friedmann equations (2.6) and (2.7) can be combined into one by differentiating the first equation and substituting the left-hand-sides for the right-hand-sides of both equations, in which case the homogeneous continuity equation is produced:

$$\dot{\bar{\rho}} = -3\frac{\dot{a}}{a}(\bar{\rho} + \bar{P}). \quad (2.8)$$

This equation tell us the rate at which the energy density $\bar{\rho}$ dilutes given a value for the expansion rate \dot{a}/a as well as values of the density $\bar{\rho}$ and pressure \bar{P} . We could also have obtained the continuity equation from the requirement of local energy conversation, $\bar{T}^{0\alpha}_{;\alpha} = 0$, where a semicolon denotes covariant differentiation with respect to the following indices.

The homogeneous continuity equation (2.8) makes it clear that the density and pressure are not independent variables. In fact, for our set of non-interacting species, what defines a given species is the exact relation between density and pressure; an equation of state. With $\bar{\rho}(t)$ and $\bar{P}(t)$ depending only on time, it is clear that there exists some $w(t)$ such that $\bar{P}(t) = w(t)\bar{\rho}(t)$. Furthermore, since the homogeneous continuity equation really describes local conservation of energy, it will hold true for any single species (specified by the label α) with mean density $\bar{\rho}_\alpha$, not just for the total energy density $\bar{\rho}$. Substituting $\bar{\rho} \rightarrow \bar{\rho}_\alpha$ and $\bar{P} \rightarrow w_\alpha \bar{\rho}_\alpha$ in (2.8), performing separation of variables and integrating between today and some other time, we obtain

$$\bar{\rho}_\alpha(a) = \exp\left(3 \int_a^1 \frac{1 + w_\alpha(a')}{a'} da'\right) \bar{\rho}_{\alpha,0}, \quad (2.9)$$

where now a ‘0’ subscript refer to the present day value, i.e. $\bar{\rho}_{\alpha,0} \equiv \bar{\rho}_\alpha(a=1)$, where the scale factor normalisation is chosen such that it

equals unity at the present time. In writing (2.9) we have also changed the explicit temporal dependency from cosmic time t to the scale factor a , which we can freely do if $a(t)$ is bijective, i.e. a grows monotonically with t . We can see from (2.6) that this is indeed the case since $a(t) > 0$, $\bar{\rho}(t) > 0$, and so $\dot{a}(t)$ can never change sign. In the common case of constant w_α , (2.9) reduces to

$$\bar{\rho}_\alpha(a) = a^{-3(1+w_\alpha)} \bar{\rho}_{\alpha,0} \quad (\text{constant } w_\alpha), \quad (2.10)$$

which is applicable for all species discussed in this thesis except for (massive) neutrinos.

Looking back at the first Friedmann equation (2.6), we can evaluate it at the present to get $H_0^2 = 8\pi G \rho_{c,0}/3$, where $H \equiv \dot{a}/a$ is the Hubble parameter and the critical density $\rho_c = \bar{\rho}$ has been introduced, so named as this is the exact total density required to make the universe flat, given some value of H . The Hubble constant is observed [24] to be $H_0 \approx 67 \text{ km s}^{-1} \text{ Mpc}^{-1}$, from which we get $\rho_{c,0}$ equal to the mass of 5 hydrogen atoms per cubic meter of space. To express the first Friedmann equation (2.6) in terms of present-day values, we simply divide it by itself evaluated at today. Furthermore we split the total background density into contributions from separate species:

$$\frac{H^2(a)}{H_0^2} = \sum_\alpha \exp\left(3 \int_a^1 \frac{1+w_\alpha(a')}{a'} da'\right) \Omega_{\alpha,0}, \quad (2.11)$$

where the sum is over all species and the density parameters $\Omega_\alpha \equiv \bar{\rho}_\alpha/\rho_c$ have been introduced. In our flat universe, $\sum_\alpha \Omega_\alpha = 1$ and so Ω_α is simply the fraction of the total energy budget taken up by species α . Given the current composition of the universe, $\{\Omega_{\alpha,0}\}$, together with the equation of state for each species and the current expansion rate H_0 , we can now solve for the expansion history $H(a)$ or $a(t)$.

2.1.1 Cosmic Inventory

As it stands, the Friedmann equation (2.11) is very general*. Let us now decide on which species to include in our model universe. We shall follow that of standard Λ CDM with three massive neutrinos, i.e. our universe contain dark energy in the form of a cosmological constant ($\Omega_{\Lambda,0} \approx 0.68$), cold dark matter ($\Omega_{\text{cdm},0} \approx 0.27$) and baryons ($\Omega_{\text{b},0} \approx 0.049$) [24], leaving

*It even allows for curvature by thinking of this as an energy density with an equation of state $w_k = -1/3$, should be wish to relax our assumption of a flat universe.

trace amounts of photons and the three neutrino mass states, of which at least two are massive.

The cosmological constant is defined to have constant energy density $\bar{\rho}_\Lambda$. From (2.9), it is clear that this requires

$$w_\Lambda = -1, \quad (2.12)$$

which then implies the constant negative pressure $\bar{P}_\Lambda = -\bar{\rho}_\Lambda$.

The mean density $\bar{\rho}_\alpha$ and pressure \bar{P}_α generally emerge as statistical properties from microscopic physics, i.e. they can be computed from weighted integrals over the underlying distribution function $\bar{f}_\alpha(t, \mathbf{p})$, with \mathbf{p} the 3-momentum. Here the overbar is to remind us that this is the zero-order distribution function, i.e. any spatial dependencies (inhomogeneities) have been integrated out. Similarly, isotropy dictates that really $\bar{f}_\alpha(t, \mathbf{p}) = \bar{f}_\alpha(t, |\mathbf{p}|)$. From the distribution function, the mean density and pressure are given as

$$\bar{\rho}_\alpha = g_\alpha \int \frac{d^3 p}{(2\pi)^3} \bar{f}_\alpha(\mathbf{p}) \sqrt{m_\alpha^2 + \mathbf{p}^2}, \quad (2.13)$$

$$\bar{P}_\alpha = g_\alpha \int \frac{d^3 p}{(2\pi)^3} \bar{f}_\alpha(\mathbf{p}) \frac{\mathbf{p}^2}{3\sqrt{m_\alpha^2 + \mathbf{p}^2}}, \quad (2.14)$$

where $\sqrt{m_\alpha^2 + \mathbf{p}^2}$ is the energy of a single particle with momentum \mathbf{p} and mass m_α , while g_α is the degeneracy of species α (i.e. number of single-particle quantum states). With (2.13), (2.14) and some key knowledge of the different species at our disposal, we can find the individual equations of state for all remaining species. The distinguishing feature of matter, whether it being cold dark or baryonic, is that $m \gg \sqrt{\langle \mathbf{p}^2 \rangle}$, i.e. the particle mass is much larger than typical momenta (values of \mathbf{p} for which $\bar{f}(\mathbf{p})$ is non-negligible). This translates to a vanishing pressure when compared to the density, and so since $w_\alpha \equiv \bar{P}_\alpha/\bar{\rho}_\alpha$, we get

$$w_{\text{cdm}} = w_b = 0. \quad (2.15)$$

Without even bothering with the integrals of (2.13) and (2.14), we then have $\bar{\rho}_{\text{cdm}} = \bar{\rho}_{\text{cdm},0} a^{-3}$, $\bar{\rho}_b = \bar{\rho}_{b,0} a^{-3}$ and $\bar{P}_{\text{cdm}} = \bar{P}_b = 0$, where the density scalings are purely the result of the Hubble expansion (no redshifting of the energy as matter is non-relativistic).

For photons we have the opposite limit, $m_\gamma = 0$, which transform $\bar{P}_\gamma \rightarrow \bar{\rho}_\gamma/3$ and so

$$w_\gamma = \frac{1}{3}, \quad (2.16)$$

implying $\bar{\rho}_\gamma = \bar{\rho}_{\gamma,0}a^{-4}$, $\bar{P}_\gamma = \bar{\rho}_{\gamma,0}a^{-4}/3$.

In the case of massless neutrinos, the result is of course identical to that of photons. Often, photons and neutrinos are collectively referred to as “radiation” since they (may) behave similar in this respect. As we shall generally work with massive neutrinos, we shall refrain from this collective labelling and keep these species distinct.

From the observed effects of neutrino oscillations [25] we know that at least two of the three mass eigenstates of the neutrinos have non-zero masses. These masses are however very small, with a bound [24] on the sum of masses $\sum m_\nu < 0.12 \text{ eV}$. With masses in this regime, neutrinos will be ultrarelativistic and hence photon-like in the early universe, but non-relativistic and hence matter-like in the late universe, as $\sqrt{\langle \mathbf{p}_\nu^2 \rangle}$ will be redshifted from $\sqrt{\langle \mathbf{p}_\nu^2 \rangle} \gg m_\nu$ to $\sqrt{\langle \mathbf{p}_\nu^2 \rangle} \ll m_\nu$ over the course of cosmic history, and so the equation of state for massive neutrinos vary in time. In order to obtain this $w_\nu(a)$, we must then carry out the integrals of (2.13) and (2.14), using some suitably defined distribution function \bar{f}_ν .

Leaving $w_\nu(a)$ undefined for now, we can write out the Friedmann equation (2.11) for our given cosmic inventory:

$$\frac{H^2(a)}{H_0^2} = \Omega_{\Lambda,0} + \frac{\Omega_{\text{cdm},0} + \Omega_{\text{b},0}}{a^3} + \frac{\Omega_{\gamma,0}}{a^4} + \exp\left(3 \int_a^1 \frac{1+w_\nu(a')}{a'} da'\right) \Omega_{\nu,0}, \quad (2.17)$$

where specific values for $\Omega_{\gamma,0}$ and $\Omega_{\nu,0}$ have yet to be specified.

2.2 Neutrinos

In the case of photons and neutrinos, their mean densities are usually not specified by their density parameters Ω_γ and Ω_ν . We can measure the temperature of the cosmic microwave background with pristine precision, and so most often this temperature T_γ is given in place of Ω_γ . We can relate these quantities if we know the distribution function \bar{f}_γ . Before recombination (electrons and light nuclei forming atoms), the photons were in thermal equilibrium with the plasma, and so followed a Bose-Einstein distribution (due to photons being bosons) with some definite temperature. After recombination, the photons decouple from the baryons and are now able to stream freely. The photons then keep a Bose-Einstein distribution, but the temperature of this distribution is

redshifted over time due to the Hubble expansion. A similar story holds for neutrinos; they too were once in equilibrium with the cosmic plasma (though through weak as opposed to electromagnetic interactions), but has since (prior to the photons) decoupled. Generally then, we may write the zero-order distribution function* of species α as

$$\bar{f}_\alpha(a, \mathbf{p}) = \left[\exp\left(\frac{\sqrt{m_\alpha^2 + \mathbf{p}^2}}{T_\alpha(a)}\right) + \begin{cases} -1 & \alpha \in \text{bosons} \\ +1 & \alpha \in \text{fermions} \end{cases} \right]^{-1}, \quad (2.18)$$

where the ∓ 1 depending on the bosonic/fermionic nature of species α chooses between a Bose-Einstein or a Fermi-Dirac distribution. Plugging \bar{f}_γ into (2.13) using $g_\gamma = 2$ due to the two spin degrees of freedom of the photon, one obtain

$$\bar{\rho}_\gamma = \frac{\pi^2}{15} T_\gamma^4. \quad (2.19)$$

With $T_{\gamma,0} = 2.7255$ K [26] we then get $\Omega_{\gamma,0} = 5.5 \times 10^{-5}$.

We can similarly get $\bar{\rho}_\nu(T_\nu)$ by choosing the Fermi-Dirac version of (2.18) and inserting into (2.13). At early times where the neutrinos were relativistic, we can neglect their mass and get

$$\bar{\rho}_\nu(a \ll 1) = \frac{7\pi^2}{40} T_\nu^4, \quad (2.20)$$

where I have used $g_\nu = 6$ due to the three generations of neutrinos, all of which have antiparticles. No spin factor is used since the neutrinos only have a single spin degree of freedom due to their chiral nature with respect to the weak interaction.

Unlike the temperature of the CMB, we have not measured the temperature of the CvB, as we in fact have not directly observed a single cosmic neutrino. We can however link T_ν to T_γ . At a time where the neutrinos were in thermal equilibrium with the plasma and hence the photons, the whole combined system had a single temperature. Then the neutrinos decouple, locking their distribution though with a redshifting temperature. After this, the temperature of the plasma drops below the electron mass, ending e^+e^- pair production. The remaining annihilations then dump energy into the photons, heating them up, and so at this point (as today), $T_\gamma > T_\nu$. A full analysis [27] yields $T_\nu = (4/11)^{1/3} T_\gamma$ right after decoupling. For this relation to hold throughout time, the neutrino and photon temperature have to have identical scalings with

*In reality, the zero-order distribution functions will differ slightly due to non-equilibrium physics. Also, all chemical potentials are assumed to be zero.

a. From (2.16) and (2.19) we have $T_\gamma \propto \bar{\rho}_\gamma^{1/4} \propto a^{-1}$. It turns out that this scaling holds for the massive neutrinos as well, which we can see by calculating the number density,

$$\begin{aligned}\bar{n}_\nu(a) &= g_\alpha \int \frac{d^3\mathbf{p}}{(2\pi)^3} \left[\exp\left(\frac{|\mathbf{p}|}{T_\nu(a)}\right) + 1 \right]^{-1} \\ &= \frac{9\zeta(3)}{2\pi^2} T_\nu^3(a),\end{aligned}\quad (2.21)$$

where ζ is the Riemann zeta function. We thus indeed have $T_\nu \propto \bar{n}_\nu^{1/3} \propto a^{-1} \propto T_\gamma$, where the middle proportionality results from the expansion of space and the conservation of neutrino number after decoupling. Note that even though no neutrino mass appears in the distribution function explicitly written in (2.21), this equation still holds for massive neutrinos throughout time. The massless distribution function must be used as this is the distribution set at the time of thermal equilibrium, where the mass is negligible.

In actuality, the neutrino decoupling and electron-positron annihilation does not take place sufficiently separated in time so that the two processes can be considered completely separate, and so a few “relic” neutrinos will be around to take away some of the energy from the annihilations. This is parameterised through N_{eff} , so that $\bar{\rho}_\nu/\bar{\rho}_\gamma \rightarrow (N_{\text{eff}}/3)\bar{\rho}_\nu/\bar{\rho}_\gamma$, i.e. $N_{\text{eff}} = 3$ corresponds to the limit of complete separability of neutrino decoupling and e^+e^- annihilation. Simultaneous simulation [28] of neutrino decoupling and e^+e^- annihilation, taking neutrino flavour oscillations into account, yields $N_{\text{eff}} = 3.046$. Absorbing this N_{eff} into the neutrino temperature, we have

$$T_\nu = \left(\frac{N_{\text{eff}}}{3}\right)^{1/4} \left(\frac{4}{11}\right)^{1/3} T_\gamma. \quad (2.22)$$

We can now put everything together, obtaining

$$\bar{\rho}_{\nu_i} = 2 \int \frac{d^3\mathbf{p}}{(2\pi)^3} \left\{ \exp\left[\left(\frac{3}{N_{\text{eff}}}\right)^{1/4} \left(\frac{11}{4}\right)^{1/3} \frac{a|\mathbf{p}|}{T_{\gamma,0}}\right] + 1 \right\}^{-1} \sqrt{m_{\nu_i}^2 + \mathbf{p}^2}, \quad (2.23)$$

$$\bar{P}_{\nu_i} = 2 \int \frac{d^3\mathbf{p}}{(2\pi)^3} \left\{ \exp\left[\left(\frac{3}{N_{\text{eff}}}\right)^{1/4} \left(\frac{11}{4}\right)^{1/3} \frac{a|\mathbf{p}|}{T_{\gamma,0}}\right] + 1 \right\}^{-1} \frac{\mathbf{p}^2}{3\sqrt{m_{\nu_i}^2 + \mathbf{p}^2}}, \quad (2.24)$$

Table 2.1 – Cosmological background parameters used throughout this thesis, unless explicitly stated otherwise. Dark energy is assumed to take the form of a cosmological constant Λ , with $\Omega_{\Lambda,0} = 1 - (\Omega_{b,0} + \Omega_{cdm,0} + \Omega_{\gamma,0} + \Omega_{\nu_1,0} + \Omega_{\nu_2,0} + \Omega_{\nu_3,0})$, making the universe flat.

H_0	$67 \text{ km s}^{-1} \text{ Mpc}^{-1}$
$\Omega_{b,0}$	0.049
$\Omega_{cdm,0}$	0.27
$T_{\gamma,0}$	2.7255 K
m_{ν_1}	0
m_{ν_2}	$8.7 \times 10^{-3} \text{ eV}$
m_{ν_3}	$5.2 \times 10^{-2} \text{ eV}$

where now I have used $g_\nu = 6/3 = 2$ to separate out the three neutrino species ν_i , $i \in \{1, 2, 3\}$. Note that since the temperature does not depend on the mass, all three mass states share the same temperature throughout time, $T_{\nu_i} = T_\nu$. Should we wish to think of the three neutrinos $\{\nu_i\}$ as a collective species ν (as in e.g. the Friedmann equation (2.17)), we can use the additivity of density and pressure, $\bar{\rho}_\nu \equiv \sum_{i=1}^3 \bar{\rho}_{\nu_i}$ and $\bar{P}_\nu \equiv \sum_{i=1}^3 \bar{P}_{\nu_i}$, from which we get $\Omega_\nu = \sum_{i=1}^3 \Omega_{\nu_i}$. Note however that with regards to the equation of state used in the Friedmann equation, things are not so straight forward. In particular, $w_\nu \neq \sum_{i=1}^3 \bar{P}_{\nu_i} / \sum_{i=1}^3 \bar{\rho}_{\nu_i}$, but rather some sum of w_i with combinations of Ω_i as weights. Though we may sometimes want to think of all neutrino species (or some other set of species) as a single, collective species, in practice it is easier to keep all species separate.

From studies [29] of neutrino oscillations and assuming the by now slightly favoured normal neutrino mass hierarchy, it is found that $m_{\nu_2} = m_{\nu_1} + 8.7 \times 10^{-3} \text{ eV}$ and $m_{\nu_3} = m_{\nu_1} + 5.0 \times 10^{-2} \text{ eV}$, leaving m_{ν_1} unconstrained except for the cosmological upper bound [24] on $\sum m_\nu < 0.12 \text{ eV}$, implying $m_{\nu_1} < 2.0 \times 10^{-2} \text{ eV}$. For no other reason than the added interest from having one of the neutrinos be massless, we choose $m_{\nu_1} = 0$ as our standard value for this thesis, resulting in $\sum m_\nu = 5.9 \times 10^{-2} \text{ eV}$. With this, we have completely specified the background cosmology. For future reference, our choices of cosmological parameters scattered around this chapter is collected together in table 2.1.

Figure 2.1 shows the evolution of the mean density of all species, computed as described in this chapter. The mean densities are parameterised

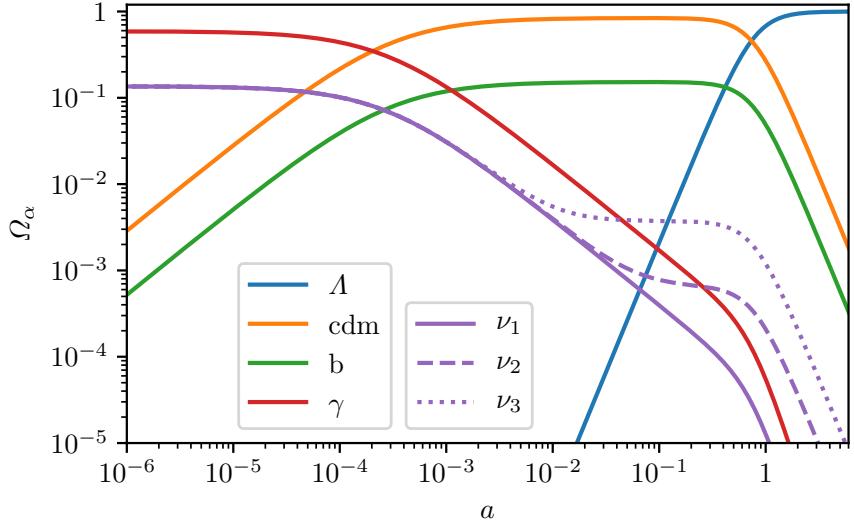


Figure 2.1 – Energy density parameters Ω_α for all the different species as function of scale factor a . Cosmological parameters are as defined in table 2.1.

as $\Omega_\alpha(a) \equiv \bar{\rho}_\alpha / \rho_c$. Note that we do not have to solve the Friedmann equation in order to obtain $\rho_c(a)$, as this is simply the sum of the individual background densities. We do however need $\rho_{c,0}$ (or equivalently H_0) to convert from $\{\Omega_{\Lambda,0}, \Omega_{\text{cdm},0}, \Omega_{\text{b},0}\} \rightarrow \{\bar{\rho}_{\Lambda,0}, \bar{\rho}_{\text{cdm},0}, \bar{\rho}_{\text{b},0}\}$. The figure clearly shows that cold dark matter and baryons behave the same; they are merely scaled (translated, on the logarithmic plot) versions of each other. The same goes for photons and the massless neutrino. As discussed earlier, massless neutrinos and photons should indeed scale similarly. The massive neutrinos ν_2 and ν_3 follow the massless ν_1 at early times when all three neutrinos are relativistic, but obtain an enhancement when transitioning to the non-relativistic regime, surpassing the photons. The more massive ν_3 goes non-relativistic much earlier than the lighter ν_2 . Lastly we see that the cosmological constant has just recently begun to dominate over matter, which in turn began to dominate over “radiation” (photons and neutrinos) around $a_{\text{rm}} = 3 \times 10^{-4}$.

We have yet to numerically solve (2.24) for the neutrino pressure. This is done for figure 2.2 which shows the equations of state $w_{\nu_i} = \bar{P}_{\nu_i} / \bar{\rho}_{\nu_i}$ for the three neutrino mass states. We see that all neutrinos obey the relativistic limit $w_{\nu_i}(a \ll 1) = 1/3$ at early times and that for the massive neutrinos, w_{ν_2} and w_{ν_3} follow decaying power laws at late times, approaching the matter limit of 0.

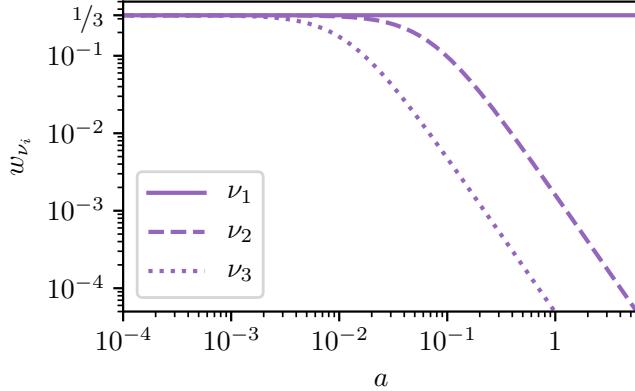


Figure 2.2 – Neutrino equation of state parameters w_{ν_i} as function of scale factor a . Cosmological parameters are as defined in table 2.1.

With the neutrino equations of state at hand we can solve the Friedmann equation* (2.17), either algebraically for $H(a)$ or via integration for $a(t)$ and $H(t)$. These are shown in figure 2.3, where the history of the universe has been separated into a radiation dominated era, a matter dominated era and a dark energy dominated era. As a bonus we also get the current age of the Universe from $a(t_0) = 1$; $t_0 = 13.8$ Gyr. The scalings of $a(t)$ and $H(t)$ in the different epochs displayed on the figure can be easily found from the single component Friedmann equation with constant w ; $H^2/H_0^2 = a^{-3(1+w)}$, where I have left out a density parameter which equals unity for a flat, single component universe. Integrating from $t = a = 0$ to some other t and a gives

$$a(t) = \left[\frac{9}{4}(1+w)^2(H_0 t)^2 \right]^{\frac{1}{3(1+w)}}, \quad (2.25)$$

which we see fails for $w = -1$. Simply setting $w = -1$ and *then* doing the indefinite integral yields $a = C \exp(H_0 t)$, with C being some undefined constant of integration. Thus a Λ -only universe has no $a = t = 0$ a finite time ago, i.e. no Big Bang, hence the failure of (2.25). Plugging

*Only a single neutrino species appears in equation (2.17). To include all three species, we simply replace the neutrino term with three identical copies, with $\{\omega_\nu, \Omega_\nu\} \rightarrow \{\omega_{\nu_i}, \Omega_{\nu_i}\}$.

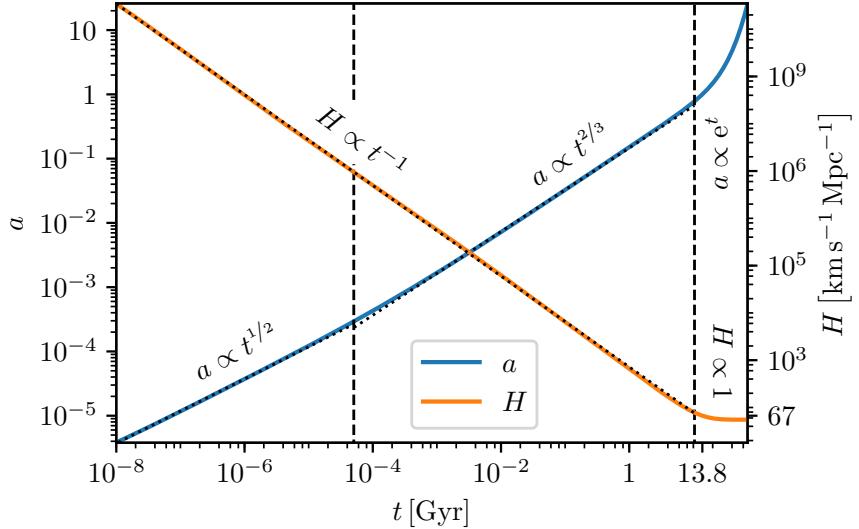


Figure 2.3 – Scale factor a and Hubble parameter H as function of cosmic time t . Cosmological parameters are as defined in table 2.1. The vertical lines are situated at matter-radiation and dark energy-matter equality, with the proportionalities showing the evolution in each epoch. Dotted trendlines show fits to these proportionalities.

radiation ($w = 1/3$) and matter ($w = 0$) into (2.25) we then have

$$\begin{aligned}
 a &= \begin{cases} (2H_0 t)^{1/2} & \text{(radiation)} \\ \left(\frac{3}{2} H_0 t\right)^{2/3} & \text{(matter)} \\ C \exp(H_0 t) & (\Lambda) \end{cases} \\
 \dot{a} &= \begin{cases} \left(2 \frac{t}{H_0}\right)^{-1/2} = H_0 a^{-1} & \text{(radiation)} \\ \left(\frac{3}{2} \frac{t}{H_0^2}\right)^{-1/3} = H_0 a^{-1/2} & \text{(matter)} \\ H_0 C \exp(H_0 t) = H_0 a & (\Lambda) \end{cases} \\
 H &= \begin{cases} (2t)^{-1} = H_0 a^{-2} & \text{(radiation)} \\ \frac{2}{3} t^{-1} = H_0 a^{-3/2} & \text{(matter)} \\ H_0 & (\Lambda), \end{cases} \quad (2.26)
 \end{aligned}$$

which indeed agrees with the numerical findings stated on figure 2.3.

We use the terms “matter” and “radiation” to refer to species with equation of state $w_m = 0$ and $w_r = 1/3$, respectively. The instance of matter-radiation equality a_{rm} is then defined from $\bar{\rho}_m(a_{rm}) = \bar{\rho}_r(a_{rm})$ and corresponds to the leftmost vertical line on figure 2.3. We see that matter-radiation equality takes place around $a_{rm} = 3 \times 10^{-4}$, at which point all the neutrinos were still relativistic according to figure 2.2. We should thus include all neutrinos as part of the radiation when computing a_{rm} , i.e. $\bar{\rho}_r(a_{rm}) = \bar{\rho}_\gamma(a_{rm}) + \bar{\rho}_{\nu_1}(a_{rm}) + \bar{\rho}_{\nu_2}(a_{rm}) + \bar{\rho}_{\nu_3}(a_{rm})$. Contrary, the dark energy-matter equivalence at a_{mA} takes place in the recent past as seen from figure 2.3 ($a_{mA} = 0.8$), which now means that the two massive neutrinos are non-relativistic (c.f. figure 2.2) and so should now be included when counting up the matter; $\bar{\rho}_m(a_{mA}) = \bar{\rho}_{cdm}(a_{mA}) + \bar{\rho}_b(a_{mA}) + \bar{\rho}_{\nu_2}(a_{mA}) + \bar{\rho}_{\nu_3}(a_{mA})$.

With the Hubble expansion as well as the evolution of the background density and pressure for every species known, we are done describing the homogeneous universe and ready to begin perturbing it.

3 The Perturbed Universe

With the homogeneous universe from the previous chapter, this chapter sets out to study the linear growth of structure given some initial perturbations. We shall first look at a perturbed matter-only universe in which we can successfully carry out the analysis in a Newtonian manner. We can even apply the results from this to cosmologies containing other species as well, as long as these do not cluster much themselves. We shall then discuss the general relativistic fluid dynamics applicable for any species. Linearising this fluid dynamics will produce linear perturbation theory, which is capable of solving for the linear growth of any species in the context of general relativity.

3.1 Newtonian Matter Perturbations

Allowing for inhomogeneities in the universe switches on gravity*, as overdense regions can now attract each other. Gravity is global in the sense that all species share the gravitational interaction, i.e. the gravitational effect on a given species cannot be computed without knowledge of all other species. For now though, we shall consider a universe containing only a matter fluid, i.e. a species which is described completely by its energy (or mass) density field ρ and its momentum density field.

3.1.1 Eulerian Equations of Motion in Comoving Space

To obtain a description of Newtonian gravity, we consider the (temporally) local limit of the FLRW metric (2.2), which then becomes the metric of special relativity (i.e. we disregard the Hubble expansion), $g_{\mu\nu} \rightarrow \eta_{\mu\nu} = \text{diag}(-1, 1, 1, 1)$. To allow for Newtonian (weak) gravity, we now perturb the metric, $g_{\mu\nu} \rightarrow \eta_{\mu\nu} + h_{\mu\nu}$ with $|h_{\mu\nu}| \ll 1$. As matter is pressureless,

*Here ‘gravity’ is meant in the usual (Newtonian) sense as an attractive force. In the larger context of general relativity, the overall Hubble expansion may also be referred to as gravity.

the only non-zero diagonal element of $T_{\mu\nu}$ is $T_{00} = \rho$. The Einstein field equations (2.3) then reduces [22] to the single equation*

$${}_r\partial_i {}_r\partial^i \phi = 4\pi G\rho, \quad (3.1)$$

which we recognize as the Poisson equation for gravity with ϕ being the gravitational potential. The seemingly unnecessary left subscripts r on the derivatives are to remind us that these derivatives are to be taken with respect to the static space coordinates r as opposed to comoving coordinates x , as we have removed the Hubble expansion in order to derive (3.1).

In static space with physical coordinates r , the equations of motion for a matter fluid specified completely by its energy density $\rho(r)$ and velocity $\dot{r}(r)$, comes down to the simple continuity and Euler equations

$$\begin{cases} {}_r\partial_t \rho &= - {}_r\partial_j \rho \dot{r}^j, \\ {}_r\partial_t \rho \dot{r}^i &= - {}_r\partial_j \rho \dot{r}^i \dot{r}^j - \rho {}_r\partial^i \phi, \end{cases} \quad (3.2)$$

where subscript t means time (the 0'th index). The first equation (continuity) states that any change in local energy density ρ must be the result of mass transfer. The second equation (Euler) states that changes in the local momentum density $\rho \dot{r}$ are due to either momentum transfer or momentum generation through the gravitational potential. In writing the Euler equation (3.2), the choice has been made to group the product $\rho \dot{r}^i$ together. This is convenient not only because we can think of $\rho \dot{r}^i$ as a fluid variable (the momentum density) in its own right, but it keeps the equation in so-called *conservation form*, where fluxes (transfer) of the fluid variable are easily recognized as divergences over the variable, while all other terms are source terms.

To transform the continuity and Euler equation (3.2) from static space r to comoving space x , we upgrade r from static to *physical* space (which is allowed to expand), related to comoving space by $r = ax$ (c.f. the FLRW metric (2.2)). Allowing the fluid variables to take in both physical and comoving arguments, e.g. $\rho(t, r) = \rho(t, ax) \equiv \rho(t, x)$, the transformation is then handled solely by the derivatives, e.g. ${}_r\partial \rightarrow {}_x\partial$. From $r = ax$ we immediately get the spatial transformation ${}_r\partial_i = a^{-1} {}_x\partial_i$. For the temporal transformation, compare the comoving and physical differential of some function $f(t, r) = f(t, x)$;

$$df = {}_x\partial_t f dt + {}_x\partial_i f dx^i$$

*Here we have made the association $h_{00} \equiv 2\phi$ in order for the potential to have the same scaling as we are accustomed to from Newtonian gravitation.

$$\begin{aligned}
&= {}_r \partial_t f dt + {}_r \partial_i f dr^i \\
&= \left(\frac{\dot{a}}{a} x^i {}_x \partial_i f + {}_r \partial_t f \right) dt + {}_x \partial_i f dx^i \\
\Rightarrow {}_r \partial_t &= {}_x \partial_t - \frac{\dot{a}}{a} x^i {}_x \partial_i,
\end{aligned} \tag{3.3}$$

where $dr^i = x^i da + a dx^i = \dot{a} x^i dt + a dx^i$ has been used. Knowing how ${}_r \partial_\mu$ transforms, the continuity and Euler equation (3.2) in comoving space become

$$\begin{cases} {}_x \partial_t a^3 \rho = -a^2 {}_x \partial_i \rho u^i, \\ {}_x \partial_t a^4 \rho u^i = -a^3 {}_x \partial_j \rho u^i u^j - a^3 \rho {}_x \partial^i \left(\phi + \frac{1}{2} a \ddot{a} x_j x^j \right), \end{cases} \tag{3.4}$$

where the fluid variables are now chosen to be the comoving energy density $a^3 \rho$ and momentum density $a^4 \rho u^i$, with $u^i \equiv a \dot{x}^i$ being the peculiar velocity. Stating the comoving continuity and Euler equation (3.4) in terms of these variables makes them appear very similar to their static space versions (3.2), with the Hubble expansion only* appearing through multiplicative factors of a (we shall deal with the \ddot{x} term below).

The Peculiar Potential

The gravitational source term of (3.4) looks almost nonsensical due to the $x_j x^j$ term. To fix this, define the peculiar potential φ as the parenthesis in (3.4);

$$\varphi(\mathbf{x}) \equiv \phi(\mathbf{r}) + a \ddot{a} \mathbf{x}^2 / 2, \tag{3.5}$$

where I have changed to vector notation as the following will be easier expressible this way. Though the \mathbf{x}^2 term is still hiding inside the definition of φ , it disappears (due to the Laplacian) when writing out the Poisson equation for φ ;

$${}_x \nabla^2 \varphi = 4\pi G a^2 (\rho - \bar{\rho}), \tag{3.6}$$

where we have used (3.1) with ${}_x \partial_i = a {}_r \partial_i$ and the subtraction of the mean density appears from substitution of \ddot{a} via the second Friedmann equation (2.7). As before, the left subscript x reminds us that the differentiation happens in comoving space. The Poisson equation (3.6) shall be our new definition of φ , and so we can forget that the strange $x_j x^j$

*Had we stated e.g. the continuity equation in terms of ${}_x \partial_t \rho$, a new Hubble drag term $-3H\rho$ would appear, just as we had in (2.8).

term ever appeared. The price to pay is that now gravity is determined from the density perturbation $\delta\rho \equiv \rho - \bar{\rho}$ rather than just the density ρ . This is referred to as ‘Jeans swindle’. Intuitively we would expect a (spatially) constant offset in the density field to make no difference. Inverting the Laplacian on the left-hand-side to some integral over all of space on the right-hand-side of (3.6), it is clear that such an integral can only be convergent if the integrated density has zero spatial mean, demonstrating the importance of the subtraction of the mean.

We now want to carry out the above mentioned inversion of the Laplacian in the Poisson equation (3.6). The inverse Laplacian will be an integral over all of space with some kernel $\mathcal{G}_{\mathbf{x}}\nabla^2$ (the Green’s function of the Laplacian), i.e. $\varphi = 4\pi G a^2 \int d^3\mathbf{x}' \mathcal{G}_{\mathbf{x}}\nabla^2(\mathbf{x}, \mathbf{x}') \delta\rho$. Reapplying the Laplacian should convert back to the Poisson equation, and so $_{\mathbf{x}}\nabla^2 \mathcal{G}_{\mathbf{x}}\nabla^2 = \delta(\mathbf{x}' - \mathbf{x})$, with δ the Dirac delta function. The solution is $\mathcal{G}_{\mathbf{x}}\nabla^2 = -1/(4\pi|\mathbf{x}' - \mathbf{x}|)$, and so the potential alone can be expressed as

$$\varphi(\mathbf{x}) = -G a^2 \int d^3\mathbf{x}' \frac{\delta\rho(\mathbf{x}')}{|\mathbf{x}' - \mathbf{x}|}. \quad (3.7)$$

Similarly, we want an expression for the force $_{\mathbf{x}}\nabla\varphi$, which we can get from the Poisson equation in a similar manner to what we just did, or more easily by simply taking the gradient of (3.7):

$$_{\mathbf{x}}\nabla\varphi(\mathbf{x}) = -G a^2 \int d^3\mathbf{x}' \frac{\mathbf{x}' - \mathbf{x}}{|\mathbf{x}' - \mathbf{x}|^3} \delta\rho(\mathbf{x}'). \quad (3.8)$$

3.1.2 Lagrangian Equations of Motion in Comoving Space

In the previous subsection we found the non-linear equations of motion for matter in comoving coordinates. We described matter as a fluid through its density $\rho(t, \mathbf{x})$ and peculiar velocity $u^i(t, \mathbf{x})$, i.e. we can choose some spatial location \mathbf{x} and ask about the density and pressure at this point. This is called an Eulerian description of the fluid. Though this is the description we will need for now, we shall later be using the Lagrangian description, where we follow specific fluid parcels — or particles — as they move through space. Labelling these particles with an n , the system is then described by $\{\mathbf{x}_n(t), \mathbf{q}_n(t)\}$, where $\mathbf{q}_n \propto m_n \mathbf{u}_n$ is the comoving momentum canonical to \mathbf{x}_n , with m_n the mass of particle n .

One can obtain the Lagrangian description from the Eulerian description. As our system of gravitationally interacting matter particles is

rather simple, we shall instead derive the Lagrangian equations directly from the single-particle Lagrangian in static space;

$$\mathcal{L}_n(t, \mathbf{r}_n, \dot{\mathbf{r}}_n) = \frac{1}{2}m_n\dot{\mathbf{r}}_n^2 - m_n\phi(\mathbf{r}_n). \quad (3.9)$$

From $\mathbf{r} = a\mathbf{x} \Rightarrow \dot{\mathbf{r}} = \dot{a}\mathbf{x} + a\dot{\mathbf{x}} = \dot{a}\mathbf{x} + \mathbf{u}$ and the relation (3.5) between the physical (or proper) potential ϕ and the peculiar potential φ , we can immediately write the above Lagrangian in comoving space;

$$\begin{aligned} \mathcal{L}_n(t, \mathbf{x}_n, \dot{\mathbf{x}}_n) &= \frac{1}{2}m_n\mathbf{u}_n^2 - m_n\varphi(\mathbf{x}_n) + \frac{d}{dt}\frac{1}{2}a\dot{a}m_n\mathbf{x}_n^2 \\ &\rightarrow \frac{1}{2}m_n\mathbf{u}_n^2 - m_n\varphi(\mathbf{x}_n), \end{aligned} \quad (3.10)$$

where the spurious term $\propto \mathbf{x}_n^2$ is removed by a gauge transformation, which correspond to Jeans swindle introduced for the Eulerian case.

With the Lagrangian in comoving space (3.10) defined, the comoving momentum \mathbf{q}_n canonical to \mathbf{x}_n comes out to be

$$\mathbf{q}_n \equiv \frac{\partial \mathcal{L}_n(t, \mathbf{x}_n, \dot{\mathbf{x}}_n)}{\partial \dot{\mathbf{x}}_n} = a^2m_n\dot{\mathbf{x}}_n = am_n\mathbf{u}_n. \quad (3.11)$$

The Hamiltonian $\mathcal{H}_n(t, \mathbf{x}_n, \mathbf{q}_n) \equiv \dot{\mathbf{x}}_n\mathbf{q}_n - \mathcal{L}_n(t, \mathbf{x}_n, \mathbf{q}_n)$ is then

$$\mathcal{H}_n(t, \mathbf{x}_n, \mathbf{q}_n) = \frac{\mathbf{q}_n^2}{2a^2m_n} + m_n\varphi(\mathbf{x}_n), \quad (3.12)$$

from which the equations of motion $\dot{\mathbf{x}}_n = \partial\mathcal{H}_n/\partial\mathbf{q}_n$, $\dot{\mathbf{q}}_n = -\partial\mathcal{H}_n/\partial\mathbf{x}_n$, become

$$\begin{cases} \dot{\mathbf{x}}_n = \frac{\mathbf{q}_n}{a^2m_n}, \\ \dot{\mathbf{q}}_n = -m_n \nabla \varphi|_{\mathbf{x}=\mathbf{x}_n} \equiv \mathbf{f}_n, \end{cases} \quad (3.13)$$

where the left index on the gradient is once again to remind us that the gradient it to be taken in comoving space, and $\mathbf{f}_n(t)$ is the comoving force acting on particle n .

With the Hamiltonian equations established, let us now explicitly express the Eulerian fluid fields in terms of the Lagrangian particles. For point-like particles, we have

$$\begin{cases} a^3\rho(\mathbf{x}) = \sum_n m_n\delta(\mathbf{x}_n - \mathbf{x}), \\ a^4\rho\mathbf{u}(\mathbf{x}) = \sum_n \mathbf{q}_n\delta(\mathbf{x}_n - \mathbf{x}). \end{cases} \quad (3.14)$$

Integrating (3.14) over some comoving volume will result in the enclosed mass (first equation) and the enclosed comoving momentum (second equation), both of which should be constant in time for a homogeneous universe. The exponents of a on the left-hand-sides are then chosen such that the resulting energy (here simply mass) and momentum densities are in conservation form as in (3.4).

The Hamiltonian equations (3.13) together with the Poisson equation (3.6) (or equivalently one of (3.7) or (3.8)) completely describe the system of particles. For a Poisson equation for φ expressed in terms of particles, we can replace $\delta\rho$ in the Poisson equation with the sum over particles as given in (3.14).

3.1.3 Linear Perturbations

From this point on we will stay in the comoving frame with coordinates \mathbf{x} , and so we shall drop the left subscript on derivatives. We shall also go back to the Eulerian description of the matter fluid. Collecting the results from subsection 3.1.1, the continuity, Euler and Poisson equations for the perturbed matter-only universe are

$$\begin{cases} \partial_t a^3 \rho = -a^2 \partial_i \rho u^i, \\ \partial_t a^4 \rho u^i = -a^3 \partial_j \rho u^i u^j - a^3 \rho \partial^i \varphi, \\ \partial_i \partial^i \varphi = 4\pi G a^2 \delta\rho. \end{cases} \quad (3.15)$$

The assumption going into equations (3.15) were that of weak (Newtonian) gravity. The fluid variables ρ and u^i have not themselves been linearised, and so (3.15) holds true in non-linear theory as well.

We define the *density contrast* $\delta(\mathbf{x})$ as the dimensionless “perturbation” to $\rho(\mathbf{x})$,

$$\rho(a, \mathbf{x}) = \bar{\rho}(a)[1 + \delta(a, \mathbf{x})], \quad (3.16)$$

although generally we do not assume δ to be small. Right now however that is exactly what we shall do. Keeping only linear terms of δ and u^i (meaning $\delta^2 = u^i u^j = \delta u^i = 0$) in (3.15), we can combine the three equations into one by taking the temporal derivative of the continuity equation and the spatial derivative of the Euler equation, resulting in

$$\ddot{\delta} + 2H\dot{\delta} - 4\pi G\delta\bar{\rho} = 0 \quad (\delta \ll 1). \quad (3.17)$$

Note that only for $\delta \ll 1$ is the variable $\delta\rho \equiv \delta\bar{\rho}$ equal to the product $\delta\rho$. Generally we wish to solve for the growth of structure (3.17) throughout history, not just during matter domination. We thus relax the assumption

of the universe being matter-only. The δ in (3.17) is still the density contrast of just the matter, but $\delta\bar{\rho}$ is really the density perturbation of all species, as it is what provides gravity. On the other hand we know for a fact that it is only the matter in our Universe which cluster strongly, and so we may take $\delta\bar{\rho}$ in (3.17) to only include matter as well. Expressing the background matter density as $\Omega_{m,0}\rho_{c,0}a^{-3}$ with the current critical density given by the Friedmann equation (2.6), we can write (3.17) as

$$\ddot{D} + 2H\dot{D} - \frac{3}{2}\Omega_{m,0}H_0^2a^{-3}D = 0, \quad (3.18)$$

where I have substituted the linear growth factor $D(a)$ for $\delta(a)$, which is somewhat nicer as δ generally is a spatial field, whereas (3.17) and (3.18) do not depend on \mathbf{x} . The growth of structure given by (3.17) and (3.18) are then completely scale independent.

In the radiation era, the background matter density was relatively insignificant (see figure 2.1), and we know that initially the matter contrast δ have to be very small. We can then neglect the $\delta\bar{\rho}$ term in the radiation era. We similarly ignore the $\delta\bar{\rho}$ term during the Λ epoch, this time because $\bar{\rho}$ exponentially decays with time. For matter domination, we keep the term and approximate $\bar{\rho} = \rho_c$. With these approximations we now plug H from (2.26) into (3.17) and get [22]

$$D(a) = \begin{cases} C_r \log a + C'_r & (\text{radiation}) \\ C_m a + C'_m a^{-3/2} & (\text{matter}) \\ C_\Lambda + C'_\Lambda a^{-2} & (\Lambda), \end{cases} \quad (3.19)$$

where the C 's are constants of integration. Disregarding the slow logarithmic growths during radiation domination, we see that δ only grows during matter domination. The large-scale structure in the universe then only has a finite temporal window within which to form.

Figure 3.1 shows the numerical solution of the matter growth factor $D(a)$, in our complete universe simultaneously containing matter, photons, neutrinos and Λ . As we learned from the analytical approximations (3.19), we do indeed find that matter perturbations only grow during matter domination, and that this growth very precisely follows $D \propto a$. As the matter epoch spans ~ 3 orders of magnitude in a , linear perturbation theory then predicts a similar ~ 3 orders of magnitude growth of δ . This suggests that we cannot treat the perturbations as small throughout time, establishing the need for non-linear simulations.

What we have considered “matter” in this section has been a pressureless fluid, i.e. cold dark matter and baryons. In a universe with

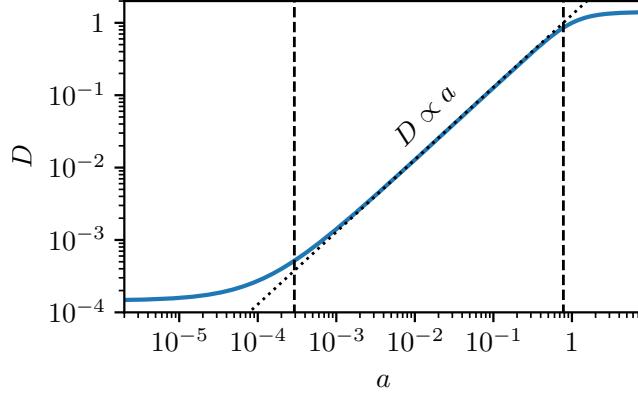


Figure 3.1 – Growth factor D as function of scale factor a . Cosmological parameters are as defined in table 2.1. The growth factor has been normalised so that $D(a = 1) = 1$. The vertical lines are situated at matter-radiation and dark energy-matter equality. The dotted trendline shows a perfect $D \propto a$ relation.

massive neutrinos however, we know that these will act matter-like at late times, including the late phase of matter domination, depending on the mass (see figure 2.2). Similar to cold dark matter and baryons, we then expect structure in the massive neutrino distribution to also grow significantly during matter domination, increasing $\delta\rho_\nu$ to a point where it can no longer be neglected from the total density perturbation $\delta\bar{\rho}$ in (3.17). The growth factor D as defined through (3.18) is then a good description of the linear growth of matter only in universes without massive neutrinos.

3.2 General Relativistic Fluid Dynamics

In this section we will study the general relativistic fluid dynamics of any species. For this we will first need to define a common set of fluid variables by which any fluid can be fully described, and then we need to find evolution equations for each of these variables. As we will not have any particular set of species in mind, we will leave out the α index on the various variables.

3.2.1 Fluid Variables and the Boltzmann Hierarchy

The biggest mathematical structure we have so far encountered, holding fluid variables such as the energy density, momentum density and

pressure, has been the stress-energy tensor $T_{\mu\nu}$. As this is a symmetric 4×4 tensor, it generally contains 10 independent* variables. We may take these to be† [31]

$$\begin{cases} T^0_0 = -\rho, \\ T^i_0 = -(\rho + P)u^i, \\ T^i_j = P\delta^i_j + (\rho + P)u^i u_j + \Sigma^i_j, \end{cases} \quad (3.20)$$

where δ^i_j is the Kronecker delta and the symmetric and traceless shear stress Σ^i_j is the only new variable, from which we further define

$$\Sigma^i_j \equiv (\rho + P)\sigma^i_j. \quad (3.21)$$

In the case of complete homogeneity we saw in (2.13) and (2.14) how the density and pressure arose as weighted integrals — or *moments* — of the underlying distribution function f , though the same formalism holds true in the general case. Completely analogous to (2.13) and (2.14), the stress-energy tensor can be constructed as

$$T^\mu_\nu = \sqrt{-g} \int d^3P f \frac{P^\mu P_\nu}{P_0}, \quad (3.22)$$

where g is the determinant of the metric, $d^3P = dP^1 dP^2 dP^3$ and P^μ the conjugate momentum to the comoving x^μ . Indeed, setting $\mu = \nu = 0$ in (3.22) we obtain $\rho \propto \int d^3P f P^0$ with P^0 the energy density, which matches (2.13), while setting $\mu = \nu = i$ in (3.22) yield $3P \propto \int d^3P f P^i P_i / P_0$ (with P the pressure), where the 3 comes from the sum (trace), which again matches (2.14).

Equation (3.22) can be seen as a description for generating fluid variables as momentum/velocity moments over f . Let us write these

*Really only 6 due to the 4 Bianchi identities.

†Note the inclusion of the term $(\rho + P)u^i u_j$ which is missing in e.g. (25) of [30]. This term is clearly needed when we consider a perfect fluid for which $T^i_j = P\delta^i_j + (\rho + P)u^i u_j$. It is left out of (25) of [30] and much other literature because $u^i u_j$ vanish to linear order in perturbations.

variables down in a list:

$$\left\{ \begin{array}{l} T_0^0(t, \mathbf{x}) = \sqrt{-g} \int d^3 \mathbf{P} f P^0, \\ T_0^i(t, \mathbf{x}) = \sqrt{-g} \int d^3 \mathbf{P} f P^i, \\ T_j^i(t, \mathbf{x}) = \sqrt{-g} \int d^3 \mathbf{P} f \frac{P^i P_j}{P_0}, \\ \Pi_{jk}^i(t, \mathbf{x}) = \sqrt{-g} \int d^3 \mathbf{P} f \frac{P^i P_j P_k}{P_0 P_0}, \\ \vdots \end{array} \right. \quad (3.23)$$

We see that each new variable is an object with one additional index, and that the rule for constructing the next variable is just to multiply the integrand by P_ℓ/P_0 . This enable us to construct higher-order moments such as Π_{jk}^i , which are new fluid variables not contained in the stress-energy tensor. The infinite list of variables (3.23) is called the Boltzmann hierarchy. The temporal and spatial dependency is contained in the fluid variables, while no momentum dependency is left due to the integrations. Viewing the infinite hierarchy as a monomial expansion of f in velocity ($P_\ell/P_0 \sim u_\ell$) moments then guarantees that all the information of f is preserved in the infinite list of fluid variables, assuming only that f be analytic.

3.2.2 Fluid Equations

When analysing Newtonian fluid dynamics in section 3.1, we did not use the underlying distribution function but instead the fluid variables ρ and ρu^i . These exactly correspond to the first two fluid variables in the Boltzmann hierarchy (3.23) for a fluid with no pressure (c.f. (3.20)). In fact, having no pressure is the same as $T_i^i = 0$, while having no shear stress means $\Sigma_j^i = 0$, which in turn leaves T_j^i as just some combination of the lower order fluid variables, and is hence not an independent variable. If the even higher moments of the fluid in question also vanish, the hierarchy becomes finite. In this subsection we will not assume that any fluid variable vanishes. The goal is then to construct the evolution equation for each variable to see how the different orders are related.

As mentioned in section 2.1, local energy-momentum conservation is expressed by the vanishing covariant derivative of the stress-energy tensor,

$$0 = T^\alpha_{\mu;\alpha} = \partial_\alpha T^\alpha_\mu + \Gamma^\beta_{\beta\alpha} T^\alpha_\mu - \Gamma^\beta_{\alpha\mu} T^\alpha_\beta, \quad (3.24)$$

which indeed amounts to four equations, corresponding to the continuity and Euler equations. As (3.24) refer only to the stress energy tensor and the metric, we are guaranteed that the continuity and Euler equation can contain nothing but the fluid variables ρ , u^i , P and Σ^i_j , together with gravity from the metric. For the evolution of e.g. the pressure itself, (3.24) cannot help us. For a general approach to constructing the evolution equations of fluid variables then, we have to go beyond (3.24).

As the fluid variables are constructed from the distribution function f , we can use the evolution of f itself to obtain the evolution of any given fluid variable. In GR, the non-manifestly covariant collisionless Boltzmann equation takes the form [32, 33]

$$P^\alpha \frac{\partial f}{\partial x^\alpha} - P^\alpha P^\beta \Gamma_{\alpha\beta}^i \frac{\partial f}{\partial P^i} = 0. \quad (3.25)$$

Multiplying this equation by the contravariant invariant volume element [18] $\sqrt{-g} d^3P/P_0$ and integrating, we obtain integrals of the sort we see in (3.23) but with spatial and importantly temporal derivatives of f . Using this procedure while introducing the additional needed factors of P_ℓ/P_0 according to (3.23) in the integrand, we can find the evolution equations of the fluid variables.

To carry out the integrations over (3.25), we need to settle on a gauge. In [18] this is done in conformal Newtonian gauge [30] with line element

$$ds^2 = -(1 + 2\psi) dt^2 + a^2(1 - 2\phi) dx^2. \quad (3.26)$$

With the metric perturbations being small, $\psi, \phi \ll 1$, we recognize (3.26) as a perturbed FLRW metric (2.2). Multiplying the Boltzmann equation (3.25) by the invarianet volume element together with P_0 and then integrating, [18] finds the continuity equation

$$\begin{aligned} \delta' = & - (1 + w)(\theta - 3\phi') - 3 \frac{a'}{a} \left(\frac{\delta P}{\delta \rho} - w \right) \delta \\ & - \theta \delta - u^i \partial_i \delta \\ & + 3 \left(1 + \frac{\delta P}{\delta \rho} \right) \phi' \delta - \frac{\delta P}{\delta \rho} \theta \delta - u^i \partial_i \left(\frac{\delta P}{\delta \rho} \delta \right) \\ & - (\partial_i \psi - 3\partial_i \phi) \left(1 + \delta + w + \frac{\delta P}{\delta \rho} \delta \right) u^i, \end{aligned} \quad (3.27)$$

where $\delta P \equiv P - \bar{P}$, $\theta \equiv \partial_i u^i$ and a prime refers to differentiation with respect to conformal time, $' \equiv \partial_\tau = a\partial_t$. Applying the same procedure

but this time multiplying by $P_0 P^i / P^0$, [18] finds

$$(u^i)' = - \left[\frac{a'}{a} (1 - 3w) - \psi' - 5\phi' \right] u^i - \frac{\left[\delta' + w' + \left(\frac{\delta P}{\delta \rho} \delta \right)' \right] u^i + \delta^{ij} (1 + \delta) \partial_j \psi}{1 + \delta + w + (\delta P / \delta \rho) \delta} - \frac{[\delta^{ik} (\partial_j + \partial_j \psi - 3\partial_j \phi) + \delta_j^k \delta^{il} \partial_l \phi] T_k^j}{\bar{\rho} [1 + \delta + w + (\delta P / \delta \rho) \delta]} . \quad (3.28)$$

Both (3.27) and (3.28) are also derived in [31] from the vanishing of the covariant derivative of the stress-energy tensor (3.24).

We could continue down the Boltzmann hierarchy and find $(T_j^i)'$, corresponding to P' and $(\Sigma^i_j)'$. As laid out in [18], $(T_j^i)'$ will contain the divergence of Π^i_{jk} as well as the double contraction of Π^i_{jkl} , where I have reused the symbol Π for the rank 4 fluid variable (the fluid variable specification is completely contained in the number of indices). For the massive neutrinos studied during my PhD work, it seems as though the two lowest order fluid equations (3.27) and (3.28) are enough at the non-linear level, and so I have not explored further up the Boltzmann hierarchy.

As they stand, (3.27) and (3.28) are definitely not in conservation form. As we wish to solve these partial differential equations numerically it is much preferable to have them in conservation form, as this respects the conserved quantities/underlying symmetries. That is, if we use the conserved form numerically, the physical conserved quantities are guaranteed to also be conserved numerically, regardless of numerical inaccuracies, improving numerical stability of the solutions. In transforming (3.27) and (3.28) to conservation form, we simultaneously switch from conformal time τ to cosmic time t ; $' = \partial_\tau \rightarrow a \partial_t = a'$. We do this for no other reason other than the fact that CONCEPT uses cosmic time internally, and so in order to implement the equations in CONCEPT they have to be expressed with respect to cosmic time.

We saw in (3.15) how the conserved energy density in comoving coordinates was $a^3 \rho$ for a matter fluid. In the static limit where flux terms disappear, we can understand this a^3 as coming from the scaling $\bar{\rho} \propto a^{-3}$ for matter. The product $a^3 \bar{\rho}$ is then a constant, i.e. it is conserved. In the case of radiation, it is $a^4 \bar{\rho}$ that would be constant. With these limits in place, we define the *conserved energy density* as

$$\varrho \equiv a^{3(1+w)} \rho , \quad (3.29)$$

satisfying the criteria for conservation for both matter and radiation limit. In (3.29), w is the *effective equation of state*, which equal the normal equation of state w when this is constant, as in the case of matter and radiation. In the general case of varying $w(a)$, $w(a)$ is simply defined to be whatever function makes (3.29) be the conserved variable. If we compare (2.9) with (2.10), we see that we must have

$$\begin{aligned} a^{-3(1+w)} &\equiv \exp\left(3 \int_a^1 \frac{1+w(a')}{a'} da'\right) \\ &\Rightarrow w = \frac{1}{\ln a} \int_1^a \frac{w(a')}{a'} da', \end{aligned} \quad (3.30)$$

which indeed satisfy $w = w$ in the case of constant w .

For the momentum density, (3.15) shows that $a^4\rho u^i$ is the conserved quantity for matter. This makes sense since for matter, $\rho \propto a^{-3}$, $u^i \propto a^{-1}$, providing the need for the a^4 . For radiation, we have $\rho \propto a^{-4}$ and constant peculiar velocity u^i , and so we could define the *conserved momentum density* as just $a^4\rho u^i$, independent of w (or w). Looking back at the equation for $(u^i)'$ (3.28), we see that it contains a temporal derivative of the pressure, the expression for which we do not know, although we do know what it takes to find it, as described earlier. We can bypass this difficulty by including the pressure into our definition of the conserved momentum density;

$$J^i \equiv a^4(\rho + P)u^i, \quad (3.31)$$

where the addition of P does not break the previous arguments of conservation as $\bar{P} = w\bar{\rho}$, with w constant for matter and radiation.

As we do not concern ourselves with the time evolution of the pressure and shear stress in this work, we do not need to find conserved versions of these. However, I have found that the equations simplify if we introduce

$$\mathcal{P} \equiv a^{3(1+w)}P, \quad (3.32)$$

$$\varsigma_j^i \equiv a^{3(1+w)}\Sigma_j^i = (\varrho + \mathcal{P})\sigma_j^i, \quad (3.33)$$

which we shall refer to as the *conserved pressure* and the *conserved shear stress*, respectively, although it might turn out that (3.32) and (3.33) are not actually the conserved quantities once we know the equations for \dot{P} and $\dot{\Sigma}_j^i$.

Writing the continuity (3.27) and Euler (3.28) equation only in terms of the conserved variables, we get

$$\left\{ \begin{array}{l} \dot{\varrho} = -a^{3w-2}\partial_i J^i \\ \quad + 3H(w\varrho - \mathcal{P}) \\ \quad + a^{3w-2}J^i\partial_i(3\phi - \psi) \\ \quad + 3(\varrho + \mathcal{P})\dot{\phi}, \\ J^i = -\partial^j \left[a^{3w-2} \frac{J^i J_j}{\varrho + \mathcal{P}} + a^{-3w} (\mathcal{P}\delta^i_j + \varsigma^i_j) \right] \\ \quad - a^{-3w}(\varrho + \mathcal{P})\partial^i\psi - a^{3w-2} \frac{J^j J_j}{\varrho + \mathcal{P}} \partial^i\phi \\ \quad + \left[a^{3w-2} \frac{J^i J_j}{\varrho + \mathcal{P}} + a^{-3w} \varsigma^i_j \right] \partial^j(3\phi - \psi) \\ \quad + J^i(\dot{\psi} + 5\dot{\phi}), \end{array} \right. \quad (3.34)$$

where we have switched back to cosmic time derivatives. The equivalent equations with conformal time derivatives can be found in [18]. Note that the many factors of a to various powers are not the result of switching to cosmic time, but instead a necessary price we pay for using the conserved variables.

I have written equations (3.34) so that similar physical terms appear together on the same line. The first line of both equations contains the entire flux; for the continuity equation this is a single energy flux, while for the Euler equation we have momentum, pressure and shear flux.

The second line in the continuity equation is a Hubble source term, the very sort of term we are trying to avoid by working in conserved quantities. I conjecture that no choice of energy density variable can completely rid the continuity equation for such a Hubble term, at least not without introducing similar terms in the Euler equation. Note that the specific Hubble source term in the continuity equation (3.34) is as benign as can be, as it vanishes in the homogeneous limit since $\bar{P} = w\bar{\rho}$, as well as in the perfect fluid limit where $P = w\rho$.

The third line of the continuity equation and the second and third line of the Euler equation are gravitational source terms coming from spatial derivatives of the potentials. As $J^i J_j \sim u^i u_j$ and ς^i_j similarly can be expected to be of the order of $u^i u_j$, the $(\varrho + \mathcal{P})\partial^i\psi$ term in the Euler equation dominate. This is indeed precisely the Newtonian gravitational source term, with $\psi = \varphi$ being the Newtonian peculiar gravitational potential. Remembering that a P is hiding inside J^i , explaining the

unusual appearance of the pressure in the Newtonian gravitational source term.

The last line of both the continuity and the Euler equation are still more general relativistic gravitational source terms, this time through temporal derivative of the potentials. As with the other relativistic gravitational source terms, we may neglect these when working with Newtonian gravity.

Note that the claim of (global) “conservation” of ϱ and J^i can now be explicitly checked by spatially averaging the right-hand-sides of (3.34): This makes the flux terms vanish as we have no gradients, the gravitational potentials vanish as they are produced by inhomogeneities, and we have already discussed how the Hubble source term vanish in the case of homogeneity. Thus ϱ and J^i as defined in (3.29) and (3.31) with w defined in (3.30) are indeed good choices for conserved energy and momentum density variables.

As a sanity check, we should make sure that the general continuity and Euler equations (3.34) reduces to their Newtonian matter-versions (3.15). The transformations that we need are $\varsigma_j^i = 0$ and $w = w = \mathcal{P} = 0 \Rightarrow \varrho \rightarrow a^3 \rho$, $J^i \rightarrow a^4 \rho u^i$, which once we throw away the general relativistic source terms exactly converts the continuity and Euler equations (3.34) into those of (3.15).

The non-linear continuity and Euler equation (3.34) needs to be augmented with equations for the metric perturbations ϕ and ψ . Such equations can be found by inserting the perturbed metric (3.26) and the non-linear stress-energy tensor (3.20) into the Einstein equations (2.3). The linear-order results are given later in subsection 3.3.2. Generally when working in non-linear theory, we assume that gravity can be well described by its Newtonian form*, i.e. the Poisson equation given in e.g. (3.15). As noted above, this corresponds to removing all gravitational source terms from the continuity and Euler equation (3.34), except for the $(\varrho + \mathcal{P})\partial^i\psi$ term in the Euler equation. With $\psi = \varphi$, the Poisson equation is then simply that of (3.15), which expressed using conserved variables is

$$\partial_i \partial^i \varphi = 4\pi G a^{-3w-1} \sum_{\alpha} \delta \varrho, \quad (3.35)$$

where the explicit sum over all species reminds us that the peculiar potential $\varphi \approx \psi \approx \phi$ is shared amongst all species, each of which have

*We shall later put in GR corrections, but effectively this will be done by modifying the density field present in the Poisson equation. We will not (re)introduce a second potential with a separate Poisson-like equation.

their own continuity equation, Euler equation and entire Boltzmann hierarchy in general.

3.3 Linear Perturbation Theory

The non-linear continuity and Euler equations (3.27) and (3.28), or equivalently their conservation form (3.34), hold true for any *collisionless* species (meaning one that only interacts through gravity) as they are derived from the collisionless Boltzmann equation (3.25). For matter, we saw how the Boltzmann hierarchy (3.25) ended at the momentum/Euler equation level, meaning that the continuity and Euler equation only refer to fluid variables at or below this level, i.e. the energy and momentum density. Appending to the continuity and Euler equation some equations for the gravitational potentials ψ and ϕ , which we can get by just plugging the perturbed metric (3.26) into the (perturbed version of) the Einstein equations (2.3), we have a fully specified system of equations governing the non-linear evolution of matter, with the only approximations being that matter is collisionless* and that the metric perturbations are linear, i.e. the potentials ψ and ϕ are small, which is an extremely good approximations at cosmic scales.

The most general species is one which have non-zero pressure and shear stress while also having a time dependent equation of state $w(a)$, which is the case for massive neutrinos. Here, the continuity and Euler equations require us to know these additional quantities. We could continue up the Boltzmann hierarchy (3.23), deriving equations for the pressure and shear, which in turn would require the knowledge of even higher-order variables, and so on ad infinitum. As solving just the non-linear continuity and Euler equation is already computationally demanding, another approach is needed.

As the Universe becomes more and more homogeneous when we go up in scale or back in time, we are able to successfully apply perturbation theory at these scales and times. In this section we will outline the basics of *linear* perturbation theory, where we seek to find evolution equations for the perturbed quantities at the linear level, i.e. we throw away any term containing two or more perturbed variables. We will find that the linearised version of the Boltzmann hierarchy is principally manageable up to arbitrary order.

*This is not true for the baryons and photons as these interact with each other. We shall not worry about this complication here.

3.3.1 Continuity and Euler equation

It will turn out that δ and $\theta \equiv \partial_i u^i$ are the natural ‘‘energy’’ and ‘‘momentum’’ variables to use in linear perturbation theory. We can get their time evolution to linear order simply by linearising (3.27) and (3.28), where we have to take the divergence of (3.28) to convert from u^i to θ :

$$\begin{cases} \delta' = (1+w)(-\theta + 3\phi') - 3\frac{a'}{a}\left(\frac{\delta P}{\bar{\rho}} - w\delta\right), \\ \theta' = \left[\frac{a'}{a}(3w-1) - \frac{w'}{1+w}\right]\theta + \frac{k^2\delta P}{\bar{\rho}(1+w)} + k_i k^j \sigma_j^i + k^2\psi, \end{cases} \quad (3.36)$$

where we have gone back to conformal time derivatives just as in (3.27) and (3.28), and also changed to Fourier space by $\partial^i \rightarrow ik^i$. As in writing the equations above, we will most often not distinguish between a variable in real space and its Fourier space counterpart. In ambiguous circumstances we shall write e.g. $\delta(\mathbf{x})$ and $\delta(\mathbf{k})$ for the real and Fourier space variable, respectively.

Doing linear perturbation theory in Fourier space turns out to be far easier than keeping the equations in real space. The reason is that in Fourier space, the linear equations decouple into modes. That is e.g. $\delta(\mathbf{k})$ and $\theta(\mathbf{k})$ does not depend on anything at some other $\mathbf{k}' \neq \mathbf{k}$. The same was not true in real space due to the derivatives. Each equation then has to be solved over and over again at various \mathbf{k} , whereas in real space we effectively had to solve all of these equations simultaneously, as they are not decoupled.

In linearising the Euler equation (3.36) we see that any mention of the velocity (or momentum) is through the velocity divergence θ . This effectively reduces the vector u^i to the scalar θ , and hence lower the dimensionality of the Euler equation to a scalar equation. For the velocity vector u^i to be fully specified by its divergence, it has to be irrotational*. To invert the divergence in $\theta = \partial_i u^i$ we make use of this irrotationality, from which we know that u^i can be written as the gradient of some scalar field Υ ; $u^i \equiv \partial^i \Upsilon$. We then have $\theta = \partial_i \partial^i \Upsilon \xrightarrow{\mathcal{F}} -k^2 \Upsilon \Rightarrow \Upsilon = -\theta/k^2$,

*A rotational part would be generated by vector perturbations, whereas the divergence is generated by scalar perturbations. As these perturbations are generated from a very nearly isotropic state of the universe, vector perturbations are extremely hard to induce and so we neglect them entirely. Although tensor perturbations are not nearly as suppressed as vector perturbations, we similarly neglect these, meaning that we construct σ_j^i from a single scalar field as well.

where \mathcal{F} denotes the Fourier transform. From the implicit definition of Υ in Fourier space, $u^i \equiv ik^i\Upsilon$, we then get

$$u^i = -i\frac{k^i}{k^2}\theta, \quad (3.37)$$

allowing us to reconstruct the irrotational, linear velocity vector from θ .

It is clear that since the velocity now only has scalar degree of freedom, the linear Fourier space equations (3.36) are “isotropic” in the sense that they can only depend on the magnitude of \mathbf{k} , $|\mathbf{k}| \equiv k$, not the direction. Thus, $\delta(\mathbf{k}) \rightarrow \delta(k)$, $\theta(\mathbf{k}) \rightarrow \theta(k)$, and so on for all the other spatial variables.

Just as we have replaced the velocity vector u^i with its divergence $\sim k^i\theta$, we now want to replace the shear stress tensor σ^i_j with something like $\sim k_i k^j \sigma$, with $\sigma(k)$ some scalar. When doing so we must however remember to preserve the properties of σ^i_j , i.e. its symmetry $\sigma^i_j = \sigma^j_i$ and its tracelessness $\sigma^i_i = 0$, the latter of which is not consistent with merely doing the replacement $\sigma^i_j \rightarrow k_i k^j \sigma$. The way to define such a symmetric, traceless, 3×3 tensor from some scalar Υ (not related to the temporary scalar field of the same name used to define u^i) is $\sigma^i_j = (\partial^i \partial_j - 1/3 \delta^i_j \partial^k \partial_k) \xrightarrow{\mathcal{F}} -k^2(\hat{k}^i \hat{k}_j - \delta^i_j / 3)\Upsilon$, where $k^i \equiv k \hat{k}^i$. We shall define the scalar shear stress σ following the convention of [30];

$$\begin{aligned} \sigma &\equiv -\left(\hat{k}_i \hat{k}^j - \frac{1}{3} \delta_i^j\right) \sigma^i_j \\ &= \frac{2}{3} k^2 \Upsilon \\ \Rightarrow \sigma^i_j &= -\frac{3}{2} \left(\hat{k}^i \hat{k}_j - \frac{1}{3} \delta^i_j\right) \sigma, \end{aligned} \quad (3.38)$$

where $\hat{k}^i \hat{k}_i = 1$ and $\delta^i_j \delta_i^j = \delta^i_i = 3$ has been used. Note that once we got the symmetric and traceless σ^i_j , multiplying (3.38) by $k_i k^j$ and summing over i and j reveals the simple relation

$$\sigma = -\frac{k_i k^j}{k^2} \sigma^i_j, \quad (3.39)$$

as we might expect on the basis of analogy to $\theta = ik_i u^i$.

Inserting (3.38) into the Euler equation (3.36), we arrive at our final incarnation of the general linearised Euler (and continuity) equation,

$$\begin{cases} \delta' = (1+w)(-\theta + 3\phi') - 3\frac{a'}{a}\left(\frac{\delta P}{\bar{\rho}} - w\delta\right), \\ \theta' = \left[\frac{a'}{a}(3w-1) - \frac{w'}{1+w}\right]\theta + \frac{k^2 \delta P}{\bar{\rho}(1+w)} - k^2 \sigma + k^2 \psi. \end{cases} \quad (3.40)$$

Though a lot of terms have dropped, the linear system of equations (3.40) has a similar structure to its non-linear counterpart with regards to the Boltzmann hierarchy, i.e. the interconnectedness of the fluid variables is much the same. A big difference is that the velocity and shear variable are now scalars, making the Euler equation a scalar equation. Linearising the fluid equations (while again ignoring vector and tensor perturbations) thus helps tremendously in reducing the dimensionality of the system, as the otherwise growing number of indices on the tensor variables in the Boltzmann hierarchy (3.23) now vanish completely, reducing every higher-order fluid equations to scalar equations. Extending the Boltzmann hierarchy beyond the Euler equation then seems reasonable in linear perturbation theory.

3.3.2 Beyond the Euler Equation

Let us look at the linear continuity and Euler equation (3.40) for matter, where $w = \delta P = \sigma = 0$:

$$\begin{cases} \delta' = -\theta + 3\phi', \\ \theta' = -\frac{a'}{a}\theta + k^2\psi. \end{cases} \quad (\text{matter}) \quad (3.41)$$

Except for the general relativistic ϕ' term, we can easily obtain the same results by linearising the non-linear Newtonian matter continuity and Euler equation (3.15) (removing the $\partial_j \rho u^i u^j$ term entirely), where we will also need the homogeneous continuity equation (2.8) to get from ρ' to δ' and remember that $\psi = \varphi$. Importantly, the system of equations (3.41) is closed*, as we have already seen from the non-linear case.

In the general case of non-vanishing δP and σ , the Euler equation will include these on its right-hand-side. To find the equations of such higher-order variables, set aside the Boltzmann hierarchy for now and go back to the distribution function f , the perturbed version of which we may define as

$$f(\tau, \mathbf{x}, \mathbf{q}) = \bar{f}(q)[1 + \Psi(\tau, \mathbf{x}, \mathbf{q})], \quad (3.42)$$

where \bar{f} is the background level distribution function from section 2.2 and Ψ is the perturbation. The perturbation is allowed to vary in space x^i and depend on the direction of the momentum. We follow [30] and use the comoving momentum q_j as the momentum variable. We now

*The Poisson equation is of course also needed.

expand the Fourier space version of the perturbation Ψ in a Legendre series,

$$\Psi(\tau, \mathbf{k}, \mathbf{q}) = \sum_{\ell=0}^{\infty} (-i)^{\ell} (2\ell + 1) \Psi_{\ell}(\tau, \mathbf{k}, q) P_{\ell}(\hat{\mathbf{k}} \cdot \hat{\mathbf{q}}), \quad (3.43)$$

which create a Boltzmann-like hierarchy of variables Ψ_{ℓ} . The relation between these and our usual variables follows the prescription (3.23). Using the orthogonality of the Legendre polynomials, [30] finds

$$\left\{ \begin{array}{l} \delta = \frac{4\pi}{a^4 \bar{\rho}} \int q^2 dq \sqrt{q^2 + a^2 m^2} \bar{f} \Psi_0, \\ \theta = \frac{4\pi k}{a^4 \bar{\rho}(1+w)} \int q^2 dq q \bar{f} \Psi_1, \\ \delta P = \frac{4\pi}{3a^4} \int q^2 dq \frac{q^2}{\sqrt{q^2 + a^2 m^2}} \bar{f} \Psi_0, \\ \sigma = \frac{8\pi}{3a^4 \bar{\rho}(1+w)} \int q^2 dq \frac{q^2}{\sqrt{q^2 + a^2 m^2}} \bar{f} \Psi_2. \end{array} \right. \quad (3.44)$$

Solving the Boltzmann equation with the perturbed distribution function (3.42), [30] are able to find the dynamical equations

$$\left\{ \begin{array}{l} \Psi'_0 = -\frac{qk}{\sqrt{q^2 + a^2 m^2}} \Psi_1 - \phi' \frac{d \ln \bar{f}}{d \ln q}, \\ \Psi'_1 = \frac{qk}{3\sqrt{q^2 + a^2 m^2}} (\Psi_0 - 2\Psi_2) - \frac{\sqrt{q^2 + a^2 m^2} k}{3q} \psi \frac{d \ln \bar{f}}{d \ln q}, \\ \Psi'_{\ell \geq 2} = \frac{qk [\ell \Psi_{\ell-1} - (\ell+1) \Psi_{\ell+1}]}{(2\ell+1) \sqrt{q^2 + a^2 m^2}}, \end{array} \right. \quad (3.45)$$

where crucially the last equation holds recursively for any $\ell \geq 2$. Equations (3.45) then allow us to solve the dynamics of any collisionless species. Note that the time evolution of moment Ψ_{ℓ} only depends on the neighbouring moments $\Psi_{\ell \pm 1}$. We can thus choose some maximum ℓ , ℓ_{\max} , after which we close the hierarchy, either through simple truncation ($\Psi_{\ell > \ell_{\max}} \rightarrow 0$) or by some closed-form algebraic approximation for $\Psi_{\ell_{\max}+1}$.

Note also that gravity only acts on the two lowest moments of (3.45), corresponding to energy, momentum and pressure, i.e. the gravitational effects we already know from the continuity and Euler equations. Even in the case of non-linear gravity, where the weak-field metric (3.26) is replaced by a completely general metric, only the next level Ψ_2 will be

further affected, as the Einstein field equations (2.3) only couple the metric to variables within the stress-energy tensor, i.e. the highest-order variable affected is the second-order variable σ_j^i , equivalent to its scalar version σ in linear perturbation theory.

While equations (3.45) hold for any collisionless species, we saw that the much simpler equations (3.41) hold for matter. Indeed, one can regain (3.41) from (3.45) in the matter limit. Similarly, a simpler (though still infinitely recursive) system of equations can be found by considering the case of massless species [30], where the q dependence of the distribution function can be integrated out. The resulting recursively defined variables are then not the Ψ_ℓ moments under the integrals of (3.45), but instead the integrals themselves, i.e. the fluid variables. Thus, the massless counterpart to the general hierarchy (3.45) is much less computational demanding, as we can solve for the whole integral for each ℓ in one go, rather than separately solving multiple sample points of the integrand.

The massless version of the general (3.45) can be used to describe massless neutrinos and photons. In actuality, the photons are more complicated since these are not collisionless, and so a collision term need to be added to the Boltzmann equation (3.25). This interaction is between the photons and the baryons/electrons, which is extremely important prior to recombination. The interaction is also important later however, due to reionization. This interaction then also means that we cannot use the simple “matter” equations (3.41) for baryons.

With all moments defined by (3.45), the only unspecified quantities left are the metric perturbations ψ and ϕ . These are found from the (perturbed version of) the Einstein equation (2.3), where we plug in the perturbed metric (3.26) and the perturbed stress-energy tensor. These

take the form of the following four equations [30]:

$$\left\{ \begin{array}{l} k^2\phi + 3\frac{a'}{a}\left(\phi' + \frac{a'}{a}\psi\right) = 4\pi Ga^2 \sum_{\alpha} (-\delta\rho_{\alpha}), \\ k^2\left(\phi' + \frac{a'}{a}\psi\right) = 4\pi Ga^2 \sum_{\alpha} \bar{\rho}_{\alpha}(1 + w_{\alpha})\theta_{\alpha}, \\ \phi'' + \frac{a'}{a}(\psi' + 2\phi') \\ + \left(2\frac{a''}{a} - \frac{a'^2}{a^2}\right)\psi + \frac{k^2}{3}(\phi - \psi) = 4\pi Ga^2 \sum_{\alpha} \delta P_{\alpha}, \\ k^2(\phi - \psi) = 4\pi Ga^2 \sum_{\alpha} 3\bar{\rho}_{\alpha}(1 + w_{\alpha})\sigma_{\alpha}. \end{array} \right. \quad (3.46)$$

Each equation above is analogous to the Poisson equation found in (3.15). In fact, in the Newtonian limit where $a' = \phi' = 0$, $\phi = \varphi$, the first equation above is identical to the Poisson equation. From the last equation above, we see that in general $\phi = \psi$ unless the universe contains shear stress.

This concludes our description of the inner workings of linear perturbation theory. From now on, we shall not deal directly with any of the linear equations, but instead use numerical tools which can solve the linear evolution equations for all species simultaneously, including the photon-baryon coupling, while also solving their gravitational interaction as described by the weak-field limit of GR. Such codes are known as ‘Einstein-Boltzmann solvers’. The main two used today are CAMB [34] and CLASS [17], where I have used the latter throughout my PhD work. A slightly modified version of CLASS is included when installing CONCEPT.

3.3.3 Transfer Functions

As the result of linearisation of the dynamical equations, we ended up with the linear continuity and Euler equations (3.40), as well as the general linear hierarchy (3.45). As these are linear differential equations, the absolute scale of the dynamical variables does not matter. That is, if $\{\delta(a, k), \theta(a, k), \delta P(a, k), \sigma(a, k), \dots\}$ is the solution to the linear equations, so is $\{C\delta(a, k), C\theta(a, k), C\delta P(a, k), C\sigma(a, k), \dots\}$ for any constant C . Of course the real Universe has chosen one specific scaling, and so in the end we will need to care about this C .

To distinguish between the honest physical variables and the variables resulting from solving the equations of linear perturbation theory, we

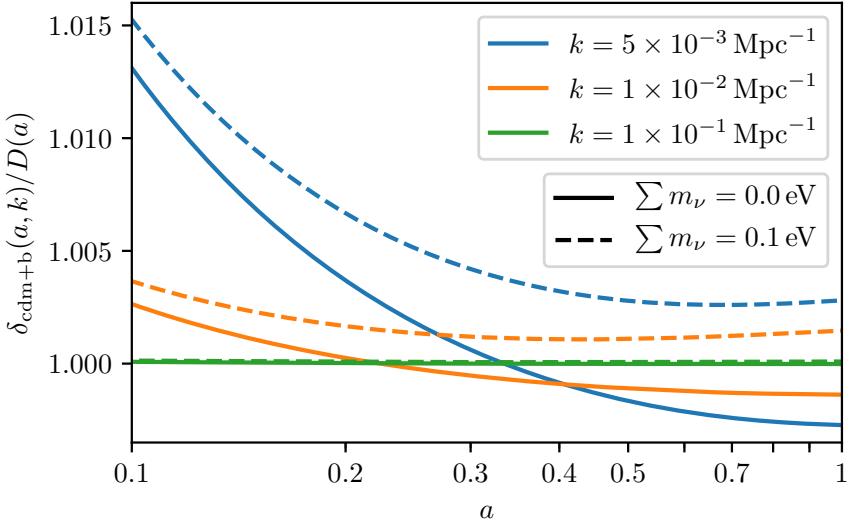


Figure 3.2 – Ratio of the linear matter (combined cold dark matter and baryon) density contrast $\delta(a, k)$ and the growth factor $D(a)$ as function of a . Three cases of k are shown. Cosmological parameters are as defined in table 2.1, though with either three massless neutrinos or with a degenerate neutrino mass hierarchy with $\sum m_\nu = 0.1 \text{ eV}$. Both D and δ have been computed in CLASS, though with the normalisation of D removed. Also, δ is shown in conformal Newtonian gauge. The ratios have been scaled so that they equal ~ 1 at $a = 1$, with individual scaling for each k but with the same scaling applying across the two cosmologies for any given k .

refer to the latter as *transfer functions*. Given a set of transfer functions and the knowledge of the physically correct value of any of the variables at any^{*} point $\{a, k\}$, we can then rescale the corresponding transfer function accordingly, while also scaling all other transfer functions by the same amount.

In the rest of this thesis, we shall denote transfer functions as e.g. T_δ , while reserving δ for the corresponding properly scaled physical quantity. To be clear, this does not mean that e.g. the variables on the left-hand-sides of (3.44) should have been written as T_δ , T_θ , $T_{\delta P}$ and T_σ , rather the equations hold true for both the physical quantities and their transfer functions, and the scaling applies to Ψ_ℓ (or T_{Ψ_ℓ}) as well.

We have previously encountered the same scaling freedom in the equations (3.18) when discussing the growth factor $D(a)$, which we arbitrarily scaled so that $D(a = 1) = 1$ in figure 3.1. Figure 3.2 compares

*As long as this point is well within the linear regime, of course.

D to the linear matter δ computed in CLASS. In the Newtonian regime, these should be very close, provided that we do not have strong perturbations outside the matter sector, i.e. heavy massive neutrinos. A critical difference between $D(a)$ and $\delta(a, k)$ is the k -dependence of δ . The Newtonian regime then corresponds to large k . Indeed, figure 3.1 shows excellent agreement between D and δ for $k = 0.1 \text{ Mpc}^{-1}$, while the same is not true for the larger scales. The linear growth of matter is then really somewhat k dependent, even in the case of massless neutrinos. Adding mass to the neutrinos alters this k dependence.

Increasing the neutrino mass (while keeping Ω_{cdm} and Ω_b fixed) raises δ directly due to the added gravitational interacting from the clustering of massive neutrinos. On the other hand, D is only affected indirectly by the neutrinos through the Hubble expansion; an effect which is of course also taken into account by δ . The net result is then that δ/D increases with the neutrino mass, which is what we see in figure 3.2.

4 Power Spectra

This chapter introduces the notion of a *power spectrum*, the absolute measure used to describe the amount of structure at a given scale. We then go on to describe the primordial power spectrum, i.e. the initial conditions set very near the time of the Big Bang from which all future structure spring.

4.1 Power Spectra as Measures of Correlation

We seek a measure for the amount of structure in the universe. Equation (3.16) defines the density contrast $\delta(\mathbf{x})$ which associates each point in 3D real space with a number, fully specifying the energy density distribution. As all spatial locations are statistically equivalent (homogeneity on large scales), a specific point \mathbf{x} is not very meaningful. What *is* meaningful is instead the distance scale $x = |\mathbf{x}|$ or its Fourier equivalent $k = |\mathbf{k}|$. A statistical measure of the amount of structure on the length scale x can e.g. be taken as the two-point correlation function

$$\xi(x) = \langle \delta(\mathbf{x}')\delta(\mathbf{x} - \mathbf{x}') \rangle_V , \quad (4.1)$$

where $\langle \bullet \rangle_V$ denotes averaging over a (comoving) volume V , i.e. let \mathbf{x}' take on all values satisfying $(4\pi/3)|\mathbf{x} - \mathbf{x}'|^3 \leq V$. For $\xi(x)$ to truly describe the statistics of the distribution $\delta(\mathbf{x})$, V has to be large enough* so that the averaging has converged, making ξ independent of V . The fact that (4.1) has x on the left and \mathbf{x} on the right just means that any chosen direction of \mathbf{x} will lead to the same result, guaranteed by large-scale isotropy.

Replacing the real space δ 's in (4.1) with their Fourier decompositions and exploiting the reality of $\delta(\mathbf{x})$, $\delta_{-\mathbf{k}} = \delta_{\mathbf{k}}^*$ with the asterisk denoting complex conjugation, we can derive

$$\xi(x) = V^{-2} \sum_{\mathbf{k}} \langle |\delta_{\mathbf{k}}|^2 \rangle_V e^{i\mathbf{k}\mathbf{x}}$$

*Large-scale homogeneity guarantees the existence of a large enough V .

$$\begin{aligned} &\equiv V^{-1} \sum_{\mathbf{k}} P_{\delta}(k) e^{i \mathbf{k} \cdot \mathbf{x}}, \\ &= \int \frac{d^3 k}{(2\pi)^3} P_{\delta}(k) e^{i \mathbf{k} \cdot \mathbf{x}}, \end{aligned} \quad (4.2)$$

where the middle line define the power spectrum $P_{\delta}(k) \equiv V^{-1} \langle |\delta_{\mathbf{k}}|^2 \rangle_V$, where the averaging in Fourier space should be done “in shells”, i.e. average over all \mathbf{k} with the same $|\mathbf{k}|$. It is understood that V should be taken large enough so that there are enough 3D Fourier modes of scale k to make the averaging converge for any given k . In the final line of (4.2) we have removed any reference to V , allowing \mathbf{k} to take on arbitrarily small values, corresponding to $V \rightarrow \infty$.

To be precise, the above power spectrum is really the δ power spectrum as it is build from the δ field, whereas we can imagine a similar construction done for e.g. the θ field. Furthermore, we could have chosen to work with the $\delta\rho$ power spectrum $P_{\delta\rho} = \bar{\rho}^2 P_{\delta}$ instead of the δ power spectrum P_{δ} . When leaving out further specification, *the* power spectrum will refer to the δ power spectrum. Including a factor V^{-1} in the averaging above, i.e. $\langle \bullet \rangle_V \equiv V^{-1} \int_V \bullet d^3x$, the δ power spectrum then has units of volume.

As the power spectrum is just the Fourier transform of the two-point correlation function, they contain the same information. Furthermore, if the full 3D distribution $\delta(\mathbf{x})$ is a Gaussian random field, i.e. each mode has mean zero and is fully defined by its variance, then $\langle |\delta_{\mathbf{k}}|^2 \rangle_V = V P_{\delta}(k)$ contains the entire statistical information of $\delta(\mathbf{x})$. In the general case of an arbitrary distribution $\delta(\mathbf{x})$, a complete statistical description is captured only by the set of all correlation functions, i.e. $\{ \langle \delta(\mathbf{x}) \delta(\mathbf{x} - \mathbf{x}') \rangle_V, \langle \delta(\mathbf{x}) \delta(\mathbf{x} - \mathbf{x}') \delta(\mathbf{x} - \mathbf{x}'') \rangle_V, \dots \}$.

4.2 The Primordial Power Spectrum

The current structure in the Universe is largely the product of gravity doing its work over billions of years, exaggerating small, linear overdensities into galaxy clusters. Ultimately however, these early time linear fluctuations cannot themselves be produced in this way, as gravity needs inhomogeneities to work. The common explanation for the primordial structure of the Universe is that of inflation, proposing a hypothetical scalar field which dominated the Universe at very early times. This inflaton field makes the Universe expand exponentially, blowing up quantum fluctuations in the inflaton field to macroscopic scales, after which the field decays into the standard model particles.

Table 4.1 – Primordial cosmological parameters used throughout this thesis, unless explicitly stated otherwise.

A_s	2.1×10^{-9}
n_s	0.96
α_s	0
k_{pivot}	0.05 Mpc^{-1}

Many models of inflation exists, but the general predictions of interest to us here is the shape of the primordial power spectrum, the fact that all current species necessarily shared this primordial power spectrum, and the fact that the primordial fluctuations were (at least very nearly) Gaussian, implying that the primordial distribution of all species are fully specified by their power spectrum.

One generic prediction of inflation is that the primordial power spectrum P_s is close to being “flat”, meaning that the dimensionless power spectrum $k^3 P_s$ does not vary much across a wide range of k . One often model this as

$$P_s(k) = 2\pi^2 A_s k^{-3} \left(\frac{k}{k_{\text{pivot}}} \right)^{n_s-1} \exp \left[\frac{\alpha_s}{2} \ln \left(\frac{k}{k_{\text{pivot}}} \right)^2 \right], \quad (4.3)$$

where the ‘tilt’ or ‘spectral index’ $n_s \approx 1$ in order for $k^3 P_s$ to be flat. The overall scale is set by the amplitude A_s with $k^3 P_s(k_{\text{pivot}}) = 2\pi^2 A_s$, the ‘pivot scale’ k_{pivot} being some chosen scale at which to define A_s . Finally, α_s is the ‘tilt running’. Choosing $k_{\text{pivot}} = 0.05 \text{ Mpc}^{-1}$, current best-fit values [24] of the three parameters are $A_s = 2.1 \times 10^{-9}$, $n_s = 0.96$, $\alpha_s \approx 0$. For future reference, these parameters are listed in table 4.1.

Just as a general power spectrum $P_\delta(k)$ was given in terms of the square of a Fourier space perturbation, we write the primordial power spectrum $P_s(k)$ as the square of the *primordial curvature perturbation* ζ , $P_s(k) = \zeta^2(k)$, implying

$$\zeta(k) = \pi \sqrt{2A_s} k^{-3/2} \left(\frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \exp \left[\frac{\alpha_s}{4} \ln \left(\frac{k}{k_{\text{pivot}}} \right)^2 \right]. \quad (4.4)$$

Note that we define $\zeta(k)$ as time independent, i.e. we only define it at some initial instance right after inflation. Also note that the presence of the definite scale A_s ensures that $\zeta(k)$ is absolutely defined. This is the opposite behaviour of the transfer functions T introduced in

subsection 3.3.3, which were defined throughout time but lacked an absolute scale. With the absolute scale of primordial fluctuations in any species determined by ζ , we can write a general linear power spectrum as*

$$P_{\delta_\alpha}(a, k) = \zeta^2(k) T_{\delta_\alpha}^2(a, k), \quad (4.5)$$

where α labels the species. This link between transfer functions and absolute power spectra, themselves defined by (4.2), finally makes it clear how to interpret the k dependence of the linear variables/transfer functions of section 3.3, as now $T_\delta(k) \propto \sqrt{\langle \delta_k^2 \rangle_V}$, i.e. a dependence on k can be thought of as the root-mean-square of the corresponding 3D \mathbf{k} dependent quantity, with the mean being carried out in shells of constant $|\mathbf{k}|$.

To nail down the normalization of transfer functions and power spectra and thus imbue the value of A_s with absolute meaning, we can pick some a and k for which we simply define $T(a, k)$ to have some definite value, where T is some specific transfer function. In our notation here, the choice of CLASS corresponds to $T_\zeta(a = 0, k) = -1$, with T_ζ the transfer function for the primordial curvature perturbation, which in the above is defined only at the initial time, corresponding to $a = 0$. From (4.5), this results in the wanted $P_s(a = 0, k) = \zeta^2(k)$. As the curvature perturbation is really a dynamic perturbation in its own right, $\zeta = \zeta(a, k)$, the transfer function $T_\zeta(a, k)$ is linked to all other transfer functions, and so once a normalisation is chosen for this, the normalisation of every transfer function is specified.

4.2.1 The Variance as Amplitude

In (4.5), the normalisation of the power spectrum was delegated to A_s within the primordial curvature perturbation ζ . This make A_s the fundamental parameter responsible for the amplitude of fluctuations. A different approach is to just scale the power spectrum until it matches some predefined criterion. A common approach is to specify a definite variance of the density field, $\sigma^2 = \langle \delta^2(\mathbf{x}) \rangle_V - \langle \delta(\mathbf{x}) \rangle_V^2 = \langle \delta^2(\mathbf{x}) \rangle_V = \xi(0)$. From (4.2) then, σ^2 is just the integral over the power spectrum. As the observable power spectrum is not available to us for arbitrarily large k , one chooses a scale at which to smooth out $\delta(\mathbf{x})$ before doing the integration. The smoothing is done by convolving $\delta(\mathbf{x})$ by a simply

*This is the CLASS convention. Transfer functions from other sources may use slightly different convention.

top-hat function

$$W_R(\mathbf{x}) = \begin{cases} \left(\frac{4\pi}{3}R^3\right)^{-1} & |\mathbf{x}| < R \\ 0 & \text{otherwise} \end{cases} \quad (4.6)$$

$$\Rightarrow W_R(k) = \frac{3}{(kR)^3} [\sin(kR) - kR \cos(kR)], \quad (4.7)$$

which in Fourier space turns into simple multiplication,

$$\sigma_R^2 = \frac{1}{2\pi^2} \int_0^\infty dk k^2 W_R^2(k) P_\delta(k), \quad (4.8)$$

where the angular part has been integrated out. The standard choice is $\sigma_{R=8 \text{ Mpc } h^{-1}} \equiv \sigma_8$, where $h \equiv H_0/(100 \text{ km s}^{-1} \text{ Mpc}^{-1})$. That is, a given value for σ_8 can be used instead of A_s . As written in (4.8), σ_8 is really a function of time since $P_\delta(k) = P_\delta(a, k)$. Whenever a value for σ_8 is provided, it is always the present day value. A further sloppiness of (4.8) is that it does not specify from which species the power spectrum is taken. In practice, σ_8 is always the variance of the matter density field.

For any otherwise specified cosmology, there is then a one-to-one relationship between A_s and σ_8 . However, while A_s is a proper input parameter, σ_8 (as it is defined at $a = 1$) is really only obtained after solving all of the dynamical equations. Furthermore, as σ_8 is the variance of the linear density field at the non-linear time and scale $\{a, k\} = \{1, 8 \text{ Mpc } h^{-1}\}$, σ_8 is just as A_s not a directly observable quantity.

4.3 Realisations

We have seen how we can express the statistical information of a 3D distribution $\delta(\mathbf{x})$ by its power spectrum $P_\delta(k)$, even capturing the entirety of this information when $\delta(\mathbf{x})$ is Gaussian. We now want to reverse the process, creating a 3D distribution from the power spectrum. This amounts to inverting the averaging operation in the definition (4.2) of the power spectrum, which of course cannot be done without supplying the missing 3D information by hand. We encode this missing information in $\mathcal{R}(\mathbf{k})$, which we know should be a Gaussian random field with zero mean. Then,

$$\delta(a, \mathbf{k}) \equiv V^{1/2} \delta(a, k) \mathcal{R}(\mathbf{k}), \quad (4.9)$$

from which we see that $\mathcal{R}(-\mathbf{k}) = \mathcal{R}^*(\mathbf{k})$ as $\delta(\mathbf{x}) \in \mathbb{R}$.

If we specify the variance of the random Gaussian field $\mathcal{R}(\mathbf{k})$, its properties will be fully specified. Taking this to be unity, it follows from (4.9) that

$$\langle |\mathcal{R}(\mathbf{k})|^2 \rangle_V \equiv 1 \Rightarrow \delta(k) = V^{-1/2} \sqrt{\langle |\delta(\mathbf{k})|^2 \rangle_V}, \quad (4.10)$$

which then finally defines the exact meaning of 1D Fourier space quantities like $\delta(k)$ as simply the shell-average of the corresponding 3D quantity. From the definition of the power spectrum (4.2) and its relation to transfer functions (4.5), we now also have

$$\delta(a, k) = T_\delta(a, k)\zeta(k), \quad (4.11)$$

which explains why we wanted to include a factor of $V^{1/2}$ in (4.9). Note that as $\zeta(k)$ has units of length^{3/2}, so does $\delta(k)$. Also, $\delta(\mathbf{k})$ has units of volume.

The dimensionless 3D real space quantity $\delta(a, \mathbf{x})$ is given as the Fourier transform of $\delta(a, \mathbf{k})$,

$$\begin{aligned} \delta(a, \mathbf{x}) &= V^{-1} \sum_{\mathbf{k}} \delta(a, \mathbf{k}) e^{i\mathbf{k}\mathbf{x}} \\ &= V^{-1/2} \sum_{\mathbf{k}} \delta(a, k) \mathcal{R}(\mathbf{k}) e^{i\mathbf{k}\mathbf{x}} \\ &= V^{-1/2} \sum_{\mathbf{k}} T_\delta(a, k) \zeta(k) \mathcal{R}(\mathbf{k}) e^{i\mathbf{k}\mathbf{x}}, \end{aligned} \quad (4.12)$$

where the volume V should reflect the minimum $|\mathbf{k}|$ used in the sum; $\mathbf{k} = 2\pi\mathbf{h}/V^{1/3}$, $\mathbf{h} \in \mathbb{Z}^3 \setminus \mathbf{0}$. Considering the $V \rightarrow \infty$ limit, we could write (4.12) as a Fourier integral rather than a series, ridding of us any mention of V and hence a scale above which $\delta(\mathbf{x})$ is periodic. However, (4.12) is in the form in which we will use it numerically, so we shall leave it be.

To be clear, we now have five different versions of each quantity, so far demonstrated only for the density contrast δ . Together with (4.9), their relations are* $P_\delta(a, k) = V^{-1} \langle |\delta(a, \mathbf{k})|^2 \rangle_V = \delta^2(a, k) = T_\delta^2(a, k)\zeta^2(k)$, along with the realisations (4.12) which incorporate $\delta(a, \mathbf{x})$. Here only the Fourier space transfer function $T_\delta(k)$ and the real space 3D field $\delta(\mathbf{x})$ are

*It is uncommon to see this explicit distinction between e.g. $\delta(k)$ and $T_\delta(k)$ in the literature. When working at the interface between linear (and hence 1D) theory and non-linear (3D) theory, I have found it very useful to clearly separate out the different usages of the same physical quantity.

dimensionless. When thinking of a 3D grid of $\delta(a, \mathbf{x})$ values, we write this as $\delta_{\mathbf{n}}(a)$, with $\mathbf{n} \in \mathbb{N}^3$ labelling the grid points. Correspondingly, we write the grid version of $\delta(a, \mathbf{k})$ as $\delta_{\mathbf{h}}(a)$ with $V^{1/3}(2\pi)^{-1}\mathbf{k} = \mathbf{h} \in \mathbb{Z}^3$ labelling the grid points. Sometimes we might instead use the physical but non-integer labelling $\delta_{\mathbf{k}}(a)$. Thus, $\delta(a, \mathbf{x}) = \delta_{\mathbf{n}}(a)$ and $\delta(a, \mathbf{k}) = \delta_{\mathbf{k}}(a) = \delta_{\mathbf{h}}(a)$ for matching $\{\mathbf{x}, \mathbf{n}\}$ and $\{\mathbf{k}, \mathbf{h}\}$.

The instantiation of a specific $\delta(\mathbf{x})$ from the random field $\mathcal{R}(\mathbf{k})$ is called a *realisation*. There exists an infinite number of equally valid random fields \mathcal{R} , each giving rise to its own realisation, all with the same power spectrum, as this is controlled solely by ζT_δ . In the case of finite V , which we will have numerically, we can then think of a given choice for \mathcal{R} as corresponding to translation within an infinite universe, enabling us to sample different patches of this infinite space.

As written in (4.12), both $\zeta(k)$ and $\mathcal{R}(\mathbf{k})$ are defined independent of time, meaning that they are both defined at the time right after inflation. We can then understand $\mathcal{R}(\mathbf{k})$ as representing the 3D distribution of fluctuations left behind by inflation, from which the fluctuations in all other species arose. That is, all species share the same \mathcal{R} . Moreover, the same \mathcal{R} apply for variables other than δ . Substituting δ (both the variable and the subscript) for e.g. δP in (4.12) would then lead to the correct realisation of the pressure. The same would be true for θ and σ , although typically we are not interested in realising $\theta(\mathbf{x})$ and $\sigma(\mathbf{x})$ but rather $u^i(\mathbf{x})$ and $\sigma_j^i(\mathbf{x})$. From (3.37) and (3.38), we have

$$u^i(a, \mathbf{x}) = V^{-1/2} \sum_{\mathbf{k}} \left[-i \frac{k^i}{k^2} T_\theta(a, k) \right] \zeta(k) \mathcal{R}(\mathbf{k}) e^{i\mathbf{k}\mathbf{x}}, \quad (4.13)$$

$$\sigma_j^i(a, \mathbf{x}) = V^{-1/2} \sum_{\mathbf{k}} \left[-\frac{3}{2} \left(\hat{k}^i \hat{k}_j - \frac{1}{3} \delta^i_j \right) T_\sigma(a, k) \right] \zeta(k) \mathcal{R}(\mathbf{k}) e^{i\mathbf{k}\mathbf{x}}, \quad (4.14)$$

which alongside (4.12) shall enable us to realize scalar, vector and rank-2 tensor fields.

It should be clear that the realisations listed in this section can only ever produce linear fields, as they are built from linear transfer functions together with the primordial field $\zeta \mathcal{R}$. In contrast, given some non-linear field $\delta(\mathbf{x})$, the power spectrum $P_\delta(k) = V^{-1} \langle |\delta_{\mathbf{k}}|^2 \rangle_V$ will be fully applicable, resulting in the non-linear power spectrum, which is still a useful measure of the amount of structure, though it will miss out on the non-Gaussianity developed during the non-linear evolution. In chapter 7 we shall further develop the basic linear realisations of this section, while also extending the idea to non-linear realisations.

Part II

Numerics

5 Overview of Standard N-body Techniques

This chapter introduces the overall framework of non-linear cosmological N -body simulations and gives an overview of the main numerical methods utilised within them. The main goal of such N -body simulations is to evolve a system of N particles (or bodies) forward in time under mutual gravity, and so this is the main problem discussed in this chapter. For a more thorough and pedagogical walk-through of the basic techniques covered, see [22]. Other façades of cosmological simulations, such as fluid dynamics and generation of initial condition, will be discussed in chapters 6 and 7, respectively.

Special attention will be given to the techniques which I have implemented into CONCEPT, though this chapter does not serve as an overview of the CONCEPT code. For that, see chapter 8.

5.1 Basic Set-up

This section briefly describe the general set-up of cosmological N -body simulations, introducing terms and notions that we will need later.

Newtonian Framework

Most of the widely used cosmological simulation codes, such as GADGET [35], PKDGRAV [20] and RAMSES [19], as well as CONCEPT [16] operate in a Newtonian setting, disregarding* GR corrections at all scales. An exception is the GEVOLUTION N -body code [36] which evolve the system in the weak-field limit of GR.

*In fact this is not quite true, as during my PhD I have implemented weak-field GR corrections into both CONCEPT and PKDGRAV. This however is done in a seemingly roundabout manner, exactly for the reason as to keep the overall setting Newtonian. These GR corrections are discussed in subsection 7.3.2.

The Hubble Expansion

Though the local gravitational field is treated as Newtonian, the Hubble expansion is however accounted for, which as we have seen in e.g. (3.15) really just amounts to multiplying the different terms of the equations of motion by a raised to various powers. Thus, all cosmological N -body codes are in need of $a(t)$. This is usually obtained by solving the Friedmann equation (2.11) internally by the N -body code itself. This is of course fine, but in practice the Friedmann implementations are all very limited, often only accounting for matter and Λ (GADGET) and perhaps radiation (PKDGRAV). In CONCEPT, the Friedmann equation (including matter, Λ and radiation) may be solved internally, but really one should use its excellent integration with CLASS. Indeed, the CONCEPT default is to run CLASS in order to get $a(t)$, allowing for the inclusion of other species such as massive neutrinos.

The Particles

Usually, N -body simulations only contain within them a single component; matter. The N particles thus sample the phase space distribution function $f_m(t, \mathbf{x}, \mathbf{q})$. Additionally, this ‘matter’ is often taken to behave like cold dark matter*, which then removes the \mathbf{q} dependence of the distribution function, leaving it a function of only time and space. This is the reason as to why particles are a numerically feasible representation of (cold) matter, as we only need to sample 3D space as opposed to 6D phase space. At early times of almost homogeneity, each point in (discretised) space is then populated with a single particle, which is assigned some suitable momentum. Each particle n is fully specified by its comoving position $\mathbf{x}_n(t)$ and its comoving canonical momentum $\mathbf{q}_n(t) = m_n a^2 \dot{\mathbf{x}}_n(t)$, with m_n the particle mass. Usually the particle mass is shared amongst all particles, $m_n = m$.

The Box

Sprinkling N particles throughout an infinite space will of course do us no good. Instead, we focus on a cubic box of comoving volume $V = L^3$, in which the particles live. This box is made periodic so that any particle

*Some codes like GADGET and RAMSES implement both cold dark and baryonic matter, with the complicated gas physics of the baryons included. We shall not deal with this complication in this work. For us, the baryons are then treated exactly as cold dark matter by the N -body simulation, which is fine as long as we do not probe small (sub-galactic) scales where the gas physics is important.

flying through one face instantly reappears at the opposing face, keeping the particles within the box. One effect of this is that it removes any notion of a centre (or edge) of the box. Less trivially, this periodicity apply to gravity as well, meaning that two particles at opposite sides of the box feel a strong mutual gravity, as the largest separation between a pair of particles is now $L/2$ in any direction. The pairwise gravity is not just given by some modular arithmetic though, as the gravitational attraction from each of the infinite ‘mirror images’ of a single particle needs to be taken into account. The periodicity is then what allows us to emulate an infinite self-gravitating universe, though with the constraint of enforced homogeneity above the box size. Fortunately, this is exactly what we seek in cosmology, provided that we pick a large enough box.

5.2 Hamiltonian Dynamics

In this section we describe the numerical time evolution of the particles. This comes down to integrating the Hamiltonian equations (3.13) for $\dot{\mathbf{x}}_n$ and $\dot{\mathbf{q}}_n$ in time, which is complicated by the fact that these equations are coupled: $\dot{\mathbf{x}}_n$ depends explicitly on \mathbf{q}_n , while $\dot{\mathbf{q}}_n$ depends implicitly on $\{\mathbf{x}_{m \neq n}\}$ through the comoving force \mathbf{f}_n . This section does not account for the computation of this force, which we shall simply assume known to us. Remaining agnostic regarding the nature of this force means that the following will apply for a general force, not just that of gravity.

5.2.1 The Leapfrog Integrator

Numerically, the two coupled Hamilton equations has to be solved iteratively, evolving $\{\mathbf{x}_n\}$ slightly and only then $\{\mathbf{q}_n\}$ slightly. Exact simultaneity is out of the question. We can write down operators which when acting on the system only evolves either $\{\mathbf{x}_n\}$ or $\{\mathbf{q}_n\}$ forward in time by the amount Δt ;

$$D(\Delta t)\{\mathbf{x}_n(t), \mathbf{q}_n(t)\} = \left\{ \mathbf{x}_n(t) + \frac{\mathbf{q}_n(t)}{m_n} \int_t^{t+\Delta t} \frac{dt'}{a^2(t')}, \mathbf{q}_n(t) \right\}, \quad (5.1)$$

$$K(\Delta t)\{\mathbf{x}_n(t), \mathbf{q}_n(t)\} = \left\{ \mathbf{x}_n(t), \mathbf{q}_n(t) + a(t) \mathbf{f}_n(t) \int_t^{t+\Delta t} \frac{dt'}{a(t')} \right\}, \quad (5.2)$$

where $D(\Delta t)$ and $K(\Delta t)$ are known as the ‘drift’ and ‘kick’ operators. As evaluation of the scale factor $a(t)$ is not constrained to the discrete time steps, a better approximation to the true drift $\mathbf{x}_n(t + \Delta t) =$

$\mathbf{x}_n(t) + m_n^{-1} \int_t^{t+\Delta t} \mathbf{q}_n(t')/a^2(t') dt'$ is obtained by keeping the scale factor integration as seen in (5.1), rather than approximating this integral as just $\Delta t/a^2(t)$. A similar thing is going on for the kick operation (5.2). Here, the naïve kick would be just $\mathbf{q}_n(t+\Delta t) \approx \mathbf{q}_n(t) + \mathbf{f}_n(t)\Delta t$. However, the comoving force \mathbf{f}_n as defined in (3.13) has an intrinsic scaling of a^{-1} , inherited from the peculiar potential; $\mathbf{f}_n \propto \nabla\varphi \propto a^2\bar{\rho} \propto a^2a^{-3} = a^{-1}$. We thus integrate this scaling over the time step in (5.2) and introduce a factor* $a(t)$ to compensate for the introduced $a^{-1}(t')$.

For an infinitesimal time step Δt , $D(\Delta t)K(\Delta t)\{\mathbf{x}_n(t), \mathbf{q}_n(t)\} = K(\Delta t)D(\Delta t)\{\mathbf{x}_n(t), \mathbf{q}_n(t)\} = \{\mathbf{x}_n(t + \Delta t), \mathbf{q}_n(t + \Delta t)\}$. For any finite Δt though, the interdependence of the two canonical quantities makes the drift and kick operator noncommutative. Looking at (5.1) and (5.2), we see that the right-hand sides only refer to the canonical variables at the time t , though really we should be integrating these over the time step, just as we do for a . We of course cannot do this numerically, as this removes the time discretisation. Keeping to a discretised formalism, exact versions of (5.1) and (5.2) can be obtained by replacing $\mathbf{q}_n(t) \int_t^{\Delta t} dt'/a^2(t')$ in (5.1) with the average value of \mathbf{q}_n/a^2 over the time step (and multiplying by Δt), and similarly replacing $\mathbf{f}_n(t)$ in (5.2) with its average value over the time step. At linear order in Δt , this averaging corresponds to just evaluating \mathbf{q}_n and \mathbf{f}_n at $t + \Delta t/2$. We can incorporate this shift of $\Delta t/2$ simply by applying $D(\Delta t)$ and $K(\Delta t)$ out-of-phase;

$$U(\Delta t) = K\left(\frac{\Delta t}{2}\right)D(\Delta t)K\left(\frac{\Delta t}{2}\right), \quad (5.3)$$

where this U is known as the ‘leapfrog’ KDK time evolution operator, because it leads to $\{\mathbf{x}_n\}$ and $\{\mathbf{q}_n\}$ being constantly out-of-phase but interchanging which of the two canonical variables are ‘ahead’ after each kick and drift operation, when applied multiple times in succession.

Instead of the KDK leapfrog integrator (5.3), we could just as well have settled for the DKD version. In the limit of many successive applications of $U(\Delta t)$, these are equivalent. We can see this from the observation that $K^2(\Delta t/2) = K(\Delta t)$ and similarly $D^2(\Delta t/2) = D(\Delta t)$. Using (5.3) to integrate the system forward by N time steps then looks like

$$U^N(\Delta t) = K\left(\frac{\Delta t}{2}\right) \left[D(\Delta t)K(\Delta t) \right]^{N-1} D(\Delta t)K\left(\frac{\Delta t}{2}\right), \quad (5.4)$$

*One might then argue that the combination $a\mathbf{f}_n$ is really what should be defined as the comoving force. We shall however stick to our original definition (3.13).

i.e. the entire time evolution just consists of successive kicks and drift over time steps Δt , but with half a kick at the beginning and end of the time evolution.

Symplecticity

The naïve time evolution without the intentional out-of-phasing by $\Delta t/2$, $[K(\Delta t)D(\Delta t)]^N$, leads to double the phase shift Δt , while also always keeping $\{\mathbf{x}_n\}$ ahead of $\{\mathbf{q}_n\}$ (the other way around for $[D(\Delta t)K(\Delta t)]^N$). It can be shown [22, 35] that these naïve time evolutions are first order accurate in Δt , while the symmetric leapfrogging (5.4) is second order accurate.

Perhaps most important is the fact that (5.3) has time reversal symmetry, i.e. $U^{-1}(\Delta t) = U(-\Delta t)$, which is clear since the same property holds for K and D individually (note that the naïve integrators $K(\Delta t)D(\Delta t)$ and $D(\Delta t)K(\Delta t)$ do not have this property due to $D(\Delta t)$ and $K(\Delta t)$ not commuting with each other). This means that the numerical time evolution resulting from $U(\Delta t)$ actually do correspond to some Hamiltonian, which we know can only deviate from the true Hamiltonian (3.12) to second order in Δt . The Hamiltonian (or *symplectic*) structure of the system is then preserved when switching from true time evolution to numerical leapfrogging, from which we are ensured that the overall system will conserve energy. More generally, the symplecticity of the leapfrog integrator guarantees that we do not break Liouville's theorem (volume conservation of any closed surface in phase space), and so the numerical trajectory of each particle through phase space cannot diverge rapidly from its true trajectory.

5.2.2 Adaptive Time Stepping

In the end, the leapfrog integrator (5.3) is simply the only time evolution operator constructable from D and K which treats $\{\mathbf{x}_n\}$ and $\{\mathbf{q}_n\}$ symmetrically. Any improved version we might imagine can only correspond to lowering the time step size Δt .

A new range of possibilities arise if we split D and K into $\{D_n\}$ and $\{K_n\}$, i.e. allowing for drifting and kicking of individual particles rather than always operating on the entire system. For systems of large dynamic range such as the late time Universe with its non-linear structure formation, such time stepping at a per-particle basis allows for fine time steps of fast moving particles in strong gravity, while at the same time

making it possible to only update slower particles within voids more rarely, saving clock cycles without lowering the overall accuracy.

As the gravitational force is often computed collectively, for such adaptive time stepping to work we need the system to synchronise itself at frequent intervals. This is achieved by having a hierarchical time stepping scheme, where single-particle time step sizes are allowed only to take the form $\Delta t/2^{N_t}$, $N_t \in \mathbb{N}$, where Δt is now the maximally allowed time step for any particle. This power of two hierarchy, as opposed to e.g. just $\Delta t/N_t$, makes it possible for particles at given ‘rung’ N_t to jump to rung $N_t - 1$ after at most one time step, as inter-rung synchronisation will happen every other time step for neighbouring rungs.

In principle, the rung of a given particle n should depend on both $|\mathbf{q}_n|$ and $|\mathbf{f}_n|$, since a particle with e.g. large $|\mathbf{q}_n|$ but zero $|\mathbf{f}_n|$ can be integrated exactly using arbitrarily large Δt . In practice however, most particles with large momentum will be found in the vicinity of a large overdensity, and so a particle with large $|\mathbf{f}_n|$ can be assumed to also have large $|\mathbf{q}_n|$. The rung of a particle is then typically set by the momentum alone.

All the main cosmological simulation codes employ adaptive time stepping in some form or another, which fit into the given gravitational scheme used by the code in question. Here the CONCEPT code lacks behind, having only a globally defined time step size Δt . Due to the lack of a proper P³M (or similar) implementation which can handle the short-range component of gravity, the dynamic range in CONCEPT is however considerably lower than it is in these other codes. For CONCEPT to lack adaptive time stepping is thus not a major issue in and of itself.

5.3 Newtonian Gravitation

This section demonstrates the basics of the core numerical methods used to solve gravity, i.e. obtain $\{\mathbf{f}_n(t)\}$ from $\{\mathbf{x}_n(t)\}$. Solving gravity is the main problem of any cosmological N -body code, and so several sophisticated methods has been developed. Broadly, we can classify these methods into two categories, which I shall refer to as *direct summation* methods and *grid based* methods, the key difference being that while direct summation basically computes pair-wise forces between the particles, grid based methods first solve for the global potential, the local gradient of which is then used to compute the forces.

We shall study the most basic method from each of the two categories in some detail, after which we will discuss more advanced extensions to

these basic primitives. The basic methods are known as the particle-particle method (direct summation) and the particle-mesh method (grid based). From a mathematical point of view, the problem that both of these methods has to solve is how to deal with the infinity that arises from the periodicity. Ultimately, both methods solve the problem using Fourier techniques.

5.3.1 The Particle-Particle Method

In the particle-particle (or PP) method we simply compute the direct force between all pairs of particles. With N particles, this is then an $\mathcal{O}(N^2)$ operation. The tricky part is that each particle at \mathbf{x}_n has an infinite number of image particles due to the periodicity, located at $\mathbf{x}_n + \mathbf{n}L$, $\mathbf{n} \in \mathbb{Z}^3$, with $\mathbf{n} = \mathbf{0}$ corresponding to the particle itself. Note that the integer particle label n and the integer triple \mathbf{n} are otherwise unrelated.

Equation (3.14) writes the density $\rho(\mathbf{x})$ in terms of particles. For N particles within a periodic cubic box of comoving side length L , this becomes

$$\delta\rho(\mathbf{x}) = \frac{1}{a^3} \sum_{n=1}^N m_n \sum_{\mathbf{n} \in \mathbb{Z}^3} [\delta(\mathbf{x}_n + \mathbf{n}L - \mathbf{x}) - L^{-3}], \quad (5.5)$$

where \mathbf{x} may be taken as any position in \mathbb{R}^3 . To get $\delta\rho = \rho - \bar{\rho}$, the mean density $\sum_{n=1}^N m_n / L^3$ is subtracted once “per box” (once per value of \mathbf{n}). We can now express the peculiar potential (3.7) by plugging in (5.5);

$$\varphi(\mathbf{x}) = -\frac{G}{a} \sum_{n=1}^N m_n \sum_{\mathbf{n} \in \mathbb{Z}^3} \frac{1}{|\mathbf{x}_n + \mathbf{n}L - \mathbf{x}|}, \quad (5.6)$$

where the delta functions are used to evaluate the integral. I have removed the divergent term resulting from the constant L^{-3} . This amounts to switching back to using ρ rather than $\delta\rho$. Before we are done with the PP method, we will have to remove another, very similar divergence of the opposite sign.

Ewald Summation

Our goal is now to find a way to compute the infinite but periodic sum (5.6). For this, the Ewald [37, 38] summation method is deployed. The sum over images in (5.6) is split into two parts by multiplying by

$1 = \text{erf}[(\mathbf{x}_n + \mathbf{n}L - \mathbf{x})/(2x_s)] + \text{erfc}[(\mathbf{x}_n + \mathbf{n}L - \mathbf{x})/(2x_s)]$, where $x_s > 0$ is some chosen length scale. Because the error function approaches unity for large input (and so the complementary error function necessarily vanish), this corresponds to splitting each term of (5.6) into a short-range and a long-range part. If we simultaneously cast the long-range part into Fourier space using $\int d^3\mathbf{x} |\mathbf{x}|^{-1} \text{erf}(|\mathbf{x}|/(2x_s))e^{-ikx} = 4\pi \exp(-k^2 x_s^2)/k^2$, we get

$$\varphi(\mathbf{x}) = -\frac{G}{a} \sum_{n=1}^N m_n \left\{ \sum_{\mathbf{n} \in \mathbb{Z}^3} \frac{\text{erfc}[(\mathbf{x}_n + \mathbf{n}L - \mathbf{x})/(2x_s)]}{|\mathbf{x}_n + \mathbf{n}L - \mathbf{x}|} + \frac{4\pi}{L^3} \sum_{\mathbf{k} \in \{2\pi L^{-1} \mathbf{h} \mid \mathbf{h} \in \mathbb{Z}^3 \setminus \mathbf{0}\}} \frac{\exp(-k^2 x_s^2)}{k^2} \cos[\mathbf{k}(\mathbf{x}_n - \mathbf{x})] \right\}, \quad (5.7)$$

where the antisymmetric sine part of the complex exponential in the Fourier series vanish under the symmetric \mathbf{k} summation. Note that we leave out the divergent $\mathbf{k} = \mathbf{0}$ term. This term corresponds exactly to the already removed divergent part, but has the opposite sign, somewhat justifying these manual removals.

We can now take the gradient of (5.7) and obtain the comoving force $\mathbf{f}_n = -m_n \nabla \varphi|_{\mathbf{x}=\mathbf{x}_n}$,

$$\begin{aligned} \mathbf{f}_n = \frac{Gm_n}{a} \sum_{\substack{n'=1 \\ n' \neq n}}^N m_{n'} \left\{ \sum_{\mathbf{n} \in \mathbb{Z}^3} \frac{\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n}{|\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n|^3} \left[\text{erfc}\left(\frac{|\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n|}{2x_s}\right) \right. \right. \\ \left. \left. + \frac{|\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n|}{\sqrt{\pi} x_s} \exp\left(-\frac{|\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n|^2}{4x_s^2}\right) \right] \right. \\ \left. + \frac{4\pi}{L^3} \sum_{\mathbf{k} \in \{2\pi L^{-1} \mathbf{h} \mid \mathbf{h} \in \mathbb{Z}^3 \setminus \mathbf{0}\}} \frac{\mathbf{k}}{k^2} \exp(-k^2 x_s^2) \sin[\mathbf{k}(\mathbf{x}_{n'} - \mathbf{x})] \right\}. \end{aligned} \quad (5.8)$$

As both the complementary error function and the Gaussian in the short-range part of (5.8) fall off rapidly with increasing $|\mathbf{n}|$, we can truncate the short-range \mathbf{n} sum, keeping only terms with $|\mathbf{n}|$ below some suitable threshold. The same applies for the long-range \mathbf{k} (or \mathbf{h}) sum, where a Gaussian again ensures rapid convergence. Both GADGET and CONCEPT uses

$$\begin{cases} x_s = \frac{L}{4}, \\ |\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n| < 3.6L, \\ \mathbf{h}^2 < 10, \end{cases} \quad (5.9)$$

as suggested in [38].

With the numerical parameters (5.9), the force computation between each pair of particles involve between 170 and 220 short-range terms as well as 122 long-range terms, given by (5.8). Though finite, this is still a large calculation. As the Ewald force between a pair of particles is a smooth function of their separation, one can precompute a table (3D grid) of Ewald forces, which once the \mathbf{n} and \mathbf{k} sums are carried out depend only on $\mathbf{x}_{n'} - \mathbf{x}_n$. Expressing the force in units of $Gm_n m_{n'} a^{-1} L^{-1}$, the same Ewald table can even be used for all simulations regardless of the box size and particle masses, as well as throughout time within the simulations. Furthermore, only one octant of the $\mathbf{x}_{n'} - \mathbf{x}_n$ space needs to be tabulated due to symmetry. We can see this by placing one particle at the centre of the box. A particle at any of the 8 corners will now receive the same force from the central particle, but with different signs of the three force components, of which there are exactly $2^3 = 8$ combinations. All of GADGET, PKDGRAV and CONCEPT utilise such an Ewald table.

Finally, with the comoving force \mathbf{f}_n at hand, it is used to update the comoving momentum \mathbf{q}_n of the n 'th particle as described by the kick operator (5.2). We now have everything we need to integrate the system forward in time (5.4). In practice though, the asymmetry arising from our discretisation of time into finite time steps of size Δt while leaving space continuous leads to extremely bad numerical behaviour, which we shall now address.

Softening

Using the Ewald summation technique (5.8) we can compute the periodic force between two point particles. For an infinitesimal time step size Δt , this would be a complete description of the PP method. In actuality with a very much finite Δt , this force has to be modified slightly. A finite Δt allow the point particles to suddenly become arbitrarily close to each other, resulting in unrealistic large forces which then generates energy. To ensure that the pairwise force is “soft”, the mass of each particle needs to be distributed continuously over a finite volume, as opposed to being modelled by Dirac delta functions.

A typical and easy to implement solution is to model the particles as ‘Plummer spheres’ [39], with a single-particle density

$$\rho_P(\mathbf{x}) \equiv m \left(\frac{4\pi}{3} \epsilon^3 \right)^{-1} \left(1 + \frac{(\mathbf{x}_P - \mathbf{x})^2}{\epsilon^2} \right)^{-5/2}, \quad (5.10)$$

where the particle is placed at \mathbf{x}_P and the softening length ϵ determines its size. Note that the density is defined in terms of the comoving \mathbf{x} , meaning that the particles “grow along with the box”. We should then not read too much meaning into the actual shape given to the particles, it is merely a numerical necessity.

The seemingly odd exponent of $-5/2$ in (5.10) is chosen because it leads to the nice looking potential

$$\varphi_P(\mathbf{x}) = -\frac{Gm}{a} \frac{1}{\sqrt{(\mathbf{x}_P - \mathbf{x})^2 + \epsilon^2}}, \quad (5.11)$$

with the only change from the corresponding point particle potential being the appearance of ϵ^2 in the denominator. The force on a point particle located at \mathbf{x}_{δ} from a Plummer sphere is now

$$\mathbf{f}_{\delta,P} = \frac{Gm_{\delta}m_P}{a} \frac{\mathbf{x}_P - \mathbf{x}_{\delta}}{[(\mathbf{x}_P - \mathbf{x}_{\delta})^2 + \epsilon^2]^{3/2}}, \quad (5.12)$$

where I have now labelled the two masses. Importantly, the force (5.12) has a finite maximum, which happens to be at $|\mathbf{x}_P - \mathbf{x}_{\delta}| = \epsilon/\sqrt{2}$. Again, the only change we need in order to go from the force between two point particles and this softened Plummer force is the addition of ϵ^2 in the denominator. The Plummer softening used in N -body codes refer to the form (5.12), i.e. the particles are not actually modelled as Plummer spheres (5.10), as the force between two Plummer spheres is somewhat different from (5.12).

The Plummer softening should be incorporated into the Ewald summation (5.8). As the softening only matters at small particle separations, we only need it for the $\mathbf{n} = \mathbf{0}$ term in the short-range sum. The way this is usually handled is to introduce what me might call the *mirror force*; the part of the periodic force that results from all mirror particles, but not from the actual particle within the box itself. From (5.8), we can

introduce the mirror force on particle n from particle n' like

$$\begin{aligned} \mathbf{f}_{n,n'}^{\text{mirror}} = & \frac{Gm_n m_{n'}}{a} \left\{ \sum_{n \in \mathbb{Z}^3} \frac{\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n}{|\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n|^3} \left[\operatorname{erfc}\left(\frac{|\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n|}{2x_s}\right) \right. \right. \\ & + \frac{|\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n|}{\sqrt{\pi} x_s} \exp\left(-\frac{|\mathbf{x}_{n'} + \mathbf{n}L - \mathbf{x}_n|^2}{4x_s^2}\right) \left. \right] \\ & + \frac{4\pi}{L^3} \sum_{\mathbf{k} \in \{2\pi L^{-1} h \mid h \in \mathbb{Z}^3 \setminus \{0\}\}} \frac{\mathbf{k}}{\mathbf{k}^2} \exp(-\mathbf{k}^2 x_s^2) \sin[\mathbf{k}(\mathbf{x}_{n'} - \mathbf{x})] \\ & \left. \left. - \frac{\mathbf{x}_{n'} - \mathbf{x}_n}{|\mathbf{x}_{n'} - \mathbf{x}_n|^3} \right\}, \right. \end{aligned} \quad (5.13)$$

where the direct (non-mirror) force is subtracted on the last line. This direct force can now be readded in its softened version. The complete, softened force on particle n from all particles is then

$$\mathbf{f}_n = \sum_{\substack{n'=1 \\ n' \neq n}}^N \left[\mathbf{f}_{n,n'}^{\text{mirror}} + \frac{Gm_n m_{n'}}{a} \frac{\mathbf{x}_{n'} - \mathbf{x}_n}{[(\mathbf{x}_{n'} - \mathbf{x}_n)^2 + \epsilon^2]^{3/2}} \right]. \quad (5.14)$$

The precomputed Ewald table is then really over $\mathbf{f}_{n,n'}^{\text{mirror}}$, allowing ϵ to be a free parameter, independent of this table. Given that this table only covers the cases $|\mathbf{x}_{n'} - \mathbf{x}_n| < L/2$, it is crucial that $\mathbf{x}_{n'}$ in (5.14) is chosen accordingly for each dimension, i.e. if particle n is closer to one of the mirror versions of particle n' than it is to n' itself, we should swap $\mathbf{x}_{n'}$ for the position of this closests mirror particle when doing the Ewald lookup.

Good values for ϵ lie in the range [40, 41]

$$\epsilon \approx (2-4)\% \frac{L}{N^{1/3}}, \quad (5.15)$$

i.e. a small fraction of the mean, comoving inter-particle distance.

The simple Plummer softening is implemented in CONCEPT. More involved softening is utilised by e.g. GADGET [35], but it really just comes down to a more complicated single-particle density distribution (5.10).

5.3.2 The Particle-Mesh Method

The previous subsection made it clear that we face two distinct problems when implementing gravity numerically. First, the periodicity of the box

transforms each pairwise interaction into an infinite sum of interactions. Second, the force has to be softened, i.e. we have to introduce some smallest scale, below which gravity weakens rather than increases. Both of these problems are tackled simultaneously in the PM method by introducing a comoving grid (or mesh) of some fixed comoving grid spacing, on which we solve gravity using Fourier techniques. Besides the grid aspect, the PM method is then analogous to the $x_s \rightarrow 0$ limit of the Ewald technique (5.8), with the entire computation delegated to the long-range Fourier sum.

The softening length ϵ of the PP method is then replaced by the grid spacing Δx_φ , below which gravity is not resolved. The introduced grid is a cubic $N_\varphi \times N_\varphi \times N_\varphi$ grid and so the spacing (the cell width) in each dimension is $\Delta x_\varphi = \Delta y_\varphi = \Delta z_\varphi = L/N_\varphi$. As we can choose N_φ as we please, the PM method can in principle be used to resolve gravity to arbitrarily small scales. If we want the PM method to resolve gravity as good as the softened PP method, we must require $\Delta x_\varphi \approx \epsilon$. From (5.15), this means that $N_\varphi^3 \approx N/[(2\text{--}4)\%]^3$, i.e. the needed number of grid points is at least tens of thousands times as large as the number of particles, which is not numerically feasible. In practice then, $\Delta x_\varphi \gg \epsilon$, making the gravitational resolution of the PM method much less than that of the PP method.

What makes the PM method intriguing is its speed. As it solves the collective potential φ from all particles on the grid, the particles are never paired up, removing the $\mathcal{O}(N^2)$ scaling. As the PM method works by solving the Poisson equation on the grid in Fourier space, the most expensive part of the method is the Fourier transformations needed. Deploying the fast Fourier transform (FFT) [42], the PM method then scales as $\mathcal{O}(N_\varphi^3 \log N_\varphi^3) \approx \mathcal{O}(N \log N)$, where I have reasonably assumed the choice of $N_\varphi^3 \propto N$. Disregarding all constant factors, the largest current simulations [20] having $N \sim 10^{12}$ then run billions of times faster using the PM method than they would using the PP method.

In (3.7), the peculiar potential $\varphi(\mathbf{x})$ is given as a convolution between $\delta\rho(\mathbf{x})$ and $|\mathbf{x}|^{-1}$. In Fourier space, this convolution turns into multiplication. With the Fourier transform of $|\mathbf{x}|^{-1}$ being $4\pi/\mathbf{k}^2$, the Fourier space Poisson equation becomes the simple algebraic relation

$$\varphi(\mathbf{k}) = -\frac{4\pi G a^2}{\mathbf{k}^2} \delta\rho(\mathbf{k}). \quad (5.16)$$

The strategy of the PM method is to first construct $\delta\rho(\mathbf{x})$ on the $N_\varphi \times N_\varphi \times N_\varphi$ grid from the particle distribution, Fourier transform the grid in-place to get $\delta\rho(\mathbf{k})$, solve the Poisson equation (5.16) by multiplying

each grid point \mathbf{k} by $-4\pi Ga^2/\mathbf{k}^2$, turning the grid values into $\varphi(\mathbf{k})$, and then transform back to real space, obtaining $\varphi(\mathbf{x})$. Though this single grid in memory is used to hold $\delta\rho(\mathbf{x})$, $\delta\rho(\mathbf{k})$, $\varphi(\mathbf{k})$ and $\varphi(\mathbf{x})$ in turn, we shall consistently refer to it as the φ grid. Once $\varphi(\mathbf{x})$ is constructed on the grid, we can differentiate it to obtain the force/mass, which can then be interpolated back to the particle locations and applied.

In Fourier transforming the particle distribution within the finite box, periodicity is achieved automatically, in stark contrast to the PP method where the Ewald machinery were needed to inject the periodicity. For Fourier based PM methods then, periodicity is the natural boundary condition. If we instead wanted vacuum boundary conditions (if e.g. the box contained just a single, isolated galaxy), the Ewald part would be removed from the PP method making it much more straight forward, whereas we would have to complicate the PM method by removing the periodicity.

Mesh Interpolation

We shall now describe how to go about discretising (5.16) so that the fields $\delta\rho$ and φ live on the fixed grid, rather than as continuous functions in (real or Fourier) space. We first need to decide how the particles should come together to form $\delta\rho(\mathbf{x})$. The simplest choice is to model each particle as a delta function, giving rise to (3.14). We then say that each particle has the *shape* $S_0(\mathbf{x}) = \delta(\mathbf{x})$. Generally, the shape is a function localised around the origin describing the unit-mass single-particle density field. We can construct higher-order shapes S_O , $O \in \mathbb{N}$, by smoothing out S_0 , which we may do through repeated convolution with some simple function. A somewhat subtle criterion of these shapes is that in the end, a particle should not receive any force due to its own presence. With a cubic grid, this pins down [43] this simple function to be that of a cubic top-hat function Π :

$$S_O(\mathbf{x}) = \Delta x_\varphi^{-3O} \underset{o=1}{\overset{O}{\bigast}} \Pi\left(\frac{\mathbf{x}}{\Delta x_\varphi}\right), \quad (5.17)$$

where the big operator \bigast means repeated convolution of the function to the right, with the empty convolution ($O = 0$) defined as the δ function. To be clear, the cubic top-hat function is

$$\Pi(x, y, z) \equiv \begin{cases} 1 & |x|, |y|, |z| < \frac{1}{2}, \\ 0 & \text{otherwise.} \end{cases} \quad (5.18)$$

The shape S_1 is then $\Delta x_\varphi^{-3} \Pi(\mathbf{x}/\Delta x_\varphi)$, the shape S_2 is $\Delta x_\varphi^{-6} \Pi(\mathbf{x}/\Delta x_\varphi) * \Pi(\mathbf{x}/\Delta x_\varphi)$, and so on.

As any $S_O(\mathbf{x})$ integrate to one, so does $\Delta x_\varphi^{-3} \Pi(\mathbf{x}/\Delta x_\varphi) * S_O(\mathbf{x})$ (which is really just $S_{O+1}(\mathbf{x})$). We then define the dimensionless weight function

$$\begin{aligned} W_O(\mathbf{x}) &\equiv \Pi(\mathbf{x}/\Delta x_\varphi) * S_O(\mathbf{x}) \\ &= \Delta x_\varphi^{-3O} \underset{o=1}{\overset{O+1}{\mathbin{\text{\large $*$}}}} \Pi\left(\frac{\mathbf{x}}{\Delta x_\varphi}\right), \end{aligned} \quad (5.19)$$

which integrate the shape function over a cell volume centred at the origin. We can then use this weight function to interpolate a fraction of the mass of a particle at \mathbf{x}_n onto a grid point at $\mathbf{x}_n = \Delta x_\varphi \mathbf{n}$, where $\mathbf{n} \in \mathbb{N}^3$ labels the grid points and is not related to the particle index n . This fraction will be $W_O(\mathbf{x}_n - \mathbf{x}_n)$, corresponding to the overlap of the particle shape centred at the particle and a cell volume centred at the grid point.

The lowest order weight $W_{\text{NGP}} \equiv W_0 = \Pi(\mathbf{x}/\Delta x_\varphi)$ is known as the ‘nearest grid point’ weight, as it corresponds to simply snapping the entire particle onto the nearest grid point. The first order weight $W_{\text{CIC}} \equiv W_1 = \Pi(\mathbf{x}/\Delta x_\varphi) * \Pi(\mathbf{x}/\Delta x_\varphi)$ is known as the ‘cloud in cell’ weight, and can be written out as

$$W_{\text{CIC}}(x, y, z) = \begin{cases} \left(1 - \frac{|x|}{\Delta x_\varphi}\right) \left(1 - \frac{|y|}{\Delta x_\varphi}\right) \left(1 - \frac{|z|}{\Delta x_\varphi}\right) & |x|, |y|, |z| < \Delta x_\varphi, \\ 0 & \text{otherwise.} \end{cases} \quad (5.20)$$

The CIC weights are what is used for the PM method in GADGET and CONCEPT. In PKDGRAV*, the NPG, the CIC, the second-order TSC (‘triangular shaped cloud’) and the third-order PCS (‘piecewise cubic spline’) weights are all implemented. As the order increases, a particle is smeared over a larger volume and thus correspondingly more grid points, which further softens the force.

The weight assigned to the grid point \mathbf{n} from a particle at \mathbf{x}_n was $W(\mathbf{x}_n - \mathbf{x}_n) \equiv W_n(\mathbf{x}_n)$, where the order O does not matter and so we will not keep carrying it around. With the “true” density field $\rho(\mathbf{x})$ given

*Note that PKDGRAV does not contain a complete PM method. The grid interpolations of particles are implemented for the power spectrum computation, as here we need $\delta(\mathbf{k})$ on a grid, $\delta_{\mathbf{k}}$, as described in section 4.1.

in (3.14) as a sum of delta functions, the density assigned to each grid point, ρ_n , is

$$\rho_n = \frac{1}{a^3} \sum_{n=1}^N W_n(\mathbf{x}_n) \frac{m_n}{\Delta x_\varphi^3} \quad (5.21)$$

$$= \frac{[W_n(\mathbf{x}) * \rho(\mathbf{x})]_{\mathbf{x}=\mathbf{x}_n}}{\Delta x_\varphi^3}, \quad (5.22)$$

where the first equality is used in the actual implementation and the second equality is a convenient observation which we will use later. The specification of $\mathbf{x} = \mathbf{x}_n$ (note the boldness of n) in (5.22) is important as otherwise the right-hand-side would be a function of \mathbf{x} , which only equals the number ρ_n at the point \mathbf{x}_n .

Note that even for a smooth (i.e. not a sum of delta functions) $\rho(\mathbf{x})$, $\rho_n \neq \rho(\mathbf{x}_n)$ in general. There is thus a small (for slowly spatially varying $\rho(\mathbf{x})$) but important difference between the actual field $\rho(\mathbf{x})$ and its discretised grid version ρ_n .

Solving the Poisson Equation

With the φ grid populated with ρ_n values via interpolating of particle masses as given by (5.21), an in-place FFT results in ρ_h , $\mathbb{Z}^3 \ni \mathbf{h} = \mathbf{k}L/(2\pi)$, i.e. the grid version of $\rho(\mathbf{k})$. Fourier transforming (5.22), we obtain the exact relation

$$\rho_h = W_h \rho(\mathbf{k}) / \Delta x_\varphi^3, \quad (5.23)$$

where W_h is the Fourier transformed $W_n(\mathbf{x})$ evaluated at $\mathbf{k} = (2\pi/L)\mathbf{h}$. Importantly, for any given $\mathbf{h} \in \mathbb{Z}^3$, ρ_h is just a (complex) number and not a function, and so the right-hand-side of (5.23) is also just a number. That is, \mathbf{k} is not allowed to vary in this expression, due to it being specified through \mathbf{h} (this is exactly the same reason as for the explicit evaluation $\mathbf{x} = \mathbf{x}_n$ in (5.22)). We thus cannot take an equation like (5.23) and solve for the function $\rho(\mathbf{k})$.

Inserting (5.23) into (5.16), we get

$$\varphi_h = -\frac{4\pi G a^2}{\mathbf{k}^2} \frac{\Delta x_\varphi^3}{W_h} \rho_h, \quad \varphi_{\mathbf{h}=\mathbf{0}} = 0, \quad (5.24)$$

where I have exchanged $\varphi(\mathbf{k})$ for φ_h to reflect the fact that once again, the expression as a whole is only meaningful at the grid points. Also,

the removal of the background density (as (5.16) specify $\delta\rho$, not ρ) is handled simply by setting the “DC” ($\mathbf{k} = \mathbf{0}$) mode to zero.

We have yet to calculate $W_h = W(\mathbf{k} = (2\pi/L)\mathbf{h})$, $W(\mathbf{k})$ being the Fourier transform of $W(\mathbf{x})$. With the repeated convolution of (5.19) turning into repeated multiplication under the Fourier transform, we just need to calculate the Fourier transform of $\Pi(\mathbf{x}/\Delta x_\varphi)$, which come down to a product of cardinal sine functions;

$$W_{O,h} = \Delta x_\varphi^3 \operatorname{sinc}^{O+1}\left(\frac{\Delta x_\varphi k_x}{2}\right) \operatorname{sinc}^{O+1}\left(\frac{\Delta x_\varphi k_y}{2}\right) \operatorname{sinc}^{O+1}\left(\frac{\Delta x_\varphi k_z}{2}\right), \quad (5.25)$$

with $\operatorname{sinc}(x) \equiv \sin(x)/x$ and $(2\pi/L)\mathbf{h} = \mathbf{k} = (k_x, k_y, k_z)$.

Comparing the continuous Poisson equation (5.16) to its grid form (5.24), we see that they are not related simply by the exchange of fields for grids $\{\rho(\mathbf{k}), \varphi(\mathbf{k})\} \leftrightarrow \{\rho_h, \varphi_h\}$, but that the additional factor $\Delta x_\varphi^3/W_h$ appears in the grid equation. This factor came from the relation (5.23) between the field $\rho(\mathbf{k})$ and the grid ρ_h , or equivalently the field $\rho(\mathbf{x})$ and the grid ρ_n , where the latter was constructed from the first via convolution. The factor $\Delta x_\varphi^3/W_h$ then corrects for this convolution, and is known as the deconvolution factor.

We could now perform an in-place inverse FFT on the φ grid, converting the φ_h values to φ_n values. We could then convert the potential into forces, still locating at the grid positions, and finally interpolate these forces back to the particle positions. This additional interpolation* would then have to be corrected for by another deconvolution, which we know corresponds to multiplication by the factor $\Delta x_\varphi^3/W_h$ in Fourier space. Before leaving Fourier space, we thus carry out this second deconvolution, leaving the φ grid with the values

$$\varphi'_h = -\frac{4\pi G a^2}{\mathbf{k}^2} \frac{\Delta x_\varphi^6}{W_h^2} \rho_h, \quad \varphi'_{h=0} = 0, \quad (5.26)$$

where the prime is there to remind us that these grid values carry around one additional deconvolution. Now we do the in-place inverse FFT, leaving the φ grid with the values φ'_n .

*It may not be obvious that interpolation from the particles onto the grid in this respect is equivalent to interpolation from the grid onto the particles. This is however clear from the convolution between the grid and the particles in (5.22), convolution being commutative.

Obtaining the Forces

We now differentiate the φ'_n values stored in the φ grid using symmetric finite differencing, in order to obtain a grid of force values. The full force field require three new grids, one for each dimension, each the size of the φ grid. Carrying out the computation one dimension at a time, we can however get by using just a single additional grid. A general symmetric difference operator D_{2O}^x of order $2O$, $O \in \mathbb{N}$, acting along the x dimension, may be written [44]

$$D_{2O}^x \varphi'_{n_x, n_y, n_z} = \Delta x_\varphi^{-1} \sum_{\substack{j=-O \\ j \neq 0}}^O \left[\frac{d}{d\xi} \prod_{\substack{i=-O \\ i \neq j}}^O \frac{\xi - i}{j - i} \right]_{\xi=0} \varphi'_{n_x+j, n_y, n_z}, \quad (5.27)$$

where O does not have to match the order of the weight function. Similar difference operators can of course be constructed for the y and z dimensions. The full symmetric gradient operator is then $\mathbf{D}_{2O} = (D_{2O}^x, D_{2O}^y, D_{2O}^z)$. Both GADGET and CONCEPT uses the fourth-order operator, which boils down to

$$D_4^x \varphi'_{n_x, n_y, n_z} = \frac{\varphi'_{n_x-2, n_y, n_z} - 8\varphi'_{n_x-1, n_y, n_z} + 8\varphi'_{n_x+1, n_y, n_z} - \varphi'_{n_x+2, n_y, n_z}}{12\Delta x_\varphi}. \quad (5.28)$$

Note that we could have carried out the differentiation in Fourier space by simply multiplying by $i\mathbf{k}$, which again could be done one dimension at a time, also requiring just a single grid in addition to the φ grid. This would however require three separate inverse FFTs instead of one, \sim doubling the computation time. At least for smooth fields, low-order (e.g. 4) real space finite differencing leads to very similar results, and so it is generally preferable to do the differentiation in real space.

Interpolating the differentiated φ'_n back to the particle positions is then done in a similar manner to (5.22), though now the particle index n is fixed while the sum is over grid points \mathbf{n} :

$$\mathbf{f}_n = -m_n \sum_{\mathbf{n}} W_{\mathbf{n}}(\mathbf{x}_n) \mathbf{D} \varphi'_{\mathbf{n}}, \quad (5.29)$$

where I have left out the order specification on \mathbf{D} . Importantly, as the sum (5.29) corresponds to convolution (c.f. (5.22)), the additional deconvolution present inside of φ'_n is now paired up, and so we lose the prime.

All of GADGET, PKDGRAV and CONCEPT rely on FFTW* [45] to perform the FFTs, delegating the performance critical part of the code to this external library. Thus, as far as the PM method is concerned, all three codes are of comparable performance. While CONCEPT does not contain a (well-implemented version of) a more advanced gravitational method with the same scaling but with higher resolution, GADGET and PKDGRAV do, giving them a large advantage.

5.3.3 Advanced Methods

We are now familiar with the basic $\mathcal{O}(N^2)$ PP and $\mathcal{O}(N \log N)$ PM method. The bad scaling of PP makes it practically unusable for actual use[†]. The PM method on the other hand suffers from the lack of high spatial resolution. Both of these methods can be extended, improving their usability. They can also be combined into hybrid methods, harvesting the strength of both the PP and PM method, without inheriting too much of their bad aspects.

Tree Methods

The $\mathcal{O}(N^2)$ scaling of the PP method came about because we insisted on pairing up each particle with every single other particle. From the point of view of some particular particle, a far away dense cluster of particles act as a single, more massive particle. Thus, by somehow grouping particles together prior to the direct summation, we can reduce the number of particle-particle (or now particle-group) interactions. This is the basic idea behind ‘tree methods’, which extend the basic PP method with a tree, i.e. a recursive grouping of the particles into some hierarchical structure. The gravitational force on a single particle from an entire branch of the tree (i.e. all particles at a given level of the hierarchical structure) is then either computed as the sum of forces from all daughter-branches (‘twigs’) of the given branch, or as the single, collective force due to the entire branch if the branch is deemed “far away” from the

*Specifically, MPI-parallelised 3D in-place real-to-complex (and its complex-to-real dual) FFTs are needed. The pyFFTW Python package wraps FFTW, enabling it to be called from Python. However it does not include this particular pair of transforms, and so CONCEPT calls FFTW from a small C function which is then incorporated into CONCEPT via Cython. To this day this is the only C source code in CONCEPT.

[†]A noticeable counter example is GRAPESPH [46], a smoothed-particle hydrodynamics (a particle-based and mesh-free technique where continuous fluid quantities like $\rho(\mathbf{x})$ are described through convolution of some kernel $W(\mathbf{x})$ with the particles) code running on hardware specifically designed for the direct summation problem.

particle in question. In the simplest version, this collective branch force would be computed as the force from a single particle situated at the centre of mass of the particles within the branch and with a mass equal to the total mass of these particles.

Tree methods are then generally constructed by extending the PP method with such a tree structure. For each round of force computation, the tree must first be built. The lowest level of the tree is of course the particles themselves, which we may refer to as ‘leaves’. The highest level is the collection of all particles within the box, which we may refer to as the ‘root’. Building the tree then amounts to grouping together branches (starting with the leaves) into ever greater branches, until we reach the root. This grouping can be done in many different ways, e.g. based on the actual clusters of particles as described earlier, through nearest neighbour searches. A much simpler approach is to just divide up space recursively. For our cubic box, the simplest approach is to do a recursive division into octants, forming what is known as an octree. The recursion stops once a branch contain just a single particle.

For each branch, the particle distribution within now needs to be boiled down to some simpler but approximate form, like the position of the centre of mass and the total mass. More generally, we could compute and store the first few terms needed for a multipole expansion of the gravitational potential arising from the particles within the branch, with the monopole corresponding exactly to just the specification of centre of mass and total mass.

To compute the force on a given particle, the tree now has to be ‘walked’ with respect to that particle. Starting from the root, each branch is recursively considered for direct summation, i.e. either the approximate branch force is computed and applied after which the recursion ends, or the branch is ‘opened’ meaning that the recursion continues down the twigs. The last ingredient of such a tree algorithm is then some criterion which can decide whether or not to open a branch with respect to a given particle. Such a criterion should at least take into account the solid angle extended by the branch as viewed from the particle location, but can also depend directly on the particle itself, e.g. through the total force the particle received in the last time step, from which one can construct an estimate of the error introduced by not opening a branch further in the current time step.

Both GADGET and PKDGRAV makes extensive use of tree methods. In the case of GADGET-2, a simple octree equipped with monopoles is what is used. Regardless of the specifics, tree methods generally scale like $\mathcal{O}(N \log N)$ as each of the N particles now only receive forces from

a logarithmically growing (due to the recursion) number of branches. Tree methods thus scale similarly to the PM method. By tightening the opening criterion, the accuracy (and computational cost) of tree methods can be tuned. In the limit of always opening each branch until a leaf is reached, the tree methods reduce to (an even slower version of) the PP method. The generally claimed $\mathcal{O}(N \log N)$ scaling of tree methods is then only true for a “reasonable” opening criterion. In this reasonable regime, tree methods can indeed allow for much higher spatial resolution than the PM method. Although these have the same scaling, the PM method is generally quite a bit faster in absolute terms, as the construction of the tree itself also require significant computational resources. To speed up tree methods, one may then reuse (parts of) the tree over several time steps.

Adaptive Mesh Refinement

Just as a tree can be used to bunch particles in the PP method together into larger groups, we can equip the PM method with a tree of ever more refined meshes. This is once again most simply done using an octree. Adaptive Mesh Refinement (AMR) [19, 47] methods can then be thought of as being very analogous to tree methods, but with each branch equipped with a mesh. To be truly useful in cosmology, this multi-levelled mesh needs to be able to adaptively refine itself in regions of high density variations.

The key problems any adaptive mesh method has to overcome are that of non-trivial boundary conditions between cells of difference sizes, as well as how to go about solving the Poisson equation on a multi-levelled grid. An example of an N -body code using an AMR-extended PM method is RAMSES. Here, the Fourier technique described in subsection 5.3.2 is used to compute the potential at the root level, i.e. on a global mesh with low spatial resolution. At finer levels, the Laplacian is approximated in real space by finite differences, from which the potential is solved for via relaxation techniques.

Perhaps more commonly, AMR techniques are used not alongside PM in particle codes, but on its own in fluid dynamical codes where the cells of the mesh are all there ever are. After all, once cells can adaptively be made arbitrarily small, the need for an explicit particle description diminishes. We shall look closer into numerical fluid dynamics in chapter 6, though we shall keep our attention to grids of fixed size.

Hybrid Methods

The PP and the PM method complement each other in that the first is accurate and even fast for a small number of particles due to its low overhead, while the latter is very efficient for large number of particles but suffer from low spatial resolution. We can combine the two methods in several ways forming ‘hybrid methods’, aiming to get the best of both worlds. To do this, we want to split gravity into a short-range and a large-range part, just as was done for the Ewald summation in subsection 5.3.1. With the exact same splitting based on the error function and its complement, the long-range potential solved through the PM method is then the Fourier sum in (5.7), i.e. the potential in Fourier space is changed from $\sim 1/\mathbf{k}^2$ to $\sim \exp(-\mathbf{k}^2 x_s^2)/\mathbf{k}^2$, x_s being the force split scale, now determining the scale below which the PP method should take over from the PM method. The Gaussian rapidly reduces the long-range potential for $|\mathbf{k}| > x_s^{-1}$, leaving the small scale contribution of gravity to the PP method.

With the long-range part of gravity efficiently taken care of by the PM method, the remaining short-range contributions can be dealt with using direct summation. This short-range force is exactly that of the real space part of (5.8), though now we have no need for the sum over particle images \mathbf{n} , as we are guaranteed that only the nearest particle/image has any measurable short-range contribution. This then reduces the PP part of the hybrid scheme to a single N^2 sum over particle pairs. To gain any real benefit, we must now take into account that only nearby particles have any non-negligible short-range interaction, i.e. we can ignore any pair for which the separation is much larger than x_s .

A critical part of a PP-PM hybrid scheme is then some sort of grouping mechanism, allowing for efficient determination of neighbour particles. In CONCEPT, such a hybrid scheme is implemented with a simple grouping based only on which MPI process the particles belong to (as a given MPI process only takes care of particles within a given region). The resulting grouping of particles disregard the value of x_s entirely and generally results in much too few groups, cutting down on the N^2 operations by a suboptimal amount. This unfortunately leaves CONCEPT with the PM method as the only scalable force implementation.

A hybrid method between the PP and PM method is unimaginatively dubbed PPPM or P³M. Another approach is to pair PM with a tree method, resulting in a treePM method, which is the approach of GADGET. The tree structure then serves as the needed grouping mechanism, with a tweaked opening criterion accommodating the expo-

nential falloff of the short-range force. Instead of completely ignoring the short-range interaction between distance particles as in the P³M method, the treePM method do account for these, though through its use of truncated multipole expansions of now large groups of particles.

6 Fluid Dynamics

In this chapter we shall take a look at numerical implementations of fluid dynamics. While chapter 5 gave a broad overview of N -body techniques, i.e. standard numerical methods used when working with particles, this chapter does not aim to do the same for computational fluid dynamics (CFD). Instead we shall focus just on the two CFD methods implemented in CONCEPT, known as the MacCormack (MC) scheme and the Kurganov-Tadmor (KT) scheme.

6.1 Introduction

Conceptually, the particle (N -body) methods of chapter 5 can be used on any species, as the particles simply trace out trajectories in phase space. In the case of cold matter, the general distribution function $f(t, \mathbf{x}, \mathbf{q})$ loses the \mathbf{q} dependence, massively reducing the phase space volume needed to be traced by the particles. For some phase space density of particles needed in order to obtain a given precision, removing the \mathbf{q} dependence of f then drastically cuts down on the number of needed particles. In the general case of species with non-vanishing pressure, e.g. neutrinos, the \mathbf{q} dependence remain and so each point in space \mathbf{x} now needs an entire \mathbf{q} distribution of particles.

Consider for a moment a species with non-vanishing pressure, but vanishing shear and higher-order moments. Due to the \mathbf{q} dependence of f from the pressure, a particle description would require each point \mathbf{x} to contain several particles, sampling the \mathbf{q} distribution at that point in space. Through the Boltzmann hierarchy formalism (3.23), we can integrate this \mathbf{q} distribution to obtain a list of fluid variables. From the assumption of vanishing shear and higher-order fluid variables, all the information of the \mathbf{q} distribution is then contained in the pressure. By changing from the Lagrangian (particle) description to an Eulerian (fluid) description, $\{\mathbf{x}_n, \mathbf{q}_n\} \rightarrow \{\rho_n, u_n^i, P_n\}$, the three \mathbf{q} dimensions effectively collapse into the pressure variable, itself depending only on space \mathbf{x} . Here, the integer $1 \leq n \leq N$ labels the particles whereas the integer

triple $\mathbf{n} = (n_x, n_y, n_z)$, $1 \leq n_x, n_y, n_z \leq N_m$, labels grid coordinates on an $N_m \times N_m \times N_m$ grid. With the particles strewn out more or less uniformly throughout a hypercubical phase space volume with a density comparable to that of the grid points, we see that the fluid description allows us to encode the system using much less information; $6N \sim 6N_m^6 \gg 5N_m^3$, with $6 = 3 + 3$ the number of canonical variables of each particle (the dimensionality of phase space) and $5 = 1 + 3 + 1$ the number of 3D scalar fields/Grids used for the fluid representation.

For a fluid with non-zero shear and higher-order fluid variables, corresponding to large ℓ_{\max} , these will also have to be represented as numerical grids, enlarging the amount of information needed to fully describe the state. Eventually then, the advantage of a fluid description seemingly diminishes. For species with large ℓ_{\max} , such as neutrinos, the hope on which much of this work is based is that only the lower moments will behave non-linearly, effectively partitioning the Boltzmann hierarchy into a non-linear part (lower ℓ) and a linear part (higher ℓ). We define this split to be at ℓ_{nl} so that moments with $\ell \leq \ell_{\text{nl}}$ are treated non-linearly whereas moments with $\ell > \ell_{\text{nl}}$ are treated linearly. Only a few moments ($\ell_{\text{nl}} + 1$, as the first moment is numbered as 0) then need to exist as numerical 3D grids, whereas all higher-order moments are solved linearly as described in subsection 3.3.2, with no non-linearity being fed back from the non-linear evolution of the lowest moments.

6.1.1 Numerical Fluid Equations

Most of chapter 3 was spent on the development of the fluid equations needed for the Eulerian description, with the primary equations being the continuity and Euler equations. To solve such equations numerically, we discretise space \mathbf{x} , which is most simply done by placing a cubical mesh throughout the simulation box, exactly akin to the φ grid of the PM method describe in subsection 5.3.2. As mentioned in subsection 5.3.3, adaptive mesh geometries can also be used, enhancing performance and spatial resolution, but at the cost of much complication to the algorithms. As only fixed, cubical meshed are implemented into CONCEPT, these are the only ones we shall deal with here.

The system of fluid equations that is solved in CONCEPT is the non-linear continuity and Euler equation in conservation form (3.34), but

with GR source terms removed. This leaves us with

$$\begin{cases} \dot{\varrho} = -a^{3w-2}\partial_i J^i + 3H(w\varrho - \mathcal{P}), \\ J^i = -a^{3w-2}\partial^j \frac{J^i J_j}{\varrho + \mathcal{P}} - a^{-3w}\partial^j(\mathcal{P}\delta^i_j + \varsigma^i_j) - a^{-3w}(\varrho + \mathcal{P})\partial^i\varphi, \end{cases} \quad (6.1)$$

with the peculiar potential φ given by the Poisson equation (3.35). These are then the equations we wish to solve numerically in this chapter. We shall think of having $\ell_{\text{nl}} = 1$, corresponding to evolving ϱ and J^i non-linearly according to the continuity and Euler equation (6.1), but with \mathcal{P} and ς^i_j somehow given by linear theory. What linear theory provides are the transfer functions $T_{\delta P}(k)$ and $T_\sigma(k)$, from which we can construct linear 3D fields $P(\mathbf{x})$ (or $\mathcal{P}(\mathbf{x})$) and $\sigma^i_j(\mathbf{x})$ (or $\varsigma^i_j(\mathbf{x})$) using the scalar realisation (4.12) and rank-2 tensor realisation (4.14). At least some of the missing non-linearity can be injected using non-linear realisation schemes, which are explored in section 7.4.

We can exploit the symmetric property of ς^i_j so that we only realise its 6 unique components, rather than all 9. Using its tracelessness we could further reduce this number to 5, as we can then construct the remaining diagonal component from the other two. This however require us to have these two components of ς^i_j in memory simultaneously. Since terms in (6.1) contains at most a single factor of ς^i_j , we can choose to reuse the same memory to store each of the 6 components in turn. This is the strategy adopted in CONCEPT.

6.1.2 General Considerations

This subsection serves as a very brief general introduction to the subset of the field of computational fluid dynamics that we will need. CFD can be viewed as a subset of the field of numerical analysis of partial differential equations (PDEs). When studying PDEs analytically (i.e. non-numerically), a given PDE is grouped into one of several major classes with different behaviour. For us, the most important class is that of *hyperbolic* PDEs, which can always be cast in the form of a conservation equation, i.e. the time derivative of some quantity is given exclusively by a sum of *fluxes*, meaning divergences of other quantities. This has the physical interpretation that any change to the time varying quantity is due to transport of this quantity. In our numerical set-up, we think of small cubical cells each containing a single value of each quantity. In the case of (6.1), each cell then has a single value of ϱ and J^i , corresponding to a definite energy and momentum through integration

over the cell. The energy is then changed due to energy fluxes $\partial_i J^i$ over the 6 cell faces, while momentum is changed due to momentum fluxes $\partial^j J^i J_j$ and pressure/shear fluxes $\partial^j (\mathcal{P} \delta^i_j + \varsigma^i_j)$, again over the 6 cell faces.

With hyperbolic PDEs we can then start from some initial condition, i.e. a specification of all spatial fields at some time, after which we can run the system forward (or backward) in time. In contrast, no analogous evolution is to be found in e.g. the Poisson equation (3.35), which is classified as an *elliptical* PDE. Here a static solution must be found from a given boundary condition (periodicity, in our case).

The whole reason for using the ‘conserved’ fluid variables $\{\varrho, J^i, \mathcal{P}, \varsigma^i_j\}$ instead of a more familiar set like $\{\rho, u^i, P, \sigma^i_j\}$ was to exploit the conservative nature of the system, casting the fluid equations into what we might call its manifestly hyperbolic/conservative form (6.1). A huge complication is the fact that (6.1) of course contain terms other than flux terms on the right-hand-side. Specifically, the Hubble term in the continuity equation and the gravitational term in the Euler equation are “proper” source terms. The classification of the pressure/shear flux as either a flux term or a source term is less clear due to the pressure and shear being treated linearly. At early, linear times the pressure/shear flux will yield the correct physical fluxes, but at non-linear times these will be uncorrelated (or at least less correlated) with the non-linearly evolved J^i , effectively converting the pressure/shear flux term into a source term. Our actual equations (6.1) are then not purely hyperbolic. Our approach is then to consider the various source terms as small relative to the flux terms, only slightly perturbing the hyperbolicity of the system, though this might not really be the case.

Stability

There exists a wealth of different numerical methods for solving hyperbolic PDEs. Given some problem it is generally not at all straight forward to determine which method will lead to best results. Most modern methods concern themselves with the capturing of shock waves, which is particularly tricky to get right. As our fluids are always collisionless, this is not a concern of ours and so we should be able to get by using less complicated methods.

A basic criterion for any method is that it ought to be *stable*, meaning that the numerical solution does not simply blow up after a few time steps. To demonstrate this concept of stability of the numerical solver, we consider integrating the continuity equation over a small time, i.e. just

a few time steps. In this limit, the Hubble expansion can be neglected, removing the source term. We furthermore consider the velocity field to be slowly varying in space, so that $\partial_i u^i \approx 0$. We then have just

$$\dot{\rho} \approx -a^{-1} u^i \partial_i \rho, \quad (6.2)$$

where the velocity has been excluded from the differentiation due to it being roughly constant over the short interval we are considering.

We now need discrete versions of the temporal and spatial derivatives. For the temporal derivative, the most obvious discretisation is that of *forward Euler* differencing;

$$\dot{\rho}|_{\mathbf{n}}^j = \frac{\rho_{\mathbf{n}}^{j+1} - \rho_{\mathbf{n}}^j}{\Delta t} + \mathcal{O}(\Delta t). \quad (6.3)$$

Here $\rho_{\mathbf{n}}^j$ means the value of the density in the numerical grid at grid point $\mathbf{n} \in \mathbb{N}^3$ at time step $j \in \mathbb{N}$, i.e. the numerical approximation to $\rho(j\Delta t, \mathbf{n}\Delta x)$. The forward Euler difference (6.3) then simply means that we can perform a time step by adding $\Delta t \dot{\rho}|_{\mathbf{n}}^j$ to the current state, which indeed seems very reasonable. The forward Euler method is called an *explicit* method since it gives us a direct algebraic equation for the state at the next time step, given just the current state.

For the spatial derivative of (6.2), the forward Euler method would treat left and right asymmetrically, for which we have no reason (unlike in the temporal case where the future is unknown). Instead we should use some symmetric difference, the general form of which is given in (5.27). The lowest order version of this is the second order derivative

$$\partial_x \rho|_{\mathbf{n}}^j = \frac{\rho_{n_x+1,n_y,n_z}^j - \rho_{n_x-1,n_y,n_z}^j}{2\Delta x} + \mathcal{O}(\Delta x^2). \quad (6.4)$$

Substituting (6.3) and (6.4) into (6.2) we have

$$\begin{aligned} \rho_{\mathbf{n}}^{j+1} \approx \rho_{\mathbf{n}}^j & - \frac{\Delta t}{2a\Delta x} \left[u_x (\rho_{n_x+1,n_y,n_z}^j - \rho_{n_x-1,n_y,n_z}^j) \right. \\ & + u_y (\rho_{n_x,n_y+1,n_z}^j - \rho_{n_x,n_y-1,n_z}^j) \\ & \left. + u_z (\rho_{n_x,n_y,n_z+1}^j - \rho_{n_x,n_y,n_z-1}^j) \right]. \end{aligned} \quad (6.5)$$

A tool for analysing whether a scheme like (6.5) is stable is von Neumann stability analysis [48]. In the limit $\Delta t \rightarrow 0$, our equation becomes just $\dot{\rho} \approx 0$, the Fourier space solution of which is of the form $\rho_{\mathbf{n}}^{j+1} = \xi^j(k) e^{ikn\Delta x}$, i.e. each eigenmode grow independently with some *amplification factor* ξ .

Remembering that j is our time step index, we see that having $|\xi(k)| > 1$ will lead some modes to grow exponentially, rendering the numerical scheme unstable.

Replacing $\rho_{\mathbf{n}}^{j+1}$ with $\xi^j(\mathbf{k})e^{ikn\Delta x}$ in (6.5) and treating all directions as equal, meaning $u_x = u_y = u_z = \sqrt{(u_x^2 + u_y^2 + u_z^2)/3} \equiv u/\sqrt{3}$ and $k_x = k_y = k_z \equiv k$, we end up with

$$\xi = 1 - i\sqrt{3} \frac{u\Delta t}{a\Delta x} \sin k\Delta x, \quad (6.6)$$

for which we see that $|\xi| > 1$ for every finite k , making the forward Euler second-order symmetric method (6.5) unconditionally unstable.

It turns out that our simple scheme can be stabilised using the *Lax method* [49], which simply smooths out the solution by replacing the original value $\rho_{\mathbf{n}}^j$ on the right-hand-side of (6.5) by its spatial average from neighbouring cells:

$$\begin{aligned} \rho_{\mathbf{n}}^{j+1} \approx & \frac{1}{6} \left(\rho_{n_x+1,n_y,n_z}^j + \rho_{n_x-1,n_y,n_z}^j \right. \\ & + \rho_{n_x,n_y+1,n_z}^j + \rho_{n_x,n_y-1,n_z}^j \\ & + \rho_{n_x,n_y,n_z+1}^j + \rho_{n_x,n_y,n_z-1}^j \left. \right) \\ & - \frac{\Delta t}{2a\Delta x} \left[u_x (\rho_{n_x+1,n_y,n_z}^j - \rho_{n_x-1,n_y,n_z}^j) \right. \\ & + u_y (\rho_{n_x,n_y+1,n_z}^j - \rho_{n_x,n_y-1,n_z}^j) \\ & \left. + u_z (\rho_{n_x,n_y,n_z+1}^j - \rho_{n_x,n_y,n_z-1}^j) \right]. \end{aligned} \quad (6.7)$$

Now doing the replacement $\rho_{\mathbf{n}}^{j+1} \rightarrow \xi^j(\mathbf{k})e^{ikn\Delta x}$ and treating all directions equally, we end up with

$$\xi = \cos k\Delta x - i\sqrt{3} \frac{u\Delta t}{a\Delta x} \sin k\Delta x, \quad (6.8)$$

with the stability criterion $|\xi| \leq 1$ leading to

$$\Delta t \leq \frac{a\Delta x}{\sqrt{3}u}, \quad (6.9)$$

which is known as the Courant-Friedrichs-Lowy (or just Courant) condition, most often seen in its 1D version where the $\sqrt{3}$ is absent and in static space $a = 1$.

We might have argued intuitively that we indeed ought to keep the time step size below $\Delta x/\dot{x} = a\Delta x/u$, as otherwise information from

one cell has time to travel further than to just its immediate neighbour cells, which should be disallowed as our scheme only directly couples neighbouring cells. The factor of $1/\sqrt{3}$ is less obvious, but since in practice the Courant condition is only a necessary and not sufficient constraint, the real time step size is usually chosen as a small fraction of what is allowed by this condition anyway. The most important information from the stability analysis is the knowledge of whether a method is stable at all.

The replacement of ρ_n^j with its spatial average from neighbour cells in (6.7) corresponds to keeping the numerical method as in (6.5) but modifying the underlying differential equation (6.2), specifically as [48] $\dot{\rho} \approx -u^i \partial_i \rho + \Delta x^2 / (2\Delta t) \partial_i \partial^i \rho$, with a suitable symmetric second-order finite difference approximation for the Laplacian. That is, the Lax trick which stabilised the method corresponds exactly to the addition of a diffusion term to the underlying differential equation. One may argue then that though the method is stable, it no longer solves the differential equation that we wanted to solve. We can understand this in terms of the amplification factor ξ (6.8). For $k\Delta x \ll 1$ corresponding to large scales, we have $\xi \approx 1$, whereas for larger k , ξ decreases. Remembering that the eigenmodes grow as $|\xi|^j$, this then means that the added diffusion only change the solution at small scales, which is what we may expect given that the averaging is spatially local. Generally then, we can safely introducing numerical diffusion into a hyperbolic PDE without worrying that it perturbs the solution much beyond the grid scale.

6.2 The MacCormack Method

The simplest CFD method implemented in CONCEPT is that of *MacCormack* (MC) [50]. This is a reformulation of the Lax-Wendroff method, which itself is the second-order version of the (temporally) first-order Lax-improved forward Euler method analysed in the previous section. I originally decided to implement a Lax-Wendroff-type method simply because it was amongst the simplest possible second-order methods. The specific MacCormack version was chosen because it is easier to implement, requiring no explicit Jacobian and less memory than other Lax-Wendroff-type methods.

We remind ourselves that the system of equations under consideration is that of (6.1), with only the first term of both equations considered as flux terms. Unlike the leapfrog method used to evolve the Hamiltonian particle system in subsection 5.2.1, the MacCormack method updates

the ϱ_n and the J_n^i grids synchronously. To do this, secondary grids are needed to store intermediate values, which one might think of as ϱ_n and J_n^i evaluated halfway through the time step. We shall denote these new grids with a star, ϱ_n^* and J_n^{i*} , and call them simply the ‘starred’ grids. The introduction of this additional set of grids then doubles the memory usage.

In the previous section we used the notation ϱ_n^j to refer to the density grid at time $j\Delta t$ and grid point $n\Delta x$. As only a single ϱ grid (now two, with the starred grids) is present in memory, keeping the temporal index around maps badly to the actual numerical implementation, and so we shall drop it from now on. The MacCormack method consists of a predictor step, populating the starred grids from the unstarred (normal) grids, followed by a corrector step in which the unstarred grids are updated according to the starred grids. For our system of equations (6.1), the predictor step takes the form

$$\begin{cases} \varrho_n^* = \varrho_n - \frac{1}{\Delta x} \int_t^{t+\Delta t} a^{3w-2} dt \sum_{i=1}^3 (J_{n+[\delta_{i1}, \delta_{i2}, \delta_{i3}]}^i - J_n^i), \\ J_n^{i*} = J_n^i - \frac{1}{\Delta x} \int_t^{t+\Delta t} a^{3w-2} dt \sum_{j=1}^3 \left(\frac{J^i J_j}{\varrho + \mathcal{P}} \Big|_{n+[\delta_{j1}, \delta_{j2}, \delta_{j3}]} - \frac{J^i J_j}{\varrho + \mathcal{P}} \Big|_n \right). \end{cases} \quad (6.10)$$

This is very much analogous to the forward Euler method (6.5), the difference being that (6.10) approximate the divergence asymmetrically with respect to left and right. The notation $n + [\delta_{i1}, \delta_{i2}, \delta_{i3}]$ should be understood as vector addition, where the vector in the bracket takes on the values of the unit vectors along the coordinate axes; $\hat{i}, \hat{j}, \hat{k}$, for $i = 1, i = 2, i = 3$, respectively. The divergence is thus approximated as the difference between the local cell value and that of the right/upper/forward neighbouring cells.

Just as in the particle drift operation (5.1), the state variables are treated as constant over the predictor (as well as the corrector) step, whereas the scale factor $a(t)$ and the effective equation of state $w(t)$ are integrated properly.

The corrector step is quite similar to the predictor step, where now we account for the left/lower/backward neighbouring cells. Crucially, the corrector fluxes are constructed from the predictor (starred) values,

which is what allows for second-order behaviour:

$$\left\{ \begin{array}{l} \varrho_n \rightarrow \frac{\varrho_n + \varrho_n^*}{2} - \frac{1}{2\Delta x} \int_t^{t+\Delta t} a^{3w-2} dt \sum_{i=1}^3 (J_n^{i*} - J_{n-[{\delta}_{i1}, {\delta}_{i2}, {\delta}_{i3}]}^{i*}), \\ J_n^i \rightarrow \frac{J_n^i + J_n^{i*}}{2} - \frac{1}{2\Delta x} \int_t^{t+\Delta t} a^{3w-2} dt \sum_{j=1}^3 \left(\left. \frac{J^{i*} J_j^*}{\varrho^* + \mathcal{P}} \right|_n - \left. \frac{J^{i*} J_j^*}{\varrho^* + \mathcal{P}} \right|_{n-[{\delta}_{j1}, {\delta}_{j2}, {\delta}_{j3}]} \right), \end{array} \right. \quad (6.11)$$

where the arrows indicate in-place updates of the unstarred grids, completing the (flux part of the) time step. Due to the factors of 2 in the denominators of (6.11) (no such factor appears in (6.10)), the forward and backward differences are accounted for by the same amount as in (6.4), but with the forward and backward differences evaluated effectively half a time step out of phase.

For non-linear problems such as ours, the asymmetry between left and right (and similar for the other dimensions) introduced by the MacCormack method (6.10) and (6.11) will manifest as spurious anisotropic errors after multiple time steps. To counteract this, the direction specified by $[{\delta}_{i1}, {\delta}_{i2}, {\delta}_{i3}]$ should rotate in between time steps, taking on all 8 possible directions over a cycle of 8 time steps. That is, we should really replace $[{\delta}_{i1}, {\delta}_{i2}, {\delta}_{i3}]$ for $[\pm {\delta}_{i1}, \pm {\delta}_{i2}, \pm {\delta}_{i3}]$ in (6.10) and (6.11), with the three \pm signs swapping back and forth between being a $+$ and a $-$ with a period of 1, 2 and 4 time steps.

As we treat \mathcal{P} (and ς_j^i) linearly due to our choice of $\ell_{nl} = 1$, the evolution of the pressure is not solved by the MacCormack method, but somehow given to us using linear theory. Thus, no starred version \mathcal{P}^* appear in (6.11). Viewing the predictor step not as “half a spatial step” as only three of the six faces are treated, but rather as half a time step, one may argue that while the predictor step uses $\mathcal{P}_n(t) = \mathcal{P}_n(j\Delta t) \equiv \mathcal{P}_n^j$, the corrector step ought to use $\mathcal{P}_n^{j+1/2}$, i.e. the pressure given at a time halfway in between time step j and $j+1$. As \mathcal{P}_n does not match the correct non-linear pressure anyway, we choose to ignore this possible slight gain in consistency and simply use the exact same grid \mathcal{P}_n in both the predictor and corrector step. After a full MacCormack (predictor plus corrector) step, \mathcal{P}_n is then re-realised from linear theory at the time $j+1$.

6.2.1 Source Terms

The standard MacCormack method given by (6.10) and (6.11) treat our system of fluid equations (6.1) as purely hyperbolic, ignoring all source terms. These source terms include the “real” Hubble and gravitational source terms, but also the pressure and shear flux, which is converted to an affective source term because we do not treat the pressure and shear as non-linear variables.

The MacCormack method can be extended [51] to include source term evaluation in the predictor and corrector steps. This would then require us to compute the gravitational potential φ twice for each time step, which is not only more computationally demanding but also does not play well with the leapfrog integration of the particle component, as here gravity is only applied once per time step. Instead we choose to view the MacCormack step as the fluid analogue of the particle ‘drift’. The drifting of the particle component (updating of particle positions) are then carried out simultaneously with the MacCormack step for the fluid component. Note that these operations are independent of each other, as the particle drift and the fluid MacCormack step each operate entirely within the given component. The particle kick step is similarly associated with the application of fluid source terms, which are not independent operations since both cases involves the computation of the global potential φ , due to both particle and fluid components.

The Hubble source term $3H(w\varrho_n - \mathcal{P}_n)$ of the continuity equation and the pressure and shear source term $-a^{-3w}\partial^j[\mathcal{P}_n\delta_j^i + (\varsigma_j^i)_n]$ of the Euler equation are what is referred to as “internal source terms” in the CONCEPT source code, meaning that they are constructed entirely from quantities defined on the fluid component itself together with background quantities (no interactions). Since the grids storing the fluid variables all share the same grid size*, these internal source terms can be evaluated and applied directly for each cell. We approximate the divergence of the pressure and shear using the second-order symmetric difference (6.4), to keep it at the same order as the divergences handled by the MacCormack steps.

The gravitational potential also lives on a grid, φ_n . As this grid is shared among all particle and fluid components, its grid size may be different from that of a given fluid component. To obtain the contribution to this potential from a fluid component, the full machinery of the PM

*In CONCEPT, multiple fluid components can coexist with different grid sizes N_m (or equivalently different grid spacings Δx). Within a given component however, all fluid grids share the same grid size.

method (see subsection 5.3.2) then has to come into play. Each fluid cell \mathbf{n} then act as a particle with mass $a^{-3\omega} \varrho_n \Delta x^3$ situated at $\Delta x \mathbf{n}$. If the fluid grids and the potential grids happen to have the same grid size, $N_m = N_\varphi$, and we are using either NGP or CIC interpolation, the interpolations between the fluid grid and the potential grid reduces to simple copying of cell values, as each pair of grid points of the two grids lie exactly on top of each other. That is, the density values assigned to the φ grid through interpolation (5.21) from a fluid ϱ_n grid always just amounts to $\rho_n = a^{-3(1+\omega)} \varrho_n$. In these cases then, no interpolation convolutions are actually carried out, and so the explicit deconvolutions $\Delta x_\varphi^3 / W_h$ of (5.26) actively harm the accuracy of the resulting gravitational forces, why we are better off without them. It seems reasonable that for $N_m \approx N_\varphi$, less correction in the form of deconvolutions $\Delta x_\varphi^3 / W_h$ is needed than for e.g. $N_m \gg N_\varphi$. I have not found a totally self-consistent way of dealing with these deconvolutions in the general case of a fluid with arbitrary grid size.

As CONCEPT uses CIC interpolation and the vast majority of the simulations I have run during my PhD has had $N_m = N_\varphi$, the standard case is that of no actual interpolation. The choice is then made to never perform any deconvolution of the potential grid with regards to the fluid density field. Exactly how this is made to work in the general case where both particle and fluid components are present in the same simulation is described in section 6.4.

6.2.2 Vacuum Corrections

The standard MacCormack method does not contain numerical diffusion akin to that of Lax (6.7). Unlike the Lax-free forward Euler method however, the MacCormack method remains stable in the von Neumann sense without such added diffusion, though it does introduce dispersive errors around large gradients. For a neutrino fluid the gravitational source term does lead to somewhat strong gradients, but in practice these behave nicely, perhaps due to the diffusive effects of the pressure.

Gravity also introduces a seemingly unrelated problem: The formation of cosmic voids may rob some cells of their entire energy content. Due to the finite time step size, this means that ϱ_n may become slightly negative, which in turn leads to spurious unphysical behaviour as $\varrho_n \geq 0$ is implicitly assumed throughout CONCEPT. Simply enforcing a minimum value of zero for ϱ_n is not a working solution, as this introduces the dreaded large gradients. Instead, some local smoothing scheme which guarantees positivity of energy and conservation of energy and momentum

is needed. After quite some trial and error, I found the following scheme — which I have dubbed “vacuum corrections” — to be working well.

After each of the predictor and corrector step, each grid point \mathbf{n} is checked for negative densities, which are then corrected. In fact, after the predictor step, the condition which triggers vacuum correction is not just $\varrho_{\mathbf{n}} < 0$, but that a cell \mathbf{n} has lost a significant fraction of its energy. Symbolically then, vacuum corrections for cell \mathbf{n} takes place if

$$\begin{cases} \varrho_{\mathbf{n}} + \frac{n_f}{2}(\varrho_{\mathbf{n}}^* - \varrho_{\mathbf{n}}) < \varrho_{\text{vacuum}} & (\text{after predictor step}), \\ \varrho_{\mathbf{n}} < \varrho_{\text{vacuum}} & (\text{after corrector step}), \end{cases} \quad (6.12)$$

where a density of zero is replaced by the small number* ϱ_{vacuum} , below which the density should never drop. After the predictor step, the change to the density is $(\varrho_{\mathbf{n}}^* - \varrho_{\mathbf{n}})/2$. The ‘foresight’ parameter n_f then controls the number of similar future time steps which should trigger vacuum correction now. This is a user parameter which may be specified for each individual component. It has a default value of $n_f = 30$.

Once vacuum correction has been triggered for a cell \mathbf{n} , either after the predictor or after the corrector step, the correction is carried out as follows:

$$\varrho_{\mathbf{m}} \rightarrow \varrho_{\mathbf{m}} + f_s f_f \sum_{\substack{\mathbf{m}' | 1 \leq (\mathbf{m}' - \mathbf{m})^2 \leq 3 \\ \mathbf{m}' \neq \mathbf{m}}} \frac{\varrho_{\mathbf{m}'} - \varrho_{\mathbf{m}}}{(\mathbf{m}' - \mathbf{m})^2} \quad \text{for } \mathbf{m} | 1 \leq (\mathbf{m} - \mathbf{n})^2 \leq 3. \quad (6.13)$$

The above describe how the cells \mathbf{m} of the $3 \times 3 \times 3$ cell block with cell \mathbf{n} at the centre are pairwise smoothed. The smoothing corresponds to averaging the density over pairs of cells, but with a weight proportional to the inverse square of the distance between the cells. It should be noted that as it stands, (6.13) really only works for vacuum corrections taking place after a corrector step, as here the latest updated values are stored in the unstarred $\varrho_{\mathbf{n}}$ grid. For predictor step vacuum corrections, (6.13) should really be written exclusively using the starred grid $\varrho_{\mathbf{n}}^*$.

The ‘smoothing factor’ f_s sets the overall amount of smoothing. With a small f_s and a large n_f , the vacuum correction will take place “adiabatically”, i.e. a gentle smoothing will take place over many time steps. The smoothing factor is further parameterised as

$$f_s = f'_s \left(\frac{6}{1} + \frac{12}{2} + \frac{8}{3} \right)^{-1}, \quad (6.14)$$

*In CONCEPT, this is defined to be $\varrho_{\text{vacuum}} = 10^2 \epsilon_m$, where $\epsilon_m \approx 2 \times 10^{-16}$ is the double precision machine epsilon.

with f'_s being a user parameter which defaults to unity. With $f'_s = f_f = 1$, the expression (6.14) results in the bad value in cell \mathbf{n} being completely distribution among its 26 neighbour cells, while cell \mathbf{n} itself gets populated with the inverse square weighted sum of the same neighbour cells.

The last factor, $f_f \leq 1$, controls the adiabaticity of the smoothing in the case of a vacuum correction taking place after a predictor step. For a corrector step vacuum correction then, $f_f = 1$. In case of predictor vacuum correction, linear extrapolation of the evolution leads to $\varrho_{\mathbf{n}} < \varrho_{\text{vacuum}}$ after $2(\varrho_{\mathbf{n}} - \varrho_{\text{vacuum}})/(\varrho_{\mathbf{n}} - \varrho_{\mathbf{n}}^*)$ time steps. The factor f_f is defined as the inverse of this, producing less smoothing for less imminent vacuum problems;

$$f_f = \begin{cases} \frac{1}{2} \frac{\varrho_{\mathbf{n}} - \varrho_{\mathbf{n}}^*}{\varrho_{\mathbf{n}} - \varrho_{\text{vacuum}}} & (\text{after predictor step}), \\ 1 & (\text{after corrector step}). \end{cases} \quad (6.15)$$

With f_s and f_f defined in (6.14) and (6.15), the smoothing of the density is fully specified by (6.13).

Though negative values taken on by the momentum J^i is of course not unphysical, the momentum grids $J_{\mathbf{n}}^i$ has to be smoothed as well, as otherwise discontinuities will be induced in the velocity field $u^i \propto J^i/\varrho$. The smoothing of $J_{\mathbf{n}}^i$ follows the exact same prescription (6.13) with the same values for f_s and f_f .

The vacuum corrections (6.13) cannot be carried out in-place, as the value of cell \mathbf{m} is generally needed to compute the corrections for cells $\mathbf{m}' \neq \mathbf{m}$. Also, any number of cells may need to be corrected during the same time step. To implement the vacuum corrections, four new grids needs to be allocated, storing the corrections to $\varrho_{\mathbf{n}}$ and $J_{\mathbf{n}}^i$.

As all the MacCormack method (6.10) and (6.11) does is to exchange energy and momentum between neighbouring cells, the $3 \times 3 \times 3$ smoothing scheme of (6.13) is guaranteed to result in positive densities. In extreme cases where several cells within a $3 \times 3 \times 3$ block have gone negative however, a single sweep of vacuum corrections may not suffice. Allowing arbitrarily many such sweeps, positivity is indeed guaranteed for all cells. However, repeatedly smoothing out the fluid damps structure on larger and larger scales. By default, CONCEPT only performs a single vacuum correction sweep after the predictor step, whereas it attempts up to*

*Letting the maximum allowed number of repeated vacuum correction sweeps be proportional to N_m seems sensical. A value of N_m is however properly much too large, as every pair of cells in the entire grid is able to communicate after just $N_m/6$ sweeps.

N_m vacuum sweeps after a corrector step, after which it exits on error if vacuum cells are still present.

6.3 The Kurganov-Tadmor Method

I have found that the MacCormack method with the vacuum correction extension generally works well for evolving neutrino fluids. For neutrinos with large masses ($\sum m_\nu \gtrsim 1\text{ eV}$), bad behaviour was however seen at late times, as described in the paper of chapter 10. We originally thought that this was due to the somewhat simplistic vacuum correction smoothing as described in subsection 6.2.2, as the erroneous behaviour appeared precisely in the case of large neutrino mass which leads to more structure and hence more extreme voids. To cure the issue, a second CFD method was implemented into CONCEPT; the *Kurganov-Tadmor* (KT) method. In the end the KT method did not succeed in curing the bad behaviour, and because it is slower than the MC method it was never used for much. As quite some time was spent implementing the KT method into CONCEPT, I feel that a description of this method do belong in this thesis.

The Total Variation Diminishing Feature

The Kurganov-Tadmor method [52] belongs to the class of MUSCL* schemes, which in turn can be viewed as higher-order versions of Godunov's scheme [54]. These are finite-volume methods, meaning that the values stored at the grid points are thought of as cell-averaged values of the quantity in question. The idea of MUSCL schemes is to reconstruct the fluxes at the cell boundaries using piecewise polynomial reconstruction of the continuous field based on the sampled values at the grid points.

The KT scheme is semi-discrete, meaning that it provides a method for computing the needed fluxes, but does not otherwise restrict time evolution. For a complete CFD method then, the KT method must be augmented with some time evolution scheme. In CONCEPT, simple second-order Runge-Kutta is used as the time integrator. With the KT method being second-order in space, this makes it of exactly the same order as the MC method. The possible benefits of the KT method are then not ascribed simply to it being a higher-order method. Instead, it guarantees that the evolution of the system is *total variation diminishing*

*‘Monotonic upwind scheme for conservation laws’, [53].

(TVD) [55]. This effectively means that small wrinkles at the grid scale cannot form, allowing for stable evolution across steep gradients and even good capture of shocks.

Godunov's theorem states that linear numerical schemes for solving PDEs are necessarily *monotonicity preserving*, i.e. they never produce spurious oscillations. For our use case we may think of this feature as the same as being TVD. We can construct such a linear scheme using the MUSCL reconstruction of fluxes at the cell boundaries, modelling all fluid quantities as piecewise constant. The resultant first-order scheme is then TVD. Sticking to this first-order scheme is no good as even though it is numerically very stable, it poorly resolves the physics. We can generate a corresponding second-order MUSCL scheme using piecewise linear reconstruction of the fluxes. Within some small time window, this second-order method is better at tracking the physics accurately, but it may generate oscillations as it is not TVD. The idea of the KT scheme (and others before it) is to mix together the first- and second-order fluxes to create a new flux which is both second-order and TVD.

As the KT method does not concern itself with the time integration, we shall ignore the details of the equations we are actually trying to solve (the continuity and Euler equation (6.1)). Instead, we write the time evolution of any variable as

$$\begin{aligned} \dot{\varrho}_n = -\frac{1}{\Delta x} \Big(& F_{n_x+1/2,n_y,n_z}^* - F_{n_x-1/2,n_y,n_z}^* \\ & + F_{n_x,n_y+1/2,n_z}^* - F_{n_x,n_y-1/2,n_z}^* \\ & + F_{n_x,n_y,n_z+1/2}^* - F_{n_x,n_y,n_z-1/2}^* \Big) + (\text{source terms}), \end{aligned} \quad (6.16)$$

which is indeed completely general. The semi-discrete nature of (6.16) is clearly visible with the continuous time derivative on the left-hand-side and the discrete, numerical fluxes on the right-hand-side. Just as for the MC method, the KT method does not (in its standard form) consider source terms.

The Numerical Flux

We shall now briefly lay out the steps needed to compute the numerical fluxes appearing in (6.16). The six fluxes correspond to the six faces of the cell $\mathbf{n} = (n_x, n_y, n_z)$, where half-integer indices refer to interface values. The recipe for the flux is of course similar for all dimensions, i.e. $F_{n_x,n_y-1/2,n_z}^*$ is obtained from $F_{n_x-1/2,n_y,n_z}^*$ by permuting the first and second index. Similarly, $F_{n_x+1/2,n_y,n_z}^*$ is obtained from $F_{n_x-1/2,n_y,n_z}^*$ by

adding one to the first index. We shall then consider just $F_{n_x-1/2,n_y,n_z}^*$, which for brevity we can write as $F_{n_x-1/2}^*$ as all dimensions decouple.

The MUSCL reconstruction of the flux at interface $n_x - 1/2$ can be done using cell information on either the left or the right side of the interface, resulting in two different fluxes $F_{n_x-1/2}^{L/R}$ at this same interface. The KT method defines the numerical flux to be the average of this left and right flux, plus an additional term:

$$F_{n_x-1/2}^* = \frac{1}{2} \left[F_{n_x-1/2}^L + F_{n_x-1/2}^R - \max_{T \in \{L,R\}} |v_{n_x-1/2}^T| (\varrho_{n_x-1/2}^R - \varrho_{n_x-1/2}^L) \right], \quad (6.17)$$

where the density ϱ appears as we have the continuity equation in mind, though so far the prescription is completely general. The $\varrho_{n_x-1/2}^{L/R}$ are then the MUSCL reconstructed values of ϱ at the $n_x - 1/2$ interface, computed from the left and from the right. For a smoothly varying field these will be close to equal, as they are estimations of the same field at the same position. The difference $\varrho_{n_x-1/2}^R - \varrho_{n_x-1/2}^L$ thus represent the disagreement of the left and right reconstructions, which we can use as a measure of non-smoothness. We see that if the right estimate of $\varrho_{n_x-1/2}$ is larger than the left estimate, the numerical flux $F_{n_x-1/2}^*$ is corrected by a negative amount proportional to the difference of the estimations. This lowering of the flux means that after the current time step, cell ϱ_{n_x} will receive a slightly smaller fraction of the energy of cell ϱ_{n_x-1} than it otherwise would (assuming $F_{n_x-1/2}^* > 0$), serving to minimise the difference $\varrho_{n_x-1/2}^R - \varrho_{n_x-1/2}^L$ in the next time step and thereby smoothing out the density field.

In (6.17), the difference $\varrho_{n_x-1/2}^R - \varrho_{n_x-1/2}^L$ is multiplied by the absolute value of the maximum possible comoving speed of information* at the interface $n_x - 1/2$, estimated using reconstructed values both from the left and from the right. In CONCEPT this speed is computed as $|\dot{x}_{n_x-1/2}^{L/R}| + c_s/a$, with $\dot{x} = u_x/a$ the comoving velocity and $c_s = \sqrt{w}$ the physical sound speed. In terms of the conserved fluid variables, this becomes

$$|v_{n_x-1/2}^{L/R}| = a^{3w-2} \frac{|J_{n_x-1/2}^{x,L/R}|}{\varrho_{n_x-1/2}^{L/R} + \mathcal{P}_{n_x-1/2}^{L/R}} + \frac{\sqrt{w}}{a}. \quad (6.18)$$

The above expression is not really correct, as the sound speed $\sqrt{\delta\mathcal{P}/\delta\varrho}$ is not spatially constant and hence only approximately equal to \sqrt{w} .

*Due to the entire system of fluid equations, not just the continuity equation in isolation, though only in the direction perpendicular to the cell interface (here x).

Also, the shear ς_j^i is completely absent, although it does contribute to propagation of information. As (6.18) is really just a proportionality factor used for numerical smoothing, this should not be critical. I have experimented with including a term like $a^{-1} [|(\varsigma_x^x)_{n_x-1/2}^{L/R}| / (\varrho_{n_x-1/2}^{L/R} + \mathcal{P}_{n_x-1/2}^{L/R})]^{1/2}$ on the right-hand-side of (6.18), but it made no appreciable difference. As this require ς_x^x (and ς_y^y and ς_z^z for the other directions) to be realised even for the continuity equation, it dramatically increases the computation time. We thus stick to the simpler (6.18).

We now specify what the MUSCL reconstruction looks like. We write this in terms of ϱ , but it is exactly the same for all other fluid variables:

$$\begin{aligned} \varrho_{n_x-1/2}^{L/R} &= \varrho_{n_x-1/2 \mp 1/2} \\ &\pm \frac{1}{2} \phi \left(\frac{\varrho_{n_x-1/2 \mp 1/2} - \varrho_{n_x-3/2 \mp 1/2}}{\varrho_{n_x+1/2 \mp 1/2} - \varrho_{n_x-1/2 \mp 1/2}} \right) (\varrho_{n_x+1/2 \mp 1/2} - \varrho_{n_x-1/2 \mp 1/2}), \end{aligned} \quad (6.19)$$

where the choice of L/R determines the signs of \pm and \mp . In either case, (6.19) does indeed express the interface values purely in terms of the cell values. The *flux limiter* function $\phi(r)$ can take on a variety of different forms, controlling the exact MUSCL reconstruction. We shall discuss these soon.

We still have not explicitly written down the fluxes $F_{n_x-1/2}^{L/R}$ of (6.17). These are simply the fluxes as dictated by the fluid equation in question, i.e. the continuity equation (6.1) in our case, constructed at the interfaces using (6.19). That is,

$$F_{n_x-1/2}^{L/R} = J_{n_x-1/2}^{L/R} \int_t^{t+\Delta t} a^{3\omega-2} dt, \quad (6.20)$$

with $J_{n_x-1/2}^{L/R}$ constructed using the procedure (6.19).

Flux Limiters

The last remaining missing piece of the KT method is the specification of the flux limiter function $\phi(r)$ appearing in the MUSCL reconstruction (6.19). We see that $\phi(r) = 0$ reduces the MUSCL reconstruction to that of the piecewise constant reconstruction, i.e. the first-order TVD scheme. To obtain the second-order generally non-TVD reconstruction, we must put $\phi(r) = 1$.

More generally, we seek some smoothly varying function $\phi(r)$ which results in the numerical flux being both second-order and TVD. It can

be shown [56] that this is the case when

$$\begin{cases} \phi(r < 0) = 0, \\ r \leq \phi(0 \leq r \leq 1) \leq 2r, \\ 1 \leq \phi(1 \leq r \leq 2) \leq r, \\ 1 \leq \phi(r > 2) \leq 2. \end{cases} \quad (6.21)$$

Additionally, any limiter should have the property

$$\frac{\phi(r)}{r} = \phi\left(\frac{1}{r}\right) \quad (6.22)$$

which ensures that left and right are treated symmetrically.

Effectively (6.21) defines piecewise upper and lower bounds for $\phi(r)$. The two extreme functions which lie exactly on these boundaries are the ‘minmod’ limiter [57]

$$\phi(r) = \max[0, \min(1, r)] \quad (6.23)$$

and the ‘superbee’ limiter [58]

$$\phi(r) = \max[0, \min(1, 2r), \min(2, r)], \quad (6.24)$$

both of which also satisfy (6.22). Using the minmod limiter (6.23) for the MUSCL reconstruction (6.19) then results in the weakest second-order and strongest TVD behaviour, making it rather dissipative but good for problems with smooth solutions. At the other end of the spectrum, the superbee (6.24) limiter has very little dissipation and so is good for problems with very steep gradients. It is less suited for problems with smooth solutions, as it tends to artificially steepen smooth regions.

A large set of standard limiters in between the minmod and superbee limiter exists, though any decent function satisfying (6.21) and (6.22) will be just as applicable. One of the earliest and most widely used flux limiters which reside very much right in between the minmod (6.23) and superbee (6.24) limiters are that of van Leer [59];

$$\phi(r) = \frac{r + |r|}{1 + |r|}. \quad (6.25)$$

For a given problem, exactly what flux limiter to use to a large extent comes down to trial and error.

Figure 6.1 shows non-linear neutrino power spectra produced by the Kurganov-Tadmor method with the different flux limiters (6.23), (6.24)

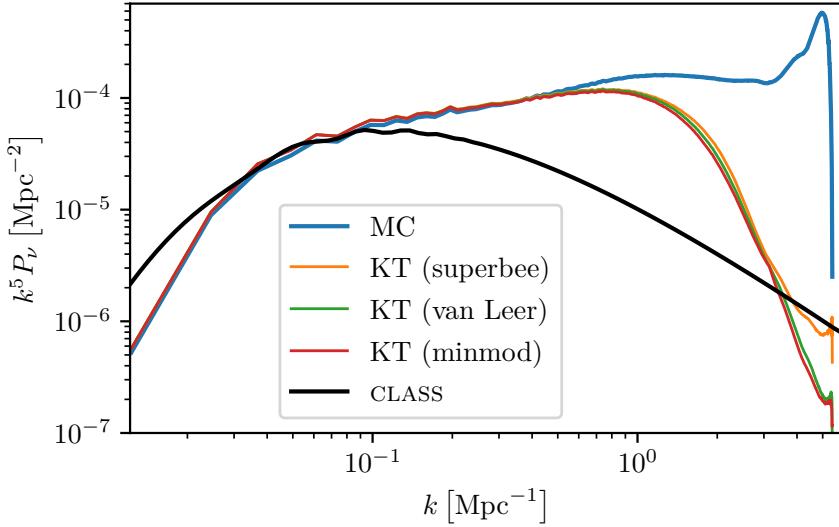


Figure 6.1 – Neutrino power spectra at $a = 1$ from cosmological simulations with non-linear matter particles and a non-linear neutrino fluid. The neutrino mass hierarchy is degenerate with $\sum m_\nu = 0.1 \text{ eV}$. Other cosmological parameters are as defined in tables 2.1 and 4.1. The coloured lines are the non-linear spectra produced using CONCEPT, whereas the corresponding linear spectrum is shown in black. For the blue line, the extended MacCormack method of section 6.2 was used, whereas for the yellow, green and red lines the Kurganov-Tadmor method of this section was used, with flux limiters as indicated on the legend.

and (6.25). We see that overall the three flux limiters results in very similar power spectra, though a slight difference in power at small scales can be seen in accordance to the proclaimed relative dissipation levels of these limiters. Figure 6.1 also shows the corresponding power spectra using the MacCormack method, which follow those of the Kurganov-Tadmor method closely at large and intermediate scales, but does not have the heavy suppression at small scales.

Figure 6.1 includes the entire k range on which the non-linear power spectra have been measured. In both ends, spurious effects can be seen which are normally not included by simply showing a narrower range in k . At the lowest k , the non-linear spectra drops severely below the linear spectrum. This is mostly due to ‘cosmic variance’, meaning that too few \mathbf{k} modes with very small $k = |\mathbf{k}|$ exist for their average to have converged. This effect is further described in chapter 7. For the smallest scales, $k \gtrsim 3 \text{ Mpc}^{-1}$, bends in the non-linear spectra can be seen at the same values of k in all non-linear spectra, though most strongly pronounced for

the MacCormack spectrum as this is not already superficially damped by strong numerical viscosity. These bends stem from the CIC deconvolution present in the matter potential felt by the neutrinos. Exactly how gravity is dealt with for fluids is described in the next section. In an attempt to minimise this effect from the deconvolution, we will end up having separate particle and fluid potential grids, as we shall see.

6.4 Gravity

As both the MacCormack method of section 6.2 and the Kurganov-Tadmor method of section 6.3 do not handle source terms, such terms must be added to the fluid time evolution by hand. Subsection 6.2.1 laid out the basic principles of this with regards to the MC method, but the same principles hold true for the KT method. The only source term not described there is that of gravity, which we shall now deal with. This will again be applicable regardless of which method is used to integrate the flux terms.

Section 5.3 described several methods for computing the Newtonian gravitational forces within a system of particles. The basic methods were that of PP and PM, the first of which computed the gravitational force on a particle by directly pairing it up with every other particle, whereas the latter solved for the collective gravitational potential, which could then be applied to each particle. Treating the fluid cells as particles, the same basic two options are available when it comes to computing gravity in the context of fluids. The single benefit of the PP method over the PM method was its larger dynamic range. As the exact same spatial discretisation imposed by the PM method is already present for the Eulerian fluid grids, no additional resolution is there to be gained by direct summation. Only the PM method* (and more advanced versions thereof) is then sensible when it comes to fluids.

Fluid-only Simulations

As discussed in subsection 6.2.1, the PM method of subsection 5.3.2 is almost readily applicable to fluid components. The two deconvolutions of the φ grid (5.26), each corresponding to multiplication by $\Delta x_\varphi^3/W_h$ in Fourier space, arose from interpolations between particle positions and grid point locations. Restricting ourselves to having the fluid grids being

*Though the first part of the name ‘particle-mesh’ loses its meaning in the context of fluids, we shall continue to refer to the method as ‘PM’.

of the same size as the potential grid, $N_m = N_\varphi$, no such interpolation takes place for fluids, at least for the NGP and CIC interpolations, the latter of which is what is used in CONCEPT. No deconvolutions should then be present in the fluid PM method.

The fluid PM method starts out by copying ϱ_n onto the equally sized potential grid φ . After an in-place FFT, the potential grid then stores what in the syntax of subsection 5.3.2 would be called φ_h . The potential values are then constructed as given by (5.26) but without the deconvolutions;

$$\varphi_h = -\frac{4\pi G a^{-1-3w}}{\mathbf{k}^2} \varrho_h, \quad \varphi_{\mathbf{h}=\mathbf{0}} = 0, \quad (6.26)$$

where $\mathbf{k} = (2\pi/L)\mathbf{h}$ and $\varrho = a^{3(1+w)}\rho$ (3.29) has been used. An inverse in-place FFT then results in φ_n , which can then be differentiated using finite difference approximations like (5.28), resulting in $D^i\varphi_n$, the numerical equivalent of $\partial^i\varphi(\mathbf{x})$.

With $D^i\varphi_n$ at hand, the question is how to apply it to J_n^i in accordance with the Euler equation (6.1). As the gravitational source term is $-a^{-3w}(\varrho + \mathcal{P})\partial^i\varphi$, the naïve thing to do is just to carry out the multiplication $(\varrho_n + \mathcal{P}_n)D^i\varphi_n$ for each grid point n , multiply by $a(t)^{-3w(t)}\Delta t$ and subtract the result from J_n^i . We can however do better by integrating the scale factor over the time step, just as in the particle kick operation (5.2). Here, the scale factor integral had an integrand of a^{-1} . This ultimately came from the potential scaling $\varphi \propto a^2\bar{\rho} \propto a^2a^{-3} = a^{-1}$, where the proportionality $\bar{\rho} \propto a^{-3}$ implicitly contains the assumption of the perturbations responsible for the potential being matter perturbations, i.e. $w = 0$. For a general species then, we would have $\varphi \propto a^2\bar{\rho} \propto a^2a^{-3(1+w)} = a^{-1-3w}$, the same factor as appears in (6.26).

Insisting that the numerical application of the gravitational source term takes the form

$$J_n^i \rightarrow J_n^i - (\varrho_n + \mathcal{P}_n)D^i \langle \varphi_n \rangle_t^{t+\Delta t} \int_t^{t+\Delta t} \frac{dt'}{a(t')^{3w(t')}} , \quad (6.27)$$

the time step integral of a^{-1-3w} is then embedded in $\langle \varphi_n \rangle_t^{t+\Delta t}$. We do not combine this embedded time step integral with that appearing in (6.27), as we want to separate the collective potential from the individual species. That is, we include the factor a^{-3w} explicitly in (6.27) because it appears in the Euler equation (6.1), whereas we stay agnostic with respect to the nature of the potential.

When having multiple fluid components α , the right-hand-side of the Poisson equation (6.26) becomes a sum over these components. Allowing for multiple fluid components then, we can write the time step averaged potential as

$$\langle \varphi_h \rangle_t^{t+\Delta t} = -\frac{4\pi G}{k^2} \Delta t^{-1} \sum_{\substack{\text{fluid} \\ \alpha \in \text{components}}} \varrho_{\alpha,h} \int_t^{t+\Delta t} \frac{dt'}{a(t')^{1+3w_\alpha(t')}} , \quad \langle \varphi_{h=0} \rangle_t^{t+\Delta t} = 0 , \quad (6.28)$$

the Fourier transform of which is $\langle \varphi_n \rangle_t^{t+\Delta t}$, needed in (6.27). In the case of a single matter fluid ($w = 0$), equations (6.27) and (6.28) reduce to $J_n^i \rightarrow J_n^i - (\varrho_n + \mathcal{P}_n) D^i \varphi_n \int_t^{t+\Delta t} dt' / a(t')$, with the integral correspondingly exactly to the one we have for the particle kick operator (5.2). Indeed, since for matter, $J = a^4 \rho$, $\varrho = a^3 \rho$, $\mathcal{P} = 0$, integrating over a cell volume we reobtain the exact form of the entire particle kick operator.

Combined Particle and Fluid Simulations

We have seen how the properly time step averaged potential due to all fluid components are constructed (6.28). For simulations having both fluid and particle components, we need a corresponding equation for particles. From (5.26), we may write this as

$$\langle \varphi'_h \rangle_t^{t+\Delta t} = -\frac{4\pi G}{k^2} \frac{\Delta x_\varphi^6}{W_h^2} \Delta t^{-1} \int_t^{t+\Delta t} \frac{dt'}{a(t')} \sum_{\substack{\text{particle} \\ \alpha \in \text{components}}} \varrho_{\alpha,h} , \quad \langle \varphi'_{h=0} \rangle_t^{t+\Delta t} = 0 , \quad (6.29)$$

where $\varrho_{\alpha,h} = a^3 \rho_{\alpha,h}$ is (the Fourier transform of) the density grid resulting from particle interpolation via (5.21).

Two deconvolutions $\Delta x_\varphi^3 / W_h$ appear in (6.29), although only one is needed for the interpolation from particle positions to grid coordinates. From subsection 5.3.2, we remember that this additional deconvolution will be needed for the force interpolation back to the particle positions. The prime in $\langle \varphi'_h \rangle_t^{t+\Delta t}$ was originally used to signal this additional deconvolution, but now it also functions as a means to discern the time step averaged fluid-only potential $\langle \varphi_h \rangle_t^{t+\Delta t}$ from the time step averaged particle-only potential $\langle \varphi'_h \rangle_t^{t+\Delta t}$.

In order to allow for fluid and particle components to interact, we must add together the fluid-only potential (6.28) and the particle-only potential (6.29) into a total potential. However, simply doing $\langle \varphi_h \rangle_t^{t+\Delta t} + \langle \varphi'_h \rangle_t^{t+\Delta t}$

results in wrong deconvolutions when applied to both fluid and particle components. Let us first consider the case of a fluid component. The potential it feels due to itself and all other fluid components is just that of $\langle \varphi_h \rangle_t^{t+\Delta t}$, with no deconvolution. The potential it feels from all particle components will be that of $\langle \varphi'_h \rangle_t^{t+\Delta t}$, but with one deconvolution instead of two. This single deconvolution is retained as it stems from the particle interpolation onto the grid, which remains even though the final force is evaluated at the grid points. Thus, we may write the total time step averaged potential felt by any fluid component as

$$\langle \varphi_h^{\text{fluid}} \rangle_t^{t+\Delta t} = \langle \varphi_h \rangle_t^{t+\Delta t} + \frac{W_h}{\Delta x_\varphi^3} \langle \varphi'_h \rangle_t^{t+\Delta t}, \quad (6.30)$$

with the factor $W_h/\Delta x_\varphi^3$ cancelling out one of the two deconvolutions of $\langle \varphi'_h \rangle_t^{t+\Delta t}$.

The total potential to be applied to particle components will be that of the twice deconvolved particle-only potential $\langle \varphi'_h \rangle_t^{t+\Delta t}$ plus a singly deconvolved fluid-only potential $\langle \varphi_h \rangle_t^{t+\Delta t}$, where the single deconvolution is needed for the upcoming force interpolation:

$$\begin{aligned} \langle \varphi'_h^{\text{particle}} \rangle_t^{t+\Delta t} &= \frac{\Delta x_\varphi^3}{W_h} \langle \varphi_h \rangle_t^{t+\Delta t} + \langle \varphi'_h \rangle_t^{t+\Delta t} \\ &= \frac{\Delta x_\varphi^3}{W_h} \langle \varphi_h^{\text{fluid}} \rangle_t^{t+\Delta t}. \end{aligned} \quad (6.31)$$

In the end, we see that the total potential to be applied to particle and fluid components only differ by a single deconvolution, reflecting the force interpolation of the particles. As (6.31) still contains the additional deconvolution, $\langle \varphi'_h^{\text{particle}} \rangle_t^{t+\Delta t}$ retains its prime.

Note that because $\langle \varphi_h \rangle_t^{t+\Delta t}$ and $\langle \varphi'_h \rangle_t^{t+\Delta t}$ are constructed from real-space quantities and contain different \mathbf{k} -dependent factors in (6.28) and (6.29), at least two grids in memory are needed for their construction. Thus, although the final relation in (6.31) seems to suggest that we can get by with a single potential grid in memory for combined particle and fluid simulations, we really cannot.

7 Realisations

This chapter takes a closer look at the CONCEPT implementation of *realisations*, i.e. the construction of a 3D real space field from its 1D Fourier space transfer function. These realisations are what provide the numerical bridge between linear and non-linear theory. The primary use for realisation is then to generate the initial conditions for the non-linear simulation, be it particle positions and momenta or fluid energy and momentum densities.

For our non-linear fluid (neutrino) simulations where we split the Boltzmann hierarchy into a non-linear and linear part at ℓ_{nl} , a second use for realisations appear. Here, the effect from the linear moments $\ell > \ell_{\text{nl}}$ on the non-linear moments $\ell \leq \ell_{\text{nl}}$ are taken into account by realising the needed linear moments at each time step, generating a full 3D field from which e.g. spatial derivatives along each direction can be computed.

Finally, for low ℓ_{nl} (in CONCEPT, only $\ell_{\text{nl}} < 2$ is implemented), realising the linear $\ell > \ell_{\text{nl}}$ moments at late times introduces errors because of the missing non-linearity. We can try to correct for this in multiple ways, as we shall see.

7.1 Numerical Details

The mathematical construction of linear realisations was carried out in section 4.3. Here, linear realisations of scalar (4.12), vector (4.13) and rank-2 tensor (4.14) fields were developed. Quite a few subtleties introduce themselves when attempting to implement these ideas numerically, as we shall see in this section.

As usual, we discretise space by introducing a grid, mapping spatial coordinates $\mathbf{x} = \mathbf{n}\Delta x$, $\mathbf{n} \in \mathbb{N}^3$, with the cell width $\Delta x = L/N_m$, L being the box size and N_m is the number of grid points along each dimension of the grid. Focusing just on the scalar field realisation (4.12) for now,

this can then be written for a numerical grid as

$$\delta_{\mathbf{n}}(a) = L^{-3/2} \sum_{\mathbf{h}} e^{2\pi i \mathbf{h}\mathbf{n}/L} T_{\delta}(a, k) \zeta(k) \mathcal{R}_{\mathbf{h}}, \quad (7.1)$$

where $V = L^3$ has been used and $k = |\mathbf{k}|$ with $\mathbf{k} = (2\pi/L)\mathbf{h}$, $\mathbf{h} \in \mathbb{Z}^3$. In (7.1), time a is treated continuously, while space (both real \mathbf{n} and Fourier \mathbf{h}) is discrete. The continuous transfer function $T_{\delta}(k)$ and primordial curvature perturbation $\zeta(k)$ are then sampled only at discrete k , determined by the values undertaken by \mathbf{h} , which are not explicitly written in (7.1). For an infinite grid, we would have $\mathbf{h} \in \mathbb{Z}^3 \setminus \mathbf{0}$, where the removal of the DC mode ensures that the real space field has zero mean, $\sum_{\mathbf{n}} \delta_{\mathbf{n}} = 0$. Alternatively, we can assign $\mathcal{R}_{\mathbf{h}=\mathbf{0}} = 0$, where we remember that $\mathcal{R}_{\mathbf{h}}$ is the grid of complex Gaussian random numbers, from which all 3D structure spring.

For our finite grid, the possible values of $\mathbf{n} = (n_x, n_y, n_z)$ are restricted to $0 \leq n_x, n_y, n_z < N_m$, while $\mathbf{h} = (h_x, h_y, h_z)$ is restricted to $0 \leq |h_x|, |h_y|, |h_z| \leq N_m/2$, with the value $N_m/2$ referred to as the Nyquist frequency. Explicitly attaching this information to the Fourier sum (7.1), we have

$$\begin{aligned} \delta_{\mathbf{n}}(a) &= L^{-3/2} \sum_{\mathbf{h} \in \{(h_x, h_y, h_z) \in \mathbb{Z}^3 \setminus \mathbf{0} \mid |h_w| \leq N_m/2\}} e^{2\pi i \mathbf{h}\mathbf{n}/L} T_{\delta}(a, k) \zeta(k) \mathcal{R}_{\mathbf{h}} \\ &\equiv \mathcal{F}_{\text{scalar}}^{-1} [T_{\delta}(a, k) \zeta(k) \mathcal{R}_{\mathbf{h}}], \end{aligned} \quad (7.2)$$

with $w \in \{x, y, z\}$ and $\mathcal{F}_{\text{scalar}}^{-1}$ defined as a convenient shorthand notation for the discrete inverse Fourier transform, including the $L^{-3/2}$ normalisation and the values undertaken by \mathbf{h} , corresponding to terms in the Fourier sum.

Though (7.2) is written in terms of δ , it serves as the general prescription for realising any scalar field from its transfer function. Thus, substituting e.g. $\delta \rightarrow \delta P$, the realised field is now the pressure perturbation.

Letting \mathbf{h} visit all existing grid points, $0 \leq |h_x|, |h_y|, |h_z| \leq N_m/2$, may in fact not be optimal, as the anisotropy of our cubic box then trickles through to Fourier space, arbitrarily sampling some but not all \mathbf{h} for which $|\mathbf{h}| > N_m/2$, or correspondingly $|\mathbf{k}| > \pi/\Delta x$. It seems that the exact effects from leaving out such ‘corner modes’ are not clearly understood, see e.g. [60]. I have experimented with not including corner modes into the CONCEPT realisations, but found no conclusive results.

7.1.1 The Complex Conjugacy Symmetry

As described in section 4.3, the reality of δ_n forces

$$\mathcal{R}_{-\mathbf{h}} = \mathcal{R}_{\mathbf{h}}^*, \quad (7.3)$$

as $\mathcal{R}_{\mathbf{h}}$ is the only complex factor inside the Fourier transform (7.2). This effectively cuts the amount of information stored in $\mathcal{R}_{\mathbf{h}}$ in half, which we can take advantage of numerically as the FFTW library [45] offers such “real” Fourier transforms. Specifically, the numerical grid $\mathcal{R}_{\mathbf{h}}$ then only contains the points

$$-N_m/2 < h_x, h_y \leq N_m/2, \quad 0 \leq h_z \leq N_m/2, \quad (7.4)$$

i.e. only half of the z dimension*. The set of values undertaken by \mathbf{h} in (7.2) is then somewhat larger than what is actually available in practice (7.4). As this is just an implementation detail of the numerical FFT, we shall not reflect this in our definition of scalar realisation (7.2).

The complex conjugacy symmetry (7.3) imposed on $\mathcal{R}_{\mathbf{h}}$ is then mostly taken care of by the FFTW internals, as most of the “negative frequencies” do not actually exist in memory. For \mathbf{h} in the $h_z = 0$ plane, $-\mathbf{h}$ is fully contained within the same plane. As this plane exist in full in memory, we have to manually ensure† that the complex conjugacy symmetry $\mathcal{R}_{-h_x, -h_y, 0} = \mathcal{R}_{h_x, h_y, 0}^*$ is satisfied.

For even‡ N_m , the N_m real-valued points along the real z dimension has to be reused for the $1 + N_m/2$ complex-valued points along the Fourier space z dimension, where the additional point comes about because the h_z range (7.4) include both 0 and $N_m/2$. Considering a complex number as a pair of real numbers, we then have $2 + N_m$ real values along the Fourier space z direction, whereas in real space we had N_m . As the amount of information must be the same for the real and Fourier representation of the grid, there must be some redundancy in the Fourier space representation. As seen in (7.4), we do not need both

*Which dimension to cut in half does not matter. The z dimension is what is chosen for the 3D MPI parallelised routines of FFTW.

†As we are using numerical Fourier transform routines specifically designed to transform such “real” fields, one might expect the complex conjugate symmetry to simply be enforced by FFTW, effectively overwriting $\mathcal{R}_{-h_x, -h_y, 0} \rightarrow \mathcal{R}_{h_x, h_y, 0}^*$ if non-symmetric values are present. At least for the MPI routines, no such enforcement is carried out. The $z = 0$ plane is simply assumed to be correctly symmetrised, with erroneous results to follow if this is not the case.

‡FFTW does support odd N_m , but to keep things simple this is disallowed by CONCEPT.

the positive and negative Nyquist frequencies for any given dimension. Thus, these must contain the same information, and are in fact equal. At the $h_z = N_m/2$ plane then, the complex conjugacy symmetry reads

$$\mathcal{R}_{-h_x, -h_y, N_m/2} = \mathcal{R}_{-h_x, -h_y, -N_m/2} = \mathcal{R}_{h_x, h_y, N_m/2}^*, \quad (7.5)$$

i.e. we have the same situation as for the $h_z = 0$ plane; the $h_z = N_m/2$ plane must be symmetrised manually. It is to be understood that any future occurrence of \mathcal{R}_h satisfy these symmetries.

With only half the initial degrees of freedom on the $h_z = 0$ and $h_z = N_m/2$ planes, we have effectively removed an amount of information corresponding to one entire complex plane, exactly matching the additional complex point along the z direction.

Measuring the Power Spectrum

From (4.2), we can compute the power spectrum numerically as $P_\delta(a, k) = V^{-1} \langle |\delta_k(a)|^2 \rangle_V$, where the angle brackets denote shell averaging over all \mathbf{k} of the same $|k|$ within the simulation box of volume V . Writing this out explicitly, we have

$$P_\delta(a, k) = V^{-1} \left[\sum_{\mathbf{h} | h^2 = L^2/(2\pi)^2 k^2} 1 \right]^{-1} \sum_{\mathbf{h} | h^2 = L^2/(2\pi)^2 k^2} \delta_{\mathbf{h}}^2(a), \quad (7.6)$$

where the first sum simply counts the number of \mathbf{k} modes with $|k| = k$. As this number is finite, the averaging is never fully converged. This is especially true for the lowest k , as here there are the fewest modes, clearly seen in figure 6.1.

The primary numerical representation of δ is that of the real space grid δ_n . This grid is realised using the $\mathcal{F}_{\text{scalar}}^{-1}$ operator defined in (7.2). Inverting the Fourier transform of this equation, we have

$$\begin{aligned} \delta_{\mathbf{h}}(a) &= L^3 \sum_{\mathbf{n} \in \{(n_x, n_y, n_z) \in \mathbb{N}^3 \mid |n_w| < N_m\}} e^{-2\pi i \mathbf{h} \cdot \mathbf{n}/L} \delta_{\mathbf{n}}(a) \\ &\equiv L^{3/2} \mathcal{F}_{\text{scalar}} [\delta_{\mathbf{n}}(a)], \end{aligned} \quad (7.7)$$

with $w \in \{x, y, z\}$ meaning that the Fourier sum takes into account all N_m^3 grid points. Equation (7.7) then defines the forward scalar Fourier transform $\mathcal{F}_{\text{scalar}}$, which constructs $\delta_{\mathbf{h}}(a)$ from $\delta_{\mathbf{n}}(a)$. Note that we indeed have $\mathcal{F}_{\text{scalar}} \mathcal{F}_{\text{scalar}}^{-1} = \mathcal{F}_{\text{scalar}}^{-1} \mathcal{F}_{\text{scalar}} = 1$. With $\delta_{\mathbf{h}}(a)$ from (7.7), we have all we need to measure the power spectrum through (7.6).

Though the numerically measured power spectrum is not extremely accurate due to the aforementioned convergence issues, the ratio between two power spectra can very accurately discern changes to the cosmological or numerical parameters. This enhanced precision comes about because the ‘cosmic variance’ due to the finite number of \mathbf{k} modes is the same in both power spectra, and so errors introduced to a large extent vanish when taking the ratio. That is, if we use the same realisation, i.e. the same random grid \mathcal{R}_h .

Generating the Random Field

Typically \mathcal{R}_h is populated using a deterministic stream of pseudo-random numbers. Which particular stream is produced is set by the so-called random seed. Keeping this seed the same for multiple simulations, the exact same \mathcal{R}_h is guaranteed. Alternatively, if we want to perform several simulations of the same cosmology but with different realisations (in order to increase the effective number of \mathbf{k} modes and hence increase the accuracy of $P_\delta(k)$) we simply use a separate random seed for each simulation.

Exactly how the stream of random numbers is mapped to the 3D grid positions of \mathcal{R}_h seems of little importance at first. The picture is complicated by the addition of parallelism, where now the grid is distributed among several processes, each of which have their own stream of random numbers. In order for \mathcal{R}_h to be independent of the number of processes used to build it, we see that all processes should use identical streams, i.e. random seeds. Furthermore, an explicit mapping from the 1D sequence of random numbers to the 3D grid positions are now needed, as we do not want the same numbers to appear multiple times throughout \mathcal{R}_h .

We often have to perform convergence tests of our simulations. Here I do not mean the convergence of the $P_\delta(k)$ computation for a given $|\mathbf{k}|$, but of the physics of the simulation. Here, convergence primarily refer to the discrete time step size Δt and grid cell width $\Delta x = L/N_m$ being small enough so that the power spectra below some k does not change by further lowering Δt and Δx . It would thus be nice if the aforementioned mapping of the random number sequence to the grid has the property that $\langle |\delta_{\mathbf{k}}|^2 \rangle_V$ is independent of the grid size* N_m , for any given \mathbf{k} . This is most straight forwardly achieved by a direct mapping between the

*In CONCEPT, the φ grid is really reused for all power spectrum computations. To be precise then, what we want is for $\langle |\delta_{\mathbf{k}}|^2 \rangle_V$ to be independent of $N_\varphi = N_m$, assuming that we keep the potential and the fluid grids of equal size.

sequence of random numbers and the physical \mathbf{k} value at each grid point, rather than the integer triple \mathbf{h} . Enlarging N_m corresponds to adding more (higher) \mathbf{k} modes, but all the lower modes are still there. The random numbers should then be drawn “in shells”, from $\mathbf{k} = 0$ and out. This also solves the problem of having the realisations be independent of the number of processes, and is the way \mathcal{R}_h is constructed in CONCEPT.

Vector and Tensor Realisations

So far we have considered the necessary conditions of \mathcal{R}_n , in order for the scalar field realisation (7.2) to result in a real field. The only actual condition was the complex conjugacy symmetry (7.3), which we then had to manually satisfy on the z DC and Nyquist planes.

The realisation of a vector field u_n^i from the transfer function of its divergence is given in (4.13), where the additional factor $-ik^i/k^2$ appears inside the Fourier transform, inverting the divergence. We wish to write down an expression for numerical vector realisation analogous to that of scalar realisations (7.2). As we shall now discuss, the factor $-ik^i/k^2$ excludes some of the grid points \mathbf{h} from the Fourier sum, leaving out specific modes. For now, we leave out the \mathbf{h} specification:

$$u_n^i(a) = L^{-3/2} \sum_{\mathbf{h}} e^{2\pi i \mathbf{h} \cdot \mathbf{n}/L} \left[-i \frac{k^i}{k^2} T_\theta(a, k) \right] \zeta(k) \mathcal{R}_h. \quad (7.8)$$

As the random field \mathcal{R}_h in (7.8) and (7.1) are the same, the presence of the imaginary unit i in (7.8) is worrying for the reality of u_n^i . However, as ik^i is invariant under simultaneous inversion $\mathbf{k} \rightarrow -\mathbf{k}$ and complex conjugation, the combination $ik^i \mathcal{R}_h$ and thus indeed the entire bracket of (7.8) satisfy the complex conjugate symmetry.

With the vector realisation (7.8) in mind, we now take a closer look at (7.5). Though this equation is written for the Nyquist frequency in the z dimension, it holds for all dimensions. Normally we do not have to worry about this symmetry condition for the x and y dimensions, as here only half of the Nyquist planes are in memory. As k^i in (7.8) changes sign under inversion $\mathbf{k} \rightarrow -\mathbf{k}$, the condition (7.5) now relates a complex point in the Nyquist half-planes to itself. As an example, consider $ik^i \mathcal{R}_h$ at the grid point $(-h_x, -h_y, -N_m/2)$ (which is not in memory). Using (7.5), this value is unchanged under a sign change of $h_z \rightarrow N_m/2$. As the new combination $ik^i \mathcal{R}_h$ must also as a whole obey a relation similar to that of (7.5), we can also flip the sign of both $h_z = N_m/2$ and k^i . Choosing $i = z$, this results in $i(N_m/2) \mathcal{R}_{-h_x, -h_y, N_m/2} = i(-N_m/2) \mathcal{R}_{-h_x, -h_y, -N_m/2} =$

$i(-N_m/2)\mathcal{R}_{-h_x,-h_y,N_m/2}$, or $\mathcal{R}_{-h_x,-h_y,N_m/2} = 0$. More generally, to ensure reality of the vector realisation (7.8) of u_n^i , we must require $\mathbf{h}_i = \frac{N_m}{2} \Rightarrow \mathcal{R}_h = 0$. That is, we have to nullify the Nyquist plane (half plane for $i \in \{x, y\}$) in dimension i , where i is given by u_n^i . As \mathcal{R}_n is supposed to be equal for all fields, one might think that it would be best to just always fill all Nyquist planes of \mathcal{R}_h with zeros, including when realising scalar fields. Alternatively* we can take this result to mean that we should just remove these particular grid points from the Fourier sum when realising vectors, i.e. leaving out the particular Nyquist plane in the frequency sampling:

$$\begin{aligned} u_n^i(a) &= L^{-3/2} \sum_{\mathbf{h} \in \{(h_x, h_y, h_z) \in \mathbb{Z}^3 \setminus \mathbf{0} \mid |h_w| \leq N_m/2 - \delta_w^i\}} e^{2\pi i \mathbf{h} \cdot \mathbf{n} / L} \left[-i \frac{k^i}{k^2} T_\theta(a, k) \right] \zeta(k) \mathcal{R}_h \\ &\equiv \mathcal{F}_{\text{vector}}^{-1} [T_\theta(a, k) \zeta(k) \mathcal{R}_h], \end{aligned} \quad (7.9)$$

where the Kronecker delta takes care of the removal of the Nyquist plane along dimension i . As for $\mathcal{F}_{\text{scalar}}^{-1}$ of (7.2), $\mathcal{F}_{\text{vector}}^{-1}$ includes all the various details of the specific Fourier transformation, including the avoidance of the Nyquist plane and the vector specific $-ik^i/k^2$ factor.

Finally, let us tackle the realisation of rank-2 tensors. Writing (4.14) out for numerical grids, we have

$$(\sigma_j^i)_n(a) = L^{-3/2} \sum_{\mathbf{h}} e^{2\pi i \mathbf{h} \cdot \mathbf{n} / L} \left[-\frac{3}{2} \left(\hat{k}^i \hat{k}_j - \frac{1}{3} \delta_j^i \right) T_\sigma(a, k) \right] \zeta(k) \mathcal{R}_h, \quad (7.10)$$

where again the exact values undertaken by \mathbf{h} is now to be worked out.

Unlike in the vector case, now the factor in front of the transfer function in (7.10) is real, and so the complex conjugate symmetry requires it to be invariant under inversion $\mathbf{k} \rightarrow -\mathbf{k}$, which is indeed the case. To ensure the reality of $(\sigma_j^i)_n$, the same additional care at the Nyquist planes must be taken as with the vector realisation. Ultimately, the additional avoidance of the Nyquist plane of dimension i came from a minus sign introduced by k^i (under inversion) in (7.9). The same condition is then required here for both i and j , with the complication that it does *not* have to be satisfied for $i = j$, as the product $\hat{k}^i \hat{k}_j$ then

*This alternative interpretation is indeed the better one, as we might imagine constructing \mathcal{R}_h not from scratch by hand but instead extracting it from a given δ_n grid (which in fact is what we shall do in section 7.4). Here of course, all Nyquist planes of \mathcal{R}_h will be populated in general.

does not change sign under inversion $\mathbf{k} \rightarrow -\mathbf{k}$. We can express these more complicated conditions through more elaborate use of Kronecker deltas:

$$(\sigma^i_j)_{\mathbf{n}}(a) = L^{-3/2} \sum_{\mathbf{h} \in \{(h_x, h_y, h_z) \in \mathbb{Z}^3 \setminus \mathbf{0} \mid |h_w| \leq N_m/2 - \delta^i_w - \delta^w_j + 2\delta^i_j\}} e^{2\pi i \mathbf{h}\mathbf{n}/L} \left[-\frac{3}{2} \left(\hat{k}^i \hat{k}_j - \frac{1}{3} \delta^i_j \right) T_\sigma(a, k) \right] \zeta(k) \mathcal{R}_{\mathbf{h}} \\ \equiv \mathcal{F}_{\text{tensor}}^{-1} [T_\sigma(a, k) \zeta(k) \mathcal{R}_{\mathbf{h}}], \quad (7.11)$$

where as usual $\mathcal{F}_{\text{tensor}}^{-1}$ encapsulates everything special about the (rank-2) tensor realisation.

7.2 Particle Realisation

In order for us to be able to realise a system of particles, we need a strategy for converting realised grids into particle positions \mathbf{x}_n and momenta \mathbf{q}_n , and we need to figure out which transfer function(s) to use.

7.2.1 The Zel'dovich Approximation

A common method for generating initial conditions for particles is that of the Zel'dovich approximation [61]. At $a = 0$, the homogeneity of the universe is achieved by positioning the particles at cubic grid locations, $\mathbf{x}_n(a = 0) = \Delta x \mathbf{n}$, establishing a relation between the Eulerian grids and the Lagrangian particles. Notationally, some implicit mapping between particle indices and grid points $n \leftrightarrow \mathbf{n}$ are now assumed.

At a later time, the particle positions are described through the *displacement field* $\psi(a, \mathbf{x})$. Keeping to the grid notation,

$$\mathbf{x}_n(a) = \Delta x \mathbf{n} + \psi_n(a), \quad \psi_n(a = 0) = \mathbf{0}. \quad (7.12)$$

Differentiation (7.12) with respect to a , we have $\dot{\mathbf{x}} = \dot{\psi}$, where the unimportant subscripts have been removed. Remembering the definition of peculiar velocity, $\mathbf{u} \equiv a\dot{\mathbf{x}}$, we can now insert this into the Newtonian limit of the continuity equation for matter (3.41), $\dot{\delta} = -a^{-1} \nabla \cdot \mathbf{u}$, obtaining $\delta(a, \mathbf{x}) = -\nabla \cdot \dot{\psi}(a, \mathbf{x})$. Using the homogeneous boundary conditions $\delta(a = 0, \mathbf{x}) = 0$, $\psi(a = 0, \mathbf{x}) = \mathbf{0}$, we can integrate this relation simply by removing the overdots;

$$\delta(a, \mathbf{x}) = -\nabla \cdot \psi(a, \mathbf{x}), \quad (7.13)$$

which is the Zel'dovich approximation for the displacement field in terms of the density contrast.

We seek to realise the displacement field on a grid, so that we may use (7.12) to produce the particle positions. As ψ is a vector, it should be realised using vector realisation (7.9). The transfer function to use should then be that of the divergence $\nabla \cdot \psi$, corresponding to $\theta = \nabla \cdot \mathbf{u}$ from T_θ of (7.9). From (7.13), this divergence is $-\delta$, and so the realisation of the displacement field takes the form

$$\psi_n^i(a) = \mathcal{F}_{\text{vector}}^{-1}[-T_\delta(a, k)\zeta(k)\mathcal{R}_h]. \quad (7.14)$$

The fact that we were able to construct particle positions from the transfer function of the density (contrast) should not be surprising. After all, we can construct the density field from the particle distribution through simple interpolation, as in e.g. (5.21).

7.2.2 Particle Momenta

The Zel'dovich approximation of the previous subsection made use of the relation $\dot{\mathbf{x}} = \dot{\psi} = a^{-1}\mathbf{u}$. The particle velocities are then given simply by the realised velocity field \mathbf{u}_n , i.e. we should not attempt to e.g. interpolate these grid values out to the newly established particle positions \mathbf{x}_n of (7.12). With the peculiar velocity field \mathbf{u}_n realised using (7.9), the particle momenta are just

$$\mathbf{q}_n = am_n \mathbf{u}_n, \quad (7.15)$$

where the relation between \mathbf{q} and \mathbf{u} is given in (3.11) and we again assume some implicit mapping between particle indices n and grid points \mathbf{n} .

Velocities from the Displacement Field

The realisation of the particle positions (7.12) and momenta (7.15) are constructed from T_δ and T_θ , respectively. Often, the momenta are constructed from T_δ as well, as it is simply more convenient to only have to provide a single transfer function to bootstrap an N -body code.

In order to relate the particle velocities to the displacement field, we consider the limit where only matter perturbations contribute to the gravitational potential, in which case we know that the evolution of δ is described by the linear growth factor $D(a)$, introduced in (3.18) and

plotted in figure 3.1. With $\dot{\psi} \propto -\dot{\delta}$ from (7.13), we can then write the displacement field as

$$\psi(a, \mathbf{x}) = \frac{D(a)}{D(a \ll 1)} \psi(a \ll 1, \mathbf{x}) \quad (7.16)$$

$$\Rightarrow \dot{\psi}(a, \mathbf{x}) = \frac{\dot{D}(a)}{D(a)} \psi(a, \mathbf{x}), \quad (7.17)$$

where $a \ll 1$ is some initial time. With $\dot{\psi} = a^{-1} \mathbf{u} = a^{-2} m^{-1} \mathbf{q}$, we can then write the particle momenta as

$$\mathbf{q}_n = a^2 H(a) f(a) m_n \psi_n(a), \quad (7.18)$$

where

$$f \equiv \frac{d \ln D}{d \ln a} = H^{-1} \frac{\dot{D}}{D} \quad (7.19)$$

is the *linear growth rate*. Often, (7.18) is considered part of the Zel'dovich approximation.

Constructing the momenta from the displacement field (7.18) instead of from the actual velocity field (7.15) corresponds to the approximation $\theta(a, \mathbf{x}) = -a H(a) f(a) \delta(a, \mathbf{x})$, i.e. the relation between momenta and density is purely a function of the background. The inclusion of e.g. massive neutrinos which affect not just the background but also has significant perturbations is then not picked up by (7.18), and so just using the honest velocity transfer functions as in (7.15) is much preferable. In CONCEPT, both options are available.

Once the background is specified, computing the growth rate f through (3.18) is a simple matter of integration. Yet, further approximations to (7.18) are often encountered, like e.g. $f(a) \approx \Omega_m^\gamma(a)$ where γ is known as the growth rate index [62].

A second-order version of the Zel'dovich approximation exists, where accelerations are taken into account. This method is referred to as 2LPT (from ‘second-order Lagrangian perturbation theory’) [63], in which case the Zel'dovich approximation might alternatively be called 1LPT. Using 2LPT thus has the potential to better capture non-linearities, allowing the simulation to begin at a later time. However, I am not aware of any 2LPT implementation which do not rely on a second-order version of the growth rate, f_2 , introduced either through a second-order version of the growth equation (3.18) or simply through relations similar to $f = \Omega_m^\gamma$. To achieve “general” 2LPT (e.g. also functioning in the case of having massive neutrinos), I suspect that we would need input from second-order perturbation theory.

7.3 Late-time Normalisation

So far we have described the realisation of fluid fields and particle systems from linear transfer functions, themselves computed using a linear Einstein-Boltzmann code like `CLASS`. As these transfer functions are computed in the framework of linear perturbation theory, they do not accurately describe the late-time non-linear universe. This is of course the whole reason why we need non-linear N -body codes at all.

Traditionally, N -body codes solve the non-linear evolution of the matter components, i.e. cold dark matter and baryons, while other components like photons and neutrinos are not evolved. Indeed perturbations of photons and of massless (or very light) neutrinos remain small all the way up to the present time, and so for these species the linear treatment remains applicable.

Leaving out such ‘relativistic species’ (photons and neutrinos) from the N -body evolution then leads to a slight decrease of the gravitational potential, affecting the clustering of matter. The first solution which comes to mind is simply to include the missing contributions to the potential from the relativistic species, which might be straightforwardly obtained using linear theory. This possibility is explored in subsection 7.3.2, where we will find that complications arise due to incompatibilities between the general relativistic linear theory and the Newtonian framework generally pervading N -body simulations. A theoretically less attractive and indeed more limited approach is that of *back-scaling*, which nevertheless historically has served the N -body community well.

7.3.1 Back-scaling

Instead of attempting to explicitly take into account the missing relativistic perturbations on the N -body particles, the back-scaling approach can be utilised to guarantee the correct general relativistic late-time matter power spectrum at linear scales, effectively faking the presence of the missing relativistic species by slightly reshaping the initial matter distribution.

We seek a matter transfer function at some initial time a_i , with the property that when evolved Newtonianly into the present, it reproduces the general relativistic effects encapsulated by the matter transfer function at the present. We thus take the present-day matter transfer function $T_{\delta_{\text{cdm+b}}}(a = 1, k)$ and ‘undo’ its Newtonian, linear time evolution. The Newtonian and linear evolution of matter is precisely what is given by

the scale-independent growth factor $D(a)$ introduced in (3.18). We may then define

$$T'_{\delta_{\text{cdm+b}}}(a_i, k) \equiv \frac{D(a = a_i)}{D(a = 1)} T_{\delta_{\text{cdm+b}}}(a = 1, k) \quad (7.20)$$

to be the back-scaled matter transfer function, the prime denoting that this is different from the actual transfer function. Thus, the realisation of $T'_{\delta_{\text{cdm+b}}}$ does not actually result in the correct density field at time a_i . However, Newtonian evolution of this field will have the effect of ‘reapplying’ the growth $D(a = 1)/D(a = a_i)$, resulting in the correct density field at the present time. While the back-scaling of the transfer function in (7.20) is done in linear theory however, the N -body evolution of the realised field will be fully non-linear. Thus, the non-linearly evolved matter field at $a = 1$ will have a power spectrum matching that from $T_{\delta_{\text{cdm+b}}}(a = 1, k)$ only at linear scales, which is precisely what we are after.

In writing (7.20), we have used the notation $T_{\delta_{\text{cdm+b}}}$ to mean the combined δ transfer function for cold dark matter and baryons. Explicitly, this is constructed from the individual transfer functions like

$$T_{\delta_{\text{cdm+b}}} = \frac{\bar{\rho}_{\text{cdm}} T_{\delta_{\text{cdm}}} + \bar{\rho}_b T_{\delta_b}}{\bar{\rho}_{\text{cdm}} + \bar{\rho}_b}, \quad (7.21)$$

where the mean densities are needed as the additive quantity is that of energy density ρ , or equivalently energy density perturbations $\delta\rho = \bar{\rho}\delta \propto \bar{\rho}T_\delta$.

We can now use the back-scaled transfer function of (7.20) to perform the realisation of the particle displacement field $\psi_n(a_i)$ (7.14), from which we can readily obtain particle positions (7.12). To generate matching momenta we use the technique of (7.18), where these are similarly obtained directly from $\psi_n(a_i)$.

Though the back-scaling technique works well in practice, it has two principle drawbacks. First, the realisation of the modified transfer function (7.20) effectively makes the particle distribution fictitious, in the sense that it does not match the actual particle distribution. The only exception is at the time $a = 1$, where the fictitious particle distribution match up with the physical one. To obtain correct results at any other time, one then has to redo the simulation, starting from a transfer function back-scaled from another time than $a = 1$. As we are usually concerned just with the present-day predictions, this is not a major issue in practice.

A more severe drawback of the back-scaling technique is the assumption that $D(a)$ correctly describes the linear matter evolution. As described in e.g. subsection 7.2.2, $D(a)$ fails to accurately predict the matter growth in cosmologies with massive neutrinos. For realistic high-precision simulations then, the back-scaling approach is not appropriate.

7.3.2 Relativistic Perturbations and the N -body Gauge

As described in the beginning of this section, a strategy seemingly more straightforward than that of back-scaling is to just realise the true transfer function at the initial time and take the gravitational contributions from the relativistic species directly into account throughout the simulation.

Gravitational Potential of Relativistic Species

Assuming light neutrinos, both these and photons remain linear throughout cosmic history, and so we can simply realise the $\delta\rho_{\gamma+\nu}(a, \mathbf{x})$ field from the linear photon and neutrino transfer functions using (7.2). The PM method, described for fluid components in section 6.4, can then be used to compute the gravitational potential contribution from these relativistic perturbations. In fact we can skip some steps, realising this potential directly using (5.16). In the syntax of (7.2), we can write this potential realisation as

$$\varphi_{\text{GR},n}(a) = -4\pi Ga^2 \mathcal{F}_{\text{scalar}}^{-1} \left[\frac{T_{\delta\rho_{\text{GR}}}(a, k)}{k^2} \zeta(k) \mathcal{R}_h \right], \quad (7.22)$$

where the ‘GR’ subscript remind us that we think of this potential as that generated by the “relativistic” species, which we for now think of as photons and neutrinos. The equation (7.22) itself is however completely general*.

The syntax used for the transfer function in (7.22) generalises that used in (7.21), where now $T_{\delta\rho} = T_{\delta\bar{\rho}} = \bar{\rho}T_\delta$. Written out in full in the case of photons and neutrinos, the transfer function of (7.22) would then be $T_{\delta\rho_{\text{GR}}} = T_{\delta\rho_{\gamma+\nu}} = (\bar{\rho}_\gamma + \bar{\rho}_\nu)T_{\delta_{\gamma+\nu}} = \bar{\rho}_\gamma T_{\delta_\gamma} + \bar{\rho}_\nu T_{\delta_\nu}$, where the neutrino term could be written out further to explicitly account for all three neutrino species.

*For a different use case of this general framework for including gravitational interactions from linear species, I would like to mention the Bachelor’s project of Katrine Alice Glasscock, which I co-supervised together with Steen Hannestad during the first half of 2018. Here she used linear realisation (7.22) to study the effects of dark energy perturbations on the non-linear matter evolution, using the CONCEPT code.

The *N*-body Gauge

Realising the relativistic potential (7.22) in each time step, finite differencing it to obtain a force and applying this force to the particles, we can successfully take the relativistic species into account. However, since GR is a gauge theory, the value of the transfer functions depends upon our gauge choice. Usually in *N*-body simulations, the particles are initialised from transfer functions in some gauge (typically synchronous), after which the notion of a gauge is completely forgotten about in the Newtonian framework. To apply the linear but general relativistic potential (7.22) to the particles throughout time, we now have to choose some gauge in which to perform the time evolution. Such a gauge choice then augments the *N*-body simulation with a dynamical space-time, in turn perturbing the Newtonian physics, the assumption of which the entire simulation code is built upon.

One solution to this problem arising from the collision of the Newtonian and relativistic worlds would be to rewrite the entire *N*-body code in a general relativistic framework, as is the approach of the GEVOLUTION code [36]. A more subtle and less radical solution is presented in [64, 65], where they demonstrate the existence of a gauge in which the Newtonian evolution of our *N*-body system is unperturbed to first order, in the matter-only case. This miracle gauge is the *N*-body gauge, which demonstrates that for normal matter-only Newtonian *N*-body simulations, the missing GR effects can be absorbed into the gauge. Should one wish to express the final results in some more conventional gauge, a simple gauge transformation can then be applied, effectively adding in GR effects after the fact.

Settling for the *N*-body gauge then seems like the natural choice. The CLASS code can operate and produce transfer functions in either the synchronous or the conformal Newtonian gauge [30], but not the *N*-body gauge. The gauge transformations relating these gauges are [66]

$$\delta_\alpha^{Nb}(a, k) = \delta_\alpha^{s/c}(a, k) + 3aH(1 + w_\alpha) \frac{\theta_{\text{tot}}^{s/c}(a, k)}{k^2}, \quad (7.23)$$

$$\theta_\alpha^{Nb}(a, k) = \theta_\alpha^{s/c}(a, k) - \frac{3}{k^2} \partial_\tau [aH\theta_{\text{tot}}^{s/c}(a, k)] + \begin{cases} h'(a, k)/2 & s \\ -3\phi'(a, k) & c, \end{cases} \quad (7.24)$$

where ‘s’, ‘c’ and ‘Nb’ respectively indicate synchronous, conformal Newtonian and *N*-body gauge, α labels the species and a prime denotes differentiation with respect to conformal time τ . The h and ϕ are the synchronous and conformal Newtonian metric perturbations, respectively,

the latter of which is defined in (3.26). The total velocity divergence is given by $\theta_{\text{tot}} = [\sum_{\alpha} \bar{\rho}_{\alpha}(1+w_{\alpha})]^{-1} \sum_{\alpha} \bar{\rho}_{\alpha}(1+w_{\alpha})\theta_{\alpha}$. Though the transfer functions for θ_{tot} , h and ψ are not by default available as outputs in CLASS, they are computed internally and so can readily be added by trivial additions to the source code. Note that the higher-order pressure and shear stress are gauge-independent to first order, and so we do not explicitly transform these to N -body gauge.

The N -body Gauge Metric Perturbation

With N -body gauge transfer functions in accordance with (7.23), we can now realise the relativistic potential (7.22) similarly in N -body gauge, which we can then apply to the Newtonian particles, similarly initialised from N -body gauge δ and θ transfer functions. As mentioned previously, the N -body gauge has a flat spacetime in the case of a matter-only universe, which was why this gauge served our Newtonian matter simulation well. As we now do have (small) perturbations in the relativistic species, non-Newtonian effects introduce themselves even in N -body gauge.

As demonstrated in [12, 64], the metric perturbations in N -body gauge can be taken into account by a new potential contribution γ , so that the total relativistic potential felt by the particles are that from the relativistic species (7.22) minus γ . This potential takes the form

$$\begin{aligned}\gamma(a, k) &= -\frac{1}{k^2}(\partial_{\tau} + aH)[H_{\text{T}}^{N\text{b}}(a, k)]' + \frac{8\pi Ga^2}{k^2}\Sigma_{\text{tot}}(a, k) \\ &= -\frac{1}{k^2}(\partial_{\tau} + aH)[H_{\text{T}}^{N\text{b}}(a, k)]' + [\phi(a, k) - \psi(a, k)],\end{aligned}\quad (7.25)$$

where ϕ and ψ are the metric perturbations in conformal Newtonian gauge (3.26) and H_{T} is a perturbation in the N -body gauge metric, specifically the trace-free component of the spatial metric, as defined in [67]. To linear order in perturbations, the total shear stress $\Sigma_{\text{tot}} = (3/2)\sum_{\alpha} \bar{\rho}_{\alpha}(1+w_{\alpha})\sigma_{\alpha}$ and is gauge invariant, with the factor $3/2$ arising from the definition (3.38) together with (3.21). The last equality of (7.25) then follows from (3.46).

The metric perturbation H'_{T} is not readily available in CLASS, though

it can be constructed from available quantities* [21];

$$\begin{aligned} [H_T^{Nb}(a, k)]' = & \frac{3aH}{\bar{\rho}_{\text{tot}} + \bar{P}_{\text{tot}}} \left\{ -\delta P_{\text{tot}}^{\text{s/c}}(a, k) + \bar{P}'_{\text{tot}} \frac{\theta_{\text{tot}}^{\text{s/c}}(a, k)}{k^2} \right. \\ & \left. + \frac{2}{3} \Sigma_{\text{tot}}(a, k) \right\}. \end{aligned} \quad (7.26)$$

The CLASS installation that comes with CONCEPT includes $[H_T^{Nb}]'$ in its output, from which $\gamma(a, k)$ (7.25) can be constructed.

Using the Poisson equation (5.16) we can relate the γ potential to a fictitious density field

$$\begin{aligned} \delta\rho_{\text{metric}}(a, k) &= \frac{k^2}{4\pi Ga^2} \gamma(a, k) \\ &= -\frac{1}{4\pi Ga^2} \left\{ (\partial_\tau + aH)[H_T^{Nb}(a, k)]' \right. \\ &\quad \left. + k^2[\phi(a, k) - \psi(a, k)] \right\}, \end{aligned} \quad (7.27)$$

where the missing minus sign compared to (5.16) comes about due to γ being defined with the opposite sign of our other potentials. With (7.27), we can treat the metric as a species on equal footing with e.g. photons and neutrinos. Adding the metric to the set of relativistic species — the potential from which affect the N -body particles — then amounts to redefining $\delta\rho_{\text{GR}} = \delta\rho_\gamma + \delta\rho_\nu + \delta\rho_{\text{metric}}$, where the photon subscript γ should not be confused with the metric potential. With $T_{\delta\rho_\gamma}$ and $T_{\delta\rho_\nu}$ obtained directly from CLASS (after application of the N -body gauge transformation (7.23)) and $T_{\delta\rho_{\text{metric}}}$ constructed from other CLASS output via (7.27), the realised relativistic potential (7.22) now contains everything needed for the N -body particles to correctly follow the general relativistic N -body gauge solution at linear scales.

Figure 7.1 shows the potential contribution from each species. Though the matter components dominate, the contributions from the relativistic species amounts to several percent of the total at large scales. At early times, the three neutrinos are barely indistinguishable. Throughout time, the massless neutrino mimic the photons, whereas we clearly see a matter-like behaviour for ν_2 and ν_3 at the present. The contribution from the metric is seen to be comparable to that from the photons and neutrinos, at least at early times.

*The first term of (7.26) has the opposite sign of the arXiv version of [21]. The sign shown here is correct.

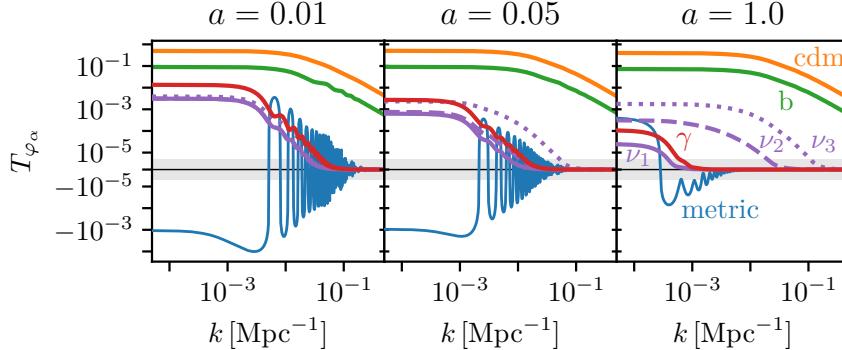


Figure 7.1 – Potential contributions from the various species at three different scale factors. The values plotted are the potential transfer functions $T_\varphi = -4\pi G a^2 k^{-2} T_{\delta\rho}$ in N -body gauge and in units of c^2 . The cosmology is as specified in table 2.1. The grey band indicates linear scaling of the vertical axis.

Due to the wildly oscillatory behaviour of the metric perturbations, the PM grid on which the relativistic potential $\varphi_{\text{GR}} = \varphi_\gamma + \varphi_\nu - \gamma$ is realised need to have much better resolution than were we to neglect the metric perturbations. The metric perturbations similarly oscillate rapidly in time, meaning that fine time stepping is needed to accurately resolve the resulting force. For N -body codes with global (and hence fine) time stepping like CONCEPT, this is not much of an issue. For codes using multi-level time stepping however, we might wish to only apply the linear force a relatively few number of times during the simulation. This is perfectly fine for all of the other slowly varying potential contributions, but not for the oscillatory metric contributions. One strategy to correct for this is to realise the average of the potential over the given time step.

In CONCEPT, T_δ is chosen as the canonical $\ell = 0$ transfer function in stead of e.g. $T_{\delta\rho} = \bar{\rho}T_\delta$ or $T_\varphi = -4\pi G a^2 k^{-2} T_{\delta\rho}$. We are thus in need of $T_{\delta_{\text{metric}}} = T_{\delta\rho_{\text{metric}}} / \bar{\rho}_{\text{metric}}$. As the fictitious metric species does not have a physical background density, we then have to decide on some $\bar{\rho}_{\text{metric}}(a)$ ourselves. Mathematically, the choice of $\bar{\rho}_{\text{metric}}(a)$ is completely irrelevant. Numerically however, I have found that $\bar{\rho}_{\text{metric}}(a) \equiv \bar{\rho}_\gamma(a)$ works noticeably better than e.g. having $\bar{\rho}_{\text{metric}}(a)$ constant, due to spline interpolations over $\delta_{\text{metric}}(\log a, \log k)$. This comes from the fact that the envelope of the oscillating $\delta\rho_{\text{metric}}(a)$ follows $\delta\rho_\gamma(a)$ reasonably well, as can be seen from figure 7.1.

7.4 Non-linear Realisations

We now go beyond the linear realisations based purely on the linear transfer functions and the static random field \mathcal{R}_h . For the photon, massless neutrino and metric realisations of the previous section, linear realisation should be adequate throughout the simulation time span. Unless the neutrino mass is not very small, the late-time behaviour of neutrinos will be non-linear, and so this is the example we will keep in mind for this section, although the developed procedures are completely general.

Remembering back to chapter 6, our strategy for non-linear neutrino simulations was to evolve the energy density $\varrho_n(a)$ and momentum density $J_n^i(a)$ non-linearly using the continuity and Euler equation (6.1). The additional dependencies of the pressure $\mathcal{P}_n(a)$ and shear stress $(\zeta_j^i)_n(a)$ will instead be solved linearly, in our case by CLASS. We referred to this particular choice as $\ell_{\text{nl}} = 1$, meaning that the $\ell = 0$ and $\ell = 1$ moments (ϱ and J^i) of the Boltzmann hierarchy are solved non-linearly, while all higher moments are solved linearly.

7.4.1 Forward Fourier Transforms

We have previously defined the operations $\mathcal{F}_{\text{scalar}}^{-1}$ (7.2), $\mathcal{F}_{\text{vector}}^{-1}$ (7.9) and $\mathcal{F}_{\text{tensor}}^{-1}$ (7.11), which construct real space scalar, vector and rank-2 tensor grids from the linear transfer function and the random field. The corresponding forwards Fourier transform $\mathcal{F}_{\text{scalar}}$ was straight forwardly defined in (7.7). We now define the forward $\mathcal{F}_{\text{vector}}$ and $\mathcal{F}_{\text{tensor}}$ as well.

The numerical vector and tensor Fourier operators are slightly trickier to define due to their embedded \mathbf{k} dependent factors. To begin, we can perfectly well use $\mathcal{F}_{\text{scalar}}$ on the individual scalar fields making up $u_n^i(a)$ and $(\sigma_j^i)_n(a)$. From (7.7) we then have $u_h^i(a) = L^{3/2} \mathcal{F}_{\text{scalar}}[u_n^i(a)]$, holding separately for each component i . Multiplying (3.37) by k_i and summing over i , we can now define a grid of Fourier space divergences $\theta_h(a)$ in terms of $u_h^i(a)$. This comes out to

$$\begin{aligned}\theta_h(a) &= ik_i u_h^i(a) \\ &= L^{3/2} ik_i \mathcal{F}_{\text{scalar}}[u_n^i(a)] \\ &\equiv L^{3/2} \mathcal{F}_{\text{vector}}[u_n^i(a)],\end{aligned}\tag{7.28}$$

where the last line of (7.28) defines $\mathcal{F}_{\text{vector}}$ to be the k_i weighted sum of three scalar Fourier transforms. With $\theta_h(a) = L^{3/2} T_\theta(a, k) \zeta(k) \mathcal{R}_h$ we see from (7.9) that we indeed have $\mathcal{F}_{\text{vector}} \mathcal{F}_{\text{vector}}^{-1} = \mathcal{F}_{\text{vector}}^{-1} \mathcal{F}_{\text{vector}} = 1$.

Using (3.39), the equivalent relation for the scalar shear stress becomes

$$\begin{aligned}\sigma_{\mathbf{h}}(a) &= -\frac{k_i k^j}{k^2} (\sigma_j^i)_{\mathbf{h}}(a) \\ &= L^{3/2} \left[-\frac{k_i k^j}{k^2} \right] \mathcal{F}_{\text{scalar}}[(\sigma_j^i)_{\mathbf{n}}(a)] \\ &\equiv L^{3/2} \mathcal{F}_{\text{tensor}}[(\sigma_j^i)_{\mathbf{n}}(a)],\end{aligned}\quad (7.29)$$

where we again have $\mathcal{F}_{\text{tensor}} \mathcal{F}_{\text{tensor}}^{-1} = \mathcal{F}_{\text{tensor}}^{-1} \mathcal{F}_{\text{tensor}} = 1$.

Using (7.28) and (7.29) within a simulation that evolves both the velocity and shear stress non-linearly (corresponding to $\ell_{\text{nl}} > 1$), we can now obtain non-linear predictions for the scalar variables $\theta(a, k)$ and $\sigma(a, k)$. We can express these in terms of power spectra exactly analogous to (7.6), i.e.

$$\begin{aligned}P_{\theta}(a, k) &= L^{-3} \langle |\theta_{\mathbf{h}}(a)|^2 \rangle_V, \\ P_{\sigma}(a, k) &= L^{-3} \langle |\sigma_{\mathbf{h}}(a)|^2 \rangle_V.\end{aligned}\quad (7.30)$$

We have not yet mentioned the pressure. As this is a scalar, it obeys the same realisation equations as the energy density;

$$\delta P_{\mathbf{h}}(a) = L^{3/2} \mathcal{F}_{\text{scalar}}[\delta P_{\mathbf{n}}(a)], \quad (7.31)$$

$$P_{\delta P}(a, k) = L^{-3} \langle |\delta P_{\mathbf{h}}(a)|^2 \rangle_V, \quad (7.32)$$

following (7.7) and (7.30).

From (4.10) and (4.11) we have the relation $T_{\delta}^2(a, k) \zeta^2(k) = \delta^2(a, k) = P_{\delta}(a, k)$. With the generalised power spectra (7.30) and (7.32), the same relation holds for θ , δP and σ . While the left expression $T_{\delta}^2(a, k) \zeta^2(k)$ is purely linear, the right expression $P_{\delta}(a, k)$ may be non-linear if it results from a power spectrum measurement in a non-linear simulation. We could opt to split the power spectrum into two variables, one specifically linear and the other non-linear. Instead we choose to keep $P_{\delta}(a, k)$ as is and just always think about it as the non-linear power spectrum, unless explicitly stated otherwise. Thus, $T_{\delta}^2(a, k) \zeta^2(k) = P_{\delta}(a, k)$ is only really true at linear times. Similarly, the variable $\delta(a, k)$ may be taken as either its linear or non-linear version. For writing equations which may be interpreted either linearly or non-linearly, we adopt the agnostic $\delta(a, k)$ in place of the specifically linear $T_{\delta}(a, k)$ or non-linear $P_{\delta}(a, k)$.

7.4.2 Dynamic “Random” Field

The linear realisations presented in section 7.1 were based on the linear transfer function $T(a, k)$ together with a static random field \mathcal{R}_h . To develop non-linear realisations, we then have to compensate for both of these.

Equations (7.7), (7.28), (7.29) and (7.31) all describe what we might call inverse realisations. That is, they produce a 3D Fourier grid — e.g. $\delta_h(a)$ — from its 3D real space counterpart. If we write these Fourier grids in terms of the scalar quantity and the random field, $\delta_h(a) = L^{3/2}\delta(a, k)\mathcal{R}_h$ from e.g. (4.9), we can solve for the random field:

$$\mathcal{R}_h^y(a) = L^{-3/2} \frac{y_h(a)}{y(a, k)}, \quad y(a, k) = \begin{cases} \sqrt{P_y(a, k)} & \text{(linear)} \\ T_y(a, k)\zeta(k) & \text{(non-linear),} \end{cases}$$

$$y \in \{\delta, \theta, \delta P, \sigma, \dots\}, \quad (7.33)$$

where a separate expression for the random field is generated for each of the quantities y , and for each y we can choose between linear $y(a, k) = T_y(a, k)\zeta(k)$ or non-linear $y(a, k) = \sqrt{P_y(a, k)}$ behaviour

By construction, all the various random fields of (7.33) coincide at the initial time of the simulation, where they all equal the usual primordial random field; $\mathcal{R}_h^y(a = a_i) = \mathcal{R}_h$. If we now consider linear evolution of $y_h(a)$ and hence also of $y(a, k)$, we see that $\mathcal{R}_h^y(a)$ stays constant as the evolution of each mode depends only on $|\mathbf{k}|$. Contrary, when $y_h(a)$ represents a non-linearly evolved grid, $\mathcal{R}_h^y(a)$ changes over time. Particularly, non-Gaussianities developing in $y_h(a)$ cannot be divided out by the use of $T_y(a, k)$ or $\sqrt{P_y(a, k)}$.

With $y_h(a)$ a genuinely non-linear grid, we still have two options for $y(a, k)$ in (7.33). If we pick $y(a, k) = \sqrt{P_y(a, k)}$, i.e. compute $y(a, k)$ as the square root of the power spectrum of the non-linear grid $y_h(a)$, the ratio $y_h(a)/y(a, k)$ in (7.33) will have a shell-variance of unity, cancelling out the non-linear growth of any particular $|\mathbf{k}|$ mode. As only entire shells are constrained in this way, the 3D grid points of $\mathcal{R}_h^y(a)$ making up a given shell will still undergo non-linear evolution. Thus, choosing $y(a, k) = \sqrt{P_y(a, k)}$ results in a dynamic $\mathcal{R}_h^y(a)$ where the phases of the random numbers correctly track the non-linear evolution of y , while none of the mode growth is captured, preserving the shell-variance $\langle |\mathcal{R}_h^y(a)|^2 \rangle_V = 1$.

The other choice in (7.33), $y = T_y(a, k)\zeta(k)$, replaces the non-linear power spectrum in the above explanation with that of the linear power

spectrum. While still tracking the evolution of the underlying phases correctly, this choice does not cancel out any non-linear deviation from linear growth. Thus with $y = T_y(a, k)\zeta(k)$, the shell-variance of the random field is now allowed to evolve away from unity, $\langle |\mathcal{R}_h^y(a)|^2 \rangle_V \neq 1$. Specifically, clustering will tend to increase this variance. This then means that $\mathcal{R}_h^y(a)$ retain the non-linear evolution of y , or more specifically the non-linear evolution of y that exists on top of the linear evolution. This is why the $y = T_y(a, k)\zeta(k)$ is labelled as ‘non-linear’ in (7.33): It divides out only the linear evolution, leaving non-linear growth in $\mathcal{R}_h^y(a)$.

7.4.3 Injecting Non-linearity

With the dynamic “random” field (7.33) at hand, we have everything we need to create non-linear realisation schemes. With scalar, vector and rank-2 tensor realisations defined as in (7.2), (7.9) and (7.11), we simply substitute the primordial random field \mathcal{R}_h for one of its dynamical versions $\mathcal{R}_h^y(a)$ defined in (7.33). Additionally, we include a species index α' , $\mathcal{R}_h^y(a) \rightarrow \mathcal{R}_h^{y_{\alpha'}}(a)$, labelling the species from which the random field has been produced. Writing the possible non-linear realisations out for all of our four quantities, we have

$$\delta_{\alpha,n}(a) \approx L^{-3/2} \begin{cases} \mathcal{F}_{\text{scalar}}^{-1} \left[y_{\alpha',h}(a) \frac{T_{\delta_\alpha}(a, k)\zeta(k)}{\sqrt{P_{y_{\alpha'}}(a, k)}} \right] & (\text{linear}) \\ \mathcal{F}_{\text{scalar}}^{-1} \left[y_{\alpha',h}(a) \frac{T_{\delta_\alpha}(a, k)}{T_{y_{\alpha'}}(a, k)} \right] & (\text{non-linear}), \end{cases} \quad (7.34)$$

$$u_{\alpha,n}^i(a) \approx L^{-3/2} \begin{cases} \mathcal{F}_{\text{vector}}^{-1} \left[y_{\alpha',h}(a) \frac{T_{\theta_\alpha}(a, k)\zeta(k)}{\sqrt{P_{y_{\alpha'}}(a, k)}} \right] & (\text{linear}) \\ \mathcal{F}_{\text{vector}}^{-1} \left[y_{\alpha',h}(a) \frac{T_{\theta_\alpha}(a, k)}{T_{y_{\alpha'}}(a, k)} \right] & (\text{non-linear}), \end{cases} \quad (7.35)$$

$$\delta P_{\alpha,n}(a) \approx L^{-3/2} \begin{cases} \mathcal{F}_{\text{scalar}}^{-1} \left[y_{\alpha',h}(a) \frac{T_{\delta P_\alpha}(a, k)\zeta(k)}{\sqrt{P_{y_{\alpha'}}(a, k)}} \right] & (\text{linear}) \\ \mathcal{F}_{\text{scalar}}^{-1} \left[y_{\alpha',h}(a) \frac{T_{\delta P_\alpha}(a, k)}{T_{y_{\alpha'}}(a, k)} \right] & (\text{non-linear}), \end{cases} \quad (7.36)$$

$$(\sigma_j^i)_{\alpha,n}(a) \approx L^{-3/2} \begin{cases} \mathcal{F}_{\text{tensor}}^{-1} \left[y_{\alpha',h}(a) \frac{T_{\sigma_\alpha}(a,k)\zeta(k)}{\sqrt{P_{y_{\alpha'}}(a,k)}} \right] & (\text{linear}) \\ \mathcal{F}_{\text{tensor}}^{-1} \left[y_{\alpha',h}(a) \frac{T_{\sigma_\alpha}(a,k)}{T_{y_{\alpha'}}(a,k)} \right] & (\text{linear}), \end{cases} \quad (7.37)$$

where both the linear and non-linear option for $y(a,k)$ from (7.33) are shown. Note that we do not require $\alpha = \alpha'$ in general, so that the random field associated with some quantity of one species may be used for the non-linear realisation of some quantity of another species.

For non-linear realisation of $\delta_{\alpha,n}(a)$ to come up in practice, none of the fluid variables of species α has to be evolved non-linearly, which we may indicate as $\ell_{\text{nl}} = -1$. In order for $y_{\alpha',h}(a)$ to exist* we then must have $\alpha' \neq \alpha$. I have not yet tested this case, but it seems plausible that the realisations of the linear photon, neutrino and metric (the so-called ‘relativistic species’) perturbations of subsection 7.3.2 can be improved by letting these be realised using the correctly evolved random phases of the matter distribution† $\mathcal{R}_h^{\delta_{\text{cdm+b}}}(a)$, enabling e.g. the exact 3D positions of neutrino overdensities to match up with the corresponding positions of matter clusters. As we do not want to further inject the non-linear growth of matter into the relativistic species, we would then opt for the linear option of using $\sqrt{P_{\delta_{\text{cdm+b}}}(a,k)}$ in (7.34).

Though all of (7.35), (7.36) and (7.37) are implemented in CONCEPT, the focus has been on the case $\ell_{\text{nl}} = 1$, meaning that only δP and σ_j^i need to be non-linearly realised. Furthermore, only the possibility $y = \delta$ has been implemented, meaning that the inherited phases are always those of the energy density. Lastly, though I did tinker with the linear option of using the numerically measured power spectrum to divide out the non-linear growth, I ended up removing its implementation as defects‡ in the power spectrum too easily propagated through to the non-linear realisation. With this limited set-up, the specific non-linear

*We can of course always construct $y_{\alpha',h}(a)$ using linear realisation in the case that $y_{\alpha'}$ is not a non-linear variable. However this reduces any of the “non-linear” realisations back to their linear versions.

†Though the matter is treated as particles, the phases $\mathcal{R}_h^{\delta_{\text{cdm+b}}}$ and $\mathcal{R}_h^{\theta_{\text{cdm+b}}}$ can still be constructed using (7.33), with $\delta_{\text{cdm+b},h}$ and $\theta_{\text{cdm+b},h}$ produced using interpolation (5.22). Here, $\theta_{\text{cdm+b},h}$ can be constructed from the grid-interpolated particle velocities using (7.28).

‡I have since improved greatly on the implementation of the power spectrum computation, so that it now yields much smoother spectra. It would thus be interesting to reintroduce the ‘linear’ options of (7.34) through (7.37).

realisations of the pressure and shear stress become

$$\delta P_{\mathbf{n}}(a) \approx L^{-3/2} \mathcal{F}_{\text{scalar}}^{-1} \left[\delta_{\mathbf{h}}(a) \frac{T_{\delta P}(a, k)}{T_{\delta}(a, k)} \right], \quad (7.38)$$

$$(\sigma^i_j)_{\mathbf{n}}(a) \approx L^{-3/2} \mathcal{F}_{\text{tensor}}^{-1} \left[\delta_{\mathbf{h}}(a) \frac{T_{\sigma}(a, k)}{T_{\delta}(a, k)} \right]. \quad (7.39)$$

For the above approximations to be exact, the content in the brackets should of course be the fully non-linearly evolved $\delta P_{\mathbf{h}}(a)$ and $(\sigma^i_j)_{\mathbf{h}}(a)$, respectively. Thus, (7.39) and (7.38) are equivalent to

$$\delta P(a, \mathbf{k}) \approx \frac{T_{\delta P}(a, k)}{T_{\delta\rho}(a, k)} \delta\rho(a, \mathbf{k}), \quad (7.40)$$

$$\sigma^i_j(a, \mathbf{k}) \approx \frac{T_{\sigma}(a, k)}{T_{\delta\rho}(a, k)} \delta\rho(a, \mathbf{k}), \quad (7.41)$$

where we have swapped the grids for fields and multiplied the numerator and denominator on the right by $\bar{\rho}$, just to make the physics clearer. Both (7.40) and (7.41) are exact at linear times, since here $y(a, \mathbf{k}) \propto T_y(a, k)$. For (7.40) to hold at non-linear times, the ratio $T_{\delta P}(a, k)/T_{\delta\rho}(a, k) = \delta P(a, k)/\delta\rho(a, k)$ corresponding to the (squared) sound speed must be independent of non-linearity. That is, though neither $\delta P(a, k)$ nor $\delta\rho(a, k)$ is well described by linear theory, (7.40) amounts to the assumption that the sound speed is. Similarly, (7.41) amounts to the assumption that the ratio $\sigma(a, k)/\delta\rho(a, k)$ is well described by linear theory at non-linear times, though neither $\sigma(a, k)$ nor $\delta\rho(a, k)$ is.

A separate assumption of (7.40) and (7.41) is that the 3D structure of the pressure and shear, corresponding to $\mathcal{R}^{\delta P}(a, \mathbf{k})$ and $\mathcal{R}^{\sigma}(a, \mathbf{k})$, follow that of the energy density $\mathcal{R}^{\delta\rho}(a, \mathbf{k})$. This indeed seems very reasonable, and at least for the pressure the only sensible choice. From the appearance of the shear stress in the energy-momentum tensor (3.20), $T^i_j = P\delta^i_j + (\rho + P)(u^i u_j + \sigma^i_j)$, we might also suggest $\mathcal{R}^{\theta}(a, \mathbf{k})$ as a good choice for the realisation of σ^i_j . We can even imagine using the product $\mathcal{R}^{u^i}(a, \mathbf{k})\mathcal{R}^{u_j}(a, \mathbf{k})$, where we use each component of the velocity field rather than their combined divergence as in (7.28) to generate the random fields.

A last option which I have not pursued in practice is to model the non-linear evolution of a given quantity analytically. As an example, the non-linear pressure realisation might then be written as

$$\delta P_{\alpha, \mathbf{n}}(a) \approx L^{-3/2} \mathcal{F}_{\text{scalar}}^{-1} \left[y_{\alpha', \mathbf{h}}(a) \frac{T_{\delta P}(a, k)\zeta(k)s_{\delta P_{\alpha}}(a, k)}{\sqrt{P_{y_{\alpha'}}(a, k)}} \right], \quad (7.42)$$

which is just (7.36) with the linear option, except for the introduction of the new variable $s_{\delta P_\alpha}(a, k)$. As the non-linear power spectrum in (7.42) divides out all the non-linear growth of $y_{\alpha', \mathbf{h}}$, we will obtain the correct non-linear pressure from (7.42) if $T_{\delta P}(a, k)\zeta(k)s_{\delta P_\alpha}(a, k) = \sqrt{P_{y_\alpha}(a, k)}$, with $\sqrt{P_{y_\alpha}(a, k)}$ the square root of the unknown non-linear power spectrum. Since at linear times this is exactly given by $T_{\delta P}(a, k)\zeta(k)$, we have $s_{\delta P_\alpha}(a \ll 1, k) = 1$, or generalised to other quantities $s_{y_\alpha}(a \ll 1, k) = 1$. The role of $s_{y_\alpha}(a, k)$ is then that of a non-linear transfer function, capturing the growth of k modes on top of that from linear theory. With a good model for $s_{\delta P_\alpha}(a, k)$, the non-linear realisation scheme (7.42) will then result in a pressure grid of correct non-linear variance, though the phases within each k shell will still be locked to those of $y_{\alpha', \mathbf{h}}$.

In light of the newly introduced $s_y(a, k)$, we can reinterpret the approximations (7.40) and (7.41) or equivalently (7.38) and (7.39) to be

$$s_{\delta P}(a, k) \approx s_{\delta\rho}(a, k), \quad (7.43)$$

$$s_\sigma(a, k) \approx s_{\delta\rho}(a, k), \quad (7.44)$$

i.e. the deviation from linearity of the non-linearly evolved $\delta\rho(a, k)$ is inherited by both $\delta P(a, k)$ and $\sigma(a, k)$ (on top of the time evolved phases, similarly inherited from $\delta\rho(a, \mathbf{k})$).

Conserved Variables

Throughout this chapter we have discussed the realisations of δ_n , u_n^i , δP_n and $(\sigma_j^i)_n$, though it is really their ‘conserved’ counterparts ϱ_n , J_n^i , \mathcal{P}_n and $(\varsigma_j^i)_n$ that we are after. To relate the two, we might simply use the definitions (3.29), (3.31), (3.32) and (3.33) of the conserved quantities, leading to

$$\varrho_n = a^{3(1+\omega)} \bar{\rho} (1 + \delta_n), \quad (7.45)$$

$$J_n^i = a^{1-3\omega} (\varrho_n + \mathcal{P}_n) u_n^i, \quad (7.46)$$

$$\mathcal{P}_n = a^{3(1+\omega)} (w \bar{\rho} + \delta P_n), \quad (7.47)$$

$$(\varsigma_j^i)_n = (\varrho_n + \mathcal{P}_n) (\sigma_j^i)_n, \quad (7.48)$$

where we see that in order to construct the ‘compound’ variables J_n^i and $(\varsigma_j^i)_n$ we first have to realise ϱ_n and \mathcal{P}_n . However, since both $(\varrho_n + \mathcal{P}_n)$ and u_n^i in the case of J_n^i or $(\varrho_n + \mathcal{P}_n)$ and $(\sigma_j^i)_n$ in the case of $(\varsigma_j^i)_n$ contain a set of underlying phases, multiplying them together alters the phases in a non-trivial manner (convolution in Fourier space). On top of

this, J_n^i and $(\varsigma_j^i)_n$ now also receive an enhancement due to the non-linear growth of ϱ_n and possibly \mathcal{P}_n .

In all, the direct multiplications in (7.46) and (7.48) lead to further enhancement of non-linearities. If we decide that we do not want this, we can simply replace $(\varrho_n + \mathcal{P}_n)$ with the background value $(\bar{\varrho} + \bar{\mathcal{P}}) = a^{3(1+\omega)}\bar{\rho}(1+w)$, leading to

$$J_n^i = a^4 \bar{\rho}(1+w) u_n^i, \quad (7.49)$$

$$(\varsigma_j^i)_n = a^{3(1+\omega)} \bar{\rho}(1+w) (\sigma_j^i)_n. \quad (7.50)$$

Importantly, (7.49) and (7.50) are identical to (7.46) and (7.48) to linear order. Both of these options are implemented in CONCEPT. During the work on the paper of chapter 10, we generally found that the added non-linearity from (7.48) led to better results.

8 The CONCEPT Code

Throughout my PhD, CONCEPT (the ‘COsmological N -body CodE in PyThon’) has been the code in which I have implemented the linear and non-linear neutrino evolution, along with all of the necessary peripherals. As mentioned in the introductory section 1.4, the code is free and open-source and available on GitHub*. The code was originally written for my Master’s thesis [22]. This chapter serves as a listing of what has been implemented into CONCEPT during my PhD. The underlying physics and the numerical details of the methods will not be discussed here, as this is done in part I and in chapters 5 through 7 of part II, respectively.

The main result of a CONCEPT computation is that of power spectra, described theoretically in chapter 4 and numerically in chapter 7. Though several types of power spectra are described, only δ power spectra are implemented in CONCEPT. When running with multiple components, e.g. matter particles and a neutrino fluid, combined power spectra of multiple species can also be outputted. Besides power spectra, snapshots[†] (i.e. complete dumps of the 3D particle and/or fluid system) can also be outputted, from which a CONCEPT simulation can restart itself. Lastly, 2D projected and full 3D renders of the particles and fluids can be produced. Examples of the former can be seen on the cover of this thesis.

Today CONCEPT consists of 16×10^3 source lines of Python code, together with 8×10^3 lines of auxiliary code used to install, compile, run and test the main code, as well as an additional 13×10^3 lines of comments. The code base is a factor ~ 4 larger now than at the beginning of my PhD.

*<https://github.com/jmd-dk/concept>

[†]The default snapshot format is an HDF5 file containing all needed information. To interoperate with other N -body codes, it can also output to (and read from) the GADGET-2 ‘unformatted binary Fortran’ snapshot type, though here only particles are supported.

8.1 Software Aspects

In this section we briefly describe the software architecture of CONCEPT, with special emphasis on the prevalent use of Cython.

8.1.1 Cython

Though CONCEPT is written in Python, it achieves C-like performance thanks to Cython [23], which is both a Python → C transpiler and a programming language in its own right, aiming to be as close to the Python language as possible. The Cython language may be viewed as a very nearly complete superset of both C and Python, allowing for a very large “dynamic range” of the level of abstraction. This in turn allows for simultaneous optimised low-level scientific code and high-level dynamic control code.

For low-level operations (e.g. pointer arithmetic), usually one has to rely on features of Cython which introduces syntax which breaks pure Python compatibility, meaning that though the code reads mostly like Python, it can only be executed after transpilation to C code using Cython. Through my insistence of keeping everything pure Python compatible, CONCEPT ended up containing within it its own transpiler* which converts the pure Python source code of CONCEPT into optimised Cython code, which then further transpiles to optimal C code. Thus, CONCEPT is one of the very few codes (possibly even the only one) that enjoys the full advantage of the dynamic nature of Python as well as of the performance and low-level control offered by C.

The CONCEPT code can then be run in both pure Python mode and in compiled C mode. Whereas the former makes for easy introspection, efficient debugging and rapid development, the latter decreases the overall computation time of a typical simulation by ∼ 2 orders of magnitude.

8.1.2 Libraries and Installation

Besides the Python/Cython/C stack and their accommodating standard libraries, CONCEPT makes further heavy use of several other libraries, including MPI for the parallelisation framework, GSL for various numerical methods such as spline interpolation, FFTW for MPI-parallelised

*On top of this, my transpiler further lowers the barrier between Python and Cython by e.g. autogenerating Cython header files. It also performs quite a few optimisations through direct manipulation of the source code, like loop unswitching and caching of constant expressions, which can otherwise be difficult for the C compiler to infer from the often cryptic C code generated by Cython.

Fourier transforms and HDF5 for MPI-parallelised file reads and writes. Furthermore, the cosmology codes CLASS and GADGET are needed for linear input and testing, respectively.

To make all of this work together, various Python/Cython packages which wraps the above C/C++/Fortran libraries must be installed and properly linked. On top of this, CONCEPT itself makes use of quite a few Python packages, including the standard numerical ones like NumPy and SciPy. In addition, CONCEPT rely on many newly included features of the various libraries and packages, so that the entire software stack must be very up-to-date for CONCEPT to run.

The Installation Script

All in all, this makes CONCEPT extremely cumbersome to install. To fix this, a complete installation script is provided, which installs CONCEPT along with all of its dependencies, the dependencies of these dependencies, and so on right down to the Linux kernel headers and C/C++/Fortran compilers, should they be missing on the system. With this, CONCEPT should be trivially installable on any Linux system, and indeed is on any such system of which I have tested.

To give some idea of the scope of this installation script, I will mention that it consists of 4.4×10^3 lines of Bash code*, not counting comments. It is also highly flexible, allowing already installed components to be used if requested. In practice when working on a computer cluster, I always let the installation script install everything from scratch, except for MPI where it is better to use an existing installation, as otherwise specific optimisations such as InfiniBand might not be enabled.

The CONCEPT installer can be downloaded and invoked directly using the following command;

```
bash <(wget -O- https://raw.githubusercontent.com\
/jmd-dk/concept/v0.2.0/installer)
```

with version 0.2.0 the latest release version as of this writing.

8.2 Numerical Methods

Throughout chapters 5 through 7, various numerical methods and techniques for implementing N -body particles, fluid flows and realisations have been described. We mostly focused on the methods which have been

*As the only hard dependency, the installation script then requires Bash (version ≥ 3.0) or another Bash-compatible shell to be preinstalled on the system.

implemented into CONCEPT, but sometimes more advanced methods were also described, and some methods in CONCEPT are not implemented in full generality. This section then serves as a specification of exactly which methods are present in CONCEPT. It also contains a description of the global time step determination and the parallelisation strategy adopted.

Though not strictly part of the actual numerical methods, let me briefly describe the unit system implemented in CONCEPT. When solving physical problems using floating point numbers, one should choose to work in a unit system so that the typical sizes of these numbers are about unity, as operations on very small or very large numbers leads to larger relative errors. For a cosmological simulation code, a good system of units might then be something like $\{\text{Mpc}, \text{Gyr}, 10^{10} M_{\odot}\}$, which is the default unit system* of CONCEPT. Within CONCEPT a complete mapping between a large set of typical cosmological units exist, including physical constants like the speed of light. Each of the three base units can then be specified programmatically in the parameter file. This comes in very handy when doing comparisons to other codes with fixed unit systems. For example, one can obtain GADGET units by specifying the set of basic units as $\{\text{kpc } h^{-1}, \text{km s}^{-1}, 10^{10} M_{\odot} h^{-1}\}$, while PKDGRAV units are obtained by $\{\text{boxsize}, \sqrt{8\pi/3} H_0^{-1}, 3H_0^2/(8\pi G)\text{boxsize}^3\}$. The choice of units will also be reflected in outputs such as power spectra, why the units are always explicitly attached along with any output.

8.2.1 Gravity

The gravitational interaction between particles are implemented in three separate ways, using the PP, PM and P³M methods, all described in section 5.3. Of these, P³M is the method with simultaneous large spatial resolution and good performance. However, its implementation in CONCEPT is very sub-optimal, as the ‘domains’† are used for the grouping of the particles, effectively cutting down the number of operations relative to the PP method by the number of CPU cores squared, but retaining the $\mathcal{O}(N^2)$ scaling. In actuality then, the PM method is the only scalable gravitational method implemented.

For fluid components also, the PM method is the only implementation of gravity. The necessary generalisation of this method to simultaneously

*Note that CONCEPT has no need for units outside of the span of the basic units of length, time and mass.

†These refer to the partitioned chunks of the box volume, each of which is designated to a unique MPI process as described in subsection 8.2.4.

incorporate particle and fluid components was laid out in section 6.4. This generalised method reduces to the standard PM method in the case of particle-only or fluid-only simulations, and is what is implemented in CONCEPT.

Throughout this thesis, the particle mass has consistently been written as m_n , allowing each particle n to have a mass separate from the others. In CONCEPT, all particles within a component shares the same mass. Usually we run simulations where matter (cold dark matter and baryons) is treated as a single component, but CONCEPT is fully capable of distributing the total mass of matter into any number of components, where now each component are allowed to have its own particle mass and number.

Generalised Interaction Framework

During the first half of 2017, together with Steen Hannestad I co-supervised Kathrine Bundgaard Henriksen on her Bachelor's project on self-interacting dark matter. As part of this project, Kathrine implemented the self-interaction into CONCEPT. In order to make such an implementation accessible, I refactored the PP and PM methods into a combined framework of interactions, in which the gravitational part was extracted from the underlying strategies adopted in these methods. As I knew that I someday would like to rewrite the P³M method, I did not care to refactor this into this new unified framework*. A new force can then be built from existing primitives, with all of the technical parallelisation aspects taken care of automatically.

Though the complete self-interaction of Kathrine has not become part of CONCEPT, this generalised interaction framework has. As mentioned in the introductory section 1.4, I plan to soon reimplement the P³M method, which will then become part of this framework.

Another benefit of this modular framework is that each force can be enabled/disabled for each individual component. Furthermore, two components interacting via some force may want to use different methods for this same force. This would be the case for simulations including both matter particles and a neutrino fluid, in the case where the particles use P³M amongst themselves, but has to interact with the neutrino fluid using PM. The interaction framework is general enough to handle such asymmetries.

*Hence the badly named `gravity_old.py` in the CONCEPT source tree.

8.2.2 Time Integration

In CONCEPT, particles are evolved using the Leapfrog integration scheme of subsection 5.2.1, with a global time step Δt . For fluids, the time integration is performed by the much more involved MacCormack (MC) or Kurganov-Tadmor (KT) methods, described in section 6.2 and 6.3, respectively.

To make the MC method positivity preserving (while also helping to stabilise the method in general) I found that it had to be augmented with numerical viscosity at grid points with an energy density close to zero. The resulting “vacuum correction” scheme of mine is described in subsection 6.2.2.

Though the KT method is much more sophisticated than the MC method, I have found that it does not lead to better results in practice, in the case of a massive neutrino fluid. In fact, the numerical viscosity obtained with the KT method is much more than that of the MC method, as seen in figure 6.1.

In an attempt to lessen the numerical viscosity of the KT method, I have additionally implemented an improved version of the same scheme [68], which uses more precise estimation of the local propagation velocities at cell interfaces. In practice, this more complicated prescription led to minute differences, and so only the basic KT method is kept in CONCEPT.

Time Step Limiters

The particle Leapfrog and the fluid MC or KT method share the same time step size Δt , which is then the same for all particles and fluid cells. We have seen how the Courant condition (6.9) for fluids limits the maximum size of Δt , above which the numerical fluid dynamics become unstable. However, the Courant condition (6.9) considers only the continuity equation. Here, u should be the maximum peculiar velocity of any fluid elements. We can replace this with the maximum of the right-hand-side of (6.18) times a , where (6.18) is an estimate of the local physical velocity. We can then write the Courant condition as

$$\Delta t < \min_{\substack{\text{fluid} \\ \alpha \in \text{components}}} \frac{\Delta x_{\alpha,m}}{\sqrt{3}} \left[a^{3w_\alpha - 2} \max_n \frac{|J_{\alpha,n}|}{\varrho_{\alpha,n} + \mathcal{P}_{\alpha,n}} + \frac{\sqrt{w_\alpha}}{a} \right]^{-1} \quad (\text{fluids}), \quad (8.1)$$

where n runs over all grid points and $\sqrt{w_\alpha}$ is the physical sound speed, approximated as a spatial constant. The minimisation over all fluid

species α means that the fluid with the highest propagation speed (compared to its grid spacing $\Delta x_{\alpha,m}$) sets Δt for all fluids.

The Courant condition (8.1) is extended to particle components by setting $w = 0$ and replacing the peculiar velocity (first term in the bracket) with $|\mathbf{u}_n| = a^{-1}m^{-1}|\mathbf{q}_n|$. Thus,

$$\Delta t < \min_{\substack{\text{particle} \\ \alpha \in \text{components}}} \frac{a\Delta x_\varphi}{\sqrt{3}m} \left[\max_n |\mathbf{q}_n| \right]^{-1} \quad (\text{particles}), \quad (8.2)$$

where now the grid spacing is taken from the potential grid. This should just be some distance specifying the smallest scale at which the particles operate. When using the PM method for gravity, Δx_φ is then appropriate. When using the PP or P³M method, we should replace Δx_φ with the comoving softening length ϵ described in subsection 5.3.1.

In addition to the fluid Courant condition (8.1) and what we might call the particle Courant condition (8.2), four further time step limiters are defined:

$$\Delta t < \begin{cases} (G\bar{\rho}_m)^{-1/2} & (\text{gravitational dynamical time scale}), \\ H^{-1} & (\text{Hubble time}), \\ t_f - t_i & (\text{simulation time span}), \\ \min_\alpha \dot{w}_\alpha^{-1} & (\text{time scale of non-relativistic transition}). \end{cases} \quad (8.3)$$

The first one restricts the time scale based on the physics of gravitation, rather than on the numerical resolution like (8.1) and (8.2). Such physics based time step limiters are helpful in odd cases of completely or very nearly static particles/fluids. The next two states that the time step size should be below the Hubble time (at the time of the given time step) and the entire simulation time span, with t_i and t_f the initial and final time, respectively. The first guarantees that the scale factor a does not change much over the time step, while the second ensures that the entire simulation consists of multiple time steps. The last time step limiter of (8.3) only comes into effect when having species with time varying equations of state, such as massive neutrinos. The condition then states that the equation of state should not change much over the time step, which is needed for a gradual transition from the relativistic to the non-relativistic regime.

In actuality, each of the time step limiters (8.1), (8.2) and (8.3) have individual constant factors multiplied on their right-hand-side, each of which is some small fraction. In CONCEPT these factors have been

manually tuned to ensure convergence over the range of simulations considered.

Every eighth time step, Δt is allowed to increase slightly, as long as all of (8.1), (8.2) and (8.3) are satisfied. Here the number 8 is chosen in order to minimise anisotropic defects induced by the MacCormack method as described in section 6.2.

8.2.3 Realisations

A high-level description of linear scalar, vector and rank-2 tensor realisations of fluid variables was described in section 4.3. Chapter 7 further elaborated on the numerical implementations, while also generalising the linear realisations to particles and introduced various non-linear fluid realisation techniques.

In CONCEPT, many of these options for realisation are implemented, though not all. For linear realisations, the primordial random grid \mathcal{R}_h is generated as described in subsection 7.1.1, i.e. the random numbers are drawn “in shells” so that the cosmic variance in the power spectrum at a given k is independent of the grid size (for fixed box size). Particles may only be realised linearly, i.e. using this primordial \mathcal{R}_h . The Zel’dovich approximation of subsection 7.2.1 is implemented, where particle positions are generated from the displacement field ψ_n . The particle momenta can either be constructed directly from the θ transfer function, or alternatively inferred from ψ_n using the Newtonian matter-only approximation (7.18).

Regarding the late-time normalisation of the power spectrum discussed in section 7.3, CONCEPT can be run either with or without back-scaling, specified for each component individually. When running without back-scaling, correct large-scale behaviour of matter must be achieved by direct interaction with linearly realised photon, neutrino and metric perturbations, as described in subsection 7.3.2. For such linear realisations, the $\delta\rho$ transfer function is averaged over the time step according to

$$\langle T_{\delta\rho}(k) \rangle_t^{t+\Delta t} = \left[\int_t^{t+\Delta t} a^2(t') dt' \right]^{-1} \int_t^{t+\Delta t} a^2(t') T_{\delta\rho}(t', k) dt', \quad (8.4)$$

where the a^2 weight is needed since what we really want to average over is the gravitational potential. It is then really $\langle T_{\delta\rho}(k) \rangle_t^{t+\Delta t}$ that is used in place of $T_{\delta\rho}(a(t), k)$ in e.g. (7.22). For slowly varying transfer

functions this does not make much of a difference, but it is important for the rapidly oscillating metric perturbations.

By default CONCEPT runs CLASS in synchronous gauge and transforms the resultant transfer functions to N -body gauge, though one can specify some other gauge to use (must be one of N -body, synchronous or conformal Newtonian).

Subsection 7.4.3 lays out several non-mutually exclusive possibilities for doing non-linear realisations of fluid variables. The only options implemented in CONCEPT is that of (7.38) and (7.39), where the pressure and shear stress of some species are realised using the non-linearly evolved energy density of the same species. Also, the scaling from energy density to respectively pressure and shear stress is done via the ratio of linear transfer functions, as opposed to using the power spectrum of the non-linearly evolved energy density.

Though we have focused on the case of $\ell_{\text{nl}} = 1$ during my PhD, meaning that both the energy density ϱ and the momentum density J^i are evolved non-linearly, non-linear realisation is also implemented fully for J^i , meaning that CONCEPT can also be run in $\ell_{\text{nl}} = 0$ mode.

The ‘compound’ fluid variables J^i and σ_j^i can be realised either from the real-space product of two 3D grids as in (7.46) and (7.48), or they might alternatively be realised just from the realised u^i and σ_j^i multiplied with background variables, as in (7.49) and (7.50).

As an alternative to using CLASS to bridge together the non-linear and linear Boltzmann hierarchy at $\ell = \ell_{\text{nl}}$, simple truncation of the non-linear hierarchy is also implemented, corresponding to disregarding fluid variables with $\ell > \ell_{\text{nl}}$ completely.

A general framework for optional approximations is built into CONCEPT, though as of now the only approximation implemented is that of $P = w\rho$. With $\ell_{\text{nl}} = 1$ and closing the non-linear hierarchy using truncation rather than linear theory input, we can obtain the perfect fluid approximation $P = w\rho \neq 0$, $\sigma_j^i = 0$ by enabling this approximation.

8.2.4 Parallelisation

We shall now briefly discuss the parallelisation strategy used in CONCEPT. The box volume is partitioned into a number of equally shaped cuboids, one for each MPI process, which in turn is mapped to a single physical CPU core. We refer to these subvolumes as ‘domains’. All of the cubic grids (including the fluid grids, the particle displacement grid and the potential grids) are then distributed according to these domains. Similarly, particles are distributed according to which domain they

currently reside in, meaning that they may be exchanged between MPI processes when crossing domain boundaries.

The cuboidal domains are chosen such that they have the largest ratio of volume to surface area possible, given the number of MPI processes. For cubic domains, we then need a cubic number of processes. This generally leads to optimal performance as it minimises particle exchanges. In this respect, the worst performance is gained for a prime number of processes, where the domains are forced to take the form of ‘slabs’, meaning that they span the simulation box along two of the three dimensions.

For fluids, this simple optimal cuboidal partitioning scheme leads to optimal performance throughout time, as the minimisation of domain boundaries imply minimisation of neighbouring grid cells belonging to separate processes. Information about neighbour cells are of course needed for the spatial derivatives during the non-linear fluid evolution. In practice, each local grid in CONCEPT is constructed with a ‘ghost layer’ around it (with a thickness of two grid cells), storing a local copy of the neighbouring cells belonging to a different process.

For particles, having a static partitioning of space is not optimal, as clustering inherently leaves some domains with a surplus of particles relative to others, serving to skew the CPU workload balance.

As CONCEPT uses FFTW for its 3D FFTs, it is subject to the memory layout criterion imposed by FFTW. Here, the volumetric partitioning has to be in the form of the aforementioned ‘slabs’, meaning that before a global FFT can be carried out, the processes must redistribute the particles and/or fluids according to this slab decomposition, which in general is an expensive operation. Were we to choose the slab decomposition as the main domain decomposition as well, we could avoid this transformation of the decomposition. It is unclear whether this would lead to an increase in the performance in actual simulations.

Muti-node CLASS Computations

The details of the integration of CLASS into CONCEPT is described in the next section. Here we note that CLASS is OpenMP parallelised over the k modes, meaning that a given k mode is solved for by a single CPU core. Also, as CLASS is not MPI parallelised, it cannot be run across multiple compute nodes. For usual CLASS runs this is no issue at all. In order to produce smooth and accurate transfer functions for the pressure and shear of massive neutrinos, as well as for the N -body gauge metric perturbation $[H_T^{Nb}]'$ given in (7.26), we found that we needed to increase

massive neutrino precision parameters a lot, as described in the papers of chapter 9 and 10. A single CLASS computation could then span several days, hence the need to distribute the CLASS computation across multiple compute nodes (effectively just increasing the number of CPU cores).

To implement this, whenever a CLASS computation is to be made from within CONCEPT, the code effectively switches from MPI to OpenMP mode, with one ‘node master’ MPI process calling CLASS on each compute node, which then spawns off threads which are assigned to the remaining MPI processes on each compute node, one thread per CPU core. When all CLASS runs on all compute nodes are completed, the results are gathered by the global master MPI process, i.e. send to a single compute node, whereafter the usual MPI mode of CONCEPT can resume.

8.3 Integration with CLASS

One of the major strengths of CONCEPT compared to other cosmological N -body codes is its complete integration with CLASS, achieved through the ‘classy’ Python wrapper.

As described in the previous section, some of the CLASS computations needed may take a very long time, and so the results of all CLASS computations are automatically dumped to HDF5 files, storing the background and perturbations of all the species and variables needed for the given CONCEPT simulation, together with a specification of the parameters fed to CLASS. The filename of these HDF5s are generated by hashing the (sorted) set of CLASS parameters, and so each CLASS run is uniquely mapped to a filename. Over time, a large library of CLASS results is then generated by CONCEPT, which will be reused whenever the same CLASS computation is requested in the future.

The CLASS background and perturbations are stored in CONCEPT using GSL cubic spline objects, with the splines being over $\log a$ and $\log k$. The CLASS transfer functions are further detrended using a power law before they are splined, yielding excellent spline results. These are then converted to N -body gauge on request.

Quite a few small modifications to the CLASS source code has been made, mostly just to construct and/or provide access to additional transfer functions like $[H_T^{Nb}]'$.

As the needs from the CLASS code grew over the course of my PhD, I also managed to discover a couple of bugs in the source code, which have been pointed out to the authors. A critical bug in the Runge-Kutta integrator still remains at the time of this writing. When installing

CONCEPT using the installation script, CLASS will be installed and patched with all the necessary modifications. I also believe that the adoption of Python 3 by CLASS to a large extent was fuelled by the need of CONCEPT for this support.

8.3.1 The CONCEPT CLASS Utility

For the project of implementing linear species into the PKDGRAV code, fully described in chapter 9, I chose not to marry PKDGRAV to CLASS in a similar fashion as I had done in the case of CONCEPT, as this would be much too time consuming. As CONCEPT already had all of the necessary facilities, including a native multi-node way of executing CLASS, caching results to disk and processing the results (including the N -body gauge transformation), I instead opted to extend CONCEPT with a side utility capable of storing processed CLASS results in a neatly organised HDF5 file. This utility was simply dubbed the ‘CLASS utility’ when referred to from within CONCEPT, though from the outside world it should be called the ‘CONCEPT CLASS utility’ to avoid confusion.

The key difference between the contents of the automatically cached HDF5s and the HDF5s produced by the CLASS utility is this: The cached HDF5s simply contain the raw results from CLASS, meaning that each k mode exist in isolation. The HDF5s of the CLASS utility produces a single 2D $\{a, k\}$ grid on which all perturbations are tabulated, making further interpolations much easier. The range and number of a values can also be specified, in addition to the gauge (N -body, synchronous or conformal Newtonian). The range of k is of course also specifiable, but that is also true for the raw HDF5s.

Additionally, which species to save to the processed HDF5 file can be specified, along with an $\ell_{\max, \text{HDF5}}$ per species indicating which moments to include, i.e. $\ell_{\max, \text{HDF5}} = 0 \Rightarrow \{\delta\}$, $\ell_{\max, \text{HDF5}} = 1 \Rightarrow \{\delta, \theta\}$ or $\ell_{\max, \text{HDF5}} = 2 \Rightarrow \{\delta, \theta, \delta P, \sigma\}$. Another difference between the raw and processed HDF5 is then that the processed HDF5 contains δP , whereas the raw HDF5 contains $\delta P/\delta\rho$ as this is what CLASS provides.

Species can also be “added together” so that e.g. the combined δ and θ transfer functions for cold dark matter and baryons are provided, rather than the individual transfer functions for each species, letting CONCEPT carry out the weighted sum at interpolated points.

Besides the CLASS data, the processed HDF5s also contain a specification of the CLASS parameters used as well as of the unit system employed by CONCEPT at the time of running the CLASS utility (the dynamic system of units in CONCEPT is described in section 8.2). This

is important as the background quantities and transfer functions will be stored in these units. Having a mutable unit system then allows users of the CLASS utility to specify whatever units they want, and so they do not have to worry about any manual unit conversion.

In all, the CONCEPT CLASS utility seeks to provide the best possible presentation of CLASS perturbations (and background quantities) for use with large cosmological simulations, making the interfacing of other simulation codes with CLASS trivial by essentially using CONCEPT as a wrapper around CLASS.

The CLASS utility is the latest addition to the collection of such side utilities of CONCEPT, of which there are presently 8. Honourable mentions include power spectra measurements directly from a snapshot*, snapshot introspection and displaying coloured animations of 2D renders directly in the terminal.

*The snapshot may either be a CONCEPT or a GADGET snapshot. The utilities may also be run on entire directories of snapshots.

Part III

Linear and Non-linear

Neutrinos

9 Linear Neutrinos

This chapter contains the paper ‘Fully relativistic treatment of light neutrinos in N -body simulations’, which I co-authored together with Thomas Tram, Jacob Brandbyge and Steen Hannestad. It was put on arXiv and submitted to JCAP in early November 2018. We are currently awaiting response from the referee.

I have carried out slight syntactic modifications to the paper in order for its notation to blend more seamlessly with that used in this thesis. The most noticeable difference not adapted for is the use of an overdot as differentiation with respect to conformal time, whereas in the main part of this thesis an overdot refers to differentiation with respect to cosmic time. Also, a sign error has been corrected in equations (9.11) and (9.13).

In the paper we demonstrate how linear species — including photons, massive but light neutrinos and the metric itself — can successfully be added to conventional N -body simulations using the N -body gauge framework of [64, 65], described in subsection 7.3.2. This results in the correct general relativistic large-scale behaviour without the use of back-scaling (see subsection 7.3.1).

To demonstrate the method, it was implemented in both CONCEPT and PKDGRAV. I wrote both of these implementations simultaneously, with the majority of the work carried out during my stay at the Institute for Computational Science at the University of Zürich, May and June of 2018. While I did the CONCEPT implementation by myself, the PKDGRAV implementation has received inputs from several other people, though the primary CLASS interfacing was carried out by me. This was done using the CONCEPT CLASS utility as a bridge between PKDGRAV and CLASS, as described in subsection 8.3.1.

Below I list the extensions to PKDGRAV itself that I have implemented in order for it to be able to run with the linear species.

- Read in and store the CLASS background and N -body gauge transfer functions from an HDF5 file produced with the CONCEPT CLASS utility.

- Replace the existing internal Friedmann solver with the tabulated CLASS background.
- Replace the existing back-scaled and σ_8 dependent method for particle initial condition generation with direct realisation based on the CLASS $\delta_{\text{cdm}+\text{b}}$ and $\theta_{\text{cdm}+\text{b}}$ transfer functions, using the Zel'dovich approximation.
- Realise the total density perturbation field from all linear species on a grid. The density perturbations are averaged over the time step according to (8.4).

Together, these items extend PKDGRAV with a ‘CLASS mode’. When running PKDGRAV, one can now choose whether to run it in its original mode or in this new CLASS mode. I could not have carried out the above without much help from the main authors of PKDGRAV; Joachim Stadel and Douglas Potter. Furthermore, the implementation of the grid on which to realise the density field from the linear species, as well as the PM technique used to compute forces on the particles from this grid, was implemented by Hugues de Laroussilhe.

Paper:

Fully Relativistic Treatment of Light Neutrinos in N -body Simulations

Thomas Tram^a, Jacob Brandbyge^b, Jeppe Dakin^b,
Steen Hannestad^b

^aAarhus Institute of Advanced Studies (AIAS), Aarhus University,
DK-8000 Aarhus C, Denmark

^bDepartment of Physics and Astronomy, Aarhus University,
DK-8000 Aarhus C, Denmark

E-mail: thomas.tram@aias.au.dk, jacobb@phys.au.dk, dakin@phys.au.dk,
sth@phys.au.dk

Abstract. Cosmological N -body simulations are typically purely run with particles using Newtonian equations of motion. However, such simulations can be made fully consistent with general relativity using a well-defined prescription. Here, we extend the formalism previously developed for Λ CDM cosmologies with massless neutrinos to include the effects of massive, but light neutrinos. We have implemented the method in two different N -body codes, CONCEPT and PKDGRAV, and demonstrate that they produce consistent results. We furthermore show that we can recover all appropriate limits, including the full GR solution in linear perturbation theory at the per mille level of precision.

Contents

9.1	Introduction	152
9.2	Method and Implementation	153
9.3	Numerical Set-up and Results	156
9.3.1	Main Results	157
9.3.2	Comparison with PKDGRAV	160
9.4	Discussion	160
9.A	Computing γ	162

9.1 Introduction

In the coming few years, new, large galaxy surveys such as those from LSST [1] and Euclid [2] will provide extremely precise measurements of the large scale structure of our Universe. This in turn requires numerical simulations of structure formation to be accurate at the sub-percent level over a wide range of scales.

One important ingredient in this quest is to include massive neutrinos which are known to make up at least 0.1 % of the total energy density at present. Even at this lower limit the inclusion of neutrinos changes the matter power spectrum at the 3–4 % level, substantially more than the required precision of these surveys.

Over the past decade, a substantial effort has been devoted to the inclusion of massive neutrinos in N -body simulations. One approach is to use a particle representation of the full neutrino distribution function (e.g. [3–12]). This, however, is very numerically challenging because of the large number of particles needed to properly follow the neutrino distribution function. Another scheme assumes that neutrino perturbations remain linear [13–15]. A simple scheme which is known to work well for small neutrino masses is to use the linear neutrino density field calculated by realising the linear neutrino transfer function on a grid [13]. An improvement on this is to solve the linear theory neutrino equations, but use the full non-linear gravitational potential calculated in the simulation [14, 15]. However, in both cases this scheme only works for relatively small neutrino masses where neutrino perturbations remain linear at all times. Finally, there are hybrid schemes coupling the two approaches [16], as well as approaches based on other approximate solutions (e.g. [17, 18])

Another effect which must be taken into account comes from the inclusion of general relativistic effects. This can be done fully relativistically

in the weak field limit (see e.g. [19]). However, as has been shown (see e.g. [20] and references therein), even N -body codes such as GADGET [21], PKDGRAV [22] and RAMSES [23] which are inherently Newtonian, can in fact be used to obtain results which are valid in the weak field limit of GR. In the case of pure Λ CDM models, i.e. models with only one matter component, this can be done via backscaling. The inclusion of massive neutrinos complicates matters, and the backscaling method becomes highly non-trivial. However, massive neutrinos can be included using the method presented in [24–26]. This requires neutrinos to be light enough that they can be treated as a purely linear component. In this case the neutrino density field can be realised at each timestep in a Newtonian simulation and thus be used to calculate the neutrino contribution to the local gravitational potential. This method was first introduced in [13] and shown to lead to sub-percent errors in the calculation of the matter power spectrum for neutrino masses up to around 0.3 eV. When neutrinos are added to the simulation using this method the N -body simulation still contains only one matter component and this makes it possible to use the framework presented in [24–26].

In this paper we show that by extending the method to include massive neutrinos we can run Newtonian N -body simulations which are fully consistent with GR, including massive neutrinos, without compromising the speed and scalability of standard N -body codes. We test our framework using two different N -body codes and demonstrate that we obtain fully consistent results.

In Section 2 we discuss the theoretical set-up needed to include massive neutrinos and GR. In Section 3 we present our numerical results, and finally Section 4 contains a discussion and our conclusions.

9.2 Method and Implementation

As was shown in [24–26], Newtonian N -body simulations containing only dark matter (or any other highly non-relativistic component) can be made compatible with general relativity.

For pure matter (i.e. a pressureless component) the continuity and Euler equations for the density contrast δ and peculiar velocity \mathbf{v} can be written as

$$\dot{\delta} + \nabla \cdot \mathbf{v} = 0, \quad (9.1)$$

$$(\partial_\tau + \mathcal{H})\mathbf{v} = -\nabla\phi + \nabla\gamma, \quad (9.2)$$

where a dot denotes differentiation with respect to conformal time τ and $\mathcal{H} = \dot{a}/a$ is the conformal Hubble parameter with a being the cosmic scale factor. The quantity γ is a correction which can be subtracted from the peculiar potential, ϕ , in the simulation. The potential ϕ is the total potential from all species, i.e.

$$\nabla^2\phi = \nabla^2 \sum_{\alpha} \phi_{\alpha} = 4\pi G a^2 \sum_{\alpha} \delta\rho_{\alpha}, \quad (9.3)$$

with $\alpha \in \{\text{cdm}, \text{b}, \gamma, \nu\}$ running over all species*.

From [20], the Fourier space equation for γ can be written as

$$\gamma k^2 = -(\partial_{\tau} + \mathcal{H})\dot{H}_{\text{T}} + 8\pi G a^2 \Sigma, \quad (9.4)$$

where Σ is the total anisotropic stress of all species and H_{T} is the trace-free component of the spatial part of the metric in N -body gauge (see e.g. [27]). In appendix 9.A we calculate γ in Fourier space with massive neutrinos included. We then have everything we need for this approach to fully consistently take massive neutrinos into account.

Concretely we split the total potential $\phi - \gamma$ experienced by the matter in the simulation into a contribution coming from the matter itself (calculable using standard techniques in the N -body simulation), ϕ_{sim} , and a contribution coming from photons, neutrinos, and the GR correction γ , ϕ_{GR} :

$$\phi - \gamma \equiv \phi_{\text{sim}} + \phi_{\text{GR}}, \quad (9.5)$$

with ϕ_{GR} given by

$$\begin{aligned} \nabla^2\phi_{\text{GR}} &\equiv \nabla^2(\phi_{\gamma} + \phi_{\nu} - \gamma) \\ &\equiv 4\pi G a^2 (\delta\rho_{\gamma} + \delta\rho_{\nu} + \delta\rho_{\text{metric}}) \\ &\equiv 4\pi G a^2 \delta\rho_{\text{GR}}. \end{aligned} \quad (9.6)$$

Here $\delta\rho_{\text{metric}}$ is a fictitious density perturbation which amounts to the GR potential correction γ ,

$$\nabla^2\gamma = -4\pi G a^2 \delta\rho_{\text{metric}}. \quad (9.7)$$

Following the same prescription as in [26], at each timestep in the simulation we realise $\delta\rho_{\text{GR}}$ in Fourier space, solve its Poisson equation (9.6), transform to real space and apply the force from ϕ_{GR} to the

*The subscript γ refers to photons and should not be confused with the variable γ , representing the relativistic potential correction.

matter particles, in addition to the usual force from the matter particles themselves (corresponding to ϕ_{sim}).

To compute $\delta\rho_{\text{GR}}$ in linear perturbation theory, a CLASS [28] computation has been run in advance, providing us with $\delta\rho_\gamma$ and $\delta\rho_\nu$ in either synchronous ('s') or conformal Newtonian gauge ('c'), \dot{H}_T in N -body gauge ('Nb') as described in appendix 9.A, as well as Σ (which is gauge independent in linear perturbation theory and can be calculated from its algebraic relation to $\phi - \psi$ in conformal Newtonian gauge), all as functions of a and k . From \dot{H}_T^{Nb} and Σ , we obtain $\delta\rho_{\text{metric}}^{\text{Nb}}(a, k)$ using (9.4) and (9.7). We then transform $\delta\rho_\gamma$ and $\delta\rho_\nu$ to N -body gauge,

$$\delta\rho_\alpha^{\text{Nb}} = \delta\rho_\alpha^{\text{s/c}} + 3\mathcal{H}(1+w_\alpha)\frac{\theta_{\text{tot}}^{\text{s/c}}}{k^2}\bar{\rho}_\alpha, \quad (9.8)$$

with θ_{tot} the total peculiar velocity divergence of all species and $w_\alpha = \bar{P}_\alpha/\bar{\rho}_\alpha$ the equation of state parameter of species α (both obtainable from CLASS), after which we add $\delta\rho_\gamma^{\text{Nb}}$ and $\delta\rho_\nu^{\text{Nb}}$ to $\delta\rho_{\text{metric}}^{\text{Nb}}$, resulting in $\delta\rho_{\text{GR}}^{\text{Nb}}$. The realisation of this $\delta\rho_{\text{GR}}^{\text{Nb}}(k)$ on a grid in real space is done using the formalism outlined in appendix A of [18].

In figure 9.1 we show the individual contributions to ϕ_{GR} from photons, neutrinos, and the GR correction γ . For the case of massless neutrinos we reproduce the results from [26]: For small (superhorizon) values of k all three contributions asymptotically approach k -independent values, while for larger k all three contributions oscillate and damp. For the case of massive neutrinos, we see that, as expected, the neutrino contribution ceases to oscillate as soon as neutrinos become non-relativistic ($T/m_\nu \sim 1/3$ around $z \sim 60$). From this point on it grows rapidly, essentially following the matter evolution. We also note that the photon and γ contributions remain almost unchanged in the case of massive neutrinos so that by far the largest difference between models with different neutrino masses comes from the neutrino component itself, rather than from photons or the GR correction γ .

We initialise the simulation using CLASS in the same way as was described in [18]. Initial conditions for the density and velocity fields are generated using the N -body gauge $\delta_{\text{cdm+b}}$ and $\theta_{\text{cdm+b}}$ transfer functions from CLASS at the initialisation time, i.e. we do not use higher order methods such as 2LPT. In this particular case this is completely unproblematic since we study effects pertaining to very large scales where structures are completely linear at the initial time.

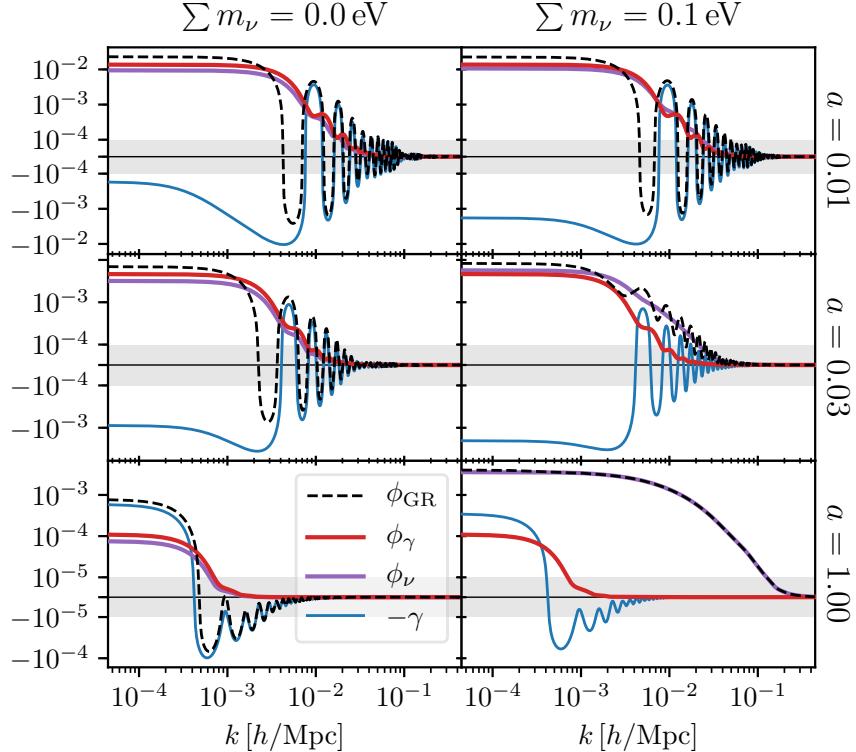


Figure 9.1 – Individual contributions to $\phi_{\text{GR}} \equiv \phi_\gamma + \phi_\nu - \gamma$ at three different scale factors. The left plot shows the case of massless neutrinos, the right plot shows the case of $\sum m_\nu = 0.1 \text{ eV}$. The potentials are all in N -body gauge. The grey bands indicate regions where the vertical axes scale linearly.

9.3 Numerical Set-up and Results

In order to test the effect of massive neutrinos including GR corrections we perform a suite of N -body simulations, primarily using the publicly available CONCEPT N -body solver [18]. All CONCEPT simulations in this work use cosmological parameters as listed in table 9.1. We use a degenerate neutrino hierarchy, i.e. three neutrinos of equal mass. The CONCEPT simulations all begin at $a = 0.01$, use 1024^3 matter particles and the potential grids (both ϕ_{sim} and ϕ_{GR}) are of size 1024^3 . All CONCEPT simulations are carried out in box sizes of either $(16384 \text{ Mpc}/h)^3$ or $(1024 \text{ Mpc}/h)^3$, the power spectra from which are patched together to give the ones shown in figure 9.2, 9.3 and 9.4.

Parameter	ΛCDM	$\sum m_\nu = 0.10 \text{ eV}$
A_s	2.215×10^{-9}	2.215×10^{-9}
n_s	0.9655	0.9655
τ_{reio}	0.0925	0.0925
Ω_b	0.049	0.049
Ω_{cdm}	0.264	0.262
Ω_ν	3.77×10^{-5}	2.37×10^{-3}
h	0.6731	0.6731
$N_{q,\nu}$	1000	1000
$l_{\max,\nu}$	1000	1000

Table 9.1 – Cosmological parameters and numerical settings for the CLASS runs used. We use the exact relation $\Omega_{\text{cdm}} = 0.2643 - \Omega_\nu$.

9.3.1 Main Results

In figure 9.2 we show ratios of matter power spectra with and without the corrections from ϕ_{GR} included. The effects from including the different terms (ϕ_γ , ϕ_ν , $-\gamma$) in ϕ_{GR} are shown. It can be clearly seen that at early times models with and without neutrino mass behave identically because the neutrinos are still close to relativistic. Once neutrinos become non-relativistic the relative contribution from ϕ_ν increases significantly and dominates over the other components, whereas the photons and the metric component (γ) are close to identical in the two cases. This is completely expected given the behaviour of ϕ_{GR} seen in figure 9.1.

In figure 9.3 we show the well-known suppression plot, comparing models with $\sum m_\nu = 0.1 \text{ eV}$ to $\sum m_\nu = 0$, with and without the full ϕ_{GR} included.

The dashed lines show results from running purely Newtonian simulations. We find the usual suppression in the semi-linear to non-linear regime (explained in detail in numerous other works, see e.g. [3–9, 11, 14–17, 29]).

Notice that in the limit of small k there are noticeable differences. At the initial time the model with non-zero neutrino mass has slightly more power, but over time the model with neutrino mass has slower growth of structure and therefore the power ratio drops with time. This phenomenon can be explained in the following way: At the initialisation time the amplitude of matter fluctuations (δ) is proportional to τ^2 , i.e. the conformal time at this particular a , squared. For the models shown

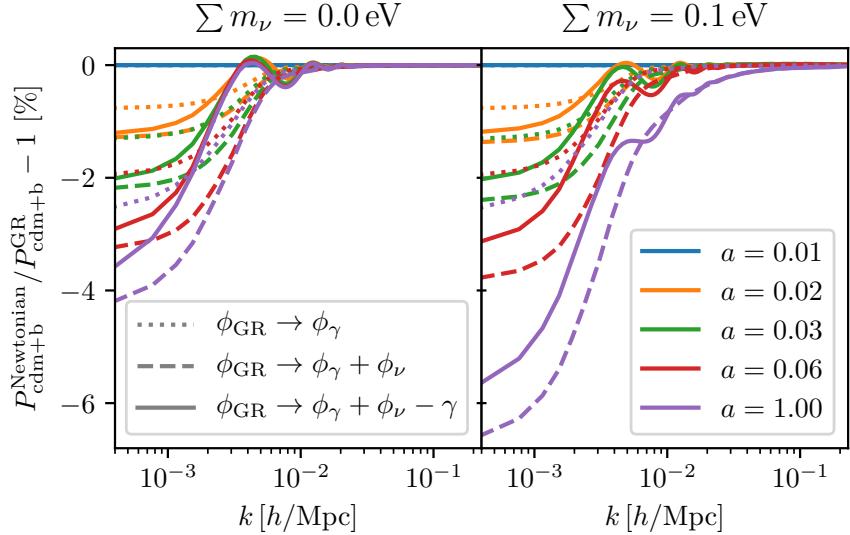


Figure 9.2 – Relative matter (CDM and baryons) power spectra with and without GR effects. Three levels of GR effects are considered; photon perturbations only (dotted lines), photon and neutrino perturbations (dashed lines) and photon, neutrino and metric perturbations (full lines). The left plot shows the case of massless neutrinos, the full lines of which are equivalent to figure 2 in [26], the right plot shows the case of $\sum m_\nu = 0.1$ eV. The power spectra are in N -body gauge.

here this τ^2 differ by approximately 0.5 %, and therefore the difference in power is approximately 1 %.

Over time the Newtonian models lack any contribution from photon, neutrino and metric perturbations on large scales and since the matter density is lower in the model with neutrino mass, the matter fluctuations grow correspondingly slower, leading to suppression of power over time.

The thin, horizontal dashed lines show the ratio of solutions to the purely Newtonian linear perturbation equations for non-relativistic matter, i.e. the ratio of the growth functions, D , squared. Both models have almost the same background evolution. However, the model with massive neutrinos has no source term from the neutrinos acting on the CDM. We normalise the ratio such that it matches exactly at the initial time. The fact that the simulations match the simple Newtonian linear theory result is a nice consistency check of the code.

The full curves show the result of simulations with ϕ_{GR} included. The thin black lines show the results from CLASS (i.e. linear theory), and

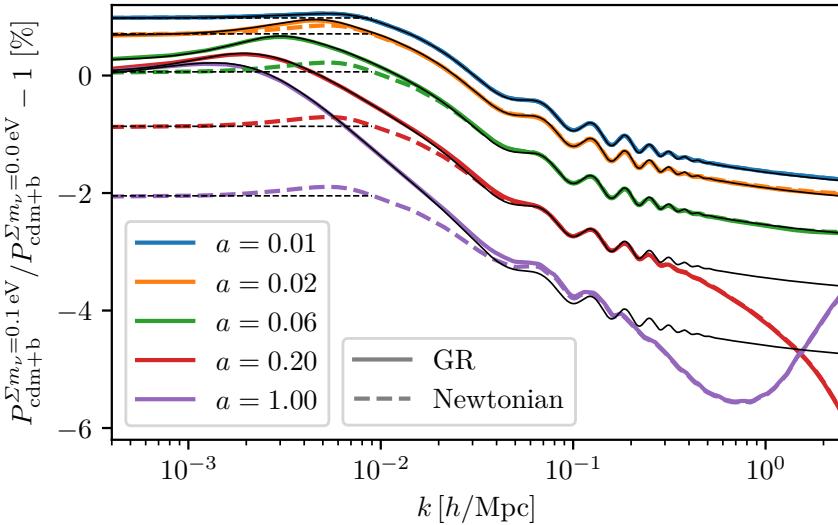


Figure 9.3 – Relative matter (CDM and baryons) power spectra between $\sum m_\nu = 0.1 \text{ eV}$ and $\sum m_\nu = 0$ cosmologies. Dashed lines are without any GR effects and full lines with all GR effects included. Black lines indicate the corresponding linear results from CLASS, where again full lines are in full GR (default CLASS) and the dashed lines show the k -independent Newtonian growth rate. The power spectra are in N -body gauge.

as can be seen the N -body results match exactly in the linear regime. As expected we see a slight increase in the ratio just before the non-linear scales (only clearly visible at $a = 1$). For large k we find the expected result, namely that there is an exact match between Newtonian and GR simulations.

For the simulations with ϕ_{GR} included the difference on large scales is far smaller. At the initialisation point the difference is the same as in the Newtonian case, since they start from the same CLASS output. However, at later times the lack of cold dark matter is, to a large extent, compensated by the presence of neutrino and photon fluctuations. On super-horizon scales these are comparable in importance to the matter fluctuations and therefore the suppression becomes much less pronounced.

Finally, we note that the bump seen around $k \sim 6 \times 10^{-3} h/\text{Mpc}$ in the initial ratio arises from the difference in matter-radiation equality between the two models (see e.g. [30]), and that it propagates differently in the two models. In the Newtonian simulations it remains fixed in k -space, whereas in the GR case it moves to the left over time. This difference is

caused by the GR corrections during evolution (i.e. it essentially amounts to the difference between the left and right panels in figure 9.2).

9.3.2 Comparison with PKDGRAV

In order to test the robustness of our calculation we have additionally implemented the GR effects in the state-of-the-art publicly available code PKDGRAV. Results from this exercise are shown in figure 9.4. As can be seen, PKDGRAV provides results which are identical to those of CONCEPT to within a very small margin, even though the two codes are fundamentally different.

PKDGRAV is a pure tree code, but with a grid structure implemented very recently precisely for the use case laid out in this paper. As seen in figure 9.4, the results from the GR implementation in PKDGRAV match those from the GR implementation in CONCEPT very accurately. Through the newly added CLASS mode of PKDGRAV, all PKDGRAV simulations use the exact same cosmology and initial conditions* as the CONCEPT simulations. Similarly, the box size is chosen as $(16384 \text{ Mpc}/h)^3$ and the simulations begin at $a = 0.01$. The number of particles is however reduced to 512^3 , as we are only interested in the linear regime.

Figure 9.4 do not show the PKDGRAV lines at the lowest k modes around $k \sim 10^{-3} h/\text{Mpc}$, as here they begin to deviate from the expected results by a few percent. We can achieve agreement in this region by increasing the box size, which then simply moves the inaccurate region to the left. We suspect that this can be explained by the time stepping scheme used by PKDGRAV, where all (major) time steps last for the same length of cosmic time. A new time stepping scheme based on the scale factor (at least at early times), rather than the cosmic time, is under construction.

Our PKDGRAV simulations ended prematurely due to a hardware failure, and so the PKDGRAV lines for $a = 1.00$ in figure 9.4 are really constructed from power spectra at $a = 0.50$, which we have extrapolated to $a = 1.00$ using linear theory.

9.4 Discussion

We have presented a framework for calculating the effect of light neutrinos, as well as photon and GR corrections in Newtonian N -body codes. The approach is based on the COSIRA code presented in [26], in which radiation

*Up to an effective change of random seed.

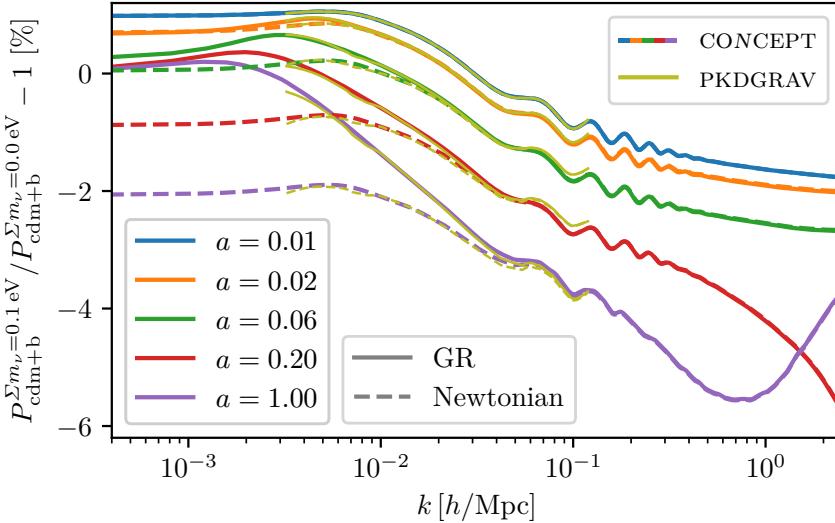


Figure 9.4 – Relative matter (CDM and baryons) power spectra between $\sum m_\nu = 0.1 \text{ eV}$ and $\sum m_\nu = 0$ cosmologies, similar to figure 9.3. The coloured lines show the CONCEPT results and are identical to those in figure 9.3. Cyan lines show the corresponding PKDGRAV results.

(photons and massless neutrinos) were included consistently to leading order in the N -body solver GADGET. The method involves the realisation at all times of the radiation perturbation field and the scalar potential quantity γ on a grid in the code. This grid is subsequently added to the ordinary potential grid to account for the effects of radiation and GR corrections to the Euler equation.

In this work we have extended the formalism to account for the possibility of massive neutrinos which complicates the calculation of γ somewhat. As in the case of massless neutrinos, we use the CLASS code to calculate the quantities necessary to construct γ in linear perturbation theory, i.e. the general relativistic potential correction γ as well as the energy density perturbations of photons and massive neutrinos. These are then realised on a grid in the N -body simulation. We have tested the implementation in two different solvers: CONCEPT, which is a PM code fully interfaced with CLASS, and PKDGRAV, which is a tree code to which has been added a potential grid in order to implement the effects.

We find that we can calculate the effects pertaining to the addition of massive neutrinos, photons and GR corrections at the per mille level on large scales, where structure formation is purely linear. On smaller scales

we find, as expected, that the effects of massive neutrinos are completely dominated by the absence of a clustering matter component, and that our results are identical to those found in a completely Newtonian N -body run.

The corrections studied here are typically at the level of a few percent on large scales, large enough that they should be included when comparing against data from future very large surveys such as Euclid [2] and LSST [1].

Acknowledgements

This work was supported by the Villum Foundation. We thank Joachim Stadel for valuable comments on the draft. Jeppe Dakin thanks Joachim Stadel, Douglas Potter and Romain Teyssier for hospitality during a stay in which much of this work was carried out.

9.A Computing γ

Here we will go through the steps necessary to calculate the quantity γ , appearing in (9.4). We will assume a spatially flat universe throughout this section. We start from the following definition of γ from equation 4.12 in [20]:

$$\gamma k^2 \equiv -\left(\partial_\tau + \frac{\dot{a}}{a}\right)\dot{H}_T + 8\pi G a^2 \Sigma \quad (9.9)$$

$$= -\left(\partial_\tau + \frac{\dot{a}}{a}\right)\dot{H}_T + k^2(\phi - \psi), \quad (9.10)$$

where the last line provides a convenient way of obtaining the total shear from quantities available in CLASS.

Given that we run the simulations using N -body gauge we will now fix our discussion to this gauge. In N -body gauge, we have $H_T^{Nb} = 3\zeta$, where ζ is the comoving curvature perturbation, leading to

$$\dot{H}_T^{Nb} = 3\dot{\zeta} = 3\frac{\dot{a}}{a} \left[-\frac{\delta P^c}{\rho + P} + \sigma \right], \quad (9.11)$$

where we have used the conservation equation for ζ in comoving gauge (equation 41 in [31]). The following gauge transformation of δP^c is valid in both the Synchronous and Newtonian gauges since $B = 0$ in those gauges:

$$\delta P^c = \delta P^{s/c} + \dot{P} \frac{\theta^{s/c}}{k^2}. \quad (9.12)$$

Combining the equations, we find the following formula:

$$\dot{H}_T^{Nb} = 3 \frac{\dot{a}}{a} \frac{1}{\rho + P} \left[-\delta P^{s/c} + \dot{P} \frac{\theta^{s/c}}{k^2} + (\rho + P)\sigma \right]. \quad (9.13)$$

Three of the quantities in this equation, δP , \dot{P} and σ are not readily available in the standard version of CLASS. Thus, it is convenient to modify CLASS slightly to output this quantity. We need a formula for \dot{P} inside CLASS that also includes non-cold dark matter. From equation 3.14 in [32] we find

$$\dot{P}_\alpha = -\frac{\dot{a}}{a} (5P_\alpha - \mathfrak{p}_\alpha), \quad (9.14)$$

where \mathfrak{p} is the pseudo-pressure defined in [32]. For any pressureless species, $\mathfrak{p}_\alpha \simeq P \simeq 0$, and for relativistic species we have $\mathfrak{p}_\alpha \simeq P$. We can then write the time-derivative of the total pressure in terms of the total pressure and $\mathfrak{p}_{\text{ncdm,tot}}$:

$$\dot{P} = \sum_\alpha \dot{P}_\alpha = -\frac{\dot{a}}{a} (4P + P_{\text{ncdm,tot}} - \mathfrak{p}_{\text{ncdm,tot}}). \quad (9.15)$$

Using this prescription we have modified CLASS to provide \dot{H}_T^{Nb} in N -body gauge, which through (9.10) provides the quantity γ .

Bibliography

- [1] P. A. Abell *et al.* [LSST Science and LSST Project Collaborations], “LSST Science Book, Version 2.0,” arXiv:0912.0201 [astro-ph.IM].
- [2] R. Laureijs *et al.* [EUCLID Collaboration], “Euclid Definition Study Report,” arXiv:1110.3193 [astro-ph.CO].
- [3] J. Brandbyge, S. Hannestad, T. Haugbølle and B. Thomsen, “The Effect of Thermal Neutrino Motion on the Non-linear Cosmological Matter Power Spectrum,” JCAP **0808** (2008) 020 [arXiv:0802.3700 [astro-ph]].
- [4] S. Agarwal and H. A. Feldman, “The effect of massive neutrinos on the matter power spectrum,” Mon. Not. Roy. Astron. Soc. **410** (2011) 1647 [arXiv:1006.0689 [astro-ph.CO]].
- [5] S. Bird, M. Viel and M. G. Haehnelt, “Massive Neutrinos and the Non-linear Matter Power Spectrum,” Mon. Not. Roy. Astron. Soc. **420** (2012) 2551 [arXiv:1109.4416 [astro-ph.CO]].

- [6] F. Villaescusa-Navarro, F. Marulli, M. Viel, E. Branchini, E. Castorina, E. Sefusatti and S. Saito, “Cosmology with massive neutrinos I: towards a realistic modeling of the relation between matter, haloes and galaxies,” *JCAP* **1403** (2014) 011 [arXiv:1311.0866 [astro-ph.CO]].
- [7] E. Castorina, C. Carbone, J. Bel, E. Sefusatti and K. Dolag, “DEMNUni: The clustering of large-scale structures in the presence of massive neutrinos,” *JCAP* **1507** (2015) no.07, 043 [arXiv:1505.07148 [astro-ph.CO]].
- [8] J. D. Emberson *et al.*, “Cosmological neutrino simulations at extreme scale,” *Res. Astron. Astrophys.* **17** (2017) no.8, 085 [arXiv:1611.01545 [astro-ph.CO]].
- [9] J. Adamek, R. Durrer and M. Kunz, “Relativistic N-body simulations with massive neutrinos,” *JCAP* **1711** (2017) no.11, 004 [arXiv:1707.06938 [astro-ph.CO]].
- [10] J. Brandbyge, S. Hannestad and T. Tram, arXiv:1806.05874 [astro-ph.CO].
- [11] M. Viel, M. G. Haehnelt and V. Springel, “The effect of neutrinos on the matter distribution as probed by the Intergalactic Medium,” *JCAP* **1006** (2010) 015 [arXiv:1003.2422 [astro-ph.CO]].
- [12] A. Banerjee, D. Powell, T. Abel and F. Villaescusa-Navarro, *JCAP* **1809** (2018) no.09, 028 [arXiv:1801.03906 [astro-ph.CO]].
- [13] J. Brandbyge and S. Hannestad, “Grid Based Linear Neutrino Perturbations in Cosmological N-body Simulations,” *JCAP* **0905** (2009) 002 [arXiv:0812.3149 [astro-ph]].
- [14] Y. Ali-Haimoud and S. Bird, “An efficient implementation of massive neutrinos in non-linear structure formation simulations,” *Mon. Not. Roy. Astron. Soc.* **428** (2012) 3375 [arXiv:1209.0461 [astro-ph.CO]].
- [15] J. Liu, S. Bird, J. M. Z. Matilla, J. C. Hill, Z. Haiman, M. S. Madhavacheril, A. Petri and D. N. Spergel, “MassiveNuS: Cosmological Massive Neutrino Simulations,” arXiv:1711.10524 [astro-ph.CO].

- [16] J. Brandbyge and S. Hannestad, “Resolving Cosmic Neutrino Structure: A Hybrid Neutrino N-body Scheme,” *JCAP* **1001** (2010) 021 [arXiv:0908.1969 [astro-ph.CO]].
- [17] A. Banerjee and N. Dalal, “Simulating nonlinear cosmological structure formation with massive neutrinos,” *JCAP* **1611** (2016) no.11, 015 [arXiv:1606.06167 [astro-ph.CO]].
- [18] J. Dakin, J. Brandbyge, S. Hannestad, T. Haugbølle and T. Tram, “ ν CONCEPT: Cosmological neutrino simulations from the non-linear Boltzmann hierarchy,” arXiv:1712.03944 [astro-ph.CO].
- [19] J. Adamek, D. Daverio, R. Durrer and M. Kunz, *JCAP* **1607** (2016) no.07, 053 [arXiv:1604.06065 [astro-ph.CO]].
- [20] C. Fidler, T. Tram, C. Rampf, R. Crittenden, K. Koyama and D. Wands, “General relativistic weak-field limit and Newtonian N-body simulations,” *JCAP* **1712** (2017) no.12, 022 [arXiv:1708.07769 [astro-ph.CO]].
- [21] V. Springel, “The Cosmological simulation code GADGET-2,” *Mon. Not. Roy. Astron. Soc.* **364** (2005) 1105 [astro-ph/0505010].
- [22] D. Potter, J. Stadel and R. Teyssier, “PKDGRAV3: Beyond Trillion Particle Cosmological Simulations for the Next Era of Galaxy Surveys,” arXiv:1609.08621 [astro-ph.IM].
- [23] R. Teyssier, *Astron. Astrophys.* **385** (2002) 337 [astro-ph/0111367].
- [24] C. Fidler, C. Rampf, T. Tram, R. Crittenden, K. Koyama and D. Wands, *Phys. Rev. D* **92** (2015) no.12, 123517 [arXiv:1505.04756 [astro-ph.CO]].
- [25] C. Fidler, T. Tram, C. Rampf, R. Crittenden, K. Koyama and D. Wands, *JCAP* **1609** (2016) no.09, 031 [arXiv:1606.05588 [astro-ph.CO]].
- [26] J. Brandbyge, C. Rampf, T. Tram, F. Leclercq, C. Fidler and S. Hannestad, *Mon. Not. Roy. Astron. Soc.* **466** (2017) L68 [arXiv:1610.04236 [astro-ph.CO]].
- [27] J. Adamek, J. Brandbyge, C. Fidler, S. Hannestad, C. Rampf and T. Tram, *Mon. Not. Roy. Astron. Soc.* **470** (2017) no.1, 303 [arXiv:1703.08585 [astro-ph.CO]].

- [28] D. Blas, J. Lesgourgues and T. Tram, “The Cosmic Linear Anisotropy Solving System (CLASS) II: Approximation schemes,” *JCAP* **1107** (2011) 034 [[arXiv:1104.2933 \[astro-ph.CO\]](#)].
- [29] S. Bird, Y. Ali-Haïmoud, Y. Feng and J. Liu, “An Efficient and Accurate Hybrid Method for Simulating Non-Linear Neutrino Structure,” [arXiv:1803.09854 \[astro-ph.CO\]](#).
- [30] J. Lesgourgues and S. Pastor, *Phys. Rept.* **429** (2006) 307 [[astro-ph/0603494](#)].
- [31] W. Hu, *ICTP Lect. Notes Ser.* **14** (2003) 145 [[astro-ph/0402060](#)].
- [32] J. Lesgourgues and T. Tram, *JCAP* **1109** (2011) 032 [[arXiv:1104.2935 \[astro-ph.CO\]](#)].

10 Non-linear Neutrinos

This chapter contains the paper ‘ ν CONCEPT: Cosmological neutrino simulations from the non-linear Boltzmann hierarchy’, which I co-authored together with Jacob Brandbyge, Steen Hannestad, Troels Haugbølle and Thomas Tram. It was put on arXiv and submitted to JCAP in December 2017. Due to issues with convergence for large neutrino masses as described in the paper, the referee has requested two major revisions. Though I doubt that we can solve this issue before then, we plan to resubmit it in late December 2018.

The version of the paper presented here includes the first batch of revisions, and so it differs somewhat from the version found on arXiv. In addition, slight syntactic modifications to the paper have been made in order for its notation to better match that used in the main thesis. The most noticeable difference not adapted for is the use of an overdot as differentiation with respect to conformal time, whereas in the main part of this thesis an overdot refers to differentiation with respect to cosmic time. An error relating to this difference has been corrected in (10.10).

In the paper we demonstrate our framework for solving the non-linear neutrino hierarchy up to $\ell_{\text{nl}} = 1$, while using linear higher order moments. The paper is rather large and touches on many aspects in this thesis, including the non-linear equations from the Boltzmann hierarchy (section 3.2), numerical fluid dynamics (sections 6.2 and 6.4), linear (section 7.2) and non-linear realisations (section 7.4). It also contains numerical results of such non-linear neutrino simulations, which other than that of figure 6.1 have been left out of the main body of this thesis.

Paper:

vCONCEPT: Cosmological Neutrino Simulations from the Non-linear Boltzmann Hierarchy

Jeppe Dakin^a, Jacob Brandbyge^{a,b}, Steen Hannestad^a,
Troels Haugbølle^b, Thomas Tram^{a,c}

^aDepartment of Physics and Astronomy, Aarhus University,
DK-8000 Aarhus C, Denmark

^bCentre for Star and Planet Formation,
Niels Bohr Institute & Natural History Museum of Denmark,
University of Copenhagen, Øster Voldgade 5–7,
DK-1350 Copenhagen, Denmark

^cAarhus Institute of Advanced Studies (AIAS), Aarhus University,
DK-8000 Aarhus C, Denmark

E-mail: dakin@phys.au.dk, jacobb@phys.au.dk, sth@phys.au.dk,
haugboel@nbi.ku.dk, thomas.tram@aias.au.dk

Abstract. In this paper the non-linear effect of massive neutrinos on cosmological structures is studied in a conceptually new way. We have solved the non-linear continuity and Euler equations for the neutrinos on a grid in real space in N -body simulations, and closed the Boltzmann hierarchy at the non-linear Euler equation using the stress and pressure perturbations from linear theory. By comparing with state-of-the art cosmological neutrino simulations, we are able to simulate the non-linear neutrino power spectrum very accurately. This translates into a negligible error in the matter power spectrum, and so our vCONCEPT code is ideally suited for extracting the neutrino mass from future high precision non-linear observational probes such as Euclid.

Contents

10.1	Introduction	170
10.2	Theory	172
10.2.1	The Non-linear Boltzmann Equation	172
10.2.2	The Moment Equations	174
10.2.3	Closing the Hierarchy	175
10.2.4	The Continuity and Euler Equations in Conservation Form	175
10.3	The Linear Computation	176
10.3.1	$\delta P/\delta\rho$ and σ in Linear Theory	178
10.4	Implementation Details	180
10.4.1	Dynamics	181
10.4.2	Gravity	184
10.5	Results	189
10.5.1	The Neutrino Power Spectra	189
10.5.2	Neutrino Suppression of the Relative Total Matter Power Spectra	192
10.5.3	Comparison with the Hybrid Code	192
10.5.4	The Effect of Anisotropic Stress	193
10.5.5	Convergence	194
10.6	Conclusions	196
10.A	N -body Realisations of the Boltzmann Hierarchy Variables	198
10.A.1	The Zel'dovich Approximation	199
10.A.2	Density and Velocity Fields	199
10.A.3	Anisotropic Stress	200
10.A.4	Non-linear Realisations	201
10.B	Comparison between CLASS and CAMB	203

10.1 Introduction

The large scale structure in our Universe has been mapped to unprecedented precision during the past decade and provided a spectacular amount of information on cosmological parameters. Within the coming few years large scale structure surveys such as Euclid [1] and LSST [2] will increase the available amount of data by yet another order of magnitude.

These surveys are likely to provide the first evidence for non-zero neutrino masses, and eventually enable a precise measurement of the neutrino mass. This is possible because neutrinos have a significant influence on the formation of structure and lead to damping of fluctuations on small scales. However, the sensitivity of large scale structure formation to the neutrino mass also requires neutrinos to be modelled accurately in e.g. N -body simulations.

In order to follow non-linear structure formation of collision-less species it is necessary to solve the collision-less Boltzmann equation,

$$\frac{d}{dt} f(\mathbf{x}, \mathbf{p}, t) = 0. \quad (10.1)$$

However, solving the equation in the full 6+1 dimensional case is currently not numerically feasible. For cold dark matter the problem can be greatly simplified because the CDM particles have no thermal velocity, reducing the problem to effectively 3+1 dimensions while the perturbations are linear. The most commonly used method for tracking structure formation with CDM is to represent the distribution function, f , with particles and follow these in phase space.

Unfortunately, neutrinos cannot easily be followed in the same way because their thermal velocities are larger than, or comparable to, the gravitationally induced streaming velocities. Several schemes have been devised for studying non-linear neutrino structure formation:

- Using a particle representation of the full neutrino distribution function (e.g. [3–10]). This requires a much larger number of particles than for CDM because the momentum dependence of the distribution function must be tracked. Furthermore, if the simulation is started early the neutrino structures will be completely noise dominated because of the large thermal velocities.
- Assuming that neutrino perturbations remain linear [11–13]. A simple scheme which is known to work well for small neutrino masses is to use the linear neutrino density field calculated by realising the linear neutrino transfer function on a grid [11]. An improvement on this is to solve the linear theory neutrino equations, but use the full non-linear gravitational potential calculated in the simulation. [12, 13]. However, in both cases this scheme only works for small neutrino masses where neutrino perturbations remain linear at all times.

- A hybrid combination of the 2 former methods where the neutrino component is initially followed with linear theory but later on, as the thermal velocities approach the gravitationally induced streaming velocities, followed with N -body particles [14].

Here, we want to take a somewhat different approach. We start from the full momentum-dependent Boltzmann equation and use the BBGKY [15–17] approach to turn this into a hierarchy of velocity moment equations. For a perfect fluid this hierarchy closes at order 1 and leads to the continuity and Euler equations. However, because neutrinos have a large anisotropic stress component we need to go beyond order 1 in the hierarchy. We demonstrate this by solving the two first moment equations in full non-linear theory while treating the stress and pressure perturbations in linear theory (scaled by the non-linear density field) leads to very accurate results.

We note that the approach of closing the equations at the second moment was also pursued in [18], where the Boltzmann equation for neutrinos is similarly recast into hierarchy form, and the integrated (fluid) equations are then solved. However, in [18] the solution is restricted to non-relativistic particles and the moment hierarchy is closed using an estimate of the second moment gained from the motion of test particles. In this work we use a version of the second moment which guarantees that the solution has the correct behaviour in the linear regime while also allowing for fully non-linear evolution of structure. Furthermore the method presented here works for both relativistic and non-relativistic fluids.

The paper is structured as follows: In section 10.2 we describe the theoretical considerations needed to formulate the hierarchy equations. In section 10.3 we discuss the needed linear theory evolution and how to set up initial conditions for the simulations. Section 10.4 contains a review of the numerical methods employed in the simulations, and section 10.5 gives a discussion of our main results. Finally, section 10.6 contains our conclusions.

10.2 Theory

10.2.1 The Non-linear Boltzmann Equation

The Boltzmann equation is an evolution equation for the distribution function (a function of 7 parameters) which we choose in their covariant form, namely x^μ and P^i . This 7-dimensional problem can be recast

into a 4-dimensional one by taking moments of the Boltzmann equation. The reduction of the dimensionality from integrating out the momentum dependence comes at the price of an infinite hierarchy of moment equations.

The symmetric energy-momentum tensor has 10 independent components which are related to the distribution function, f , via the integration

$$T^\mu{}_\nu \equiv \sqrt{-g} \int d^3P f \frac{P_\nu P^\mu}{P_0}. \quad (10.2)$$

These 10 components can be used to define 10 fluid variables, namely δ , u^i , $\delta P/\delta\rho$ and σ_j^i . The relations are given by

$$T^0{}_0 = -\bar{\rho}(1 + \delta), \quad (10.3)$$

$$T^i{}_0 = -\bar{\rho}\left(1 + \delta + w + \frac{\delta P}{\delta\rho}\delta\right)u^i, \quad (10.4)$$

$$\theta = \partial_i u^i, \quad (10.5)$$

$$T^i{}_j = \bar{\rho}\left(w + \frac{\delta P}{\delta\rho}\delta\right)\delta^i{}_j + \bar{\rho}\left(1 + \delta + w + \frac{\delta P}{\delta\rho}\delta\right)\left(u^i u_j + \sigma^i{}_j\right), \quad (10.6)$$

with $\bar{\rho}$ the average density, \bar{P} the average pressure and $w \equiv \bar{P}/\bar{\rho}$. Finally, $\delta^i{}_j$ is the Kronecker delta and $\sigma^i{}_j$ is traceless. These terms are progressively higher order in the velocity expansion. The zero order term is $\delta\rho$, first order terms are u^i and θ , while δP and $\sigma^i{}_j$ are second order terms.

The moment equations can be found by integrating the non-manifestly covariant Boltzmann equation for f (see [19, 20]),

$$P^\mu \frac{\partial f}{\partial x^\mu} - P^\mu P^\lambda \Gamma_{\mu\lambda}^i \frac{\partial f}{\partial P^i} = 0, \quad (10.7)$$

over the invariant volume element d^3P^I , which for contravariant momentum variables P^i is given by

$$d^3P^I = \frac{mc}{P_0} \sqrt{-g} d^3P, \quad (10.8)$$

with $d^3P = dP^1 dP^2 dP^3$.

The following moment equations will be derived in the conformal Newtonian gauge with line-element [21]

$$ds^2 = -a^2(1 + 2\psi)d\tau^2 + a^2(1 - 2\phi)dx^2. \quad (10.9)$$

10.2.2 The Moment Equations

Zeroth moment: The continuity equation

Multiplying the Boltzmann equation (10.7) with P_0 and integrating, we recover the general relativistic continuity equation in the weak-field limit:

$$\begin{aligned} \dot{\delta} = & - (1+w)(\theta - 3\dot{\phi}) - 3\frac{\dot{a}}{a}\left(\frac{\delta P}{\delta\rho} - w\right)\delta \\ & - \theta\delta - u^i\partial_i\delta \\ & + 3\left(1 + \frac{\delta P}{\delta\rho}\right)\dot{\phi}\delta - \frac{\delta P}{\delta\rho}\theta\delta - u^i\partial_i\left(\frac{\delta P}{\delta\rho}\delta\right) \\ & - (\partial_i\psi - 3\partial_i\phi)\left(1 + \delta + w + \frac{\delta P}{\delta\rho}\delta\right)u^i, \end{aligned} \quad (10.10)$$

where a dot implies differentiation with respect to conformal time.

First moment: The Euler equation

To get the Euler equation, we must multiply the Boltzmann equation with $P_0 P^i / P^0$, which after integration gives

$$\begin{aligned} \dot{u}^i = & - \left[\frac{\dot{a}}{a}(1-3w) - \dot{\psi} - 5\dot{\phi} \right] u^i - \frac{[\dot{\delta} + \dot{w} + \partial_\tau(\delta P/\delta\rho)\delta]u^i + \delta^{ij}(1+\delta)\partial_j\psi}{1 + \delta + w + (\delta P/\delta\rho)\delta} \\ & - \frac{1}{\bar{\rho}(1 + \delta + w + (\delta P/\delta\rho)\delta)} \left[\delta^{ik}(\partial_j + \partial_j\psi - 3\partial_j\phi) + \delta^k_j\delta^{il}\partial_\ell\phi \right] T^j_k. \end{aligned} \quad (10.11)$$

The continuity and Euler equations could also be found from energy-momentum conservation, i.e. $\nabla_\mu T^\mu_\nu = 0$.

The second moment: Pressure and anisotropic stress

The second moment equation is found by multiplying the Boltzmann equation with the factor $P_i P^j / P^0$ and then integrating, i.e. from the equation

$$\sqrt{-g} \int d^3P \frac{P_i P^j}{P_0 P^0} \left(P^\mu \frac{\partial f}{\partial x^\mu} - P^\mu P^\lambda \Gamma^k_{\mu\lambda} \frac{\partial f}{\partial P^k} \right) = 0. \quad (10.12)$$

Defining the third cumulant as

$$\Pi^{\mu\lambda}{}_\nu \equiv \sqrt{-g} \int d^3P f \frac{P_\nu P^\mu P^\lambda}{P_0 P^0}, \quad T^\mu{}_\nu \equiv \Pi^{\mu 0}{}_\nu, \quad (10.13)$$

and likewise for higher moments, and using the covariant derivative

$$\nabla_\mu \Pi^{j\mu}_i = \partial_\mu \Pi^{j\mu}_i + \Gamma^j_{\mu\nu} \Pi^{\nu\mu}_i + \Gamma^\mu_{\mu\nu} \Pi^{j\nu}_i - \Gamma^\nu_{i\mu} \Pi^{ju}_\nu , \quad (10.14)$$

one can arrive at the second-moment equation which gives the time-derivative of T^i_j

$$\nabla_\mu \Pi^{j\mu}_i = \Gamma^0_{\mu\nu} \Pi^{j\mu\nu}_i , \quad (10.15)$$

or equivalently

$$\nabla_0 T^j_i = \Gamma^0_{00} T^j_i + 2\Gamma^0_{0k} \Pi^{jk}_i + \Gamma^0_{k\ell} \Pi^{jk\ell}_i - \nabla_k \Pi^{jk}_i . \quad (10.16)$$

The time-evolution of T^j_i will depend on the third moment and a contracted version of the fourth moment of the distribution function. This is analogous to the continuity equation where $\dot{\delta}$ depends on T^i_0 and δT^i_i .

10.2.3 Closing the Hierarchy

The equation for $\nabla_0 T^j_i$ is complex and depends on several higher order moments. Instead of solving it directly to get the inputs to the continuity ($\delta T^i_i \sim \delta P$) and Euler ($T^i_j \sim \sigma^i_j$, δP) equations we will estimate these non-linear terms from their linear counterparts as follows.

Assuming the ratio $\delta P/\delta\rho$ to be independent of the amplitude of the perturbations, we find

$$\delta P_{\text{nl}}(\mathbf{k}) \simeq \delta\rho_{\text{nl}}(\mathbf{k}) \left(\frac{\delta P(k)}{\delta\rho(k)} \right)_l , \quad (10.17)$$

where ‘l’ stands for linear and ‘nl’ for non-linear.

Since σ^i_j has the same velocity order as δP we will likewise estimate

$$\sigma^i_{j,\text{nl}}(\mathbf{k}) \simeq \delta\rho_{\text{nl}}(\mathbf{k}) \left(\frac{\sigma^i_j(k)}{\delta\rho(k)} \right)_l . \quad (10.18)$$

We shall refer the reader to appendix 10.A for more details on the realisations.

10.2.4 The Continuity and Euler Equations in Conservation Form

For the numerical implementation it is preferable to express the fluid equations using conserved quantities only. To this end, we define the ‘conserved’ density ϱ , current J^i , pressure \mathcal{P} and anisotropic stress ς^i_j as

$$\varrho \equiv a^{3(1+\omega)} \rho , \quad (10.19)$$

$$J^i \equiv a^4(\rho + P)u^i, \quad (10.20)$$

$$\mathcal{P} \equiv a^{3(1+w)}P, \quad (10.21)$$

$$\varsigma_j^i \equiv (\varrho + \mathcal{P})\sigma_j^i, \quad (10.22)$$

where the effective equation of state w is given by

$$w(a) \equiv \frac{1}{\ln a} \int_1^a \frac{w(a')}{a'} da'. \quad (10.23)$$

With these variables, the continuity (10.10) and Euler (10.11) equations become

$$\begin{aligned} \dot{\varrho} = & -a^{3w-1}\partial_i J^i \\ & + 3aH(w\varrho - \mathcal{P}) \\ & + a^{3w-1}J^i\partial_i(3\phi - \psi) \\ & + 3(\varrho + \mathcal{P})\dot{\phi} \end{aligned} \quad (10.24)$$

and

$$\begin{aligned} \dot{J}^i = & -\partial^j \left[a^{3w-1} \frac{J^i J_j}{\varrho + \mathcal{P}} + a^{-3w+1} \varsigma_j^i \right] \\ & - a^{-3w+1} \partial^i \mathcal{P} \\ & - a^{-3w+1} (\varrho + \mathcal{P}) \partial^i \psi \\ & - a^{3w-1} \frac{J^j J_i}{\varrho + \mathcal{P}} \partial^i \phi \\ & + \left[a^{3w-1} \frac{J^i J_j}{\varrho + \mathcal{P}} + a^{-3w+1} \varsigma_j^i \right] \partial^j (3\phi - \psi) \\ & + J^i (\dot{\psi} + 5\dot{\phi}). \end{aligned} \quad (10.25)$$

The claim of conservation of the chosen variables can be checked by spatially averaging (10.24) and (10.25), indeed leading to $\dot{\bar{\varrho}} = \dot{\bar{J}^i} = 0$.

In our simulations we neglect the difference between ϕ and ψ (which is sourced by anisotropic stress), neglect time-derivatives of ϕ and ψ and disregard terms of order $\partial_i \phi u^i$ and higher, from which terms with $\partial_i \phi J^i$ and $\partial_i \phi \sigma_j^i$ vanish. Thus only the first two terms of (10.24) and the first three terms of (10.25) are kept.

10.3 The Linear Computation

We compute the linear evolution of all species using the Einstein-Boltzmann code CLASS [22, 23]. In the notation of [21], the distribution function is

Table 10.1 – Cosmological parameters and numerical settings for the CLASS runs used.

Parameter	ΛCDM	$\sum m_\nu = 0.15 \text{ eV}$	$\sum m_\nu = 0.3 \text{ eV}$	$\sum m_\nu = 1.2 \text{ eV}$
A_s	2.3×10^{-9}	2.3×10^{-9}	2.3×10^{-9}	2.3×10^{-9}
n_s	1.0	1.0	1.0	1.0
τ_{reio}	0.097765	0.097765	0.097765	0.097765
Ω_b	0.05	0.05	0.05	0.05
Ω_{cdm}	0.250	0.247	0.243	0.224
Ω_ν	3.48×10^{-5}	3.29×10^{-3}	6.57×10^{-3}	2.61×10^{-2}
$N_{q,\nu}$...	2310	1154	344
$\ell_{\max,\nu}$...	2000	2000	1601

expanded as

$$f(\tau, \mathbf{x}, \mathbf{p}) = f_0(q)[1 + \Psi(\tau, \mathbf{x}, \mathbf{p})], \quad (10.26)$$

and the evolution equation for $\Psi(q, k, \hat{\mathbf{q}} \cdot \hat{\mathbf{k}})$ is then solved in Fourier space. The angular dependence of Ψ is expanded in Legendre multipoles resulting in an infinite hierarchy which is then truncated at some finite ℓ_{\max} . We refer the reader to [21] for the derivation of the equations.

In this work we do not investigate effects due to the precise neutrino mass hierarchy chosen, and so each neutrino species gets assigned exactly 1/3 of this total mass $\sum m_\nu$. Though unrealistic, this is a perfectly good choice for method testing as it leaves $\sum m_\nu$ as the only free neutrino parameter, which is precisely the parameter on which the dampening of the matter power spectrum is sensitive.

Neither CLASS nor CAMB [24, 25] produce accurate neutrino transfer functions at their default precision settings since both codes are optimised to produce accurate CMB and matter power spectra, which do not depend strongly on the late-time neutrino evolution. By increasing the precision parameters of both codes, we have found that agreement can be established at the 1%-level or better. We refer the interested reader to appendix 10.B for more details.

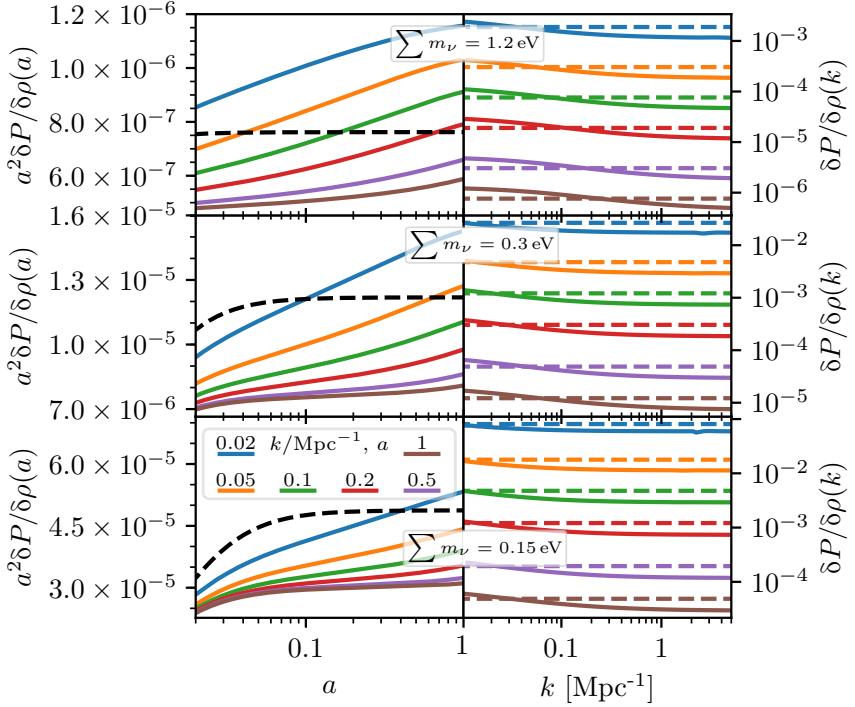


Figure 10.1 – Linear $\delta P/\delta\rho$ for $\sum m_\nu = 1.2$ (top), 0.3 (middle) and 0.15 eV (bottom), as function of a (left) and of k (right). Note that all plots share the same legend, so that e.g. the blue line on the left corresponds to $k = 0.02 \text{ Mpc}^{-1}$ while the blue line on the right corresponds to $a = 0.02$. The dashed lines show the corresponding w ($a^2 w$ for the left plots). For the CLASS precision settings used, see table 10.1.

10.3.1 $\delta P/\delta\rho$ and σ in linear theory

Following the notation of [26], δP , $\delta\rho$ and σ are related to the Legendre multipoles of Ψ_ℓ as

$$\delta\rho = 4\pi \left(\frac{T_{\nu,0}}{a} \right)^4 \int_0^\infty f_0 dq q^2 \epsilon \Psi_0, \quad (10.27)$$

$$\delta P = 4\pi \left(\frac{T_{\nu,0}}{a} \right)^4 \int_0^\infty f_0 dq \frac{q^4}{3\epsilon} \Psi_0, \quad (10.28)$$

$$(\bar{\rho} + \bar{P})\sigma = 8\pi \left(\frac{T_{\nu,0}}{a} \right)^4 \int_0^\infty f_0 dq \frac{q^4}{3\epsilon} \Psi_2. \quad (10.29)$$

The effective sound speed squared, $\delta P/\delta\rho$, turns out to be extremely challenging to compute numerically. The real $\delta P/\delta\rho$ is expected to

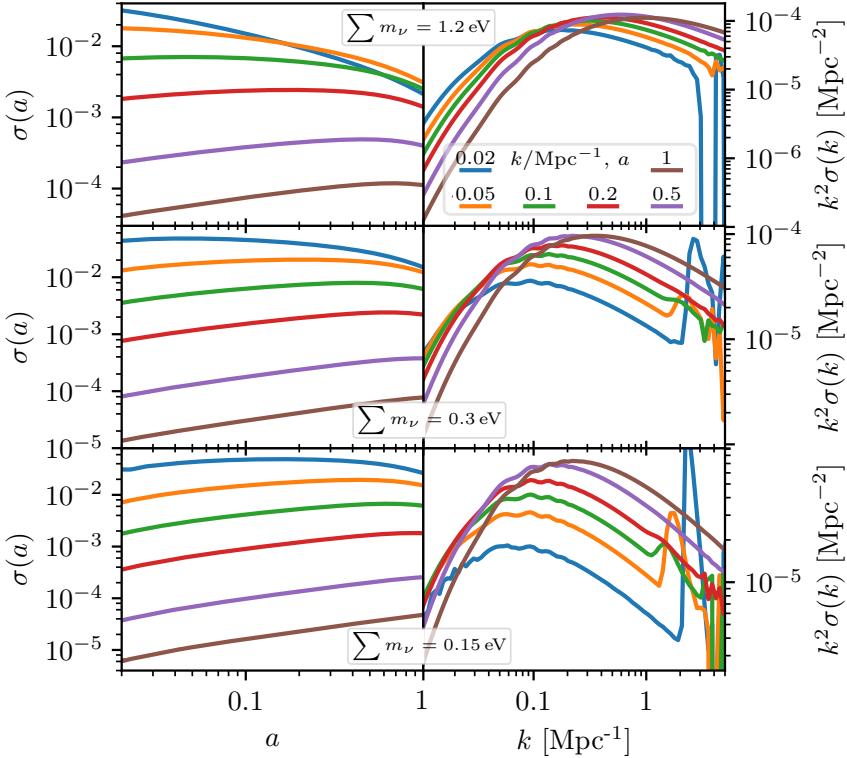


Figure 10.2 – Linear σ for $\sum m_\nu = 1.2$ (top), 0.3 (middle) and 0.15 eV (bottom), as function of a (left) and of k (right). As in figure 10.1, all plots share the same legend. For the CLASS precision settings used, see table 10.1.

be a smooth and monotonic function of a , but because the monopole perturbation Ψ_0 is highly oscillatory until $q \ll \epsilon$, the discretisation of the systems can easily lead to pathological behaviour of $\delta P/\delta\rho$ unless an extremely large number of momentum bins is used. We find that in some cases it is necessary to use well over 2000 bins (compared to the standard CLASS setting using 5 bins) before the results converge over the required range of a .

Furthermore, reflections from the large- ℓ boundary used to close the system of equations can lead to spurious effects such as coherent oscillations of $\delta P/\delta\rho$. We find that we need $\ell_{\max} \gtrsim 2000$ to ensure convergence (compared to $\ell_{\max} = 17$ in the standard setting). Running CLASS at such extreme precision-settings requires several hundred CPU-hours for a single model, which should be compared to the ~ 10 CPU

seconds required at default precision. The runtime of CLASS is still a small fraction of the total runtime, but we nevertheless store the CLASS runs to the disk in order to avoid unnecessary recalculations.

Figure 10.1 shows the linear $\delta P/\delta\rho$ resulting from three CLASS computations with different neutrino masses. These are the linear $\delta P/\delta\rho$ values used in our simulations. The slight k -dependence makes the local pressure substantially lower than what would be found by approximating $\delta P/\delta\rho \simeq w$, especially for lower (more realistic) neutrino masses. Failure of taking this k -dependence into account leads to a mismatch between the equations solved in CLASS and those solved in CONCEPT, which manifests as spurious generation of oscillations in the neutrino density field.

Figure 10.2 shows the linear σ resulting from the same three CLASS computations as was used for figure 10.1. These are the linear σ values used in our simulations. The high k modes have not quite converged at early times, but since σ falls off rapidly with k these oscillatory high k modes should only contribute a very low amount of noise to the real-space σ .

10.4 Implementation Details

The methods developed in this paper has been implemented into the CONCEPT code, a new cosmological code capable of simultaneously evolving N -body particles (matter) and fluids (neutrinos), interacting under mutual and self-gravity. We have fully integrated the CLASS code into CONCEPT, including an MPI-parallelised method of calling CLASS from CONCEPT, enabling multi-node CLASS computations. With this integration the CONCEPT code has easy access to the evolution of background and linear variables, from which realisations of particle distributions and fluid fields are made. Such realisations are used both for initial condition generation and to close the Boltzmann hierarchy during the N -body simulation.

The code is mostly written in Python. For performance, the code may optionally (and preferably) be compiled to C code via Cython. For further optimisations and to lower the Python/Cython barrier, a custom Python → Cython transpiler is built in as part of CONCEPT. The code is MPI-parallelised with a fixed spatial domain decomposition, dividing up the simulation box into rectangular boxes of equal volume. Each MPI process is then responsible for what goes on within one such domain.

10.4.1 Dynamics

In CONCEPT the collections of either N -body particles or fluid variable grids, which are to be evolved dynamically, are grouped into *components*. Particle components consists of a fixed number of particles N , of equal mass m , whereas fluid components consists of a fixed set of regular, cubic grids of fixed resolution, one for each scalar fluid variable. In the simulations carried out for this paper, baryons and dark matter are grouped together into one particle component, while the neutrino component consists of grids storing $\varrho(\mathbf{x})$, $J^i(\mathbf{x})$, $\mathcal{P}(\mathbf{x})$ and $\varsigma_j^i(\mathbf{x})$. Of these fluid variables, only ϱ and J^i are treated as non-linear variables, evolved via (10.24) and (10.25) with terms neglected as described in subsection 10.2.4. Since J^i is a vector quantity, this requires 4 grids. The higher-order variables \mathcal{P} and ς_j^i are not evolved non-linearly, but realised at each time step anew. Storing the full ς_j^i would require 6 additional grids (5 if we took advantage of the tracelessness). However, as ς_j^i is only needed once during each time step (in the Euler equation), a single grid is used to store each of its components in turn.

In each time step, all particles and fluid elements are evolved forward in time by the same amount $\Delta\tau$. A leapfrog time integration scheme is used, in which every other time step is either a ‘kick’ or a ‘drift’ step. In a ‘kick’ step, all source terms in the evolution equations are applied. For particle components, the only source term is that of gravity. For fluid components, a term is considered a source term if it is not a flux (divergence) term of one of the lower-order variables ϱ and J^i . Explicitly, the partition of terms into flux and source terms is

$$\dot{\varrho} = - \underbrace{a^{3w-1} \partial_i J^i}_{\text{flux term}} + \underbrace{3aH(w\varrho - \mathcal{P})}_{\text{source term}}, \quad (10.30)$$

$$J^i = - \underbrace{a^{3w-1} \partial^j \frac{J^i J_j}{\varrho + \mathcal{P}}}_{\text{flux term}} - \underbrace{a^{-3w+1} [\partial^j \varsigma_j^i + \partial^i \mathcal{P} + (\varrho + \mathcal{P}) \partial^i \psi]}_{\text{source terms}}. \quad (10.31)$$

The MacCormack method

In ‘drift’ steps, particle positions \mathbf{x}_i are updated according to their momenta, while the fluid grids are evolved according to the flux terms of (10.30) and (10.31). To solve these two coupled equations simultaneously, the simple MacCormack [27] finite difference method is used. This method consists of a predictor followed by a corrector step, here illustrated for

the flux term of the continuity equation:

$$\begin{cases} \varrho^*(\mathbf{x}) = \varrho(\mathbf{x}) - \sum_{i=1}^3 \frac{1}{|\Delta x_i|} [J^i(\mathbf{x} + \Delta x_i) - J^i(\mathbf{x})] \int_{\tau}^{\tau + \Delta\tau} a^{3w-1} d\tau, \\ \varrho(\mathbf{x}) \rightarrow \frac{1}{2} [\varrho(\mathbf{x}) + \varrho^*(\mathbf{x})] \\ \quad - \frac{1}{2} \sum_{i=1}^3 \frac{1}{|\Delta x_i|} [J^{i*}(\mathbf{x}) - J^{i*}(\mathbf{x} - \Delta x_i)] \int_{\tau}^{\tau + \Delta\tau} a^{3w-1} d\tau. \end{cases} \quad (10.32)$$

Here the slopes in $\partial_i J^i$ are approximated by the difference between neighbouring grid points along each dimension. The size of the spatial step $|\Delta x_i| = \Delta x$ is then just the grid spacing, which in CONCEPT is the same for all dimensions. In the predictor step, a temporary $\varrho^*(\mathbf{x})$ grid is build from “rightward” differences of $\varrho(\mathbf{x})$, whereas in the corrector step $\varrho(\mathbf{x})$ is updated from “leftward” differences of $\varrho^*(\mathbf{x})$.* In the corrector step, J^{i*} is needed, and so the predictor step needs to have been carried out on both the continuity and Euler equation before the corrector step(s) can be applied. This effectively doubles the amount of memory needed to store the non-linear ϱ and J^i variables. In (10.32), all fluid variables on the right-hand side are implicitly evaluated at the current time τ . One might argue that a more self-consistent treatment of the time-dependent function a^{3w-1} was to similarly evaluate this at time τ , rather than integrating over the time step interval. The choice of keeping the integral is inspired by [28].

For particle components, the leapfrog method ensures that \mathbf{x}_i and \mathbf{p}_i are always out-of-sync by $\Delta\tau/2$. That is, instead of letting e.g. \mathbf{x}_i evolve by an amount $\Delta\tau$ ahead of \mathbf{p}_i and then syncing up the system by letting \mathbf{p}_i evolve by $\Delta\tau$, we first ensure that \mathbf{x}_i and \mathbf{p}_i are out-of-sync by $\Delta\tau/2$, leading them to ‘leapfrog’ past one another at every time step. This then treats \mathbf{x}_i and \mathbf{p}_i symmetrically, which is the key to the stability of the leapfrog method [28]. The predictor and corrector steps in which the MacCormack scheme splits up a single (‘drift’) time step can be thought of as two half time steps, each of duration $\Delta\tau/2$. For fluids, applying gravity out-of-sync by half a time step then means that the gravitational forces applied are those matching the time right after the predictor and before the corrector step. In this way, though gravity is only applied half

*As we are in 3D, “rightwards” might be taken to refer to the $(+1, +1, +1)$ direction. In total, 8 possible directions $(\pm 1, \pm 1, \pm 1)$ exist. To avoid spurious generation of anisotropy, CONCEPT cycles through these 8 directions over a period of 8 time steps.

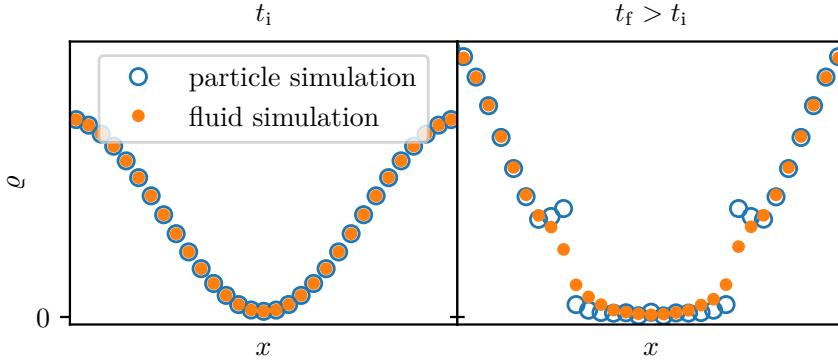


Figure 10.3 – 1D test simulations demonstrating the density smoothing extension to the MacCormack method. At time t_i , a sinusoidal wave with minimum density close to 0 is assigned a velocity field which diverges away from this minimum. This wave is then evolved to time t_f under self-gravity, leading to rapid growth of the low-density region. The same simulation is carried out using particles and a fluid with $w = 0$.

as often as flux terms, it is applied fairly with respect to the predictor and corrector step.

The bare MacCormack method as illustrated in (10.32) is not positivity-preserving, meaning that it is possible for $\varrho(\mathbf{x})$ to take on slightly negative values, regardless of the smallness of $\Delta\tau$. This can be prevented using e.g. total variation diminishing (TVD) extensions [29, 30] to the method, where flux limiters are applied in order to diminish discontinuities, effectively smoothing out the fluid. We choose a simpler solution, in which a check for negative densities is inserted after each MacCormack step. If a cell with negative density is found, ϱ and J^i of the surrounding block of 27 cells are smoothed out slightly.* This operation is quick and only perturbs the fluid in low density regions, as well as in regions with large density gradients.

To stabilise the method we found it necessary to apply the smoothing not only when the density of a fluid cell became negative, but also when a cell lost a large fraction of its total energy. This is needed because the MacCormack method introduces dispersive errors around steep gradients, which makes some kind of artificially added viscosity necessary. Figure 10.3 demonstrates the effect of this smoothing. We see that the smoothing leaves the overall evolution intact, but erases

*Here, smoothing is performed on each pair of cells in such a way as to preverse their total energy and momentum, with the amount of smoothing inversely proportional to their squared mutual distance.

discontinuities. We do not expect this to seriously perturb our neutrino fluid, as here the pressure term acts as to smooth out the fluid, making it very hard for sharp discontinuities to arise.

Time step size

The size of $\Delta\tau$ is chosen to be as large as possible without affecting the physics. Here the most important limiter is the (global) Courant condition, which we may write as

$$\Delta\tau < \frac{\Delta x}{\sqrt{3}(c\sqrt{w} + |\mathbf{u}|_{\max})}, \quad (10.33)$$

$$|\mathbf{u}|_{\max} = \begin{cases} a^{-1} \max_i |\mathbf{p}_i|/m, & \text{(particles)} \\ a^{3w-1} \max_x |J^i(\mathbf{x})/[\varrho(\mathbf{x}) + \mathcal{P}(\mathbf{x})]|. & \text{(fluids)} \end{cases}$$

Thus the particle or fluid element with largest Courant number sets the global pace of time. For fluid components, (10.33) states that $\Delta\tau$ should be small enough so that a sound wave travelling with speed $c\sqrt{w}$ on top of the bulk flow with maximal peculiar speed $|\mathbf{u}|_{\max}$ cannot traverse an entire (comoving) grid cell within a single time step.* For particles, the story is very much similar, except $w \equiv 0$ and the value of Δx has to be redefined. As particles by definition do not live on a grid, no inherent grid spacing exists. However, since gravitational interactions are implemented using the particle-mesh method (see subsection 10.4.2), we use the grid spacing of this mesh as the corresponding Δx for particles.

10.4.2 Gravity

The CONCEPT code has a rather modular interaction framework, in which different interactions and numerical methods may be assigned to different components. Currently only gravity is implemented, but several methods are available. For particle components, gravity can be computed using either the particle-particle (PP), particle-mesh (PM) or particle-particle-particle-mesh (P^3M) method. Because both the PP and P^3M methods are based on direct summation, they are virtually exact. However, in the current CONCEPT version PP and P^3M are too time-consuming for large simulations and we therefore use the code in

*As seen from figure 10.1, the actual, local sound speed $\sqrt{\delta\mathcal{P}(\mathbf{x})/\delta\varrho(\mathbf{x})}$ might be somewhat greater than $c\sqrt{w}$, and so (10.33) might not be a strong enough condition. We correct for this by simply multiplying the right-hand side of (10.33) with a small fraction.

PM set-up. As this method works by solving the Poisson equation on a mesh, it has an intrinsic limit to the force resolution.

As fluid components already have an intrinsic resolution limit, nothing is lost by using the PM method, and so only this method is implemented for fluid components. The remainder of this subsection lays out how the PM method is implemented, first for particle-only simulations and then for simulations with both particle and fluid components.

The PM method

The basic strategy of the PM method for particle-only simulations is as follows. Construct the total density field $\rho(\mathbf{x})$ on a mesh via interpolation. Now Fourier transform this mesh in-place; $\rho(\mathbf{x}) \rightarrow \tilde{\rho}(\mathbf{k})$. Convert the grid values to that of the Fourier transformed Newtonian peculiar gravitational potential, $\tilde{\rho}(\mathbf{k}) \rightarrow \tilde{\varphi}(\mathbf{k})$, using the Poisson equation

$$\tilde{\varphi}(\mathbf{k}) = -\frac{4\pi G a^2}{|\mathbf{k}|^2} \tilde{\rho}(\mathbf{k}), \quad \tilde{\varphi}(\mathbf{0}) = 0. \quad (10.34)$$

Now perform an inverse Fourier transform to obtain the potential in real space, $\tilde{\varphi}(\mathbf{k}) \rightarrow \varphi(\mathbf{x})$. Finally, use finite difference techniques to obtain approximations for $-\partial_i \varphi$ at each grid point and interpolate the resulting forces back to the particle positions and apply them. Note that a separate mesh is needed to store the forces $-\partial_i \varphi$. The fast Fourier transforms used automatically impose the needed periodic boundary conditions.*

Though the same grid in memory is used to store $\rho(\mathbf{x})$, $\tilde{\rho}(\mathbf{k})$, $\tilde{\varphi}(\mathbf{k})$ and $\varphi(\mathbf{x})$ values, we shall refer to this grid consistently as the φ grid. In CONCEPT, the cloud-in-cell (CIC) method is used for the interpolations to and from the φ grid. This method distributes each of the particles throughout the φ grid, with a weight at each grid point \mathbf{x}_m given by the geometric overlap between the particle and the grid point, where both the particle and the grid point are imagined to have a cubic shape with side lengths equal to the grid spacing of the φ mesh, Δx_φ . Denoting the weight at mesh point \mathbf{x}_m of a particle at \mathbf{x}_p by $W(\mathbf{x}_m - \mathbf{x}_p)$, we have

$$W(x, y, z) = \begin{cases} \left(1 - \frac{|x|}{\Delta x_\varphi}\right) \left(1 - \frac{|y|}{\Delta x_\varphi}\right) \left(1 - \frac{|z|}{\Delta x_\varphi}\right) & \text{if all } |x|, |y|, |z| < \Delta x_\varphi, \\ 0 & \text{otherwise.} \end{cases} \quad (10.35)$$

*CONCEPT uses the FFTW library for MPI-parallel 3D in-place real FFT's.

Note that the weights (10.35) are only non-zero for the 8 grid points closest to the particle. With the CIC weights (10.35), we can write down the interpolation as a convolution,

$$\rho_m(\mathbf{x}) = \frac{W(\mathbf{x})}{\Delta x_\varphi^3} * \rho(\mathbf{x}), \quad \rho(\mathbf{x}) = \frac{m}{a^3} \sum_{i=1}^N \delta(\mathbf{x} - \mathbf{x}_i), \quad (10.36)$$

where ρ_m is the mesh-interpolated density and the factor a^{-3} is needed since ρ is physical while \mathbf{x} is comoving. It is to be understood that numerically, all fields are only defined at the grid points. Thus, though \mathbf{x} in (10.36) is in some sense a discrete variable, this notion clashes with the convolution operation. Instead, let \mathbf{x} be continuous but make the distinction that numerical grids (e.g. ρ_m) are only defined at the grid points, unlike their physical counterparts (e.g. ρ).

In the Poisson equation (10.34), the actual density ρ is needed. Though what lives on the grid is really the interpolated values $\rho_m(\mathbf{x}) \propto W(\mathbf{x}) * \rho(\mathbf{x})$. Simply ignoring this difference leads to errors on scales comparable to the grid spacing Δx_φ . We correct for this by “undoing” the CIC convolution while in Fourier space, where the convolution with $W(\mathbf{x})$ turns into multiplication of $\widetilde{W}(\mathbf{k})$. The Poisson equation (10.34) then becomes

$$\widetilde{\varphi}_m(\mathbf{k}) = -\frac{4\pi G a^2}{|\mathbf{k}|^2} \frac{\Delta x_\varphi^3}{\widetilde{W}(\mathbf{k})} \widetilde{\rho}_m(\mathbf{k}), \quad \widetilde{\varphi}_m(\mathbf{0}) = 0, \quad (10.37)$$

where again, the subscript ‘m’ indicates that this is a numerical grid. The Fourier transform of the CIC weight (10.35) is

$$\widetilde{W}(k_x, k_y, k_z) = \Delta x_\varphi^3 \left[\text{sinc}\left(\frac{\Delta x_\varphi k_x}{2}\right) \text{sinc}\left(\frac{\Delta x_\varphi k_y}{2}\right) \text{sinc}\left(\frac{\Delta x_\varphi k_z}{2}\right) \right]^2. \quad (10.38)$$

Equation (10.37) results in a properly deconvolved potential on the φ grid, though our interest is really the resulting forces at the locations of the particles. Since another CIC interpolation is used to interpolate the forces from the grid points and onto the particles, a total of two CIC deconvolutions are actually needed. The potential actually calculated in particle-only simulations is then $\widetilde{\varphi}_m(\mathbf{k}) \Delta x_\varphi^3 / \widetilde{W}(\mathbf{k})$. Note that this is not the most accurate grid representation of the potential, but it does lead to the most accurate forces after one additional CIC convolution. Finally, just as we replaced $a^{3w-1} \Delta \tau$ with the integral of a^{3w-1} over the

time step in (10.32), here we replace the a^2 in (10.37) with its average value over the time step. Thus what is really computed on the φ grid is

$$\frac{\Delta x_\varphi^3}{\widetilde{W}(\mathbf{k})} \tilde{\varphi}_m(\mathbf{k}) = -\frac{4\pi G}{|\mathbf{k}|^2} \left[\frac{\Delta x_\varphi^3}{\widetilde{W}(\mathbf{k})} \right]^2 \tilde{\rho}_m(\mathbf{k}) \Delta\tau^{-1} \int_{\tau}^{t+\Delta\tau} a^2 d\tau, \quad (10.39)$$

where again, the DC ($\mathbf{k} = \mathbf{0}$) mode is to be disregarded.

Generalising the PM method

We shall now take a closer look at the needed generalisations to the PM method necessary when both particle and fluid components are present in the same simulation.

We can extend the CIC interpolation scheme to fluid components by specifying a coordinate x_p for each fluid element, which should be taken to be at the center of each grid cell. Just as with particles, the CIC interpolation treats fluid elements as cubes with side lengths equal to the grid spacing Δx_φ of the φ grid, regardless of the resolution of the fluid grids themselves.* If the fluid grid happens to be of the same resolution as the φ grid, all grid points coincide and the CIC interpolation (10.35) reduces to the trivial mapping, i.e. the interpolated and the original grids are equal. In effect then, for fluids with a grid size matching that of the φ grid, no CIC interpolation is carried out, and so we are actually worse off if we insist on performing the deconvolutions. Thus, to solve gravity properly, separate computations and φ grids are needed for particle and fluid components. Essentially, one grid, $\varphi^{\text{particles}}$, solves the Poisson equation as already described, including the deconvolutions, while another grid, φ^{fluids} , solves the Poisson equation without the deconvolutions.[†] Importantly, both $\varphi^{\text{particles}}$ and φ^{fluids} should account for the total gravitational potential from both particle and fluid components.

With two separate grids for particle and fluid components, we can construct the mesh-interpolated densities of all particle components

*Interpolating a homogeneous low resolution fluid grid onto a high resolution φ grid thus leaves φ with a lot of empty cells. A better fluid interpolation would distribute each fluid cell over the corresponding volume in the φ grid. As all simulations for this paper use the same grid size for fluid and potential grids, this is of no concern for our results.

[†]It is unclear whether carrying out CIC deconvolutions improves or worsens the results for fluids with resolutions different from that of the φ grid. In CONCEPT, the φ^{fluids} grid is never deconvolved, regardless of the resolution of the fluid grids.

$\rho_m^{\text{particles}}$ and of all fluid components ρ_m^{fluids} by generalisation of (10.36):

$$a^2 \rho_m^{\text{particles}}(\mathbf{x}) = \frac{W(\mathbf{x})}{\Delta x_\varphi^3} * \sum_{\alpha} m_{\alpha} \sum_{i=1}^{N_{\alpha}} \delta(\mathbf{x} - \mathbf{x}_{\alpha,i}) \Delta \tau^{-1} \int_{\tau}^{\tau + \Delta \tau} \frac{d\tau}{a}, \quad (10.40)$$

$$a^2 \rho_m^{\text{fluids}}(\mathbf{x}) = \sum_{\alpha} \varrho_{\alpha}(\mathbf{x}) \Delta \tau^{-1} \int_{\tau}^{\tau + \Delta \tau} \frac{d\tau}{a^{3w_{\alpha}+1}}, \quad (10.41)$$

with only the particles being convolved. Here, α in equation (10.40) runs over all particle components, while α in equation (10.41) runs over all fluid components. Allowing for multiple fluid components with different w introduces different integrands of the integrals over the time step, which is why these integrals must be moved from the common potential (10.39) and onto the individual densities. As both $\rho_m^{\text{particles}}$ and ρ_m^{fluids} are numerical grids, they can immediately be used in the Poisson equation for $\varphi_m^{\text{fluids}}$, while a CIC convolution is required for them to be used in the Poisson equation for $\varphi_m^{\text{particles}}$. Accounting for the single CIC convolution of (10.40), we end up with

$$\frac{\Delta x_\varphi^3}{\widetilde{W}(\mathbf{k})} \widetilde{\varphi}_m^{\text{particles}}(\mathbf{k}) = -\frac{4\pi G}{|\mathbf{k}|^2} \left[\left(\frac{\Delta x_\varphi^3}{\widetilde{W}(\mathbf{k})} \right)^2 a^2 \widetilde{\rho}_m^{\text{particles}}(\mathbf{k}) + \frac{\Delta x_\varphi^3}{\widetilde{W}(\mathbf{k})} a^2 \widetilde{\rho}_m^{\text{fluids}}(\mathbf{k}) \right], \quad (10.42)$$

$$\frac{\Delta x_\varphi^3}{\widetilde{W}(\mathbf{k})} \widetilde{\varphi}_m^{\text{fluids}}(\mathbf{k}) = -\frac{4\pi G}{|\mathbf{k}|^2} \left[\frac{\Delta x_\varphi^3}{\widetilde{W}(\mathbf{k})} a^2 \widetilde{\rho}_m^{\text{particles}}(\mathbf{k}) + a^2 \widetilde{\rho}_m^{\text{fluids}}(\mathbf{k}) \right], \quad (10.43)$$

where, as usual, we ignore the DC modes. The implementation of the two φ grids in CONCEPT is as memory efficient as possible. Equation (10.40) and (10.42) and their Fourier duals all live on one grid, while (10.41) and (10.43) and their Fourier duals live on another grid. A single additional grid is used for the final forces $-\partial_i \varphi_m^{\text{particles}}$ and $-\partial_i \varphi_m^{\text{fluids}}$.

From (10.42) and (10.43), we see that constructing the two versions of the potential from the densities requires 4 FFT's. Realising $\mathcal{P}_{\nu}(\mathbf{x})$ and all 6 components of $\varsigma_{j,\nu}^i(\mathbf{x})$ requires a total of $1 + (1 + 6) = 8$ FFT's, where the one additional FFT is due to the construction of $\widetilde{\rho}_{\nu}(\mathbf{k})$, needed for the non-linear realisations as described in subsection 10.2.3. We choose to keep $\widetilde{\rho}_{\nu}(\mathbf{k})$ around as a separate grid, which further increase the memory consumption, but saves us from having to recompute $\widetilde{\rho}_{\nu}(\mathbf{k})$ 1 + 6 times. In total, these 12 FFT's per time step take up about half (for simulations with grid size 1200^3) of the computation time.

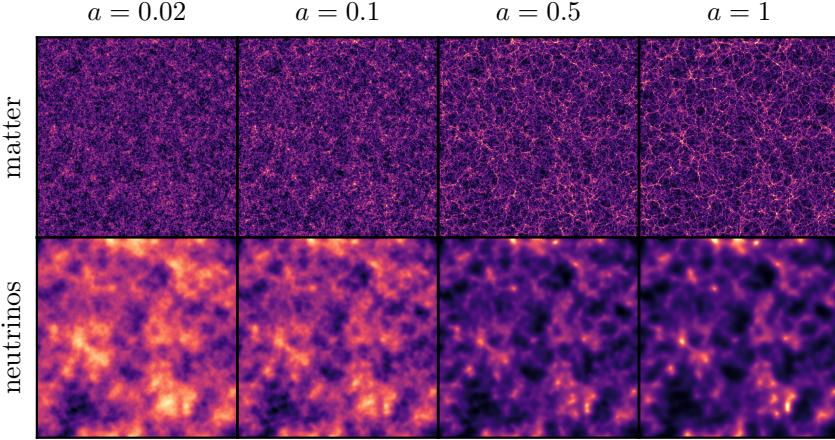


Figure 10.4 – Real space density plots for the $\sum m_\nu = 1.2 \text{ eV}$ simulation with grid size 600^3 . All plots have dimensions $512 \text{ Mpc}/h \times 512 \text{ Mpc}/h$ and a depth of $8.5 \text{ Mpc}/h$. The color scale is different for each plot. For the neutrino plots, the color scalings are linear, though with different absolute values. Each matter plot has its own non-linear color scaling.

10.5 Results

In order to test the code and compare against the hybrid neutrino method of [14], we have performed a suite of different simulations, which are presented in table 10.2 and table 10.3. Table 10.1 also specifies the Ω_ν corresponding to $\sum m_\nu$ in each simulation. We have furthermore used a flat cosmology with $\Omega_A = 0.7$, $\Omega_b = 0.05$, $\Omega_{\text{cdm}} = 0.25 - \Omega_\nu$, $h = 0.7$, $n_s = 1$ and $A_s = 2.3 \cdot 10^{-9}$.

In order to get the linear theory predictions needed for the initial condition as well as the realisation of $\sigma/\delta\rho$ and $\delta P/\delta\rho$ we have run CLASS with settings given in table 10.1, as described in section 10.3.

Figure 10.4 shows slices of the simulation volume in CONCEPT at 4 different redshifts. The density fields of both the CDM particle component and the neutrino grid component are displayed. The suppression of small scale structure in the neutrino component due to free-streaming is clearly visible.

10.5.1 The Neutrino Power Spectra

In figure 10.5 we show the absolute matter and neutrino power spectra from simulations B, D, and F, with parameters given in table 10.2.

Table 10.2 – CONCEPT simulations used in this work. All simulations are started at $z_i = 49$.

Sim	$\sum m_\nu$ [eV]	L_{box} [Mpc/ h]	$N_{\text{CDM}}^{\text{part}}$	N_ν^{grid}
A	0.15	512	600^3	600^3
B	0.15	512	1200^3	1200^3
C	0.30	512	600^3	600^3
D	0.30	512	1200^3	1200^3
E	1.20	512	600^3	600^3
F	1.20	512	1200^3	1200^3
G	1.20	1024	1200^3	1200^3
H	1.20	512	1680^3	1680^3

Table 10.3 – Simulations run with the hybrid neutrino code presented in [14]. All simulations are started at $z_i = 49$. Neutrinos with $q/T \leq 8$ are realised as N -body particles at a redshift of ~ 10 . The remaining high momentum part is kept as a linear source term on the grid. See [14] for further information on the hybrid method.

Sim	$\sum m_\nu$ [eV]	L_{box} [Mpc/ h]	$N_{\text{cdm}}^{\text{part}}$	N_ν^{grid}	N_ν^{part}
X	0	512	512^3	0	0
Y	1.2	512	512^3	512^3	1024^3

In general we find that there is a very significant increase in neutrino power beyond the linear perturbation theory prediction, in accordance with many previous investigations. In order to make a more quantitative comparison we show matter and neutrino power spectra for the 1.2 eV case, see table 10.3, predicted using the hybrid neutrino grid/particle method from [14]. Out to $k \sim 0.2\text{--}0.3 \text{ Mpc}^{-1}$ the agreement is excellent. At higher k several effects become important, which makes the comparison hard:

- CONCEPT is run as a particle mesh code and thus has no short range force included. This leads to lower matter power at late times.

- The hybrid code has a significant white noise component in the neutrino power spectrum.
- For large neutrino masses, the CONCEPT neutrino power spectra develop an unphysical bump for large k at late times. The remainder of this subsection describe the probable cause and seriousness of this bump.

Originally, we hypothesised that the unphysical bump seen in the neutrino power spectra at late times and small scales were due to our simplistic smoothing (described in subsection 10.4.1) being inadequate for larger neutrino masses, where a proper smoothing is especially important due to larger spatial gradients in the neutrino fields. We tried replacing the MacCormack+smoothing method with the Kurganov-Tadmor [31] TVD method, but the problem persisted. After much investigation, we are now convinced that the bump stems from our use of the PM method. Usually, a PM method is coupled with another method, such as direct summation (resulting in P³M) or a tree (resulting in treePM), where gravity is split into a short- and long-range part, typically by multiplying the Fourier space gravitational potential by a decaying exponential in k^2 . This ensures that the long-range part — handled by the PM method — is smooth at the grid scale regardless of how non-linear the system may become, which in turn leads to accurate (long-range only) forces via numerical differentiation. In our CONCEPT simulations, where the PM method is responsible for the entirety of the force, the gravitational contributions from \mathbf{k} modes around the grid scale will be extremely inaccurate as soon as significant structure has formed at this scale. This is a well-known effect in particle-only simulations and is in fact the whole reason to explicitly add on a separate short-range computation. Unlike with particles where this more or less random force at the grid scale leads to a relatively benign decrease in power, we see an increase in power for our neutrino fluid. This is because neighbouring fluid elements are much more tightly coupled than nearby particles: If a single fluid element receives a large kick due to a steep gradient in φ , the continuity equation will ensure that the neighbouring fluid elements will in- and decrease their density dramatically in the next time step, increasing the power at the grid scale. With time, these over- and under-densities in single fluid elements will diffuse out to neighbouring cells, propagating the unphysical power bump to larger scales.

The steep gradients in φ at late times is due to the non-linear clustering of the particles, not the neutrino field, which ought to remain

smooth at the grid scale throughout time. This further explains why we see the unphysical bump regardless of which method we use for the fluid dynamics. Furthermore, it can explain why the bump grows with the neutrino mass, and indeed is unnoticeable in the 0.15 eV case. For all our neutrino mass choices, φ is primarily shaped by the particle distribution, and so very similar steep gradients are expected at late times in all simulations. The size of the unreasonably large momentum updates to the neutrino cells is thus roughly independent on the neutrino mass. The lower the bulk velocity of the neutrino fluid, the larger the relative error due to this momentum update will be. As the heavier neutrinos move much slower than their lighter counterparts, these are affected more strongly. Additionally, because the lighter neutrinos become unrelativistic later than the heavier ones, these continue to have a strong pressure for a longer time, which serves to smooth out local over-/under-densities.

10.5.2 Neutrino Suppression of the Relative Total Matter Power Spectra

In addition to the absolute power spectra it is of interest to investigate the relative suppression of power in models with massive neutrinos relative to standard Λ CDM with massless neutrinos. We show these in figure 10.6 for $a = 0.5$ and 1 and for the three different choices of $\sum m_\nu$.

For all neutrino masses we see exactly the same trough-like shape of the suppression which was first noticed in [3] and which is a generic feature of comparing any model with suppressed structure growth to a standard Λ CDM model. As non-linear structure formation progresses, larger scales in the neutrino simulations collapse which in turn diminishes the amount of relative suppression and shifts the trough position to smaller k -values. These dynamical movements in the relative power spectrum were also found in [11].

The maximum suppression in the relative power spectrum at $z = 0$ for the lower neutrino masses can be fitted with the relation $\sim -10\Omega_\nu/\Omega_m$, which is in very good agreement with the findings in [3]. In contrast, the linear theory suppression is roughly given by $\sim -8\Omega_\nu/\Omega_m$.

10.5.3 Comparison with the Hybrid Code

Figure 10.7 shows the time evolution of the CONCEPT and hybrid absolute neutrino power spectra for $\sum m_\nu = 1.2$ eV. For $a = 0.5$ there is very good agreement out to $k \sim 0.3\text{--}0.4 \text{ Mpc}^{-1}$, where the hybrid simulation becomes dominated by neutrino particle shot noise. At lower redshift

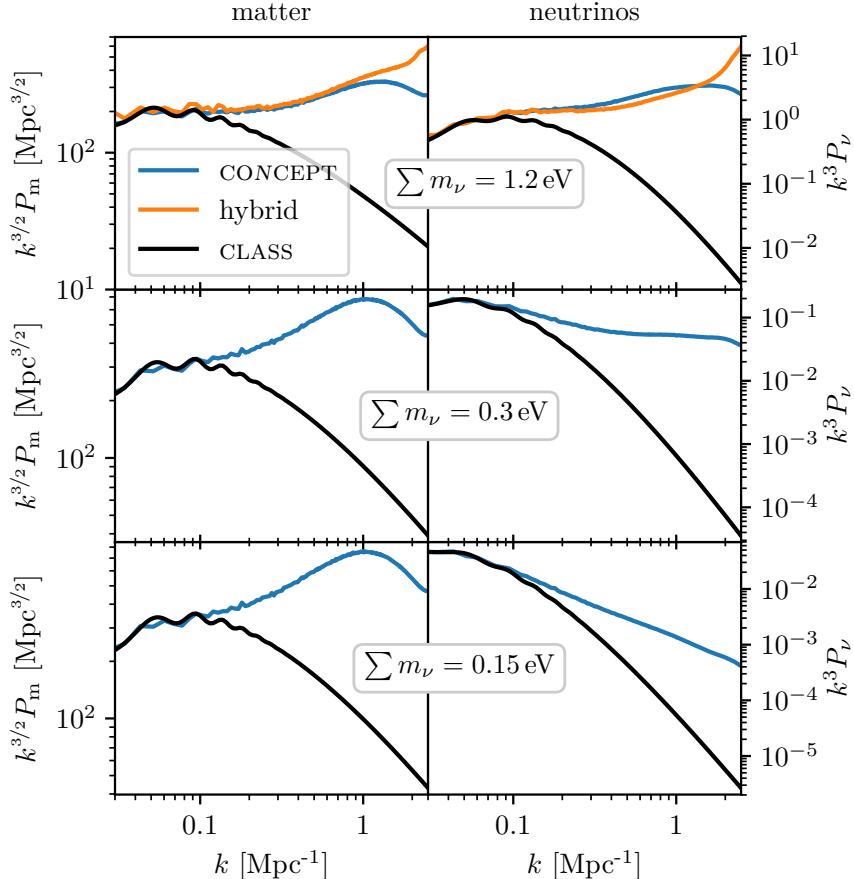


Figure 10.5 – Absolute matter (left) and neutrino (right) power spectra at $z = 0$ for $\sum m_\nu = 1.2$ eV (top), 0.3 eV (middle) and 0.15 eV (bottom).

the CONCEPT neutrino power spectrum increases faster than do the hybrid one. As described in subsection 10.5.1, we suspect this surplus of CONCEPT neutrino power to be due to inaccuracies induced by the PM method.

10.5.4 The Effect of Anisotropic Stress

In figure 10.8 we show neutrino power spectra at various a for the 1.2 eV simulation. For comparison we also show the same spectra for a simulation in which we have used $\delta P/\delta\rho = w$ and $\sigma = 0$ (i.e. the perfect fluid limit).

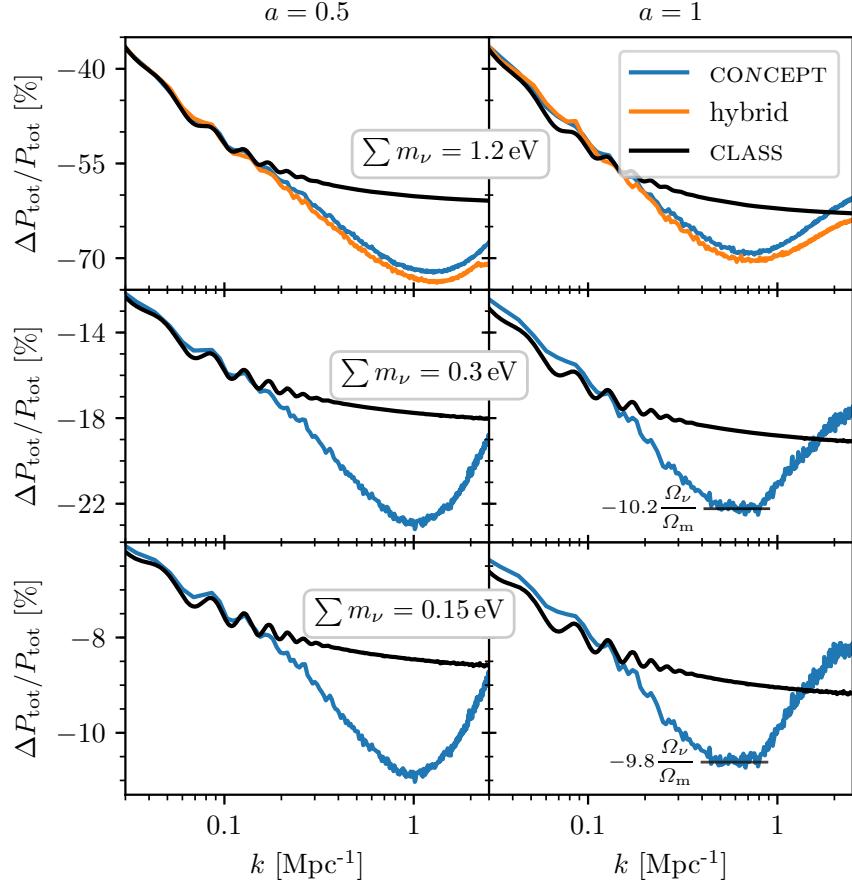


Figure 10.6 – The total power spectra (matter and neutrinos) for $\sum m_\nu = 1.2$ (top), 0.3 (middle) and 0.15 eV (bottom), relative to Λ CDM at $a = 0.5$ (left) and $a = 1.0$ (right).

From the figure it is evident that this approximation leads to an inaccurate neutrino power spectrum, with too much power on all scales. The effect of neglecting σ is known to increase neutrino fluctuations on all sub-horizon scales in linear theory [32]. This effect can be seen in the figure for small k , but the figure also demonstrates that a similar effect occurs in the non-linear regime.

10.5.5 Convergence

In figure 10.9 we show neutrino power spectra at $a = 0.5$ and $a = 1$ for the three different neutrino masses for different choices of box size and

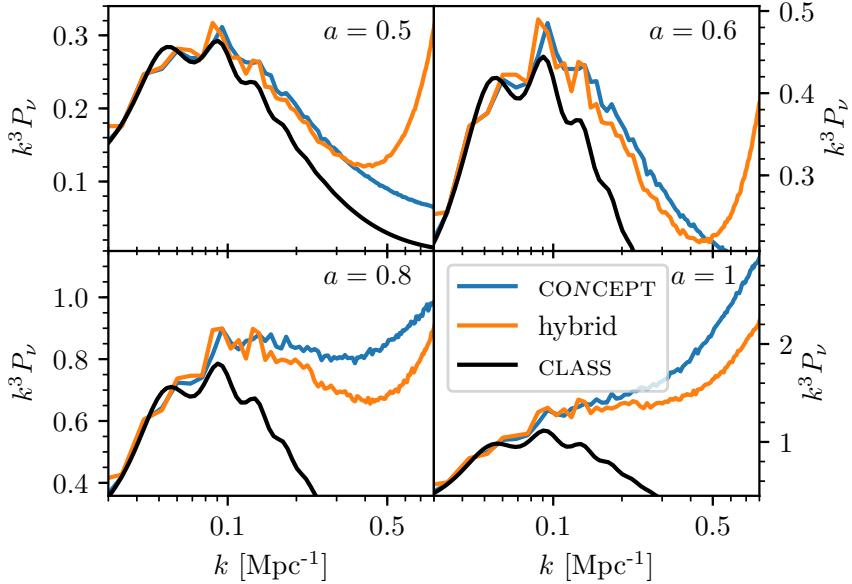


Figure 10.7 – Neutrino power spectra $\sum m_\nu = 1.2$ eV.

grid size.

Several different effects can be seen in this plot. The lowest resolution runs (512 Mpc/h box size and 600^3 grid) in general exhibit more power at large k than the other runs. This happens due to errors introduced by the CIC deconvolutions (as described in section 10.4.2), leading to more power in the neutrino component. It can be seen that this error is very small in the high resolution runs, and excellent convergence is achieved for 0.3 eV and 0.15 eV.

However, for the highest mass (1.2 eV) another effect plays a significant role. As described in subsection 10.5.1, errors arising from the failure of the PM method to resolve the actual gravitational potential leads to an increase in power at intermediate-to-high k and late times. This effect becomes even more significant when the grid size is increased, perhaps simply because of the much greater number of cells on which the error is introduced. The increased neutrino power at $a = 1$ seen in the high resolution runs for 1.2 eV is therefore in all likelihood unphysical. However, for masses more relevant to standard model neutrinos (~ 0.15 eV) this effect is absent and therefore even at this stage not a significant source of worry.

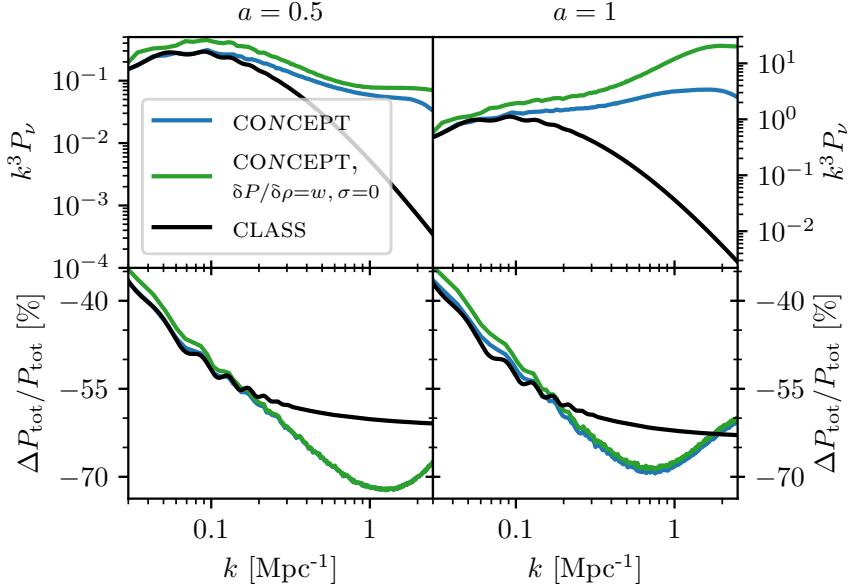


Figure 10.8 – Absolute neutrino power spectra (top) and total power spectra (matter and neutrinos) relative to Λ CDM (bottom), at $a = 0.5$ (left) and $a = 1.0$ (right), all for $\sum m_\nu = 1.2$ eV. The two CONCEPT simulations share the exact same initial conditions for both the matter particles and the neutrino fluid.

10.6 Conclusions

We have developed a new method for following neutrinos through non-linear clustering, based on the fully non-linear Boltzmann equation. Our solution is based on the equivalence of the momentum dependent Boltzmann equation to the velocity moment expansion of the same equation. Based on the assumption that moments of order v^2 and higher are mainly sourced by linear perturbations we have truncated the velocity hierarchy at this order. This amounts to solving the fully non-linear continuity and Euler equations for neutrinos, but with v^2 source terms derived from the linear perturbation theory solution.

At the starting point of the simulation these v^2 terms can be found simply from the transfer functions provided by CLASS and based on the same set of random numbers used to generate the density and velocity fields. However, at later times this method fails because terms such as δP and σ_j^i correlate with the density in the simulation at the given (not the initial) time. We have therefore developed a prescription for how

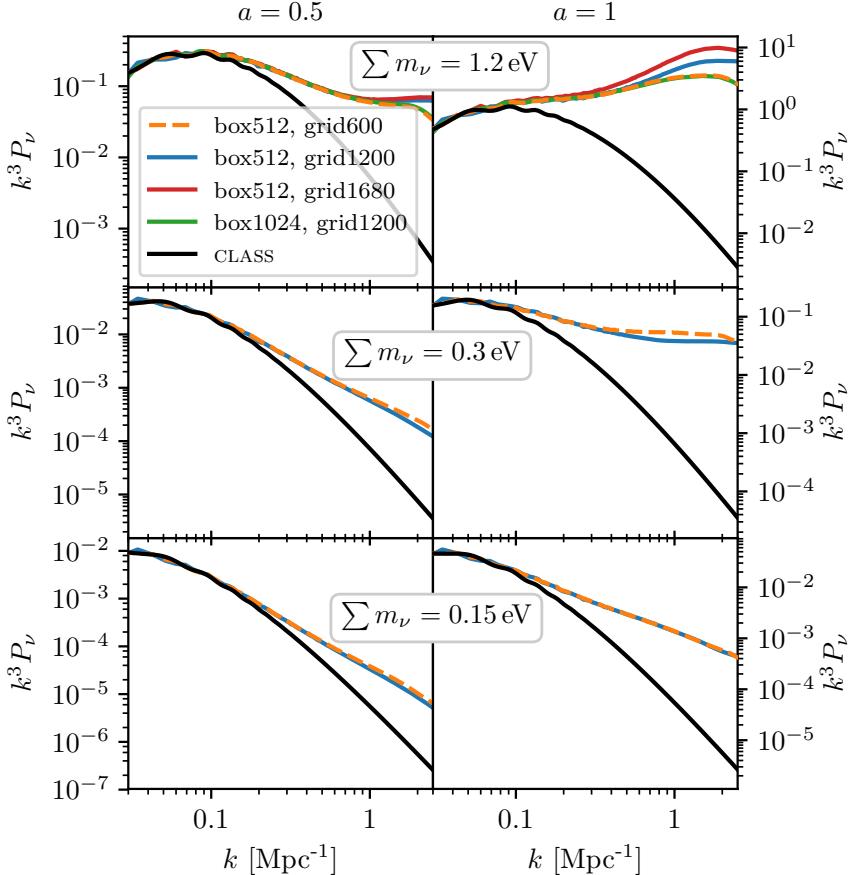


Figure 10.9 – Absolute neutrino power spectra at $a = 0.5$ (left) and $a = 1.0$ (right) for different box and grid sizes.

to generate v^2 terms correlating with the density field in the simulation and used these as source terms in the continuity and Euler equations.

We have found the method to be both very promising, and established that the method can be used to reliably calculate fully non-linear neutrino power spectra. Unlike most methods developed so far which work well for *either* small, *or* large neutrino masses, the method presented here has the potential to work equally well for all masses. We stress again that the bad convergence behaviour seen for the largest neutrino mass (1.2 eV) is most probable due to the limitations of the gravitational solver of CONCEPT, and *not* an inherent problem with the method of non-linear neutrino evolution proposed here.

Particle based methods in general suffer from noise related issues when neutrino masses are small because the thermal velocity component requires vast numbers of neutrino particles to sample properly. Conversely, linearised grid based methods break down for high neutrino masses because of non-linearities in the neutrino component. We have made a comparison between the method developed here and the hybrid method described in [14] and find that they agree well.

The platform for the implementation of neutrinos has been the CONCEPT code [33], which is made publicly available*. We have fully integrated the code with CLASS such that all linear theory calculations needed by the solver are provided by CLASS without additional input needed from the user.

Acknowledgements

We acknowledge computing resources from the Danish Center for Scientific Computing (DCSC). This work was supported by the Villum Foundation.

10.A N-body Realisations of the Boltzmann Hierarchy Variables

We will need to realise several Fourier space transfer functions in real space. The power spectrum of a quantity Y is related to the corresponding transfer function $P_Y(\tau, k)$ from CLASS by

$$P_Y(\tau, k) = 2\pi^2 Y^2(\tau, k) k^{-3} \mathcal{P}_\zeta(k), \quad (10.44)$$

$$= 2\pi^2 A_s Y^2(\tau, k) k^{-3} \left(\frac{k}{k_{\text{pivot}}} \right)^{n_s - 1}. \quad (10.45)$$

Let \mathcal{R} denote a realisation of a Gaussian random field with zero mean such that $\zeta(\mathbf{k}) = \zeta(k)\mathcal{R}(\mathbf{k})$, where ζ is the comoving curvature perturbation. The quantity $Y(\tau, \mathbf{x})$ in real space is then given as

$$Y(\tau, \mathbf{x}) = \mathcal{F}_{\mathbf{x}}^{-1} \left[\sqrt{P_Y(\tau, k)} \mathcal{R}(\mathbf{k}) \right] \quad (10.46)$$

$$= \mathcal{F}_{\mathbf{x}}^{-1} \left[Y(\tau, k) \sqrt{2\pi^2 A_s} k^{-\frac{3}{2}} \left(\frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s - 1}{2}} \mathcal{R}(\mathbf{k}) \right]. \quad (10.47)$$

*<https://github.com/jmd-dk/concept>

10.A.1 The Zel'dovich Approximation

The transfer function of the Lagrangian displacement field ψ for a given species is not directly available in CLASS. However, since $\mathbf{v} = \dot{\psi}$ the continuity equation for a non-relativistic species in N -body gauge [34] reads

$$\dot{\delta}(\tau, \mathbf{x}) = -\nabla \cdot \mathbf{v}(\tau, \mathbf{x}) = -\nabla \cdot \dot{\psi}(\tau, \mathbf{x}). \quad (10.48)$$

Using the boundary condition $\delta(0, \mathbf{x}) = 0$, $\psi(0, \mathbf{x}) = \mathbf{0}$, this equation can be integrated to give

$$\nabla \cdot \psi(\tau, \mathbf{x}) = -\delta(\tau, \mathbf{x}). \quad (10.49)$$

The divergence operator can be easily inverted in Fourier space if we introduce a scalar potential Υ such that $\psi(\tau, \mathbf{x}) = \nabla \Upsilon(\tau, \mathbf{x})$. The equation can then be written as

$$\nabla^2 \Upsilon(\tau, \mathbf{x}) = -\delta(\tau, \mathbf{x}). \quad (10.50)$$

In Fourier space we have $i k^j \Upsilon(\tau, \mathbf{k}) = \psi^j(\tau, \mathbf{k})$, which leads to

$$-k^2 \Upsilon(\tau, \mathbf{k}) = -\delta(\tau, \mathbf{k}) \quad (10.51)$$

$$\Rightarrow \Upsilon(\tau, \mathbf{k}) = \frac{\delta(\tau, \mathbf{k})}{k^2} \quad (10.52)$$

$$\Rightarrow \psi^j(\tau, \mathbf{k}) = \frac{i k^j}{k^2} \delta(\tau, \mathbf{k}), \quad (10.53)$$

where the last equality follows since an equation for quantities in Fourier space also holds for the corresponding transfer functions. Explicitly, this gives

$$\psi^j(\tau, \mathbf{x}) = \mathcal{F}_{\mathbf{x}}^{-1} \left[\left(\frac{i k^j}{k^2} \delta(\tau, \mathbf{k}) \right) \sqrt{2\pi^2 A_s} k^{-\frac{3}{2}} \left(\frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \mathcal{R}(\mathbf{k}) \right]. \quad (10.54)$$

10.A.2 Density and Velocity Fields

The density field can be directly realised from the transfer function $\delta(\tau, \mathbf{k})$:

$$\delta(\tau, \mathbf{x}) = \mathcal{F}_{\mathbf{x}}^{-1} \left[\delta(\tau, \mathbf{k}) \sqrt{2\pi^2 A_s} k^{-\frac{3}{2}} \left(\frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \mathcal{R}(\mathbf{k}) \right]. \quad (10.55)$$

Because CLASS solves for the divergence θ of the velocity field, we must invert the divergence operator in Fourier space like we do for the displacement field. We find

$$v^j(\tau, \mathbf{x}) = \mathcal{F}_{\mathbf{x}}^{-1} \left[\left(-\frac{ik^j}{k^2} \theta(\tau, k) \right) \sqrt{2\pi^2 A_s} k^{-\frac{3}{2}} \left(\frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \mathcal{R}(\mathbf{k}) \right], \quad (10.56)$$

by the identification $\psi^j \mapsto v^j$, $\delta \mapsto -\theta$ in equation (10.49). Note that (10.56) describes the velocity field of a fluid as well as the velocity field that should be used to set particle velocities when generating particle initial conditions using the Zel'dovich approximation.

10.A.3 Anisotropic Stress

CLASS solves for a quantity $\sigma(\tau, k)$ called the scalar anisotropic stress. Following [21] we define Σ^i_j as the trace-free contribution to the energy-momentum tensor*,

$$T^i_j(\tau, \mathbf{x}) = (\bar{P} + \delta P) \delta_i^j + \Sigma^i_j(\tau, \mathbf{x}). \quad (10.57)$$

As $\Sigma^i_j(\tau, \mathbf{x})$ is a symmetric, trace-free rank 2 tensor, it has 5 degrees of freedom: 2 tensor, 2 vector and one scalar degree of freedom. We can define the scalar potential γ implicitly by

$$\Sigma^i_j(\tau, \mathbf{x}) = \left(\nabla^i \nabla_j - \frac{1}{3} \delta^i_j \nabla^2 \right) \gamma(\tau, \mathbf{x}), \quad (10.58)$$

which in Fourier space becomes

$$\Sigma^i_j(\tau, \mathbf{k}) = -k^2 \left(\hat{\mathbf{k}}^i \hat{\mathbf{k}}_j - \frac{1}{3} \delta^i_j \right) \gamma(\tau, \mathbf{k}). \quad (10.59)$$

We can now compare this to the definition of σ (equation 22 in [21]):

$$\bar{\rho}(1+w)\sigma(\tau, \mathbf{k}) = - \left(\hat{\mathbf{k}}^j \hat{\mathbf{k}}_i - \frac{1}{3} \delta^j_i \right) \Sigma^i_j(\tau, \mathbf{k}) \quad (10.60)$$

$$= k^2 \left(\hat{\mathbf{k}}^j \hat{\mathbf{k}}_i \hat{\mathbf{k}}^i \hat{\mathbf{k}}_j + \frac{1}{9} \delta^j_i \delta^i_j - \frac{2}{3} \hat{\mathbf{k}}^j \hat{\mathbf{k}}_i \delta^i_j \right) \gamma(\tau, \mathbf{k}) \quad (10.61)$$

$$= \frac{2}{3} k^2 \gamma(\tau, \mathbf{k}). \quad (10.62)$$

*The $u^i u_j$ part present in (10.6) has been left out as we now work in linear theory.

The final expression then becomes

$$\Sigma^i_j(\tau, \mathbf{x}) = \mathcal{F}_x^{-1} \left[-\frac{3}{2} \bar{\rho}(1+w) \left(\hat{\mathbf{k}}^i \hat{\mathbf{k}}_j - \frac{1}{3} \delta^i_j \right) \sigma(\tau, k) \sqrt{2\pi^2 A_s} k^{-\frac{3}{2}} \left(\frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \mathcal{R}(\mathbf{k}) \right]. \quad (10.63)$$

By comparing equation (10.57) to the linearised version of equation (10.6), we find $\Sigma^i_j(\tau, \mathbf{x}) = \bar{\rho}(1+w)\sigma^i_j(\tau, \mathbf{x})$ leading to

$$\sigma^i_j(\tau, \mathbf{x}) = \mathcal{F}_x^{-1} \left[-\frac{3}{2} \left(\hat{\mathbf{k}}^i \hat{\mathbf{k}}_j - \frac{1}{3} \delta^i_j \right) \sigma(\tau, k) \sqrt{2\pi^2 A_s} k^{-\frac{3}{2}} \left(\frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \mathcal{R}(\mathbf{k}) \right]. \quad (10.64)$$

10.A.4 Non-linear Realisations

Since the linear pressure perturbation δP is a scalar, it can be realised in a manner similar to that of δ , (10.55):

$$\delta P(\tau, \mathbf{x}) = \mathcal{F}_x^{-1} \left[\delta P(\tau, k) \sqrt{2\pi^2 A_s} k^{-\frac{3}{2}} \left(\frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \mathcal{R}(\mathbf{k}) \right] \quad (10.65)$$

$$= \mathcal{F}_x^{-1} \left[\frac{\delta P(\tau, k)}{\delta(\tau, k)} \delta(\tau, \mathbf{k}) \right], \quad (10.66)$$

where $\delta(\tau, \mathbf{k}) = \mathcal{F}_x[\delta(\tau, \mathbf{x})]$ is simply the content of the bracket in (10.55). With this interpretation of $\delta(\tau, \mathbf{k})$, the resulting $\delta P(\tau, \mathbf{x})$ from (10.65) is purely linear. If we now upgrade $\delta(\tau, \mathbf{k})$ to be the non-linear density contrast present in the simulation at any time τ , (10.65) yields our estimate of the non-linear δP at any time τ . Comparing (10.65) with (10.66), we can write

$$\mathcal{R}(\tau, \mathbf{k}) = \frac{1}{\sqrt{2\pi^2 A_s}} k^{\frac{3}{2}} \left(\frac{k}{k_{\text{pivot}}} \right)^{\frac{1-n_s}{2}} \frac{\delta(\tau, \mathbf{k})}{\delta(\tau, k)}, \quad (10.67)$$

with $\mathcal{R}(\tau, \mathbf{k})$ being the time evolved random phases, coinciding with $\mathcal{R}(\mathbf{k})$ at the initialisation time. The time evolution of the estimated non-linear δP from (10.66) are then due to two effects: the time evolution of the linear transfer function of δP itself, as well as the non-linear time evolution of the underlying random field \mathcal{R} . Simply ignoring this last

effect and using the same $\mathcal{R}(\mathbf{k})$ throughout time leads to a mismatch between the actual and supposed phases, resulting in large errors.

Multiplying both $\delta(\tau, k)$ and $\delta(\tau, \mathbf{k})$ in (10.66) by $\bar{\rho}$, we can write the approximation generated by interpreting $\delta(\tau, \mathbf{k})$ as the non-linear density contrast as

$$\delta P_{\text{nl}}(\tau, \mathbf{k}) \simeq \frac{\delta P_l(\tau, k)}{\delta \rho_l(\tau, k)} \delta \rho_{\text{nl}}(\mathbf{k}), \quad (10.68)$$

where ‘l’ and ‘nl’ stands for ‘linear’ and ‘non-linear’, respectively. Thus the approximation corresponds to the assumption that $\delta P/\delta\rho$ (and hence the sound speed) is independent of the amplitude of the perturbations.

As we also want to realise $\sigma_j^i(\tau, \mathbf{x})$ throughout the simulation timespan, we similarly need an estimate of the non-linear $\sigma_j^i(\tau, \mathbf{x})$. Comparing (10.64) to (10.65), we see that the only difference is the factor $-3/2(\hat{\mathbf{k}}^i \hat{\mathbf{k}}_j - \delta_j^i/3)$, and so

$$\sigma_{j,\text{nl}}^i(\tau, \mathbf{k}) \simeq -\frac{3}{2} \left(\hat{\mathbf{k}}^i \hat{\mathbf{k}}_j - \frac{1}{3} \delta_j^i \right) \frac{\sigma_l(\tau, k)}{\delta \rho_l(\tau, k)} \delta \rho_{\text{nl}}(\tau, \mathbf{k}), \quad (10.69)$$

$$\sigma_{j,\text{nl}}^i(\tau, \mathbf{x}) = \mathcal{F}_{\mathbf{x}}^{-1} \left[\sigma_{j,\text{nl}}^i(\tau, \mathbf{k}) \right]. \quad (10.70)$$

Similarly, this approximation corresponds to the assumption that $\sigma/\delta\rho$ is independent of the amplitude of the perturbations. Since σ_j^i has the same velocity order as δP , the two approximations should be equally valid.

Interpreting (10.67) as being solely the evolved phases disregards the fact that on top of the shifting phases we also have the non-linear growth of $\delta(|\mathbf{k}|)$. Thus, equations (10.68) and (10.69) do not only supply δP_{nl} and $\sigma_{j,\text{nl}}^i$ with the correctly evolved phases, but inevitably also injects non-linearity. This can be avoided by replacing the linear transfer function $\delta\rho(\tau, k)$ with its non-linear counterpart, namely the square root of the non-linear power spectrum:

$$\delta P_{\text{nl}}(\tau, \mathbf{k}) \simeq \sqrt{\frac{P_{\delta P_l}(\tau, k)}{P_{\delta \rho_{\text{nl}}}(\tau, k)}} \delta \rho_{\text{nl}}(\tau, \mathbf{k}), \quad (10.71)$$

$$\sigma_{j,\text{nl}}^i(\tau, \mathbf{k}) \simeq -\frac{3}{2} \left(\hat{\mathbf{k}}^i \hat{\mathbf{k}}_j - \frac{1}{3} \delta_j^i \right) \sqrt{\frac{P_{\sigma_l}(\tau, k)}{P_{\rho_{\text{nl}}}(\tau, k)}} \delta \rho_{\text{nl}}(\tau, \mathbf{k}), \quad (10.72)$$

where the transfer function of the target variable also has been replaced by the (linear) power spectrum,

$$P_{Y_l}(\tau, k) = 2\pi^2 A_s k^{-3} \left(\frac{k}{k_{\text{pivot}}} \right)^{n_s-1} Y^2(\tau, k), \quad (10.73)$$

in order to cancel out the factors otherwise introduced by exchanging a transfer functions for a (square root of a) power spectrum. Equations (10.71) and (10.72) *can* then be used in place of (10.68) and (10.69). This leaves us with two separate realisation schemes with no obvious best choice.

Since what we really realize is $\zeta_j^i \propto \Sigma_j^i = (\rho + P)\sigma_j^i$, we further have the choice of whether to use $\bar{\rho}(1+w)$ as in (10.63) (inside or outside of the Fourier transform) or the non-linear $\rho + P$ (outside the Fourier transform), as in

$$\Sigma_{j,\text{nl}}^i(\tau, \mathbf{k}) \simeq (\rho_{\text{nl}}(\tau, \mathbf{k}) + P_{\text{nl}}(\tau, \mathbf{k})) \mathcal{F}_x^{-1} \left[-\frac{3}{2} \left(\hat{\mathbf{k}}^i \hat{\mathbf{k}}_j - \frac{1}{3} \delta^i_j \right) \frac{\sigma_1(\tau, k)}{\delta\rho_1(\tau, k)} \delta\rho_{\text{nl}}(\tau, \mathbf{k}) \right], \quad (10.74)$$

again with the further possibility of replacing transfer functions with power spectra. Once again, these choices come down to whether we wish to further inject non-linearity into Σ_j^i . We generally achieved better results with this added non-linearity, and all plots in this paper have been produced using (10.68) and (10.74) as non-linear realisation schemes. Further studying of these different realization schemes — which are all equivalent in linear theory but differ in non-linear theory — would be very interesting.

10.B Comparison between CLASS and CAMB

Neither CLASS nor CAMB produce accurate neutrino transfer functions at their default precision settings. The reason is simply that these precision settings are tuned for the total matter power spectrum which is usually dominated by the cold matter. The precision of the neutrino evolution is mainly controlled by the number of momentum bins, N_q and the cut-off in the Boltzmann hierarchy $\ell_{\max,\nu}$.

The momentum sampling q in CLASS is automatic as discussed in detail in [26]. An optimal choice of quadrature method is found by requiring that the distribution function multiplied by a test-function can be computed at sufficient accuracy. This has the virtue of being independent of the actual distribution function. However, in order to obtain high-precision neutrino transfer functions from CLASS, we had to use the manual quadrature strategy which was introduced in CLASS v2.6.2. We use **quadrature strategy = 3** which means that CLASS uses a trapezoidal rule on a uniform grid from 0 to q_{\max} with $N_q + 1$

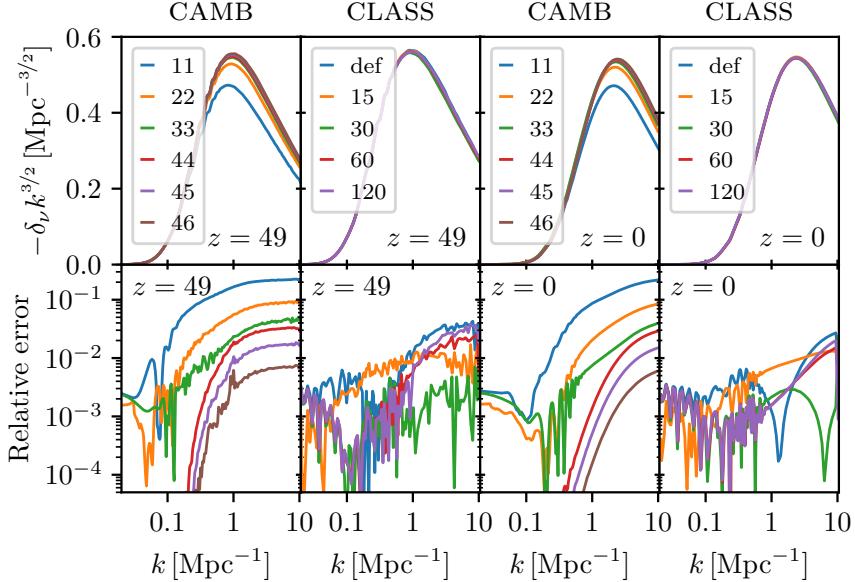


Figure 10.10 – The density transfer function $\delta_\nu(k)$ multiplied by $-k^{3/2}$ for $\sum m_\nu = 1.2 \text{ eV}$ at redshift $z = 49$ (left) and $z = 0$ (right). For CAMB, the first digit corresponds to `accuracy_boost` and the second to `laccuracy_boost`. For CLASS, the legend denotes the value of both N_q and $\ell_{\max,\nu}$, except for ‘def’ which denotes the default settings.

points. The first point at $q = 0$ is not actually evolved since all integrands would anyway vanish.

For the high-precision CLASS runs we turn off the fluid approximation by setting the parameter `ncdm_fluid_approximation = 3`, and this requires us to increase the ℓ_{\max} cutoff in the Boltzmann hierarchy considerably since the lowest multipoles now have more time to get polluted by the unphysical reflection of power at ℓ_{\max} .

The agreement between CLASS and CAMB for the neutrino transfer functions has never been checked in any detail, so we have conducted a preliminary convergence test for the case $\sum m_\nu = 1.2 \text{ eV}$. As we have shown in figure 10.10, the two codes can be brought into agreement at the 1%-level.

We matched the cosmological model in the two codes and used the precision settings given in table 10.4. For the CAMB-runs, the first and second digit in the legend of figure 10.10 refers to the value of `accuracy_boost` and `l_accuracy_boost`, respectively. For the CLASS-runs the two precision parameters, `Number of momentum bins` and

Table 10.4 – Precision settings used in the CAMB and CLASS comparison runs.

CAMB	CLASS
<code>high_accuracy_default = T</code>	<code>ncdm_fluid_approximation = 3</code>
<code>transfer_high_precision = T</code>	<code>Quadrature strategy = 3</code>
<code>massive_nu_approx = 0</code>	<code>Maximum q = 15</code>
<code>accuracy_boost = 1-4</code>	<code>Number of momentum bins = 15 (30, 60, 120)</code>
<code>l_accuracy_boost = 1-7</code>	<code>l_max_ncdm = 15 (30, 60, 120)</code>

`l_max_ncdm` were set to the same value and varied together, and the legend refers to the value of both.

We used a 47 CAMB run (not included in the figure) as the common reference for computing the relative error of both codes. We see that the 11 setting of CAMB generates a 20% error. We emphasise that this is not a problem for standard cosmological analyses, but it could be an issue when used as initial condition for neutrino simulations. For the CLASS runs we see that the error seems to increase for large k when the precision is increased beyond $\ell_{\max,\nu} = N_q = 30$, which we take as an indication that the 47 CAMB-run is not yet numerically converged. An agreement at the 1%-level that we have established is however enough for the present implementation.

Bibliography

- [1] R. Laureijs *et al.* [EUCLID Collaboration], “Euclid Definition Study Report,” arXiv:1110.3193 [astro-ph.CO].
- [2] P. A. Abell *et al.* [LSST Science and LSST Project Collaborations], “LSST Science Book, Version 2.0,” arXiv:0912.0201 [astro-ph.IM].
- [3] J. Brandbyge, S. Hannestad, T. Haugbølle and B. Thomsen, “The Effect of Thermal Neutrino Motion on the Non-linear Cosmological Matter Power Spectrum,” JCAP **0808** (2008) 020 [arXiv:0802.3700 [astro-ph]].

- [4] M. Viel, M. G. Haehnelt and V. Springel, “The effect of neutrinos on the matter distribution as probed by the Intergalactic Medium,” *JCAP* **1006** (2010) 015 [arXiv:1003.2422 [astro-ph.CO]].
- [5] S. Agarwal and H. A. Feldman, “The effect of massive neutrinos on the matter power spectrum,” *Mon. Not. Roy. Astron. Soc.* **410** (2011) 1647 [arXiv:1006.0689 [astro-ph.CO]].
- [6] S. Bird, M. Viel and M. G. Haehnelt, “Massive Neutrinos and the Non-linear Matter Power Spectrum,” *Mon. Not. Roy. Astron. Soc.* **420** (2012) 2551 [arXiv:1109.4416 [astro-ph.CO]].
- [7] F. Villaescusa-Navarro, F. Marulli, M. Viel, E. Branchini, E. Castorina, E. Sefusatti and S. Saito, “Cosmology with massive neutrinos I: towards a realistic modeling of the relation between matter, haloes and galaxies,” *JCAP* **1403** (2014) 011 [arXiv:1311.0866 [astro-ph.CO]].
- [8] E. Castorina, C. Carbone, J. Bel, E. Sefusatti and K. Dolag, “DEMNUni: The clustering of large-scale structures in the presence of massive neutrinos,” *JCAP* **1507** (2015) no.07, 043 [arXiv:1505.07148 [astro-ph.CO]].
- [9] J. D. Emberson *et al.*, “Cosmological neutrino simulations at extreme scale,” *Res. Astron. Astrophys.* **17** (2017) no.8, 085 [arXiv:1611.01545 [astro-ph.CO]].
- [10] J. Adamek, R. Durrer and M. Kunz, “Relativistic N-body simulations with massive neutrinos,” *JCAP* **1711** (2017) no.11, 004 [arXiv:1707.06938 [astro-ph.CO]].
- [11] J. Brandbyge and S. Hannestad, “Grid Based Linear Neutrino Perturbations in Cosmological N-body Simulations,” *JCAP* **0905** (2009) 002 [arXiv:0812.3149 [astro-ph]].
- [12] Y. Ali-Haimoud and S. Bird, “An efficient implementation of massive neutrinos in non-linear structure formation simulations,” *Mon. Not. Roy. Astron. Soc.* **428** (2012) 3375 [arXiv:1209.0461 [astro-ph.CO]].
- [13] J. Liu, S. Bird, J. M. Z. Matilla, J. C. Hill, Z. Haiman, M. S. Madhavacheril, A. Petri and D. N. Spergel, “MassiveNuS: Cosmological Massive Neutrino Simulations,” arXiv:1711.10524 [astro-ph.CO].

- [14] J. Brandbyge and S. Hannestad, “Resolving Cosmic Neutrino Structure: A Hybrid Neutrino N-body Scheme,” *JCAP* **1001** (2010) 021 [arXiv:0908.1969 [astro-ph.CO]].
- [15] N. N. Bogolyubov, *J. Phys. (USSR)* **10**, 256 (1946).
- [16] H. Born, H. S. Green, *Proc. Roy. Soc. A* **188**, 10 (1946).
- [17] J. G. Kirwood, *J. Chem. Phys.* **14**, 180 (1946).
- [18] A. Banerjee and N. Dalal, “Simulating nonlinear cosmological structure formation with massive neutrinos,” *JCAP* **1611** (2016) no.11, 015 [arXiv:1606.06167 [astro-ph.CO]].
- [19] F. Debbasch and W.A. van Leeuwen, “General relativistic Boltzmann equation, I: Covariant treatment,” *Physica A*, 388 (2009), 1079-1104.
- [20] F. Debbasch and W.A. van Leeuwen, “General relativistic Boltzmann equation, II: Manifestly covariant treatment,” *Physica A*, 388 (2009), 1079-1104.
- [21] C. P. Ma and E. Bertschinger, “Cosmological perturbation theory in the synchronous and conformal Newtonian gauges,” *Astrophys. J.* **455** (1995) 7 [astro-ph/9506072].
- [22] D. Blas, J. Lesgourgues and T. Tram, “The Cosmic Linear Anisotropy Solving System (CLASS) II: Approximation schemes,” *JCAP* **1107** (2011) 034 [arXiv:1104.2933 [astro-ph.CO]].
- [23] J. Lesgourgues, “The Cosmic Linear Anisotropy Solving System (CLASS) III: Comparision with CAMB for LambdaCDM,” arXiv:1104.2934 [astro-ph.CO].
- [24] A. Lewis, A. Challinor and A. Lasenby, “Efficient computation of CMB anisotropies in closed FRW models,” *Astrophys. J.* **538** (2000) 473 [astro-ph/9911177].
- [25] A. Lewis and S. Bridle, “Cosmological parameters from CMB and other data: A Monte Carlo approach,” *Phys. Rev. D* **66** (2002) 103511 [astro-ph/0205436].
- [26] J. Lesgourgues and T. Tram, “The Cosmic Linear Anisotropy Solving System (CLASS) IV: efficient implementation of non-cold relics,” *JCAP* **1109** (2011) 032 [arXiv:1104.2935 [astro-ph.CO]].

- [27] R. W. MacCormack, “The effect of viscosity in hypervelocity impact cratering,” *Frontiers of Computational Fluid Dynamics* (1969).
- [28] V. Springel, “The Cosmological simulation code GADGET-2,” *Mon. Not. Roy. Astron. Soc.* **364** (2005) 1105 [[astro-ph/0505010](#)].
- [29] D. Liang, B. Lin and R. A. Falconer, “Simulation of rapidly varying flow using an efficient tvd-maccormack scheme,” *International journal for numerical methods in fluids*, 53(5):811–826 (2007).
- [30] J. Machalinska-Murawska, M. Szydłowski, “Lax-Wendroff and McCormack Schemes for Numerical Simulation of Unsteady Gradually and Rapidly Varied Open Channel Flow,” *Archives of Hydro-Engineering and Environmental Mechanics*, 60(1–4):51–62 (2014).
- [31] A. Kurganov and E. Tadmor, “New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations,” *Journal of Computational Physics* (2000).
- [32] S. Hannestad, “Structure formation with strongly interacting neutrinos - Implications for the cosmological neutrino mass bound,” *JCAP* **0502** (2005) 011 [[astro-ph/0411475](#)].
- [33] J. Dakin, “CONCEPT - The COsmological N -body CodE in PyThon,” [arXiv:1510.07621](#) [[astro-ph.CO](#)].
- [34] C. Fidler, C. Rampf, T. Tram, R. Crittenden, K. Koyama and D. Wands, *Phys. Rev. D* **92** (2015) no.12, 123517
doi:[10.1103/PhysRevD.92.123517](#) [[arXiv:1505.04756](#) [[astro-ph.CO](#)]].

11 Conclusions and Outlook

Conclusions

My PhD work can broadly be partitioned into two major parts. The first part is my work on the inclusion of linear species (in particular massive neutrinos) into conventional N -body codes, providing the otherwise missing interactions and general relativistic effects on large scales. As demonstrated in the paper of chapter 9, we have successfully carried out implementations into both CONCEPT and PKDGRAV, yielding results that agree extraordinary well with one another and with the linear theory predictions on large scales. The N -body gauge framework together with the realisation of linear species is then a doable and accurate method for incorporating the effects of both linear species and general relativistic effects into conventional N -body codes without any real modifications to the Newtonian framework of such nodes.

The other major part of my PhD work was concerned with the inclusion of non-linear species (apart from matter itself) into N -body codes. This can be seen as an extension to the above where the assumption of linearity is removed. As this only happens appreciably in the case of massive neutrinos, we tend to think of this as being specific to that species, but really the method developed can be applied to any species.

Unlike in the linear case, the non-linear treatment contains within it several seemingly arbitrary choices, prime among which are where to cut the Boltzmann hierarchy into a non-linear and a linear part (choosing ℓ_{nl}) and which of the many non-linear realisation schemes of section 7.4 to pick for the realisations of the $\ell = \ell_{\text{nl}} + 1$ variables. One can view the linear scheme as the limiting case of having $\ell_{\text{nl}} = -1$, meaning that the evolution of the given species is entirely linear. Once the energy density of a species is realised on a grid, applying a gravitational kick to the particles due to this grid is exactly the same as for the non-linear case.

In the paper of chapter 10 we demonstrated that $\ell_{\text{nl}} = 1$ and the non-linear realisation scheme of (7.38) and (7.39) leads to results comparable to that of the ‘hybrid’ code. Whereas this code becomes very computationally demanding for small (realistic) neutrino masses, the

performance of the method developed during my PhD and implemented in CONCEPT is largely invariant to the neutrino mass.

For large neutrino masses ($\sum m_\nu = 1.2 \text{ eV}$) where CONCEPT and the hybrid code can be compared, an unfortunate defect in the non-linear neutrino evolution occurs in CONCEPT, as seen in figure 10.9. This defect does not go away but rather intensifies as the grid resolution is increased. Though we might speculate on the nature of this defect, the honest answer is that we do not understand why it occurs. Since no bad behaviour is seen for realistic neutrino masses, I am tempted to ignore this problem.

Though the non-linear light neutrino power spectra of e.g. figure 6.1 and 10.9 look convincing, we have still to directly compare them to neutrino spectra of other codes. The depth of the trough in the relative matter power spectrum at $a = 1$ as seen on figure 10.6, does agree with the non-linear prediction $\sim -10\Omega_\nu/\Omega_m$ of [3].

Outlook

As mentioned in the introduction of this thesis, I will remain in the cosmology group of Steen Hannestad for the next six months as a postdoc. Though it has not been a major issue during my PhD, it would be very good to implement ‘short-range’ particle forces into CONCEPT, so that the gravitational interaction between particles is resolved well below the grid scale of the PM method, making for more precise power spectra at high k . Implementing an optimal P³M method into CONCEPT during these six month is thus of high priority.

The implementation of the linear massive neutrinos and other linear species into CONCEPT is very much complete. The only extension to this method presented in this thesis which is not implemented is that of the linear option of (7.34), corresponding to evolving the “random” field \mathcal{R} through time based on the evolution of matter, without also imprinting the growth of matter into \mathcal{R} . This should result in better small-scale behaviour of late-time linear realisations, as structure of the linear field now follows the non-linearly evolved matter structure, rather than using the same primordial structure throughout time. As the linear species do not contribute much at small scales, this addition should not be critically important, but it would be nice to implement and see what difference this arguably improved procedure makes.

Though the paper of chapter 9 demonstrates that the implementation of linear species in PKDGRAV is fully functional, there is still some work required in order for it to play nicely with the advanced multi-level time

stepping employed by PKDGRAV. This work is primarily carried out by Joachim Stadel, Douglas Potter and Mischa Knabenhans at the Institute for Computational Science, University of Zürich. As I originally wrote this linear implementation into PKDGRAV and also am responsible for the CONCEPT CLASS utility providing the transfer functions needed for both the linear species and the initial conditions of the matter particles, our collaboration continues. Though I do not plan to further develop the linear implementation in PKDGRAV, the guys in Zürich will need me to perform comparison tests between PKDGRAV and CONCEPT. The plan is to run PKDGRAV in this ‘CLASS’ mode — including linear kicks from photons, a full neutrino mass hierarchy and metric perturbations — for the Euclid flagship 2 simulation, but this hinges on getting the final issues ironed out in time.

Tending to the non-linear treatment of massive neutrinos, I would very much like to implement a larger set of the possibilities explored in section 7.4 into CONCEPT and perform systematic tests to see how they compare. As they are all equivalent at the linear level, predicting how they differ among each other at the non-linear level is extremely difficult without such direct implementations.

One obvious member from the above mentioned set of tests is to compare the resulting matter power spectra from simulations using linearly realised and non-linear neutrinos. In particular this should be done for realistic, small neutrino masses — e.g. $\sum m_\nu = 0.06 \text{ eV}$ as will be used for the Euclid flagship simulation — in order to quantify how important the non-linear treatment of neutrinos actually is for such low masses.

Throughout this thesis and indeed during the whole of my PhD, I have kept the linear method for including photons, massive neutrinos and the metric into N -body simulations somewhat separate from the inclusion of non-linear massive neutrinos, at least conceptually. Ultimately though, these should come together so that simulations can be run with linear photons and metric perturbations but with non-linear neutrinos*. In its present stage CONCEPT is fully capable of running such simulations, mixing the linear and non-linear methods. This really comes down to the fact that the linear method is contained within the non-linear method as its limit. From the point of view of the CONCEPT implementation, there is no real distinction and only a single method exists. To carry out such simulations fully self-consistently, the non-linear neutrinos should now

*For a non-degenerate neutrino mass hierarchy, perhaps only one or two of the mass states needs to be treated non-linear.

also receive kicks from the remaining linear species. This is trivial with regards to the photons and amongst the individual neutrino mass states, but it is unclear whether the same ‘metric species’ that provides matter with general relativistic corrections may also be used to supply general relativistic corrections to the neutrinos or other species in general.

Bibliography

- [1] Paul A. Abell et al. LSST Science Book, Version 2.0. 2009.
- [2] R. Laureijs et al. Euclid Definition Study Report. 2011.
- [3] Jacob Brandbyge, Steen Hannestad, Troels Haugbølle, and Bjarne Thomsen. The Effect of Thermal Neutrino Motion on the Non-linear Cosmological Matter Power Spectrum. *JCAP*, 0808:020, 2008.
- [4] Matteo Viel, Martin G. Haehnelt, and Volker Springel. The effect of neutrinos on the matter distribution as probed by the Intergalactic Medium. *JCAP*, 1006:015, 2010.
- [5] Shankar Agarwal and Hume A Feldman. The effect of massive neutrinos on the matter power spectrum. *Monthly Notices of the Royal Astronomical Society*, 410(3):1647–1654, 2011.
- [6] Simeon Bird, Matteo Viel, and Martin G Haehnelt. Massive neutrinos and the non-linear matter power spectrum. *Monthly Notices of the Royal Astronomical Society*, 420(3):2551–2561, 2012.
- [7] Francisco Villaescusa-Navarro, Federico Marulli, Matteo Viel, Enzo Branchini, Emanuele Castorina, Emiliano Sefusatti, and Shun Saito. Cosmology with massive neutrinos i: towards a realistic modeling of the relation between matter, haloes and galaxies. *Journal of Cosmology and Astroparticle Physics*, 2014(03):011, 2014.
- [8] Emanuele Castorina, Carmelita Carbone, Julien Bel, Emiliano Sefusatti, and Klaus Dolag. Demnuni: the clustering of large-scale structures in the presence of massive neutrinos. *Journal of Cosmology and Astroparticle Physics*, 2015(07):043, 2015.
- [9] JD Emberson, Hao-Ran Yu, Derek Inman, Tong-Jie Zhang, Ue-Li Pen, Joachim Harnois-Déraps, Shuo Yuan, Huan-Yu Teng, Hong-Ming Zhu, Xuelei Chen, et al. Cosmological neutrino simulations at extreme scale. *Research in Astronomy and Astrophysics*, 17(8):085, 2017.

- [10] Julian Adamek, Ruth Durrer, and Martin Kunz. Relativistic n-body simulations with massive neutrinos. *Journal of Cosmology and Astroparticle Physics*, 2017(11):004, 2017.
- [11] Jacob Brandbyge and Steen Hannestad. Grid Based Linear Neutrino Perturbations in Cosmological N-body Simulations. *JCAP*, 0905:002, 2009.
- [12] Jacob Brandbyge, Cornelius Rampf, Thomas Tram, Florent Leclercq, Christian Fidler, and Steen Hannestad. Cosmological N -body simulations including radiation perturbations. *Mon. Not. Roy. Astron. Soc.*, 466:L68–L72, 2017.
- [13] Yacine Ali-Haïmoud and Simeon Bird. An efficient implementation of massive neutrinos in non-linear structure formation simulations. *Monthly Notices of the Royal Astronomical Society*, 428(4):3375–3389, 2012.
- [14] Jia Liu, Simeon Bird, José Manuel Zorrilla Matilla, J Colin Hill, Zoltán Haiman, Mathew S Madhavacheril, Andrea Petri, and David N Spergel. Massivenus: cosmological massive neutrino simulations. *Journal of Cosmology and Astroparticle Physics*, 2018(03):049, 2018.
- [15] Jacob Brandbyge and Steen Hannestad. Resolving cosmic neutrino structure: a hybrid neutrino n-body scheme. *Journal of Cosmology and Astroparticle Physics*, 2010(01):021, 2010.
- [16] Jeppe Dakin. CONCEPT - The COsmological N -body CodE in PyThon. 2015.
- [17] Diego Blas, Julien Lesgourges, and Thomas Tram. The cosmic linear anisotropy solving system (class). part ii: approximation schemes. *Journal of Cosmology and Astroparticle Physics*, 2011(07):034, 2011.
- [18] Jeppe Dakin, Jacob Brandbyge, Steen Hannestad, Troels Haugbølle, and Thomas Tram. ν CONCEPT: Cosmological neutrino simulations from the non-linear Boltzmann hierarchy. 2017.
- [19] Romain Teyssier. Cosmological hydrodynamics with adaptive mesh refinement: a new high resolution code called ramses. *Astron. Astrophys.*, 385:337–364, 2002.

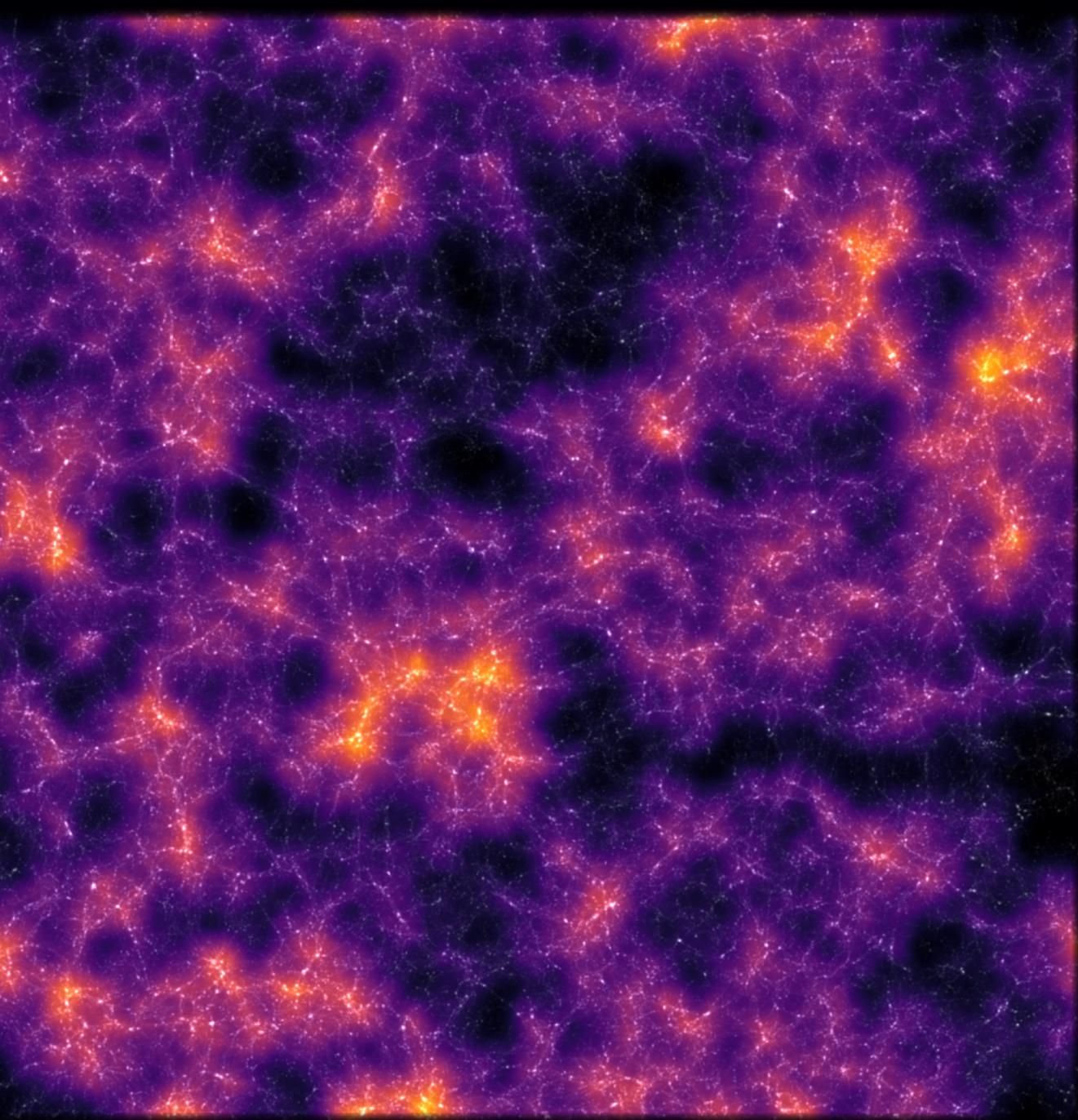
- [20] Douglas Potter, Joachim Stadel, and Romain Teyssier. PKDGRAV3: Beyond Trillion Particle Cosmological Simulations for the Next Era of Galaxy Surveys. 2016.
- [21] Thomas Tram, Jacob Brandbyge, Jeppe Dakin, and Steen Hannestad. Fully relativistic treatment of light neutrinos in N -body simulations. 2018.
- [22] Jeppe Mosgaard Dakin. Computing Dark Universes — Cosmological N -body Simulations of Dark Matter. Master’s thesis, Department of Physics and Astronomy, Aarhus University, Denmark, June 2015. http://users-phys.au.dk/jmd/github/concept/masters_thesis.pdf.
- [23] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31–39, 2011.
- [24] N. Aghanim et al. Planck 2018 results. VI. Cosmological parameters. 2018.
- [25] Y. Fukuda et al. Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.*, 81:1562–1567, 1998.
- [26] DJ Fixsen. The temperature of the cosmic microwave background. *The Astrophysical Journal*, 707(2):916, 2009.
- [27] Scott Dodelson. *Modern Cosmology*. Academic Press, Elsevier Science, 2003.
- [28] Gianpiero Mangano, Gennaro Miele, Sergio Pastor, Teguayco Pinto, Ofelia Pisanti, and Pasquale D. Serpico. Relic neutrino decoupling including flavor oscillations. *Nucl. Phys.*, B729:221–234, 2005.
- [29] P. F. de Salas, D. V. Forero, C. A. Ternes, M. Tortola, and J. W. F. Valle. Status of neutrino oscillations 2018: 3σ hint for normal mass ordering and improved CP sensitivity. *Phys. Lett.*, B782:633–640, 2018.
- [30] Chung-Pei Ma and Edmund Bertschinger. Cosmological perturbation theory in the synchronous and conformal Newtonian gauges. *Astrophys. J.*, 455:7–25, 1995.
- [31] Jacob Brandbyge and Steen Hannestad. Fourier imaging of non-linear structure formation. *JCAP*, 1704(04):032, 2017.

- [32] F Debbasch and WA van Leeuwen. General relativistic boltzmann equation, i: Covariant treatment. *Physica A: Statistical Mechanics and its Applications*, 388(7):1079–1104, 2009.
- [33] F Debbasch and WA van Leeuwen. General relativistic boltzmann equation, ii: Manifestly covariant treatment. *Physica A: Statistical Mechanics and its Applications*, 388(9):1818–1834, 2009.
- [34] Antony Lewis, Anthony Challinor, and Anthony Lasenby. Efficient computation of CMB anisotropies in closed FRW models. *Astrophys. J.*, 538:473–476, 2000.
- [35] Volker Springel. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, 364, 2005.
- [36] Julian Adamek, David Daverio, Ruth Durrer, and Martin Kunz. gevolution: a cosmological N-body code based on General Relativity. *JCAP*, 1607(07):053, 2016.
- [37] P. P. Ewald. Die berechnung optischer und electrostatischer gitterpotentiale. *Annalen der Physik*, 369, 1921.
- [38] L. Hernquist, F. R. Bouchet, and Y. Suto. Application of the Ewald method to cosmological N-body simulations. *The Astrophysical Journal*, 75:234, February 1991.
- [39] H. C. Plummer. On the problem of distribution in globular star clusters. *Monthly Notices of the Royal Astronomical Society*, 71:460–470, March 1911.
- [40] A. Knebe. Lecture notes *Computational Astrophysics: Physical Processes*, 2005.
- [41] E. Puchwein and B. Moster. Lecture notes *Cosmic Structure formation on Supercomputers (and laptops), Lecture 2: Gravity solvers and parallelization*, 2014.
- [42] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.
- [43] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. Taylor & Francis, Inc., Bristol, PA, USA, 1988.

- [44] Thomas Sauer and Yuan Xu. On multivariate lagrange interpolation. *MATH. COMP.*, 64:1147–1170, 1994.
- [45] Matteo Frigo, Steven, and G. Johnson. The design and implementation of FFTW3. In *Proceedings of the IEEE*, pages 216–231, 2005.
- [46] Ralf Klessen. Grapeshph with fully periodic boundary conditions: fragmentation of molecular clouds. *Mon. Not. Roy. Astron. Soc.*, 292:11, 1997.
- [47] Marsha J Berger and Phillip Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of computational Physics*, 82(1):64–84, 1989.
- [48] William H Press. *Numerical recipes in Pascal: the art of scientific computing*, volume 1. Cambridge University Press, 1989.
- [49] Peter Lax and Burton Wendroff. Systems of conservation laws. *Communications on Pure and Applied mathematics*, 13(2):217–237, 1960.
- [50] R. W. Maccormack. The effect of viscosity in hypervelocity impact cratering. *AIAA, American Institute of Aeronautics and Astrophysics*, 1969.
- [51] Justyna Machalinska-Murawska and Michał Szydłowski. Lax-wendroff and mcccormack schemes for numerical simulation of unsteady gradually and rapidly varied open channel flow. *Archives of Hydro-Engineering and Environmental Mechanics*, 60(1-4):51–62, 2014.
- [52] Alexander Kurganov and Eitan Tadmor. New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations. *Journal of Computational Physics*, 160(1):241–282, 2000.
- [53] B Van Leer and PR Woodward. The muscl code for compressible flow: philosophy and results. *TICOM Conference*, 1979.
- [54] Sergei K. Godunov. A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. *Matematicheskii Sbornik*, 47(07), 1959.
- [55] Ami Harten. High resolution schemes for hyperbolic conservation laws. *Journal of computational physics*, 49(3):357–393, 1983.

- [56] Peter K Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM journal on numerical analysis*, 21(5):995–1011, 1984.
- [57] PK Sweby and MJ Baines. On convergence of roe’s scheme for the general non-linear scalar wave equation. *Journal of Computational Physics*, 56(1):135–148, 1984.
- [58] Philip L Roe. Some contributions to the modelling of discontinuous flows. In *Large-scale computations in fluid mechanics*, pages 163–193, 1985.
- [59] Bram Van Leer. Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme. *Journal of computational physics*, 14(4):361–370, 1974.
- [60] B Falck, N McCullagh, MC Neyrinck, J Wang, and AS Szalay. The effect of corner modes in the initial conditions of cosmological simulations. *The Astrophysical Journal*, 837(2):181, 2017.
- [61] YB Zeldovich. Gravitational instability: An approximate theory for large density perturbations. 1970. *A&A*, 5(84):12.
- [62] Phillip James Edwin Peebles. *The large-scale structure of the universe*. Princeton university press, 1980.
- [63] Adrian Jenkins. Second-order lagrangian perturbation theory initial conditions for resimulations. *Monthly Notices of the Royal Astronomical Society*, 403(4):1859–1872, 2010.
- [64] Christian Fidler, Cornelius Rampf, Thomas Tram, Robert Crittenden, Kazuya Koyama, and David Wands. General relativistic corrections to N -body simulations and the Zel’dovich approximation. *Phys. Rev.*, D92(12):123517, 2015.
- [65] Christian Fidler, Thomas Tram, Cornelius Rampf, Robert Crittenden, Kazuya Koyama, and David Wands. General relativistic weak-field limit and Newtonian N-body simulations. *JCAP*, 1712(12):022, 2017.
- [66] Thomas Tram, private communication.
- [67] Julian Adamek, Jacob Brandbyge, Christian Fidler, Steen Hannestad, Cornelius Rampf, and Thomas Tram. The effect of early radiation in N -body simulations of cosmic structure formation. *Mon. Not. Roy. Astron. Soc.*, 470(1):303–313, 2017.

- [68] Alexander Kurganov, Sebastian Noelle, and Guergana Petrova. Semidiscrete central-upwind schemes for hyperbolic conservation laws and hamilton–jacobi equations. *SIAM Journal on Scientific Computing*, 23(3):707–740, 2001.



**Department of Physics and Astronomy
Aarhus University**