

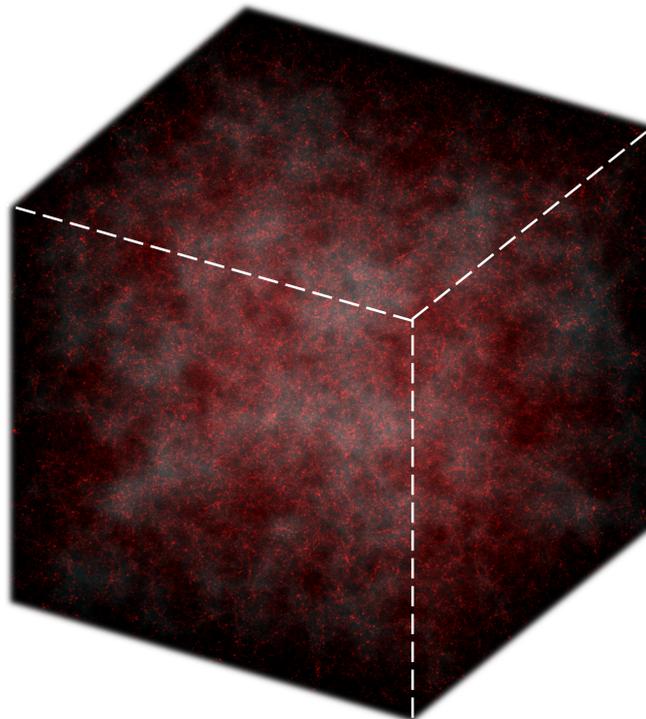
DEPARTMENT OF PHYSICS AND ASTRONOMY  
AARHUS UNIVERSITY

PART A PROGRESS REPORT

# Cosmological Simulations with Neutrinos

*Author:*  
Jeppe Mosgaard Dakin

*Supervisor:*  
Steen Hannestad



June 2017



# Contents

<b>Contents</b>	<b>i</b>
<b>Preface</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Observations . . . . .	1
1.2 Cosmological Simulations . . . . .	2
1.2.1 The CONCEPT Code . . . . .	2
<b>2 Cosmology</b>	<b>5</b>
2.1 The Hubble Expansion . . . . .	5
2.2 Different Species . . . . .	7
2.3 Linear Perturbation Theory . . . . .	9
<b>3 Fluid Dynamics</b>	<b>13</b>
3.1 Fluid Variables . . . . .	13
3.2 The Boltzmann Hierarchy . . . . .	14
3.3 Relativistic Fluid Equations . . . . .	14
3.4 Newtonian Fluid Equations . . . . .	15
3.5 Numerical Implementation . . . . .	16
3.5.1 The MacCormack Method . . . . .	17
3.5.2 Source Terms . . . . .	17
3.5.3 Time Stepping . . . . .	18
3.5.4 Vacuum Corrections . . . . .	18
3.5.5 Gravity . . . . .	20
<b>4 Bridging 3D Real and 1D Fourier Space</b>	<b>23</b>
4.1 Power Spectra . . . . .	23
4.2 Realisations . . . . .	24
4.2.1 Density, Velocity and Shear . . . . .	24
4.2.2 The Zel'dovich Approximation . . . . .	26
4.2.3 Gauge Transformations . . . . .	26
<b>5 Current State and Preliminary Results</b>	<b>27</b>
<b>6 Outlook</b>	<b>31</b>
<b>References</b>	<b>31</b>



# Preface

This report conveys the current state of my PhD research, which deals with the impact on cosmic structure formation from massive neutrinos. To study this, a new numerical method for simulating massive neutrinos has been developed and implemented into the `CONCEPT` code, a cosmological simulation code I have previously developed. In essence, this new method simulates neutrinos as a fluid on a grid, where the fluid consists of an adjustable number of fluid variables, generated by taking momentum moments of the Boltzmann hierarchy.

After a brief introduction to cosmological simulations in chapter 1, the reader will be taken through key aspects of the field of cosmology in chapter 2, outlining the homogeneous universe followed by the basics of perturbation theory, demonstrating the need for cosmological simulations. Chapter 3 is concerned with theoretical fluid dynamics in a cosmological context, as well as its numerical implementation in `CONCEPT` and the challenges I have faced during the development.

Cosmological simulation codes need several inputs from linear theory, in particular for generation of initial conditions. Through this work, `CLASS` [1] has been fully integrated into `CONCEPT` as a library module and is now used for every linear computation, ensuring self-consistency. Chapter 4 describes how this connection between linear and non-linear theory comes together.

In chapter 5, results of neutrino simulations done with `CONCEPT` are shown and compared against the predictions of linear theory and another, numerically very different neutrino implementation. Finally in chapter 6, exciting possibilities for the future are discussed.

Throughout this report we shall work in natural units,  $c = G = \hbar = k_B = 1$ . I like however to be more explicit when it comes to equations directly used as part of some numerical recipe, and so occasionally these factors will be included.



# 1 Introduction

The field of cosmology attempts to make sense of the Universe as a whole. Modern cosmology has its roots in the theory of general relativity, Einstein’s theory of gravity. Importantly, this theory links space and time to the contents within them, and so with general relativity, a definite stage is set for which the cosmic story can unfold. Exactly what story or even what actors — the different species of fundamental particles — remain however largely unconstrained, and so must be determined through observations.

## 1.1 Observations

Through observations of e.g. distant galaxies and the cosmic microwave background, we know that our universe has been expanding since its dawn in the Big Bang 13.8 billion years ago. This expansion is varying with time but pre-determined by the contents of the universe. We have found that presently, “ordinary” (often referred to as *baryonic*) matter comprises just  $\sim 5\%$  of the total energy budget, whereas the rest is composed chiefly of the mysterious (cold) dark matter ( $\sim 25\%$ ) and dark energy ( $\sim 70\%$ ). These dark components dominate our Universe on the largest of scales, yet our understanding of either is next to nothing. Dark matter appears as invisible matter, speeding up the gravitational clockwork throughout the Universe. Dark energy on the other hand has never been observed to cluster. Its effect is seen only on the Universe as a whole, where it causes the universal expansion to accelerate.

Besides matter and dark energy, the Universe also contains less dominant species, such as photons and neutrinos. Through observations of the cosmic microwave background, we understand the photon component intimately. The neutrino component is much less understood, partly because calculations are harder, but to a large extent because of a lack of direct observations. From flavour oscillations, we know that the neutrino\* have a non-zero mass, and so it could contribute somewhat to cosmic clustering, at least at late times. On the other hand, because neutrinos interact so weakly, they free-stream through the Universe, with the effect of decreasing the general amount of structure. How much of a decrease depends dramatically on the mass of the neutrino, with higher mass leading to a larger decrease in structure formation. As such, by far the lowest bound on the neutrino mass,  $\sum m_\nu < 0.23 \text{ eV}$ , comes from cosmology [2].

With the upcoming large scale surveys Euclid and LSST, far more precise maps of the cosmic structure will soon be at our disposal. To understand structure formation at this level, the details of the neutrino component cannot be ignored. In fact, it just might be possible to construct tight constraints on the neutrino mass from these observations, if only we have the theoretical predictions with which to compare. Such predictions are the outcome of large cosmological simulations; the focus point of my PhD work and of this report.

---

\*What is important for cosmology is the net sum of all neutrino masses, and so in this work we shall make little notice of the fact that neutrinos are known to come in three distinct flavours, or correspondingly in three distinct mass eigenstates.

## 1.2 Cosmological Simulations

Whenever referring to ‘cosmological simulations’ in this report, I mean full 3D simulations, evolving the components of the universe through time. Because cold dark matter has zero thermal velocity (hence the ‘cold’), each point in a lump of cold dark matter is described by a single velocity. It is therefore very naturally to simulate cold dark matter as a collection of  $N$  particles, each with a fixed mass, tracing out their individual trajectories. This is what is known as the Lagrangian approach, and a simulation based on this is referred to as an  $N$ -body simulation.

Whenever more information than just the velocity is needed at each point, the Lagrangian approach begins to break down. Thus simulating baryons in this manner is not feasible, though for many purposes (such as ours) the distinction between baryonic and dark matter becomes unimportant at large scales, and so we shall treat baryonic and dark matter on exactly the same footing.

As neutrinos have large thermal velocities, these cannot easily be traced with the Lagrangian approach, and so for these we shall use the Eulerian approach, where the flow between spatially fixed cells is considered. Including the effects of e.g. sound is easy in the Eulerian approach, as each fluid cell simply affects its neighbours depending on the pressure, which is a new variable which must be tracked through time for each cell. We shall refer to the Lagrangian and the Eulerian approach as the particle and fluid approach, respectively.

Many cosmological simulation codes exist, utilizing a variety of clever techniques to be able to simulate a large fraction of the observable Universe to a reasonable degree of detail within a reasonable amount of time. The general set-up of all of them is however the same: The simulated universe consists of a cubic box, within which the particles and fluids live. The boundaries of the box are made periodic in order to simulate an infinite space, making the topology on which the components live a 3-torus. Long-range forces (i.e. gravity) must wind infinitely around this 3-torus. Importantly, the box should be large enough so that effects of periodicity does not show up within the simulated timespan. The box itself is subject to the Hubble expansion, which is usually dealt with by solving the equations of motions in comoving coordinates.

Any given cosmological simulation code have a number of user-defined parameters, with which the physics and cosmology may be specified. Thus the same piece of software may be used to simulate a large variety of different cosmologies, which can then be compared with real-world observations and hence used to find the set of parameters that best describe the real Universe. One might therefore consider these simulation codes as laboratories in which to carry out experiments on universes.

### 1.2.1 The CONCEPT Code

The majority of my PhD work has gone into implementing neutrinos as a fluid into the CONCEPT (COsmological  $N$ -body CodE in PyThon) [3] code, which I originally wrote for my Master’s thesis. Its design was inspired by that of the publicly available GADGET-2 [4] code, and so it was capable of tracking a set of  $N$  particles through time, under their own influence of gravity and in an expanding background. The only physical properties of these particles were their mass, positions and momenta, and so effectively the code could only be used to evolve cold dark matter. The numerical sophistication required for computing the gravitational interaction between a large number of particles efficiently, is however immense. Gravity was (and still is) implemented using three different techniques, known as the PP, PM and P<sup>3</sup>M methods.

Though the part of the code dealing with interactions has been reworked, these remain the only implementations of gravity. Only the PM method has been made compatible with the fluid implementation, and so only this will be described in any detail in this report. See section 3.5.5 for details.

As the goal of my PhD work is to implement neutrinos as a fluid into cosmological simulation codes, using `CONCEPT` — which was an  $N$ -body (particle) code — was not an obvious choice. The `RAMSES` code [5] was initially considered, due to it having both particles and fluids. A very nice feature of `RAMSES` is that the mesh on which a fluid lives is adaptively refined in regions where higher resolutions become necessary. To be able to simulate the largely non-clustering neutrino component, however, this feature was not essential. Instead of having to learn a new code base, we decided that it would be simpler to implement fluid dynamics into `CONCEPT`, making it a hybrid code, though without the added complexity of adaptive mesh refinement.

Though `CONCEPT` is written in Python, the code gets transpiled to C code using Cython together with a lot of tricks of my own, making the runtime speed of the code very close to that of handwritten C code. Today the code consist\* of  $14 \times 10^3$  lines of Python code, implementing the simulation, together with  $5 \times 10^3$  lines of auxiliary code used to compile and run the code in a high-level fashion. This is almost an increase in code size by a factor of 3 compared to the state of the code right after my Master's.

---

\*Not counting comments or blank lines.



## 2 Cosmology

In this chapter we set up the basic equations of a homogeneous, expanding universe. We then perturb this universe using cosmological perturbation theory, often referred to simply as linear theory. We will find that linear theory breaks down for late times, giving rise to the need for non-linear theory and hence cosmological simulations.

### 2.1 The Hubble Expansion

Famously, space is homogeneous and isotropic on large scaled. The same is not however true for time. We are thus interested in the dynamical evolution of e.g. the energy density. As the exact same time evolution takes place throughout the Universe (due to the homogeneity), we can assign the uniform expansion of space itself as the cause of this evolution. We are thus forced to write the squared line element of our spacetime  $ds^2$  as

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

where  $t$  is time,  $d\Sigma^2$  is the squared line element of space and  $a(t)$ , the aptly named scale factor, is some function which scales space as time flow. The homogeneity of space means that  $d\Sigma^2$  can only take on the forms

$$d\Sigma^2 = \frac{d\mathbf{x}^2}{1 - \kappa\mathbf{x}^2} + \mathbf{x}^2 d\Omega^2, \quad (2.2)$$

where  $\mathbf{x}$  is a Cartesian coordinate while  $\Omega$  is a solid angle. The constant  $\kappa$  is the Gaussian curvature of space. With (2.2), (2.1) is known as the Friedmann-Lemaître-Robertson-Walker metric. Observationally [2], our Universe is very close if not perfectly flat,  $\kappa = 0$ , and so our spacetime looks like

$$ds^2 = -dt^2 + a^2(t)(d\mathbf{x}^2 + \mathbf{x}^2 d\Omega^2). \quad (2.3)$$

Our spacetime now have one undefined parameter, the scale factor. Because the Einstein equations couples derivatives of the spacetime metric to the physical stuff within the spacetime,  $a(t)$  must be calculable once we settle on exactly what to put in our Universe. The Einstein equations are

$$R_{\mu\nu} = \frac{1}{2}g_{\mu\nu}R = 8\pi T_{\mu\nu}, \quad (2.4)$$

where the Ricci tensor  $R_{\mu\nu}$  and scalar  $R \equiv R_{\mu}^{\mu}$  is constructed through derivatives of the metric through the Cristoffel symbols,

$$R_{\mu\nu} = \Gamma^{\alpha}_{\mu\nu,\alpha} - \Gamma^{\alpha}_{\mu\alpha,\nu} + \Gamma^{\alpha}_{\mu\nu} \Gamma^{\beta}_{\alpha\beta} - \Gamma^{\alpha}_{\mu\beta} \Gamma^{\beta}_{\nu\alpha},$$

which themselves are given by

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

where the metric tensor  $g_{\mu\nu}$  is read off of the squared line element (2.3) from  $ds^2 \equiv g_{\mu\nu} dx^{\mu} dx^{\nu}$ . The entire left-hand side of the Einstein equations (2.4), dealing only with the curvature of

spacetime, is now specified, with the only unknown being  $a(t)$  hiding inside the metric. The remaining ingredient is  $T_{\mu\nu}$ , the stress-energy tensor, describing the contents of the Universe.

For now, we might model the contents of the Universe as a single, perfect\* fluid, for which the stress-energy tensor is

$$T^{\mu\nu} = (\rho_c + \bar{P})u^\mu u^\nu + \bar{P}g^{\mu\nu}, \quad (2.5)$$

where  $\rho_c$  is the density,  $\bar{P}$  is the pressure and  $u^\mu$  is the 4-velocity of the fluid. The subscript ‘c’ is for ‘critical’, because the homogeneous density of the only fluid in a flat universe must have a very particular value; that which exactly balances the geometry of the Universe between being open and closed, whatever value this may be.

The isotropy of the Universe implies that  $u^\mu = (1, 0, 0, 0)$ , i.e. that the fluid is stationary. This can of course only hold true in one particular frame of reference; the rest frame of the fluid. The notion of being stationary becomes a little trickier when we remember that all of space is allowed to expand, and so the frame of interest must also expand with the universal expansion of space. That is, the fluid is viewed in the  $\mathbf{x}$  coordinate system, which is called the comoving frame. The time parameter  $t$  measured in this frame is called the cosmic time.

With our perfect, comoving fluid, the Einstein equations are ready to be solved. With  $u^\mu = (1, 0, 0, 0)$  and remembering that  $g_{\mu\nu}$  and thus its inverse  $g^{\mu\nu}$  is diagonal, it is clear from (2.5) that all off-diagonal terms in  $T^{\mu\nu}$  vanish, and that all three space-space elements are equal. There are thus two non-vanishing Einstein equations for the homogeneous Universe. These are solved in e.g. [6], with the results

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

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

which are, respectively, the Friedmann equation (resulting from the time-time equation) and the acceleration equation (resulting from the space-space equation). In these equations and the rest of this work, a dot designates differentiation with respect to cosmic time,  $\dot{\phantom{x}} \equiv \partial/\partial t$ .

If we differentiate the Friedmann equation (2.6) and insert the acceleration equation (2.7), the homogeneous continuity equation is obtained:

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

Now we separate the homogeneous fluid into multiple species, each with their own density  $\rho_\alpha$  and pressure  $P_\alpha$ . We note that  $\rho_\alpha$  and  $P_\alpha$  are fields,  $\rho_\alpha = \rho_\alpha(t, \mathbf{x})$ ,  $P_\alpha = P_\alpha(t, \mathbf{x})$ , with spatial means of  $\bar{\rho}_\alpha(t)$  and  $\bar{P}_\alpha(t)$ . Since density and pressure are additive quantities, we simply have  $\rho_c = \sum_\alpha \bar{\rho}_\alpha$  and  $\bar{P} = \sum_\alpha \bar{P}_\alpha$ . Additionally, the homogeneous continuity equation (2.8) decomposes into a set of uncoupled equations

$$\dot{\rho}_\alpha = -3\frac{\dot{a}}{a}(\bar{\rho}_\alpha + \bar{P}_\alpha). \quad (2.9)$$

To algebraically close the Friedmann and homogeneous continuity equation(s), we need a relation between density and pressure, known as an equation of state. To describe all species on large scales, a simple linear<sup>†</sup> equation of state

$$P_\alpha = w_\alpha \rho_\alpha, \quad (2.10)$$

\*A perfect fluid is one which in its rest frame is completely characterized by its density and pressure.

<sup>†</sup>The equation of state is more complicated for e.g. particles with internal degrees of freedom.

will suffice. Here, the equation of state parameter  $w_\alpha$  is some number which may or may not depend on time. Using the linear equation of state (2.10), which of course also holds for the mean values  $\bar{P}_\alpha$  and  $\bar{\rho}_\alpha$ , the homogeneous continuity equation (2.9) can be integrated to yield

$$\bar{\rho}_\alpha = a^{-3(1+w_\alpha)} \bar{\rho}_{\alpha,0}, \quad (2.11)$$

where the subscript ‘0’ indicates the present time;  $\bar{\rho}_{\alpha,0} \equiv \bar{\rho}_\alpha(t = t_0)$ , where  $t_0$  is the cosmic time at the present, i.e. the age of the universe. In writing (2.11), the scale factor has been normalized so that  $a_0 = a(t = t_0) = 1$ . We have been able to postpone this choice of normalization until now because (2.11) is the first absolute relation of a dynamic variable. With this normalization, comoving  $\mathbf{x}$  and physical coordinates  $\mathbf{r} = a\mathbf{x}$  coincide at the present. For a given set of species with measured mean densities at the present, the Friedmann equation (2.6) may now be solved for the evolution of  $a(t)$ . As is customary, we may write the Friedmann equation in terms of the Hubble parameter  $H \equiv \dot{a}/a$  and density parameters  $\Omega_\alpha \equiv \bar{\rho}_\alpha/\rho_c$ ,

$$\frac{H^2}{H_0^2} = \sum_{\alpha} a^{-3(1+w_\alpha)} \Omega_{\alpha,0}, \quad (2.12)$$

where the trend of using subscript ‘0’ for present values is continued and the sum is over all species. If we know what species the Universe contains ( $\{w_\alpha\}$ ), the observational input to the Friedmann equation (2.12) is the present mean densities and the present Hubble parameter  $H_0$  (called the Hubble “constant”). Note that once we have  $H_0$ , the present critical density is known from (2.6).

## 2.2 Different Species

We now introduce the basic species of our universe. The term “matter” is used to refer to all pressureless fluids, which then encompasses both baryonic and dark matter. From (2.10) this means that matter is defined by the equation of state  $w_m = 0$ , which through (2.11) leads to the scaling behaviour  $\rho_m \propto a^{-3}$ , which is the well-known fact that the density of everyday stuff is inversely proportional to the volume.

In the other end of the spectrum we have ultra-relativistic species, the densities of which are dominated by their momentum. Because momenta experience redshifting when space is expanded, an additional factor  $a^{-1}$  must be multiplied on the volume factor  $a^{-3}$  when writing down the density scaling relation for radiation;  $\rho_r \propto a^{-4}$ . Equation (2.11) then gives  $w_r = 1/3$ . The prime examples of radiation species are photons and massless neutrinos.

The defining property of the dark energy component of the Universe is its constantness, and so  $\rho_\Lambda \propto 1 \Rightarrow w_\Lambda = -1$ . Plugging into the acceleration equation (2.7), we see that dark energy is the only species which contribute positively to the acceleration of the universal expansion.

The different scaling behaviours of radiation, matter and dark energy implies that the dominant species (the one with the largest density) is not the same throughout time. The Universe then passes through different epochs with different expansion speeds. Since the density of radiation falls off with  $a$  the fastest, radiation dominates at early times. Since the density of dark energy does not fall off with  $a$  at all, dark energy dominate at late times. Solving the single-component Friedmann equation (2.6) for the different species, one find

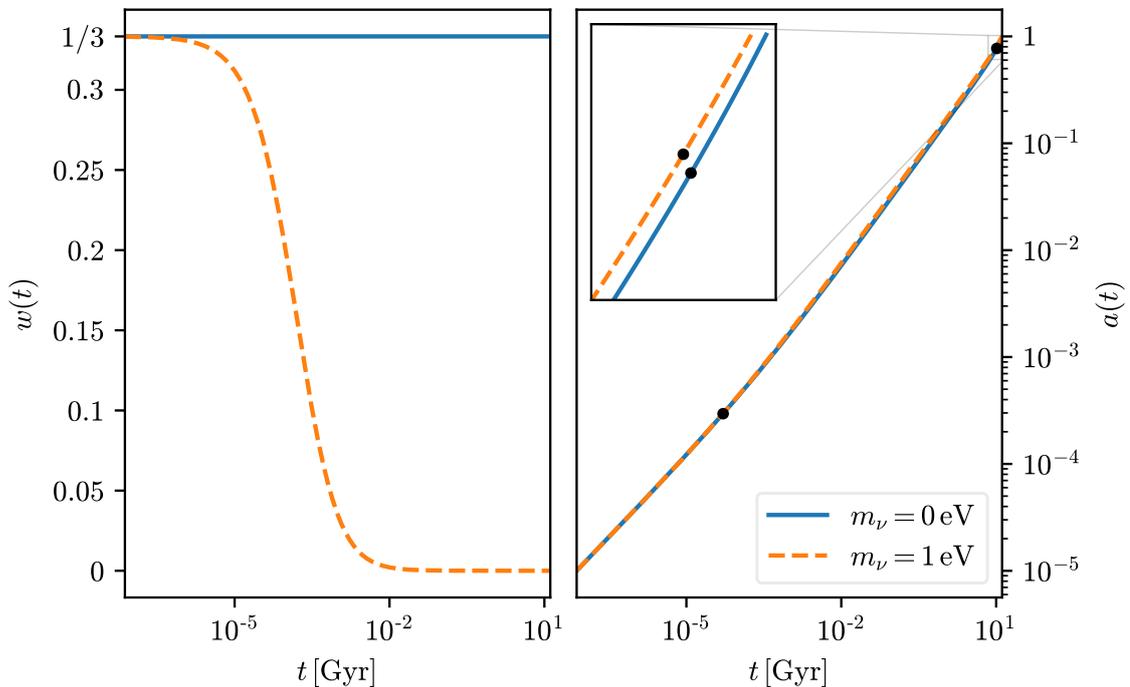
$$a \propto \begin{cases} t^{1/2} & \text{(radiation domination),} \\ t^{2/3} & \text{(matter domination),} \\ e^{H_0 t} & \text{(dark energy domination).} \end{cases} \quad (2.13)$$

The value of the  $t$  and  $a$  right between two epochs, at radiation/matter equivalence or matter/dark energy equivalence, depend on absolute values of the present densities. A plot of  $a(t)$  for the  $\Lambda$ CDM cosmology is shown to the right on figure 2.1.

Now let us turn to the massive neutrinos, our main species of interest. That neutrinos have a non-zero mass is known through the experimental existence of neutrino oscillations, where a neutrino in a particular weak flavor state “oscillate” into another. This process is only possible if the mass eigenstates are non-degenerate, so that a difference in mass exists. It is however possible that the lowest mass eigenstate is completely massless. For the sake of argument we may imagine that only a single neutrino exists and that it has a mass of  $\sim 1$  eV. This mass is large enough that the neutrino will be unrelativistic at late times. We thus have a species that starts out as radiation,  $w = 1/3$ , but as time goes will turn into matter,  $w = 0$ . For neutrinos then, the equation of state is time dependent,  $w_\nu = w_\nu(t)$ . To compute  $w_\nu(t)$  we need  $\bar{\rho}_\nu(t)$  and  $\bar{P}_\nu(t)$ . Both of these can be obtained from the underlying phase space distribution  $f_\nu(t, \mathbf{r}, \mathbf{p})$ , where  $\mathbf{p}$  is momentum. At early times when the neutrinos were in equilibrium, they were distributed according to a Fermi-Dirac distribution, just as all other fermions. Since neutrinos interact only very weakly, a good approximation is that their distribution function keeps a Fermi-Dirac shape, with a diminishing temperature, all the way to the present day. The neutrinos are then distributed as

$$f_\nu(t, \mathbf{p}) = \frac{1}{e^{|\mathbf{p}|/T_\nu(t)} + 1}, \quad (2.14)$$

where  $T_\nu$  is the temperature of the neutrino fluid. In (2.14),  $|\mathbf{p}|$  is used as a (very) good approximation for the energy  $\sqrt{m_\nu^2 + \mathbf{p}^2}$  at the time of decoupling. The mean density and



**Figure 2.1** – Left: Evolution of the equation of state  $w(t)$  for neutrino species with  $m_\nu = 0$  and  $m_\nu = 1$  eV. Right: Evolution of the scale factor  $a(t)$  in  $\Lambda$ CDM cosmologies with three neutrino species, each with mass  $m_\nu$ . The black dots mark the time of radiation/matter and matter/dark energy equivalence.

pressure can be computed by integrating over the distribution, weighted by energy and momentum as follows:

$$\bar{\rho}_\nu = 2 \int \frac{d^3\mathbf{p}}{(2\pi)^3} f_\nu(\mathbf{p}) E(\mathbf{p}), \quad (2.15)$$

$$\bar{P}_\nu = 2 \int \frac{d^3\mathbf{p}}{(2\pi)^3} f_\nu(\mathbf{p}) \frac{\mathbf{p}^2}{3E(\mathbf{p})}, \quad (2.16)$$

where the factor of two is due to the spin degree of freedom of the neutrino. To compute  $w_\nu = \bar{P}_\nu/\bar{\rho}_\nu$  we now just need to know  $T_\nu(t)$ . At early times when the neutrinos were in equilibrium with the other fermions and photons, all of these had to have the same temperature. Once a given species has decoupled, its temperature scale as  $T \propto a^{-1}$ . We thus expect the neutrino temperature  $T_\nu$  to follow the photon temperature  $T_\gamma$  closely. At some point after the neutrino decouples,  $e^+e^-$  pair production stops, dumping energy from the remaining annihilations into the photon component, heating it up. Thus the neutrinos are somewhat cooler than the photons. The exact relation [7] is

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

With the observational result  $T_{\gamma,0} = 2.725$  K and  $T_\nu \propto a^{-1}$ , we now have all we need to compute  $w_\nu(t)$  for a given value of the neutrino mass  $m_\nu$ . The resultant  $w(t)$  and  $a(t)$  for the  $\Lambda$ CDM cosmology with three massless neutrinos, as well as with three massive neutrinos, is shown in figure 2.1. We see that even in the case of three neutrinos with the large mass of 1 eV, their effect on the expansion is minuscule. The true importance of massive neutrinos lies in their effect on structure formation, as we shall see.

## 2.3 Linear Perturbation Theory

In this section we take the completely homogeneous Universe and perturb it slightly. We do this in real space and comoving coordinates. For brevity, we consider only the dark matter component, but leave out the ‘m’ subscript. As dark matter has neither pressure nor shear, this simplifies the calculation immensely. In chapter 3 a non-linear treatment of a general species will be performed.

With the physical velocity at a point  $\mathbf{r}$  given by  $\dot{\mathbf{r}}$ , the equations of motion for the matter fluid are the continuity and Euler equation, describing conservation of mass and momentum, respectively:

$$\begin{cases} \left. \frac{\partial \rho}{\partial t} \right|_{\mathbf{r}} = -\nabla_{\mathbf{r}} \cdot (\rho \dot{\mathbf{r}}), \\ \left. \frac{\partial \dot{\mathbf{r}}}{\partial t} \right|_{\mathbf{r}} = -\dot{\mathbf{r}} \cdot \nabla_{\mathbf{r}} \dot{\mathbf{r}} - \nabla_{\mathbf{r}} \phi, \end{cases} \quad (2.17)$$

where all derivatives are given a subscript ‘ $\mathbf{r}$ ’, indicating that they are to be taken with respect to the physical coordinate system. The reason that the temporal derivatives need this coordinate specification is because the coordinate transformation is time-dependent through the scale factor,  $\mathbf{r} = a(t)\mathbf{x}$ . The right-hand-side of the continuity equation represent transport of energy, whereas the first term in the right-hand-side of the Euler equation represent transport of momentum. The last term represent the influence of gravity with the potential  $\phi$ .

Our first goal is to transform the above equations of motion to the comoving coordinate system. From the definition of comoving coordinates,  $\mathbf{r} = a\mathbf{x}$ , it is clear that

$$\nabla_{\mathbf{r}} = a^{-1}\nabla_{\mathbf{x}}, \quad (2.18)$$

but how  $\partial/\partial t|_{\mathbf{r}}$  transforms is less clear. To find out, consider a function  $f(t, \mathbf{r})$  with differential

$$\begin{aligned} df &= \nabla_{\mathbf{r}}f(t, \mathbf{r}) \cdot d\mathbf{r} + \left. \frac{\partial f(t, \mathbf{r})}{\partial t} \right|_{\mathbf{r}} dt \\ &= \nabla_{\mathbf{x}}f(t, \mathbf{r}) \cdot d\mathbf{x} + \left[ \frac{\dot{a}}{a}\mathbf{x} \cdot \nabla_{\mathbf{x}}f(t, \mathbf{r}) + \left. \frac{\partial f(t, \mathbf{r})}{\partial t} \right|_{\mathbf{r}} \right] dt, \end{aligned} \quad (2.19)$$

where  $d\mathbf{r} = \mathbf{x} da + a d\mathbf{x} = \dot{a}\mathbf{x} dt + a d\mathbf{x}$  has been used. By allowing  $f$  to take in comoving arguments,  $f(t, \mathbf{r}) = f(t, a\mathbf{x}) \equiv f(t, \mathbf{x})$ , its differential in purely comoving coordinates is

$$df = \nabla_{\mathbf{x}}f(t, \mathbf{x}) \cdot d\mathbf{x} + \left. \frac{\partial f(t, \mathbf{x})}{\partial t} \right|_{\mathbf{x}} dt. \quad (2.20)$$

Comparing (2.19) and (2.20), we can read off the relation

$$\left. \frac{\partial}{\partial t} \right|_{\mathbf{r}} = \left. \frac{\partial}{\partial t} \right|_{\mathbf{x}} - \frac{\dot{a}}{a}\mathbf{x}\nabla_{\mathbf{x}}. \quad (2.21)$$

The defining feature of  $\phi$  is that it satisfies the Poisson equation

$$\nabla_{\mathbf{r}}^2\phi(\mathbf{r}) = 4\pi\rho(\mathbf{r}).$$

In comoving coordinates, the potential  $\phi(t, \mathbf{r})$  has to be replaced not just by  $\phi(t, a\mathbf{x})$ , but by the so-called peculiar potential, satisfying a Poisson equation for the density *fluctuations*,  $\nabla_{\mathbf{x}}^2\varphi(\mathbf{x}) \propto \delta(\mathbf{x})$ . Whether using density or density fluctuations ought not change the behaviour of gravity, and so this change is allowed. It is required because a solution to the Poisson equation only exists\* for a field with mean value 0. The relation between  $\phi$  and  $\varphi$  takes the form  $\varphi(\mathbf{x}) \propto \phi(a\mathbf{x}) + a\ddot{a}\mathbf{x}^2/2$ . In the literature, the proportionality factor is usually chosen as just 1, leading to

$$\begin{aligned} \varphi(\mathbf{x}) &\equiv \phi(\mathbf{r}) + \frac{1}{2}a\ddot{a}\mathbf{x}^2 \\ \Rightarrow \nabla_{\mathbf{x}}^2\varphi(\mathbf{x}) &= 4\pi a^2(\rho(\mathbf{x}) - \bar{\rho}), \end{aligned} \quad (2.22)$$

where the acceleration equation (2.7) has been used to replace  $\ddot{a}$ . In a universe with multiple inhomogeneous components,  $\rho(\mathbf{x})$  and  $\bar{\rho}$  in (2.22) must be the total mean density and density contrast.

Transforming the continuity and Euler equation (2.17) via (2.25), (2.18), (2.21) and (2.22), one obtain

$$\begin{cases} \left. \frac{\partial \rho}{\partial t} \right|_{\mathbf{x}} = -a^{-1}\nabla_{\mathbf{x}} \cdot (\rho\mathbf{u}) - 3\frac{\dot{a}}{a}\rho, \\ \left. \frac{\partial \mathbf{u}}{\partial t} \right|_{\mathbf{x}} = -a^{-1}\mathbf{u} \cdot \nabla_{\mathbf{x}}\mathbf{u} - \frac{\dot{a}}{a}\mathbf{u} - a^{-1}\nabla_{\mathbf{x}}\varphi, \end{cases} \quad (2.23)$$

where  $\mathbf{u}$  is the peculiar velocity  $\mathbf{u} \equiv a\dot{\mathbf{x}}$ .

---

\*Note that this means that this complication only arise for infinite systems.

We write the perturbations in the fluid density as

$$\rho(t, \mathbf{r}) = \bar{\rho}(t)[1 + \delta(t, \mathbf{r})], \quad (2.24)$$

where  $\delta$  is called the density contrast.

We now write the perturbations in the fluid density as

$$\rho(t, \mathbf{r}) = \bar{\rho}(t)[1 + \delta(t, \mathbf{r})], \quad (2.25)$$

where  $\delta$  is called the density contrast. So far, no approximations about the fluctuations being small have been used. If we however consider only small perturbations,  $\delta \ll 1 \Rightarrow \mathbf{u} \cdot \nabla_{\mathbf{x}} \mathbf{u} \approx \mathbf{0}$ , the non-linear continuity and Euler equation in comoving coordinates (2.23) can be combined into a single second-order differential equation for the density contrast,

$$\ddot{\delta} + 2\frac{\dot{a}}{a}\dot{\delta} - 4\pi\delta\bar{\rho} = 0. \quad (2.26)$$

With the growth of the scale factor in each domination epoch given in (2.13), we can solve (2.26) for each epoch separately:

$$\delta \propto \begin{cases} \log a & \text{(radiation domination),} \\ a & \text{(matter domination),} \\ 1 & \text{(dark energy domination).} \end{cases} \quad (2.27)$$

It is clear from (2.27) that significant matter structure may only form in the matter domination epoch. As seen from 2.1, the Universe has been matter dominated for the majority of its life, and so  $a$  grows dramatically throughout this epoch. The  $\delta \propto a$  behaviour then tells us that the density cannot be taken to be perturbative through matter domination. Perturbation theory may then only be used to evolve the Universe through the radiation epoch and some way into matter domination. From here, non-linear theory must take over, implying solving the non-linear fluid equations (2.23) (or their equivalent) using numerical codes.



## 3 Fluid Dynamics

In this chapter we will find the non-linear equations governing a general fluid with equation of state  $w(t)$ . As far as possible we will express these equations in so-called conservation form, important for numerical stability.

### 3.1 Fluid Variables

The full non-linear stress-energy tensor can be written as [8]

$$\begin{cases} T_0^0 = -\rho, \\ T_0^i = -\bar{\rho} \left( 1 + \delta + w + \frac{\delta P}{\delta \rho} \delta \right) u^i, \\ T_j^i = \bar{\rho} \left( w + \frac{\delta P}{\delta \rho} \delta \right) \delta_j^i + \Sigma_j^i, \end{cases} \quad (3.1)$$

where two new fluid variables  $\delta P/\delta \rho$  and  $\Sigma_j^i$  appear. The first is the square of the sound speed, effectively decoupling density and pressure into two separate variables. Since we approximate the pressure of any fluid to be linearly related to its density (2.10) at any point,  $P(t, \mathbf{x}) = w(t)\rho(t, \mathbf{x})$ , with  $w(t) = \bar{P}(t)/\bar{\rho}(t)$  being spatially constant, the variable  $\delta P/\delta \rho$  reduces to just  $w$ . As the equation for  $T_j^i$  in (3.1) defines the stress tensor  $\Sigma_j^i$ , we are free to modify it, at the cost of our new stress tensor being different from the one in (3.3). The equation for  $T_j^i$ , as it stands in (3.1), guarantees a traceless stress tensor. As what we have been working with so far is a perfect (shear stress-less) fluid, we are interested in a stress tensor,  $\sigma_j^i$  that vanishes for perfect fluids. If we lower an index on (2.5), we see that for a perfect fluid,

$$T_j^i = w\rho\delta_j^i + (1+w)\rho u^i u_j \quad (\text{perfect fluid}),$$

where the first term  $w\rho\delta_j^i$  is really identical to the first term on the right-hand-side of the  $T_j^i$  equation in (3.1), once we set  $\delta P/\delta \rho = w$ . Our new stress tensor, containing all deviations from a perfect fluid, can then be defined by

$$\Sigma_j^i = (1+w)\rho(u^i u_j + \sigma_j^i).$$

Our stress-energy tensor then takes the form

$$\begin{cases} T_0^0 = -\rho, \\ T_0^i = -(1+w)\rho u^i, \\ T_j^i = w\rho\delta_j^i + (1+w)\rho(u^i u_j + \sigma_j^i). \end{cases} \quad (3.2)$$

### 3.2 The Boltzmann Hierarchy

We saw in section 2.1 how the homogeneous continuity equation could be derived from the Einstein equations. We could also have found this by requiring that the covariant derivative of the stress-energy tensor vanishes:

$$0 = T^{\alpha}_{\mu;\alpha} \equiv \partial_{\alpha} T^{\alpha}_{\mu} + \Gamma^{\beta}_{\beta\alpha} T^{\alpha}_{\mu} - \Gamma^{\beta}_{\alpha\mu} T^{\alpha}_{\beta}. \quad (3.3)$$

Similarly, using the non-linear stress-energy tensor (3.2), the non-linear continuity and Euler equations can be derived. The equation of evolution for  $\sigma^i_j$  cannot be found in this way, however, as (3.3) only has a single free index and so cannot possibly contain a term like  $\dot{\sigma}^i_j$ . Instead we can use the same approach as in (2.15) and (2.16), taking moments of the distribution function. The general relativistic expression for the stress-energy tensor and higher-order variables, written as momentum moments of the distribution function  $f$ , is [9, 10]

$$\left\{ \begin{array}{l} T^0_0(t, \mathbf{x}) = \sqrt{-g} \int dP_1 dP_2 dP_3 f(t, \mathbf{x}, \mathbf{p}) P_0, \\ T_i^0(t, \mathbf{x}) = \sqrt{-g} \int dP_1 dP_2 dP_3 f(t, \mathbf{x}, \mathbf{p}) P_i, \\ T_i^j(t, \mathbf{x}) = \sqrt{-g} \int dP_1 dP_2 dP_3 f(t, \mathbf{x}, \mathbf{p}) \frac{P_i P^j}{P^0}, \\ \Pi_i^{jk}(t, \mathbf{x}) = \sqrt{-g} \int dP_1 dP_2 dP_3 f(t, \mathbf{x}, \mathbf{p}) \frac{P_i P^j P^k}{P^0 P^0}, \\ \vdots \end{array} \right. \quad (3.4)$$

where  $P_{\mu}$  is conjugate momentum to comoving  $x^{\mu}$  and  $g$  is the determinant of the (now perturbed) metric. As demonstrated by the inclusion of the rank-3 tensor variable  $\Pi_i^{jk}$  in (3.4), we can extend this hierarchy indefinitely by including another  $P^{\ell}/P^0$  at each step, regardless of the fact that the more well-known stress energy tensor is only of rank 2. We can then generate an infinite hierarchy of fluid variables, each a symmetric tensor carrying one additional index than the previous, all of which are needed in order to cover all of phase space and describe the fluid exactly. With increasing mass,  $P^{\ell}/P^0$  drops, making higher order variables less important. A valid approximation to the infinite hierarchy (3.4) is then a truncated version including only the first  $n$  variables, where  $n$  is allowed to be chosen lower the higher the mass. Alternatively, instead of simply truncating the hierarchy, one might wish for some algebraic approximation to the  $n$ 'th fluid variable, closing the hierarchy at this level. Lastly, one could close the hierarchy by computing the  $n$ 'th fluid variable using linear theory, which can be done essentially exactly. As the fluid variables become less non-linear the higher the moment, this ought to be a very precise and self-consistent way to close the hierarchy. This linear-theory closure is what has been implemented in CONCEPT.

### 3.3 Relativistic Fluid Equations

To derive the evolution of the fluid variables defined by the Boltzmann hierarchy (3.4), we are in need of the evolution of the underlying distribution function  $f(t, \mathbf{x}, \mathbf{p})$ . The equation governing the evolution of  $f$  is the Boltzmann equation, describing conservation of phase space volume:

$$0 = \frac{df(t, \mathbf{x}, \mathbf{p})}{dt} = \left[ \frac{\partial}{\partial t} + \frac{d\mathbf{x}}{dt} \frac{\partial}{\partial \mathbf{x}} + \frac{d\mathbf{p}}{dt} \frac{\partial}{\partial \mathbf{p}} \right] f(t, \mathbf{x}, \mathbf{p}), \quad (3.5)$$

where no collision term is present for the weakly interacting neutrinos. We can now create equations of motion for any fluid variable by multiplying (3.5) by the appropriate factors of  $P^\ell$  and  $P^0$  and integrating over  $\mathbf{p}$ . The fully relativistic continuity and Euler equation resulting from this procedure, only subject to the approximations of the metric perturbations being small, can be found in [8]:

$$\left\{ \begin{array}{l} \delta' = -(1+w)(\theta - 3\phi') - 3\frac{a'}{a}\left(\frac{\delta P}{\delta\rho} - w\right)\delta - \theta\delta - u^i\partial_i\delta \\ \quad + 3\left(1 + \frac{\delta P}{\delta\rho}\right)\phi'\delta - \frac{\delta P}{\delta\rho}\theta\delta - u^i\partial_i\left(\frac{\delta P}{\delta\rho}\delta\right) \\ \quad - (\partial_i\psi - 3\partial_i\phi)\left(1 + \delta + w + \frac{\delta P}{\delta\rho}\delta\right)u^i, \\ u^i{}' = -\left[\frac{a'}{a}(1-3w) - \psi' - 5\phi'\right]u^i - \frac{\left[\delta' + w' + \partial_\tau\left(\frac{\delta P}{\delta\rho}\delta\right)\right]u^i + \delta^{ij}(1+\delta)\partial_j\psi}{1 + \delta + w + \frac{\delta P}{\delta\rho}\delta} \\ \quad - \frac{1}{\bar{\rho}\left(1 + \delta + w + \frac{\delta P}{\delta\rho}\delta\right)}[\delta^{ik}(\partial_j + \partial_j\psi - 3\partial_j\phi) + \delta^k{}_j\delta^{il}\partial_l\phi]T^j{}_k, \end{array} \right. \quad (3.6)$$

where a prime denotes differentiation with respect to conformal time  $\tau$ ,  $' = \partial_\tau = a\partial_t$ , and all spatial derivatives are with respect to comoving coordinates;  $\partial_i = \partial/(\partial x^i)|_{\mathbf{x}}$ . We shall continue using this notation from here on out. An equally monstrous formula for the time evolution of  $\sigma_i{}^j$  is known, involving fluxes of  $\Pi_i{}^{jk}$ , but as I have not yet implemented this into my work, I will not discuss it.

### 3.4 Newtonian Fluid Equations

Solving systems of partial differential equations numerically is no easy task, as the problem at hand may suffer from different kinds of instabilities. Generally we can improve the stability by expressing the partial differential equations in *conservation form* (if possible). In this way, whatever numerical errors may occur, the physically conserved quantities are guaranteed to stay conserved numerically, effectively constraining the numerical solution in our favour without adding to the numerical complexity.

As an example, consider the continuity equation for matter (2.23),

$$\left.\frac{\partial\rho_{\mathbf{m}}}{\partial t}\right|_{\mathbf{x}} = -a^{-1}\nabla_{\mathbf{x}}\cdot(\rho_{\mathbf{m}}\mathbf{u}_{\mathbf{m}}) - 3\frac{\dot{a}}{a}\rho_{\mathbf{m}}. \quad (3.7)$$

Here the change in density is caused by a mass flux  $\nabla_{\mathbf{x}}\cdot(\rho_{\mathbf{m}}\mathbf{u}_{\mathbf{m}})$ , meaning that mass is flowing around, but conserved. The last term  $-3(\dot{a}/a)\rho_{\mathbf{m}}$  acts as a source (sink) term, lowering the density throughout space. This is of course not because we are actually losing mass, but rather because we are gaining space. Remembering the scaling  $\bar{\rho}_{\mathbf{m}} \propto a^{-3}$ , we expect the quantity  $a^3\rho_{\mathbf{m}}$  to fulfil a continuity equation without this Hubble term. Indeed, from (3.7), we can immediately obtain

$$\left.\frac{\partial a^3\rho_{\mathbf{m}}}{\partial t}\right|_{\mathbf{x}} = -a^2\nabla_{\mathbf{x}}\cdot(\rho_{\mathbf{m}}\mathbf{u}_{\mathbf{m}}).$$

Were we to just implement a matter fluid numerically,  $a^3\rho$  would then be an appropriate variable to track. Similarly, for a radiation fluid, the variable which continuity equation can be stated in conservation form is  $a^4\rho$ . With these two limits in place, we define the density variable

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

which should lead to a continuity equation without source terms, for any fixed  $w$ . Now looking at the Euler equation at (2.23), we see a similar Hubble term  $-(\dot{a}/a)\mathbf{u}$ . Thus for matter,  $a\mathbf{u}$  is a better variable to track numerically. In conservation form, fluid equations involving momentum density  $\rho\mathbf{u}$  rather than the bare velocity  $\mathbf{u}$  tends to involve fewer and simpler terms. Now since  $\rho \propto a^{-3}$  for matter, we thus choose our second fluid variable as  $(a^3\rho)(a\mathbf{u})$ . For radiation,  $\rho \propto a^{-4}$  but peculiar velocities  $\mathbf{u}$  have no explicit time dependence, again suggesting that the second fluid variable should be chosen as

$$\mathbf{J} \equiv a^4\rho\mathbf{u},$$

which is what we do. A similar analysis for  $\sigma^i_j$  may have to be done once it is fully incorporated as a dynamic fluid variable. For now,  $\varrho$ ,  $\mathbf{J}$  and  $\sigma^i_j$  are the variables implemented in CONCEPT.

We wish to reduce the relativistic fluid equations (3.6) to their Newtonian form. In the Newtonian limit, the metric perturbations both reduces to the peculiar potential  $\psi, \phi \rightarrow \varphi$ , which have no explicit time dependence,  $\psi', \phi' \rightarrow 0$ . Furthermore, since  $u^i, \psi, \phi$  and  $\sigma^i_j$  are small quantities, any term with two or more of these quantities can be neglected. Then we have the approximation of a spatially constant sound speed, meaning  $\delta P/\delta\rho \rightarrow w$ . As CONCEPT uses cosmic time, all conformal time differentials must also be converted accordingly. Last but not least,  $\delta$  and  $u^i$  should be converted into  $\varrho$  and  $\mathbf{J}$ , and all terms should be written as flux terms as far as possible. It took me quite some time to do these calculations, as I tried with different combinations of fluid variables (including different stress tensors) only to find spurious source terms popping up. The final result, including factors of  $c$ , is

$$\begin{cases} \partial_t \varrho = -a^{3w-2}(1+w)\partial_i \mathbf{J}^i + 3\dot{w} \ln a \varrho, \\ \partial_i \mathbf{J}^i = -c^2 a^{-3w} \frac{w}{1+w} \partial_i \varrho - a^{3w-2} \partial_j \left( \frac{\mathbf{J}^j \mathbf{J}^i}{\varrho} \right) - a^{-3w} \partial^j (\varrho \sigma^i_j) \\ \quad - \frac{\dot{w}}{1+w} \mathbf{J}^i - a^{-3w} \varrho \partial^i \varphi, \end{cases} \quad (3.8)$$

where no significance should be given to the placement of indices on the partial derivatives. Interpreting the new terms, we see that both the continuity and Euler equation has picked up a source term due to  $\dot{w}$ . Also, for  $w \neq 0$ , a pressure term  $\propto \partial_i \varrho$  appears in the Euler equation. The last equation we need is the Poisson equation written in terms of  $\varrho$ . This is straight forwardly obtained from (2.22);

$$\partial_i \partial^i \varphi = a^{-3w-1} 4\pi G \varrho. \quad (3.9)$$

### 3.5 Numerical Implementation

In CONCEPT, the inhomogeneous components of a universe can be simulated as either particles or fluids. Particle components consist of  $N$  particle positions and momenta. For fluid components these are replaced with the list of fluid variables, which can have any\* length. Each fluid

---

\*Though presently, no dynamic evolution is implemented for  $\sigma^i_j$  or higher order.

variable is thought of as a symmetric array of scalar fields. Each such scalar field consists at the moment of three equally sized cubic grids, spanning the simulation box. The fluid scalar lives on the primary grid, whereas the other two are used when solving the fluid equations (3.8).

### 3.5.1 The MacCormack Method

The numerical method adopted is the MacCormack method [11], a finite difference method, second-order in both time and space. This method is computationally cheap, generalizes easily to three dimensions and parallelizes straight forwardly. The method solves the fluid equations (3.8) in two steps, called the predictor and corrector step, here illustrated for the continuity equation:

$$\left\{ \begin{array}{l} \varrho^*(\mathbf{x}) = \varrho(\mathbf{x}) - \sum_{i=1}^3 \frac{1}{|\Delta \mathbf{x}_i|} [\mathbf{J}^i(\mathbf{x} + \Delta \mathbf{x}_i) - \mathbf{J}^i(\mathbf{x})] \int_t^{t+\Delta t} a^{3w-2}(1+w) dt, \\ \varrho(\mathbf{x}) \rightarrow \frac{1}{2} [\varrho(\mathbf{x}) + \varrho^*(\mathbf{x})] \\ - \frac{1}{2} \sum_{i=1}^3 \frac{1}{|\Delta \mathbf{x}_i|} [\mathbf{J}^{i*}(\mathbf{x}) - \mathbf{J}^{i*}(\mathbf{x} - \Delta \mathbf{x}_i)] \int_t^{t+\Delta t} a^{3w-2}(1+w) dt. \end{array} \right. \quad (3.10)$$

After the above computation has been performed,  $\varrho(\mathbf{x})$  will have been evolved by an amount  $\Delta t$  due to its flux term. The algorithm is simple to understand: In the predictor step, the slopes in  $\partial_i \mathbf{J}^i(\mathbf{x})$  are approximated by the finite difference steps\*  $\Delta \mathbf{x}_i$  to the right, for each dimension  $i$ . The results  $\varrho^*(\mathbf{x})$  are stored on a separate grid. Before the corrector step, the corresponding  $\mathbf{J}^*$  is computed from  $\varrho$  and  $\sigma_j^i$ , and so on. In the corrector step, we compute the slope to the left, and update  $\varrho(\mathbf{x})$  using the mean of the two slopes. The integral is needed because  $a$  and  $w$  depend on time, and so simply multiplying by  $a^{3w-2}(1+w)\Delta t$  would produce a small error. Of course  $\mathbf{J}$  also depends on time, but we have to take it outside the integral to produce a finite time *step*. One might argue that a more self-consistent treatment was to also take  $a$  and  $w$  outside of the integral, evaluating them only at the time step instances. In the GADGET-2 [4] code, a similar choice (though only including  $a$ ) of keeping the integral is made.

### 3.5.2 Source Terms

No source terms are implemented by the MacCormack method in CONCEPT, though it can be done straight forwardly [12]. As source terms perturbs the conservations, we wish to apply them as rare as possible. After all flux terms have been applied to all fluid variables through (3.10), all source terms are applied, including gravity. As gravity is inherently expensive to compute due to it being a long-range force, we do not want to compute this more than once per time step, which is another reason not to apply source terms intermangled with the MacCormack steps. As source terms like<sup>†</sup> the one in the continuity equation (3.10) does not involve fluid variable derivatives, they can be added on effortlessly. For the computation of gravity, see subsection 3.5.5.

\*As all fluid grids in CONCEPT are regular and cubic,  $|\Delta \mathbf{x}_i|$  is equal for all dimensions  $i$ .

<sup>†</sup>In CONCEPT, these are referred to as “internal” source terms, in contrast to e.g. gravity which is really an external force.

### 3.5.3 Time Stepping

In CONCEPT the time step size  $\Delta t$  is spatially global and determined dynamically throughout the simulation to be as high as possible without corrupting the physics. Global tracers such as the dynamical time scale and the Hubble time are used, but also more detailed information about the actual components are used. Since the MacCormack method (3.10) computes derivatives as the difference between neighbouring\* grid points only, this sets the maximum distance that an influence can travel to be one grid spacing  $\Delta x$ . The speed of the bulk flow of the fluid is set by  $\mathbf{u}(\mathbf{J})$ , but for a fluid with pressure ( $w \neq 0$ ), sound waves can propagate on top of this bulk flow. The maximum speed of information in the fluid is then the sum of the (local) bulk velocity and the (global) sound speed. The physical sound speed is  $c_s = c\sqrt{w}$ , but because we are working in the comoving frame, this has to be scaled by  $a^{-1}$ , taking the redshifting of velocities into account. For the bulk velocity, we are *not* to use  $\mathbf{u} = a^{3w-1}\mathbf{J}/\varrho$  but rather  $\dot{\mathbf{x}} = a^{-1}\mathbf{u}$ , as we are interested in velocities with respect to the comoving  $\mathbf{x}$  grid. Our time step size must then be no larger than

$$\Delta t < \frac{\Delta x}{\sqrt{3}} \left( ca^{-1}\sqrt{w} + a^{3w-2} \max_{\mathbf{x}} \left| \frac{\mathbf{J}(\mathbf{x})}{\varrho(\mathbf{x})} \right| \right)^{-1}. \quad (3.11)$$

This is what is known as the Courant condition, here generalized to comoving coordinates. It is a necessary but generally not quite sufficient condition [14], so the actual time step size is chosen to be a fraction of the right-hand side of (3.11). The  $\sqrt{3}$  in (3.11) is needed when in three dimensions, because flow running diagonally on the grid needs to “zigzag” its way forward, in the sense that no information can travel between diagonal grid cells without passing through neighbouring grid cells with shared faces.

As (3.10) is not left/right symmetric (i.e. the left slopes are always computed using results obtained from the right slopes), it leads to clear anisotropic errors, especially in higher dimensional simulations. To correct this, the 8 possible orders (3 dimensions with an ordered pair of directions for each) are periodically cycled. Additionally, the time step size  $\Delta t$  is allowed to change only after each complete cycle.

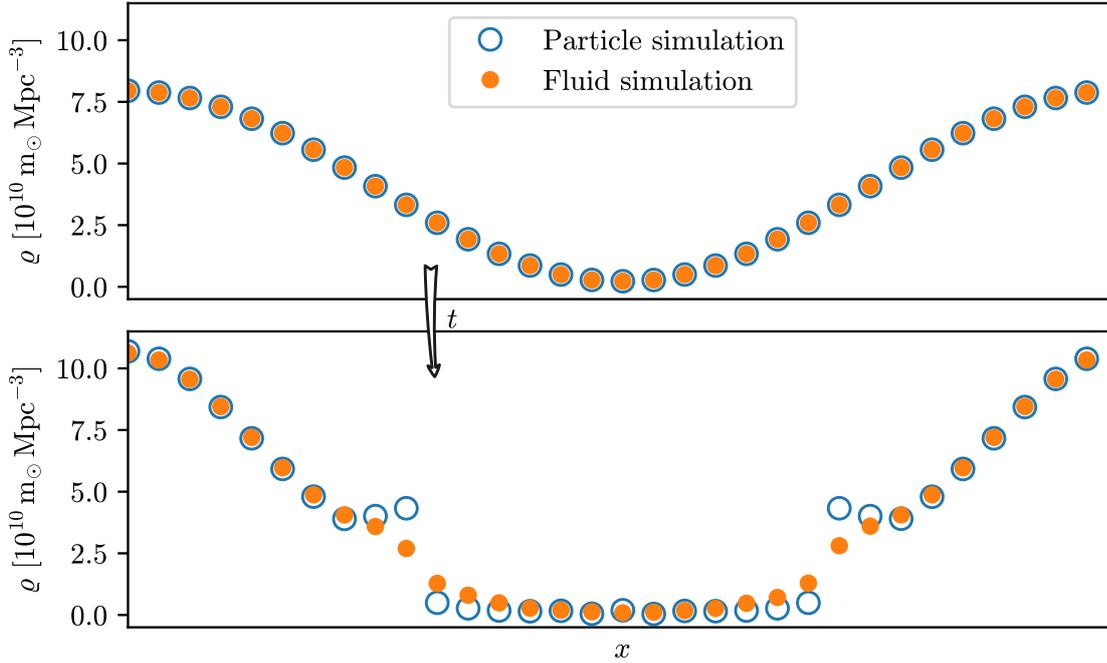
### 3.5.4 Vacuum Corrections

Since the density  $\rho$  is hiding inside both  $\varrho$  and  $\mathbf{J}$ , both the flux and the source term in the continuity equation (3.8) vanishes in the case of vacuum. Thus  $\varrho$  is bounded bounded from below, as it should. Due to the discrete time stepping however, it may happen that the density becomes slightly negative at certain points, which causes problems for the simulation and leads to blow-up of  $\mathbf{J}$ . This turns out to be a real problem even for the neutrino fluid, at least at late times and with high spatial resolution.

To counteract this problem is not trivial, as we cannot simply lower the time step size as even a slightly negative density has catastrophic consequences. The physically correct solution is to consider  $\varrho$  and  $\varrho^*$  for each fluid element during the MacCormack steps and limit the flux if it leads to a negative density. This is not so easily done however, as this limited flux has to be remembered between MacCormack steps and applied to both the left and right differentiation at a given cell interface. How such a positivity-preserving flux limiter can be implemented for single-step schemes is demonstrated in [15], where two new grids per interface (thus 6 new grids

---

\*The order of the MacCormack scheme can be made higher [13] in a fashion similar to Runge-Kutta methods, increasing both its accuracy and the allowed size of  $\Delta t$ . I have played with such higher orders, but found that the increase in computation time was too much.



**Figure 3.1** – 1D test simulations showing the vacuum corrections in action. The same initial conditions (top) are evolved as a matter fluid and as particles. The final states of the two simulations are shown at the bottom.

in 3D) are needed for storing the flux limiters. Exactly how this method can be applied to our two-step MacCormack method is unclear, as we have to accept the predictor step before we can compute the corrector step. Any detected negative densities resulting from the corrector step would have to alter the predictor step, which is not possible as this has already taken place. One could of course fix this by keeping all fluxes in memory instead of applying them immediately, resulting in the need for even for grids.

I decided that using positivity-preserving flux limiters was too complicated and memory-inefficient, and decided to simply smooth out regions with dangerously low densities. After the first MacCormack step, the  $\varrho$  and  $\varrho^*$  grids are compared and the number of time steps needed to bring each fluid element down to vacuum is computed, using linear extrapolation. If for some fluid element vacuum will be reached within a given number of time steps (currently this is set to 30), the  $3 \times 3 \times 3$  block around this element will be smoothed in a manner that respects conservation of both energy and momentum. All 27 fluid elements will give/receive some of its density to every other, the amount being inversely proportional to their squared distance and the estimated time before the vacuum state will be reached. The same smoothing is performed if after the second MacCormack step, any density is negative. That is, here we only consider low densities to be problematic if they are already negative. The smoothing cannot be applied immediately, as this would alter the remainder of the vacuum checking procedure where other elements of  $\varrho$  and  $\varrho^*$  are compared. Instead, these “vacuum corrections” are saved to a new grid and only applied afterwards, increasing the total number of grids per fluid scalar to 3. Since both  $\varrho$  and  $J$  are proportional to  $\rho$ , the values in  $J$  are smoothed alongside those of  $\varrho$ , though the check for negative values always takes place on  $\varrho$  only.

I have found that my smoothing scheme just described leads to well-controlled behaviour. The results are different from the physical correct solution only at steep boundaries between vacuum (or very low density regions) and high-density regions. A test of the smoothing is shown in figure 3.1, where the same initial conditions are evolved as particles and as a fluid. The initial conditions are a simple 1D sine curve with a period that matches the box size. As time evolves, the overdensity grows due to gravity, emptying out the region of low density. To enforce a future discontinuity in  $\rho$ , the initial velocity field diverges rather strongly from the center of the underdensity. For the particle and fluid simulation to be comparable, the fluid must be pressureless,  $w = 0$ . From the pressure term in the Euler equation (3.8),  $\propto \partial_i \rho$ , we see that pressure has a counteracting effect on the build-up of steep density gradients, and so this test shows the most extreme case. Figure 3.1 demonstrates that the fluid implementation is able to survive large regions of rapidly growing voids, though discontinuous interfaces between low and high density regions become smooth. As we do not expect such discontinuities in the cosmic neutrino fluid, this simple smoothing scheme is deemed adequate.

The vacuum correction test of figure 3.1 is included as part of the `CONCEPT` source code, together with a variety of others, including a full 3D cosmological comparison test between a particle and a pressureless fluid simulation. For the particle implementation to have validity, this is tested against full 3D cosmological simulations using the `GADGET-2` code.

### 3.5.5 Gravity

The only external force acting on the neutrino fluid is gravity. Gravity for particles was originally implemented in `CONCEPT` using the so-called PP, PM and P<sup>3</sup>M methods, detailed description of which can be found in [6]. The PM (particle-mesh) method interpolates the particles onto a mesh and then solves the Poisson equation on this mesh, after which the gravitational forces are interpolated back to the particle coordinates. As the fluids already lives on a mesh, it was obvious to extend the PM method to work for fluids also. At the same time, neither the PP nor the P<sup>3</sup>M method makes much sense for fluids, so `CONCEPT` only supports\* the PM<sup>†</sup> method for fluid components, and only this will be described further. We shall discuss the PM method for the case of a simulation involving both particles (matter) and a fluid (neutrinos).

The overall strategy of the PM method is this. Construct a grid (or mesh) of the total density  $\rho_{\text{tot}}(\mathbf{x})$  of all components via interpolation, be it particles or fluids. Now Fourier transform this mesh;  $\rho_{\text{tot}}(\mathbf{x}) \rightarrow \tilde{\rho}_{\text{tot}}(\mathbf{k})$ . Convert the grid values to that of the Fourier transformed potential,  $\tilde{\rho}_{\text{tot}}(\mathbf{k}) \rightarrow \tilde{\varphi}(\mathbf{k})$  using the Poisson equation (2.22), though now in Fourier space:

$$\tilde{\varphi}(\mathbf{k}) = -\frac{4\pi G a^2}{k^2} \tilde{\rho}_{\text{tot}}(\mathbf{k}). \quad (3.12)$$

Note that this is not defined for  $\mathbf{k} = \mathbf{0}$ , exactly corresponding to the mean density which was subtracted in (2.22). Removing the mean density then corresponds to setting  $\tilde{\varphi}(\mathbf{k}) = 0$ . Now do an inverse Fourier transform to obtain the potential in real space,  $\tilde{\varphi}(\mathbf{k}) \rightarrow \varphi(\mathbf{x})$ . Differentiate the  $\varphi$  grid using finite difference to obtain approximations for  $\partial_i \varphi$  at each grid point and interpolate this force back to the particle/fluid element positions and apply it. The fast Fourier transformed used impose periodic boundary conditions, which is exactly what we want. We

---

\*Though it *is* possible for gravity to be computed using different methods for different components in a multi-component simulation.

<sup>†</sup>The name ‘particle mesh’ is not really appropriate any more, as only the ‘mesh’ part is used for fluids. With no other commonly used name known to me, I shall continue referring to it as the PM method.

thus obtain periodicity of gravity for free, a feature which requires quite some effort using non-Fourier methods.

The last step involving differentiation requires a separate grid and so doubles the memory consumption of the method. Alternatively, the differentiation could be done in Fourier space by multiplying by  $i\mathbf{k}$ , at the cost of tripling the number of needed Fourier transforms and therefore the computation time. As the bottleneck\* of the time stepping is usually the computation of gravity, I have settled with real-space differentiation.

We shall now take a closer look at the interpolations and the complications that arises when having both particles and fluids in the same simulation. Though the same mesh in memory is used to store  $\rho_{\text{tot}}(\mathbf{x})$ ,  $\tilde{\rho}_{\text{tot}}(\mathbf{x})$ ,  $\tilde{\varphi}(\mathbf{x})$  and  $\varphi(\mathbf{x})$  values, we shall refer to this grid consistently as the  $\varphi$  grid. The CIC (cloud-in-cell) method is used for the density interpolations, which may be described as distributing each of the particles (or fluid elements) throughout the  $\varphi$  grid, with weights at each grid point given by the geometric overlap between the particles and the grid point, where both the particles and the grid points are imagined to have cubic shapes with side lengths equal to the grid spacing of the  $\varphi$  mesh,  $H$ . Denoting the weight at mesh point  $\mathbf{x}_m$  of a particle at  $\mathbf{x}_p$  as  $W(\mathbf{x}_m - \mathbf{x}_p)$ , we have

$$\begin{aligned} W(x, y, z) &= H^{-3} \Pi\left(\frac{x}{H}, \frac{y}{H}, \frac{z}{H}\right) * \Pi\left(\frac{x}{H}, \frac{y}{H}, \frac{z}{H}\right) \\ &= \begin{cases} \left(1 - \frac{|x|}{H}\right) \left(1 - \frac{|y|}{H}\right) \left(1 - \frac{|z|}{H}\right) & \text{if all } |x|, |y|, |z| < H, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \quad (3.13)$$

where  $\Pi$  is the cubic top-hat function

$$\Pi(x, y, z) = \begin{cases} 1 & \text{if all } |x|, |y|, |z| < \frac{1}{2}, \\ 0 & \text{otherwise} \end{cases}$$

and  $*$  denotes convolution. Note that the weights (3.13) are only non-zero for the 8 grid points closest to the particle. If the fluid grids happen to be of the same size as the  $\varphi$  grid, all grid points coincide and the CIC interpolation reduces to direct copying.

All source terms, including gravity, are applied to the system half a time step out-of-sync with the rest of time evolution, meaning flux terms for fluids and position updates for particles. This is a relic of the symplectic leapfrog integration originally designed for the particle evolution, where updates of positions and momenta are interwoven in this out-of-sync fashion, making the time integration for particles much more stable<sup>†</sup>. Looking back at the Poisson equation for a single-fluid universe, (3.9), we see that the right-hand-side includes a factor  $a^{-3w-1}$ , which is both time-varying and component-dependent. To generalize this to multi-component situations, such as a single fluid and a particle component, what is initially interpolated to the  $\varphi$  grid is not just the  $\rho = a^{-3(1+w)}\varrho$  for each component, but rather

$$a^2 \rho_m(\mathbf{x}) = \frac{W(\mathbf{x})}{H^3} * a^2 \rho_{\text{tot}}(\mathbf{x}),$$

---

\*The FFTW library is used for parallel, in-place fast Fourier transformations of real-valued 3D data, with computation times scaling as  $\mathcal{O}(N \log N)$ ,  $N$  being the number of grid points.

<sup>†</sup>Note that because the MacCormack scheme itself splits up a single time step into two half steps, applying the gravitational forces out-of-sync by half a time step 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 often as flux terms, it is applied fairly with respect to the predictor and corrector step.

$$a^2 \rho_{\text{tot}}(\mathbf{x}) = \sum_{\alpha} \begin{cases} m_{\alpha} \sum_{i=1}^N \delta(\mathbf{x} - \mathbf{x}_{\alpha,i}) \Delta t^{-1} \int_t^{t+\Delta t} a^{-1} dt & \text{(particles),} \\ \varrho_{\alpha}(\mathbf{x}) \Delta t^{-1} \int_t^{t+\Delta t} a^{-3w_{\alpha}-1} dt & \text{(fluids),} \end{cases} \quad (3.14)$$

where  $\rho_{\text{m}}$  is the mesh-interpolated density,  $\alpha$  runs over all components,  $m_{\alpha}$  is the mass of the particles of component  $\alpha$  and  $\mathbf{x}_{\alpha,i}$  is the position of the  $i$ 'th particle of component  $\alpha$ . Looking at (3.14), the discreteness of the particle description is obvious, while the discreteness of the fluid description is less transparent. It is to be understood that numerically, all fields are really only defined at the grid points — or equivalently, zero everywhere except at the grid points — and so a more rigorous notation would distinguish between physical fields and numerical grids, e.g.  $\varrho_{\alpha, \text{III}}(\mathbf{x}) \equiv \varrho_{\alpha}(\mathbf{x}) \text{III}(\mathbf{x}/H)$ , where  $\varrho_{\alpha, \text{III}}(\mathbf{x})$  is the numerical grid of density values of species  $\alpha$  and  $\text{III}$  is the Dirac comb;  $\text{III}(\mathbf{x}/H) = H^3 \sum_{\mathbf{n} \in \mathbb{Z}^3} \delta(\mathbf{x} - H\mathbf{n})$

In the Poisson equation (3.12), the actual density  $\rho_{\text{tot}}$  is needed. Though what lives on the grid is really the interpolated values  $\rho_{\text{m}} \propto W * \rho_{\text{tot}}$ . Simply ignoring this difference leads to errors on scales comparable to the grid spacing  $H$ . 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 (3.12) then turns into

$$\widetilde{\varphi}(\mathbf{k}) = -\frac{4\pi G a^2}{\mathbf{k}^2} \frac{H^3}{\widetilde{W}(\mathbf{k})} \widetilde{\rho}_{\text{m}}(\mathbf{k}), \quad (3.15)$$

where the Fourier transform of the CIC weighting function (3.13) is

$$\widetilde{W}(k_x, k_y, k_z) = H^3 \left[ \text{sinc}\left(\frac{Hk_x}{2}\right) \text{sinc}\left(\frac{Hk_y}{2}\right) \text{sinc}\left(\frac{Hk_z}{2}\right) \right]^2.$$

Equation (3.15) results in a properly deconvolved potential, though our interest is really the resulting forces at the locations of the particles/fluid elements. Since another CIC interpolation is used to interpolate the forces from the grid points of the  $\varphi$  grid and onto the particle/fluid elements, two CIC deconvolution are actually needed, and so the potential actually calculated in `CONCEPT` is  $\widetilde{\varphi}(\mathbf{k}) H^3 / \widetilde{W}(\mathbf{k})$ .

The CIC deconvolution is not perfect and does not take into account the exact positions of every particle/fluid element. For fluids with a grid size matching that of the  $\varphi$  grid, no CIC-interpolation is needed at all, and so we are actually worse off performing the two deconvolutions. In my simulations I tend to use the same grid size for fluids and  $\varphi$ , and I have indeed found that the small-scale results are erroneously influenced by these CIC-deconvolutions. In simulations with both a particle and a fluid component, however, we either have to deconvolve  $\widetilde{\rho}_{\text{m}} \propto \widetilde{\rho}_{\text{tot}}$  or we do not; no component-specific control is available. This limitation could be surpassed by introducing a separate  $\varphi$  grid, solving the Poisson equation separately for particle and fluid components, allowed by the linearity of the Poisson equation.

## 4 Bridging 3D Real and 1D Fourier Space

The raw\* output of a CONCEPT run, or any other similar cosmological simulation codes, is what is called a snapshot. This is simply a memory dump of the current state of the simulated universe, i.e. a list of values of all fluid variables, particle positions and momenta, together with a few informations about the overall cosmology. Such snapshots can generally not be directly compared to observations or even to other simulations. First, the overall statistics must be extracted, which is embodied in the power spectra  $P_\alpha(k)$ , measuring the amount of structure at a given Fourier mode  $k$ .

The power spectra are thus 1D (Fourier space) representations of 3D (real space) distributions of the different species. This reduction in dimensionality means that information is lost, and so we cannot recreate the unique 3D distribution from which a given power spectrum was originally computed. We *can* however construct an infinite set of 3D distributions which all map to the same power spectrum. This reverse process of *realising* a power spectrum as a 3D distribution is the key to initial condition generation, since physically accurate power spectra at early times can be computed using linear perturbation theory. Popular numerical codes for such linear theory include CAMB [16] and CLASS [1], the latter of which has been fully integrated as a library module into CONCEPT.

### 4.1 Power Spectra

The power spectrum for species  $\alpha$  may be defined simply as

$$P_\alpha(k) \propto \langle |\delta_\alpha(\mathbf{k})|^2 \rangle, \quad (4.1)$$

where angle brackets denote ensemble averaging, reducing the many instances of  $\delta_\alpha(\mathbf{k})$  with  $|\mathbf{k}| = k$  to a single<sup>†</sup> number. In CONCEPT, the normalization of all power spectra is chosen to be the (comoving) box volume. With this choice of normalization, individual power spectra are computed simply as

$$P_\alpha(k) = V \left\langle \left| \mathcal{F} \left\{ \begin{array}{l} N^{-1} W(\mathbf{x}) * \sum_{i=1}^N \delta(\mathbf{x} - \mathbf{x}_{\alpha,i}) \quad (\text{particles}) \\ \left[ \int \varrho_\alpha(\mathbf{x}) d^3\mathbf{x} \right]^{-1} W(\mathbf{x}) * \varrho_\alpha(\mathbf{x}) \quad (\text{fluids}) \end{array} \right. \right| \right|^2 \right\rangle, \quad (4.2)$$

where  $\mathcal{F}$  is the Fourier transform and  $V$  is the box volume. The  $\varphi$  grid is reused for all power spectra computations, and so the interpolation via convolution with  $W(\mathbf{x})$  is just as described in subsection 3.5.5. Just as we did there, we really ought to deconvolve for the CIC interpolation

---

\*CONCEPT is able to output power spectra, full 3D renders or even 2D projected renders directly in the terminal. The image on the front page is an example of such a 3D render, of a simulation with both matter and neutrinos.

<sup>†</sup>Actually, CONCEPT also computes the standard deviation of this mean, which is useful for further processing of the raw power spectrum.

by dividing by  $\widetilde{W}(\mathbf{k})$  after the Fourier transformation. Currently, deconvolution is done for particles but not for fluids, though fluids with grid sizes different from that of  $\varphi$  ought to be deconvolved as well.

In addition to individual power spectra, CONCEPT can compute the total power spectrum  $P_{\text{tot}}$  for all species combined. Due to the non-linear  $|\bullet|^2$  operation in (4.1) and (4.2), power spectra do not simply add, and so this total power spectrum must be computed independently. The recipe is similar to (4.2) with a sum over all species inside the Fourier transform, but now each species has to be normalized in a compatible way. This is done by normalize each component based on their total mass, essentially by replacing in (4.2)  $N^{-1}$  with  $m_\alpha$  and the reciprocal integral with  $a^{-3w_\alpha}(H_\alpha/H)^3$ , where  $H_\alpha$  is the grid spacing of fluid component  $\alpha$ . We are faced with the same dilemma of whether or not to deconvolve for the CIC interpolation when computing  $P_{\text{tot}}$  in a simulation with both a particle component and a fluid component with a grid size matching that of  $\varphi$ , just as in subsection 3.5.5

## 4.2 Realisations

In CONCEPT, a given fluid is instantiated with a given number of fluid variables, e.g.  $\varrho$  and  $\mathbf{J}$ , which are realised using the methods described below. If the Boltzmann hierarchy is simply truncated, this is the end of the story as far as realisations are concerned. If however the Boltzmann hierarchy is closed using  $\sigma^i_j$  from linear theory, a realisation of  $\sigma^i_j(t)$  is performed at each time step.

The present power spectrum is largely a result of gravity doing its work for billions of years. Though to get gravity going in the first place, some initial inhomogeneities must be present. These initial or *primordial* density fluctuations are often ascribed to quantum fluctuations expanded to macroscopic scales by inflation. The primordial power spectrum  $P_s$  is often assumed to be in the form of a power law

$$P_s(k) = A_s \left( \frac{k}{k_{\text{pivot}}} \right)^{n_s - 1}.$$

The departure from the primordial power spectrum as time goes on is captured by what is known as a transfer function  $T$ ,

$$P_T(t, k) = 2\pi^2 T^2(t, k) k^{-3} P_s(k), \quad (4.3)$$

where the exact definition of  $T$  in (4.3) is that used by CLASS. The power spectra discussed so far has all been with respect to the density, but power spectra might equally well be constructed from any other scalar quantity, such as the velocity divergence. Thus  $T$  is not just unique per species but also per quantity in question. CONCEPT is able to extract various transfer functions using CLASS, compute the matching power spectrum using (4.3) and then realise it as a 3D fluid or particle distribution. Exactly how this is done is described below.

### 4.2.1 Density, Velocity and Shear

CONCEPT is able to realise dark matter, baryons\* and neutrinos as fluids with density, velocity and shear. For dimensionless  $T$ , we see that (4.3) is normalized by volume, in line with the

---

\*It is also possible to realise a single ‘matter’ component using both transfer functions for dark and baryonic matter.

normalization in CONCEPT. We can then simply plug (4.3) into (4.1), “invert” the ensemble averaging  $\langle \bullet \rangle$  and do an inverse Fourier transform to obtain

$$\delta_\alpha(t, \mathbf{x}) = \mathcal{F}^{-1} \left[ T_{\delta_\alpha}(t, k) \sqrt{2\pi^2 A_s} k^{-3/2} \left( \frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \mathcal{R}(\mathbf{k}) \right], \quad (4.4)$$

where  $T_{\delta_\alpha}$  is the specific transfer function for the density contrast of species  $\alpha$  and  $\mathcal{R}(\mathbf{k})$  is a field of random, complex numbers, producing the 3D realisation. Specifically, we choose  $\mathcal{R}(\mathbf{k})$  to be a Gaussian\* random field, in which case it is characterized by its mean and variance, which not to alter the power spectrum must be chosen as 0 and 1, respectively. For a given realisation of  $\mathcal{R}(\mathbf{k})$  on a grid, we are then able to transform the transfer function  $T_{\delta_\alpha}$  into a the 3D real-space realisation of the corresponding density contrast  $\delta_\alpha$ , and from here to  $\varrho_\alpha = a^{3(1+w_\alpha)} \bar{\rho}_\alpha (\delta_\alpha + 1)$ .

The story is very similar for the velocity. A minor complication arises because the velocity variable solved by CLASS is the velocity *divergence*  $\theta = \nabla_{\mathbf{x}} \cdot \mathbf{u}$ , with transfer function  $T_{\theta_\alpha}(k)$ . As we are interested in the full velocity, we invert the divergence operator while in Fourier space, resulting in

$$\mathbf{u}_\alpha^i(t, \mathbf{x}) = \mathcal{F}^{-1} \left[ \left( -i \frac{k^i}{k^2} \right) T_{\theta_\alpha}(t, k) \sqrt{2\pi^2 A_s} k^{-3/2} \left( \frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \mathcal{R}(\mathbf{k}) \right], \quad (4.5)$$

from which we can get  $\mathbf{J} = a^{-3w+1} \varrho \mathbf{u}$ .

Crucially, the realisation of the Gaussian random field  $\mathcal{R}(\mathbf{k})$  must be the same when using both (4.4) and (4.7) in the same simulation (even for different species), corresponding to each point in space having one (complex) degree of freedom. This actually reduces to a single real degree of freedom, once we recognize the symmetry which must be imposed on  $\mathcal{R}(\mathbf{k})$ . Since all of our real-space fields are real (as opposed to complex), their Fourier transforms all obey the symmetry  $\delta_\alpha(\mathbf{k}) = \delta_\alpha(-\mathbf{k})^*$  (\* denoting complex conjugation), using  $\delta_\alpha$  as an example. This symmetry requirement trickles down to  $\mathcal{R}(\mathbf{k})$ , leaving one real degree of freedom per grid point, which might be physically ascribed to primordial fluctuations in the inflaton field. This symmetry must be imposed explicitly in the code, which become non-trivial in our case of a distributed grid.

CLASS solves for the anisotropic stress scalar, with transfer function  $T_{\sigma_\alpha}(t, k)$ , resulting in [17]

$$(\sigma^i_j)_\alpha(t, \mathbf{x}) = \mathcal{F}^{-1} \left[ -\frac{3}{2} \left( \frac{k^i k_j}{k^2} - \frac{1}{3} \delta^i_j \right) T_{\sigma_\alpha}(t, k) \sqrt{2\pi^2 A_s} k^{-3/2} \left( \frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \mathcal{R}(\mathbf{k}) \right].$$

As  $\sigma^i_j$  is symmetric, it consists of 6 independent elements. As each realisation consists of a fast Fourier transform, the needed realisation of  $\sigma^i_j$  at each time step becomes the bottleneck for such simulations. It is therefore important that each element is realised only once. Additionally, having 6 additional fluid grids more than doubles the memory footprint. The strategy used by CONCEPT is therefore to realise only a single element at a time, updating  $\mathbf{J}^i$  according to the stress flux  $\propto \partial^j \sigma^i_j$  in the Euler equation (3.8), and similarly updating  $\mathbf{J}^j$  according to  $\propto \partial^i \sigma^j_i$ , remembering that  $\sigma^i_j = \sigma^j_i$ . Then we need the next element of  $\sigma^i_j$ , which simply

---

\*With this choice, the power spectrum (two-point correlation function) contains all information about the 3D distribution.

reused the memory allocated for the first element. Since all flux terms resulting in the time derivative of the highest-moment fluid variable is guaranteed to only contain this variable to linear order (otherwise too many free indices will appear), it will always be possible to use up only a single grid of memory for the highest-order fluid variable, when using the strategy of continuous realisation.

### 4.2.2 The Zel'dovich Approximation

For particle components, a slightly different approach is used for realisation, known as the Zel'dovich approximation. Here, all particles are placed on regular cubic grid points, and then displaced by an amount  $\psi_\alpha(t, \mathbf{x})$ , given in the comoving frame. If the particles are imagined to have originated from the grid points and travel linearly, their velocities must be  $a^{-1}\mathbf{u}_\alpha = \dot{\mathbf{x}}_\alpha = \dot{\psi}_\alpha$ . Substituting  $a\dot{\psi}_\alpha$  for  $\mathbf{u}_\alpha$  in the continuity equation (e.g. (2.23)) and neglecting terms  $\propto \dot{\psi} \cdot \nabla_x \delta_\alpha$ , we get  $\nabla_x \cdot \dot{\psi} = -\dot{\delta}_\alpha$ , which can be integrated to

$$\nabla_x \cdot \psi_\alpha(t, \mathbf{x}) = -\delta_\alpha(t, \mathbf{x}). \quad (4.6)$$

Inverting the divergence operator of (4.6) in Fourier space just like we did for the velocity divergence in (4.7), the realised displacement field becomes

$$\psi_\alpha^i(t, \mathbf{x}) = \mathcal{F}^{-1} \left[ \left( i \frac{k^i}{k^2} \right) T_{\delta_\alpha}(t, k) \sqrt{2\pi^2 A_s} k^{-3/2} \left( \frac{k}{k_{\text{pivot}}} \right)^{\frac{n_s-1}{2}} \mathcal{R}(\mathbf{k}) \right]. \quad (4.7)$$

With the realised displacement grid, the regular particle positions are updated by an amount  $\psi_\alpha(t, \mathbf{x})$ . The velocities  $\mathbf{u}_\alpha = a\dot{\psi}_\alpha$  are computed using the approximation  $\dot{\psi}_\alpha = (\dot{D}/D)\psi_\alpha$ , where  $D$  and  $\dot{D}$  are the linear growth function and its time derivative, both computed by CLASS.

### 4.2.3 Gauge Transformations

CLASS is able to compute transfer functions in either the synchronous gauge or the conformal Newtonian gauge, but what we need is the  $N$ -body gauge. By default, CONCEPT asks CLASS for all transfer functions in the synchronous gauge and then transforms them to  $N$ -body gauge. To first order, these transformations are [17]

$$\begin{aligned} T_{\delta_\alpha}^{N\text{-body}} &= T_{\delta_\alpha}^{\text{sync}} + 3\dot{a}(1 + w_\alpha)k^{-2}T_{\theta_{\text{tot}}}^{\text{sync}}, \\ T_{\theta_\alpha}^{N\text{-body}} &= T_{\theta_\alpha}^{\text{sync}} + \frac{a}{2}\dot{h} - 3ak^{-2}\partial_t(\dot{a}T_{\theta_{\text{tot}}}^{\text{sync}}), \end{aligned}$$

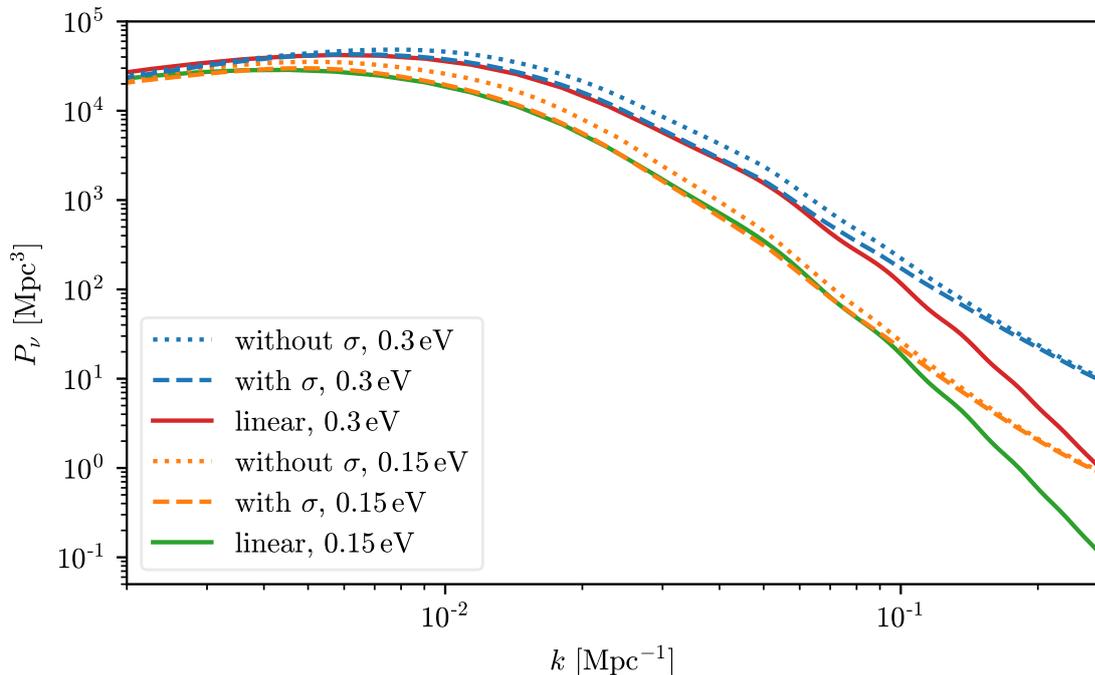
where  $h$  is the synchronous metric perturbation and  $T_{\theta_{\text{tot}}}^{\text{sync}}$  is the transfer function for the divergence of the total velocity field, both of which are directly accessible from CLASS. The stress-tensor  $\sigma_j^i$  need not be transformed, as here all of the three gauges are identical to first order.

## 5 Current State and Preliminary Results

It is only very recently that the state of [3] has progressed to the point of actually being able to run full-fledged neutrino simulations. If trying to run a neutrino-only simulation, spurious chock waves are excited at very small scales, which then propagate to larger scales. After much tinkering I have concluded that these “neutrino oscillations” arise from the pressure term in the Euler equation, and I believe that they are described by the Godunov theorem [18], stating that second-order partial differential equation solvers tend to generate such oscillations near discontinuities. Unlike the matter transfer functions, the neutrino transfer functions contain structure at all scales due to acoustic oscillations. This means that regardless of the resolution, the neutrino fluid starts out as being jagged, leading to badly defined, large slopes.

In an effort to remove these spurious oscillations, I have experimented with so-called ‘total variation diminishing’ extensions to the MacCormack scheme, specifically the method of [19]. In such methods, flux limiters (similar in essence to those of [15]) are applied in order to diminish discontinuities, effectively smoothing out the fluid. I have had some success with this, though traces of the oscillations remain.

To my surprise, when running the actual simulations using both a neutrino fluid and matter particles, the spurious oscillations never evolve. Thus somehow the effects of gravity stabilises

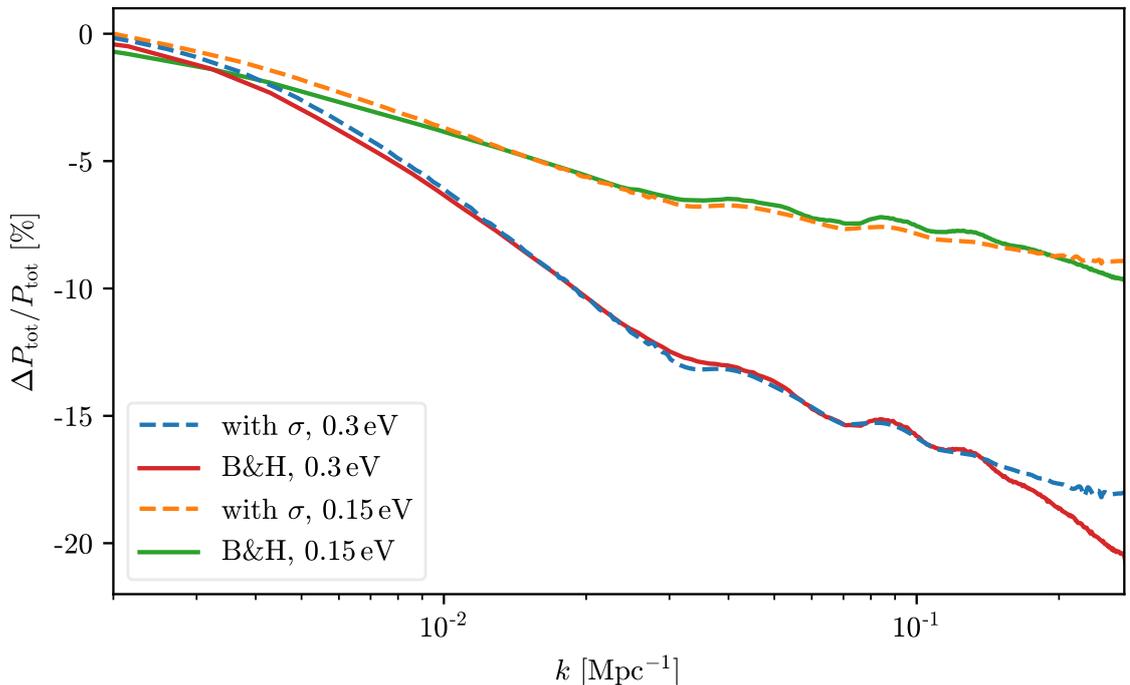


**Figure 5.1** – Neutrino power spectra at  $a = 1$ , resulting from linear theory and CONCEPT simulations with and without the inclusion of shear. A  $\Lambda$ CDM cosmology including three neutrinos with a total mass of either 0.3 eV or 0.15 eV, is used. The simulation box is 4096 Mpc/h,  $h = 0.7$ . For the CONCEPT simulations, a linear grid size of 600 is used for both the fluid grids and the potential grid, and the simulation is started at  $a = 0.02$ .

the neutrino fluid, apparently rendering the spurious oscillations harmless in practice. In figure Figure 5.1, neutrino power spectra resulting from cosmological CONCEPT simulations are compared against linear theory predictions. Both CONCEPT simulations with and without shear are shown, meaning simulations where the Boltzmann hierarchy has been closed using linear shear and where it is closed simply by truncating it after the second fluid variable. We see that the simulations without the shear lead to too much power compared with linear theory, but including linear shear reduces the power almost all of the remaining way.

As the scales shown in figure 5.1 are very large, we expect linear theory to yield correct predictions. At the smaller scales, the agreement between the linear and non-linear power spectra ends. This is at least partly due to the CIC deconvolution of the potential, which as described in subsection 3.5.5 really should not be carried out for the neutrino component. Comparing with figure 3 (right) of [20], where neutrinos are implemented using a hybrid fluid/particle scheme, we see that an increase in power is indeed expected for the non-linear power spectra at  $k \gtrsim 0.1 \text{ Mpc}^{-1}$ , though a neutrino mass of  $\sum m\nu = 1.2 \text{ eV}$  is used in [20]. Notice also how the point at which the simulation starts to deviate from linear theory happens at a larger scale for the heaviest neutrino, strongly suggesting that what we see is indeed non-linear effects.

As directly observing the cosmic neutrino structure is beyond our reach, plots like 5.1 cannot be compared with observations. For this task, a more useful plot is shown in figure 5.2, demonstrating the effect of the neutrino component on the total power spectrum. We see a very clear diminishing effect on the total power from the neutrino component, increasing with



**Figure 5.2** – Relative total power spectra at  $a = 1$ , for CONCEPT simulations and simulations of B&H [20]. The specifications are the same as those in figure 5.1. Here,  $P_{\text{tot}}$  is the power in a matter-only simulation and  $\Delta P_{\text{tot}} = P_{\text{tot, dyn}} - P_{\text{tot}}$ , where  $P_{\text{tot, dyn}}$  is the power in a simulation with both a dynamic matter and neutrino component.

---

neutrino mass. Moreover, the results from `CONCEPT` agrees very well with those of Brandbyge and Hannestad [20] for all but the smallest scales.

Though I have not yet carried out proper comparisons with other non-linear codes in the highly non-linear regime (soon to come!), I permit myself to declare the very close agreement seen in figure 5.1 and 5.2 for a proof of `CONCEPT`.



## 6 Outlook

With the self-consistent, fast and relatively simple neutrino implementation of CONCEPT, many possibilities arise. First, detailed comparisons with other codes in the non-linear regime need to be carried out, testing the accuracy and stability of the method.

Convergence tests of the closure of the Boltzmann hierarchy will also need to be tested, i.e. running simulations with full non-linear shear and closing the Boltzmann hierarchy using the linearised rank-3 variable, but also simulations where only the density is fully non-linear. We could continue implementing still higher order variables indefinitely, but I suspect that once non-linear shear is taken into account, no further precision may be gained, as other inaccuracies (e.g. from gravity) should begin to dominate.

The current simple (but accurate) method of realising every independent element of the highest order variable — i.e. 6 for shear — at each time step is computationally expensive, and will be even more so if going to higher order variables. One might then wish to spend some time replacing these realisations with faster alternatives, e.g. realising only the anisotropic stress *scalar* and from it compute all elements of the stress tensor by applying differential operators to this scalar in real space.

The problem of having a single potential grid for both particle and fluid components should be solved, so that the small-scale structure of fluids is not scrambled by the CIC deconvolutions. This should easily be taken care of by introducing separate potentials for particle and fluid components, though at the expense of doubling the computational cost of gravity.

The spurious “neutrino oscillations” appearing in neutrino-only simulations could be studied further and perhaps eradicated completely. However, since this problem does not seem to arise in actual cosmological simulations at all, this does not have a high priority.

A paper about the techniques described in this report — together with results from simulations yet to be run — is under construction and will be published this summer. Once precise comparisons with other non-linear codes has been carried out, we can start looking at more subtle things than just the power spectrum, e.g. the effects on individual dark matter halos from the neutrinos.

If we are able to demonstrate the usefulness of the technique describes in the report through CONCEPT, the hope is that it will be implemented in RAMSES and used to produce some of the enormous simulations required for the Euclid mission.

Though the fluid implementation is done with neutrinos in mind, it really is a much more general framework. As such, one could use it to study completely different things, such as various kinds of dynamical dark energies.



# References

- [1] Diego Blas, Julien Lesgourgues, and Thomas Tram. The cosmic linear anisotropy solving system (class). part ii: approximation schemes. *Journal of Cosmology and Astroparticle Physics*, 2011(07):034, 2011.
- [2] PAR Ade, N Aghanim, M Arnaud, M Ashdown, J Aumont, C Baccigalupi, AJ Banday, RB Barreiro, JG Bartlett, N Bartolo, et al. Planck 2015 results-xiii. cosmological parameters. *Astronomy & Astrophysics*, 594:A13, 2016.
- [3] Jeppe Dakin. CONCEPT - The COsmological  $N$ -body Code in PyThon. 2015. <https://github.com/jmd-dk/concept>.
- [4] Volker Springel. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, 364, 2005.
- [5] Romain Teyssier. Cosmological hydrodynamics with adaptive mesh refinement: a new high resolution code called ramses. *Astron. Astrophys.*, 385:337–364, 2002.
- [6] 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. <https://github.com/jmd-dk/concept/raw/master/thesis.pdf>.
- [7] Scott Dodelson. *Modern Cosmology*. Academic Press, Elsevier Science, 2003.
- [8] Jacob Brandbyge and Steen Hannestad. Fourier imaging of non-linear structure formation. *JCAP*, 1704(04):032, 2017.
- [9] Chung-Pei Ma and Edmund Bertschinger. Cosmological perturbation theory in the synchronous and conformal Newtonian gauges. *Astrophys. J.*, 455:7–25, 1995.
- [10] Jacob brandbyge, private communication.
- [11] R. W. Maccormack. The effect of viscosity in hypervelocity impact cratering. *AIAA, American Institute of Aeronautics and Astrophysics*, 1969.
- [12] Justyna Machalinska-Murawska and Michał 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.
- [13] Ray Hixon and R Hixon. On increasing the accuracy of maccormack schemes for aeroacoustic applications. In *3rd AIAA/CEAS Aeroacoustics Conference*, page 1586, 1997.
- [14] William H Press. *Numerical recipes in Pascal: the art of scientific computing*, volume 1. Cambridge University Press, 1989.
- [15] Xiangyu Y Hu, Nikolaus A Adams, and Chi-Wang Shu. Positivity-preserving method for high-order conservative schemes solving compressible euler equations. *Journal of Computational Physics*, 242:169–180, 2013.

- [16] Antony Lewis, Anthony Challinor, and Anthony Lasenby. Efficient computation of CMB anisotropies in closed FRW models. *Astrophys. J.*, 538:473–476, 2000.
- [17] Thomas tram, private communication.
- [18] Sergei K. Godunov. A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. *Matematicheskii Sbornik*, 47(07), 1959.
- [19] Dongfang Liang, Binliang Lin, and Roger 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.
- [20] 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.