UNIVERSITY OF OSLO
Institute of Theoretical Astrophysics

Using needlets to estimate the power spectrum

Master Thesis

Tony A. Ingebrigtsen

June 1, 2010

Preface

I would like to thank my supervisor Frode K. Hansen for providing the idea and support for this thesis, and Sandro Scodeller for the introduction to needlets.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The black, empty space between the stars in the sky contains a very faint radiation that was first detected in 1964. It had no relation to any known source, and no matter which direction one looked the same type of radiation could be observed. It was everywhere. This radiation was named the Cosmic Microwave Background radiation, or CMB for short.

The CMB is the afterglow of the Big Bang. In the beginning the universe was a hot, dense soup of elementary particles and radiation, and impenetrable to light. Then, after about 300.000 years the universe suddenly got transparent, and light was finally able to shine through space. Some of this light has continued to pass through the expanding universe unhindered for the last 13.7 billion years until it hit our telescopes, and this is the CMB we are observing today.

The CMB is remarkably uniform, showing almost exactly the same temperature in all directions. Almost. There turned out to be small fluctuations present. Some of these variations were on a small scale, some stretched over large parts of the sphere. Looking at figure 1.1 it is possible to notice this effect. The figure shows an image of the variations in the CMB, as observed by the WMAP project [2]. We clearly see the tiny fluctuations, but we also get a sense of warmer and colder areas stretching over larger scales.

By statistical studies of the variations, it is possible to characterize the power of these fluctuations on different scales, leading to what we call the power spectrum. The power spectrum estimated from the WMAP results is shown

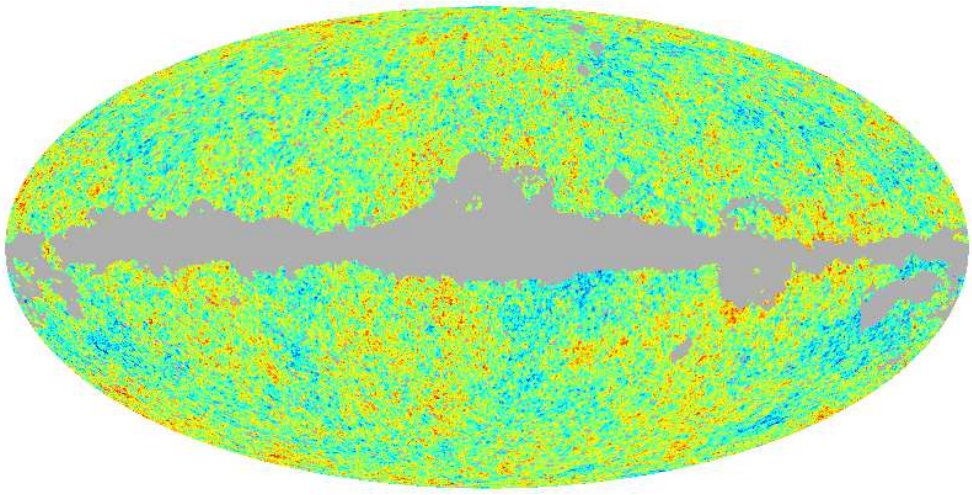

Figure 1.1: The CMB anisotropies as seen in WMAP. The grey mask covers areas to remove unwanted radiation from the galaxy and various sources.

in figure 1.2, and here we see the power of the variations for the first 1000 scales. The scales are known as multipoles.

The power spectrum is a tremendously important tool in cosmology. It can be used to test cosmological models and make constraints on the cosmological parameters. The observed variations in the CMB was created by sound waves traveling through the cosmic fluid that existed right after the big bang. These sounds were created and influenced by the behaviour and detailed composition of the early universe. The exact shape of the power spectrum is therefore heavily dependent on what the universe was like in the beginning, and the CMB is an observational window into this earliest time.

To make the best possible estimates we want better images of the CMB. The extraction of the power spectrum from these images are not a trivial matter however. It involves computationally heavy statistical calculations, and with an increasing amount of data to analyze we are rapidly approaching the limits of what we are able to handle with conventional, brute-force methods. New experiments are already underway to give even better resolution images, and to be able to handle these we need more efficient techniques to be able to estimate the power spectrum in a computationally reasonable time.

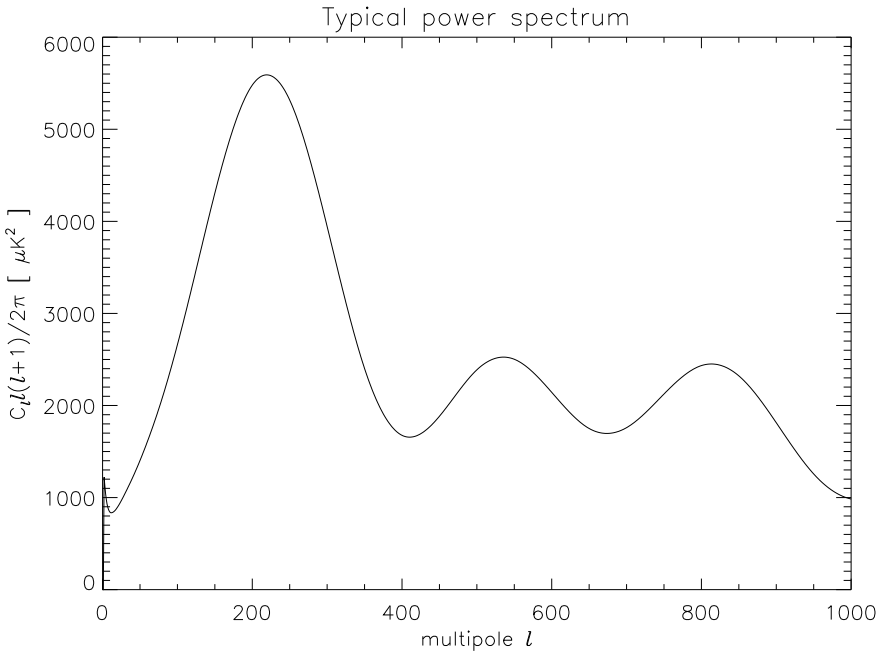

Figure 1.2: The WMAP power spectrum up to $\ell = 1000$.

One such technique is to use *needlet transformations*. Needlets belong to a special family of functions, known as wavelets. The needlets are constructed to be able to pick out selected scales from the CMB maps. Due to the special properties of the needlets it may be possible to create algorithms that promises to be much simpler and faster than conventional methods for estimating the power spectrum.

In this thesis we will look at one specific algorithm for estimating the power spectrum, developed by Oh, Spergel and Hinshaw [9]. This algorithm will be combined with needlet transformations, and it is then hoped that we will be able to get good, fast estimates for the power spectrum.

Chapter 2

Background

Before we begin with actually trying to make estimates of the power spectrum, it is useful to take a look at the cosmological model. We will examine what the model predicts, and why the power spectrum is such a useful characteristic. We will also look closer at how to get statistical information from the CMB maps.

2.1 Cosmology

The birth of modern cosmology took place in the first decades of the 20. century. In 1915 Einstein presented his theory of General Relativity, making it possible to describe the gravitational evolution of the timespace continuum. And in 1929 Hubble discovered evidence of an expanding universe. The galaxies in the universe were all moving away from us, and at a rate proportional to their distance from us. Combined these lead to the familiar Big Bang theory we have today.

With the discovery of the Cosmic Microwave Background radiation in 1964 it has become possible to gain information about the details of the birth and evolution of the universe. From the CMB we are able to extract the power spectrum, and this is greatly influenced by the details of the cosmological model. The following is based on [3] and [4].

2.1.1 Cosmological models

From observations we have evidence for a homogeneous, isotropic universe that is expanding uniformly. On large scale the mass seems to be evenly distributed, and this we call homogeneity. By isotropy we mean that everything looks the same in every direction, there are no directions in the sky that stands out.

Hubble discovered that the universe was expanding uniformly, and it is common to describe this by using the scale factor $a(t)$. The scale factor describes how the distances in the universe vary with time, and by using it we can set up a coordinate system known as a comoving grid. This is a coordinate system that follows the expansion of the universe. To illustrate the usefulness of the comoving grid we may compare it with what happens in real space. Figure 2.1 shows the expansion of the universe as we see it in real space.

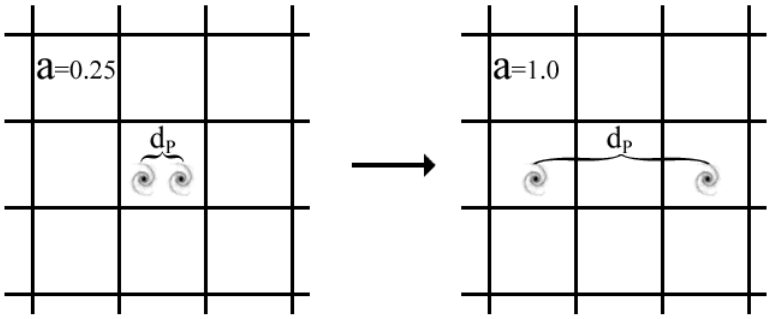

Figure 2.1: The expansion of the universe in real space

As the scale factor increases the distances in the universe grow. Figure 2.2 shows how this looks in the comoving grid. Nothing particular happens as the scale factor increases, and the comoving distance r between two galaxies remains the same as long as they do not have any proper motion relative to each other. To find the proper distance d_P at any time we would then need to multiply by the scale factor a.

$$d_P = a(t)r \tag{2.1}$$

It is common to use subscript zero to indicate values at present. Thus, t_0 means present day, and a_0 is the scale factor today. It is convenient to use $a_0 = 1$. In the past, the scale factor was smaller than one since the universe

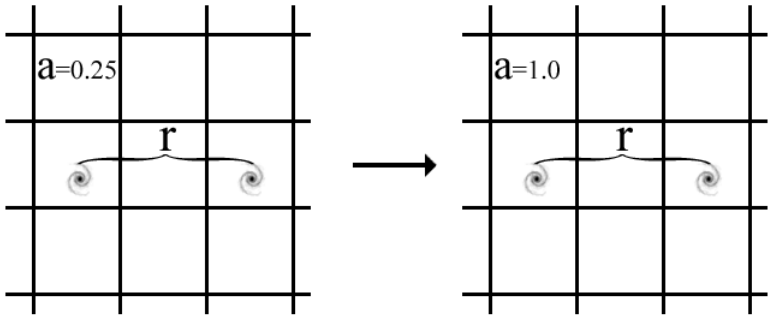

Figure 2.2: The expansion of the universe in comoving coordinates

has been expanding. If we assume continued expansion, the scale factor will be larger than one in the future.

Hubble discovered that the galaxies were moving away from us at a rate proportional to their distance. He did this by measuring their redshift. This is an effect similar to the doppler effect, where the frequency of sound or light is perceived as lowered for the observer when the object emitting it is moving away from the observer. The radial velocity (v_r) at which two points moves away from each other is then given as

$$\begin{aligned} v_r &= \frac{d}{dt} d_P = \dot{a} r \\ &= \frac{\dot{a}}{a} d_P \end{aligned} \tag{2.2}$$

The radial velocity is proportional to a factor $\dot{a}/a$, and we define this rate as the Hubble parameter:

$$H(t) = \frac{\dot{a}}{a} \tag{2.3}$$

and the radial velocity can be written as

$$v_r = H(t) d_P(t) \tag{2.4}$$

The radial velocity at which anyting moves away from us is then the product of it's proper distance and the Hubble parameter. The further away anything is, the faster it moves away from us.

In it's most compact form, the Einstein equations can be written as

$$G_{\mu\nu} = 8\pi G T_{\mu\nu} \tag{2.5}$$

Here $G_{\mu\nu}$ is the Einstein tensor, describing the space-time geometry; $T_{\mu\nu}$ is the energy-momentum tensor, describing the energy content of the universe; and G is Newton's constant.

To solve the Einstein equations we need a description of the geometry of spacetime, in the form of a line-element or a metric. In the case of an expanding universe we use the following line-element, known as the Friedmann-Robertson-Walker line-element:

$$ds^2 = -c^2dt^2 + a^2(t)\left[\frac{dr^2}{1-kr^2} + d\Omega^2\right] \tag{2.6}$$

This basically describes a spacetime geometry where the spatial part is scaled by the scale factor, and it includes the possibility for curvature of space via the parameter k.

Using Einstein's theory of General Relativity and the assumption of a homogenous and isotropic universe, it is then possible to develop the Friedmann-equations:

$$\left(\frac{\dot{a}}{a}\right)^2 + \frac{kc^2}{a^2} = \frac{8\pi G}{3}\rho \tag{2.7}$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3}\left(\rho + \frac{3p}{c^2}\right) \tag{2.8}$$

$$\tag{2.9}$$

and the adiabatic expansion:

$$\dot{\rho} = -3\frac{\dot{a}}{a}\left(\rho + \frac{p}{c^2}\right) \tag{2.10}$$

These equations describe the evolution of the scale factor a with respect to the energy density ρ, the pressure p of the various components in the universe, and the curvature of space, k. These equations are not independent. Any one of the equations can be derived from the other two. Since they contain three different variables we need a way to get rid of one of them. Luckily pressure can be expressed as a simple equation of state, relating the pressure to the energy density as: $p = w\rho c^2$. Here w is a constant and will depend on the exact nature of the particle we are describing.

The Friedmann equations will have very different solutions depending on what kind of particles we put into the model. A model of a universe consisting of radiation, matter and a cosmological constant would have to consider the

energy density of each component, and we would write this as $\rho = \rho_\gamma + \rho_m + \rho_\Lambda$ in the equations. Each of these would have a pressure component as well.

As we have seend the Hubble parameter is defined as $H(t) \equiv \dot{a}/a$, and inserting this into eq. 2.7 for present time we get

$$1 + \frac{kc^2}{a_0^2 H_0^2} = \frac{8\pi G}{3H_0^2}\rho_0$$

Since the term $\frac{3H_0^2}{8\pi G}$ have the unit of density, we define this as the present value of the critical density, written as ρ_{c0}. Further, we define a new measure of densities in units of the critical density as

$$\Omega_0 \equiv \frac{\rho_0}{\rho_{c0}}$$

and the curvature density parameter as

$$\Omega_{k0} = -\frac{kc^2}{a_0^2 H_0^2} \tag{2.11}$$

and this leads to this simplified version of eq. 2.7:

$$\Omega_0 + \Omega_{k0} = 1 \tag{2.12}$$

For a universe with no curvature we have $\Omega_{k0} = 0$, and the energy density has to be equal to the critical density today.

The most well known particles we include in the models are of course radiation and baryonic matter, but recent observations have lead to the inclusion of more exotic elements as well. To explain some observed features of galactic rotation we include dark matter, a kind of matter with the property that it does not interact electromagnetically, and this renders it virtually undetectable. It does however interact gravitationally, and has an impact on large scale structures like galaxies and matter distribution in the universe.

Using standard candles, sources of light of which we know the luminosity and therefore the distance to, and comparing this with their redshift, it appears that in the last few billion years the expansion rate of the universe has accelerated. To explain this we postulate the existence of dark energy. This energy must have the property of negative pressure, and a common way to include this in the Friedmann equations is by describing it as a cosmological constant.

The result of trying out different parameters and components leads to widely different models for the evolution of the universe. It is possible to have a static solution, the one that was preferred by Einstein and the rest of the scientific community before Hubbles discovery. Other models include universes that are expanding exponentially, leading to eternal universes with no beginning and no end; universes with a definite ending like the big rip or a collapse back to a singularity; or the one most cosmologists today belive is the correct one, a universe with a definite beginning in the Big Bang and eternal expansion afterwards.

The reason for having landed on this model is of course by matching models with observations. Using eq. 2.12 it can be expressed as

$$0.3\Omega_b + 0.7\Omega_\Lambda = 1$$

This describes a spatially flat universe where about 70% of the energy content comes from the cosmological constant, and the rest comes from matter, mostly in the form of dark matter. There is some radiation present as well, but the contribution is vanishingly small, with a value $\Omega_{r0} = 8.4 \cdot 10^{-5}$.

It should be evident that we have a lot of freedom in choosing what parameters we want to populate our equations with. We clearly need to be able to find a reliable observational test for which model to prefer. The CMB gives us a very solid way of finding bounds for the many parameters we are working with, and we will look at where the CMB came from next.

2.1.2 Recombination

Combining the Friedmann equations (eqs.2.7 to 2.10) with the Hubble expansion and the measured energy content of the universe, we have come to accept the Big Bang theory as the preferred model. In this theory the universe must have started out in a very dense, very hot state. This state can be described as a cosmic fluid consisting of elementary particles and radiation. At this early time the electrons are not bound to protons, and electromagnetic interactions takes place between various particles all the time. This means that photons can only move a short distance before hitting the very large electromagnetic cross-section of a free electron, and will then be bounced in a different direction.

When the universe expands this cosmic fluid becomes less dense and cool off. At a sufficiently low temperature, when the universe has expanded enough,

the electrons no longer have enough kinetic energy to be able to escape the protons and will instead be bound to them, forming the first atoms. An effect of this is the neutralization of the cosmic fluid. The photons are now free to move and the universe has suddenly become transparent.

This is the event known as *recombination*, and the radiation surviving from this time is what we observe as the CMB. Another name for this event is *last scattering surface*, since this represents a boundary of how far back in time we are able to observe. The last scattering surface is a star-like surface that sent out the CMB radiation, and we are unable to see beyond this boundary.

Using eq. 2.10 we may relate the energy density to the scale factor today. For matter this leads to

$$\rho_m = \rho_{0m} \left(\frac{a_0}{a} \right)^{-3} \tag{2.13}$$

and for radiation we get

$$\rho_\gamma = \rho_{0\gamma} \left(\frac{a_0}{a} \right)^{-4} \tag{2.14}$$

This means that as the scale factor a increases, the energy density of both matter and radiation decreases. But because of the difference in power, the energy density of radiation decreases much faster than for matter. Today we measure the energy density of matter to be dominant, while radiation is vanishingly small. By reversing the process however, it is clear that at some point the two must have been equal, and before this time radiation must have been the dominant factor. Setting eqs. 2.13 and 2.14 equal to each other we find the scale factor at equality as $a_{eq} = \frac{\rho_{0\gamma}}{\rho_{0m}} = \frac{\Omega_{0\gamma}}{\Omega_{0m}}$. In astronomy an unambiguous way to measure time or distance is using redshift. The higher the redshift, the more remote in time and space the event measured is. The redshift z is given as

$$1 + z = \frac{a_0}{a} \tag{2.15}$$

and using this we find that the matter-radiation equality took place at $z \approx 3570$.

The temperature of the CMB has been precisely measured to $T = 2.725 \pm 0.002$K and behaves like almost perfect black body radiation. Because of this the energy density can be calculated as

$$\rho_\gamma = 2 \int \frac{d^3p}{(2\pi)^3} \frac{1}{e^{p/T} - 1} p \tag{2.16}$$

leading to

$$\rho_\gamma = \frac{\pi^2}{15} T^4 \tag{2.17}$$

Since we know from eq. 2.14 that radiation scales as a^{-4}, the temperature must scale as a^{-1}. This makes it possible to calculate the scale factor at recombination. We know that this process takes place at about $T_* = 3000\text{K}$, and are able to find the scale factor a_* at recombination. We have $a_*/a_0 = T_0/T_*$, and using $a_0 = 1$ we get:

$$a_* = \frac{T_0}{T_*} \approx \frac{2.73}{3000} \sim 10^{-3}$$

The visible universe has expanded to a size about 1000 times greater than at recombination, and this event took place at $z \approx 1100$.

2.1.3 Inflation

The speed limit of the universe is given by the speed of light, c. By dividing a distance d_P by c we get the time it would take for light to reach us. Looking at equation 2.4 we see that this is also the inverse of the Hubble parameter. This quantity, $\frac{1}{H}$, is known as the Hubble-radius, and it sets a limit to what distances can be causally connected at present. This radius describes a sphere called the Hubble sphere, and any signal sent from outside this sphere would have to move at a speed greater than c to be able to reach us at present.

Looking at figure 2.3 we see the light from two points on opposite sides of the Hubble sphere, their light just reaching us now. But there is a problem

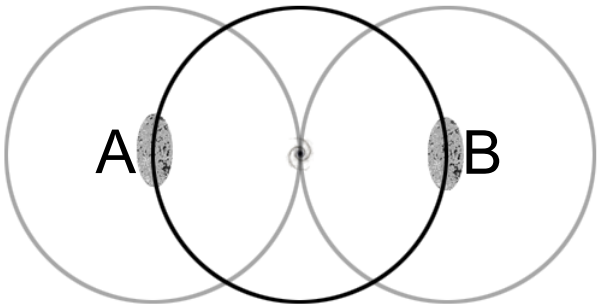

Figure 2.3: Our Hubble sphere at present.

here. The CMB from both directions are remarkably similar, but they are not within each others Hubble radius, and there is no way information from point A could have gotten to point B in the lifetime of the universe. Since the

points can not have been causally connected, there is no good explanation for why the two points are as similar. The temperature measured from two very different points at the last scattering surface should not be as similar unless they somehow were in equilibrium at some point in time.

The basic idea behind the theory of inflation is that shortly after the Big Bang, a very rapid expansion took place. During a period of time much shorter than a second the universe expanded at such a rate that a volume the size of an atom would blow up to astronomical scale. This ensures that points that are currently not causally connected could have been at an early stage. Let's look at what happens to the Hubble radius by rewriting eq. 2.3 as

$$\frac{da}{a} = H dt \tag{2.18}$$

If we assume that inflation took place in a very short period of time we may assume the Hubble radius to be constant, and the solution becomes

$$a(t) = a_e e^{H(t-t_e)} \tag{2.19}$$

where t_e is the time at the end of inflation and a_e is the scale factor at this time. As long as the duration of the inflation period is much smaller than the Hubble radius we can have a period of exponential growth of the scale factor.

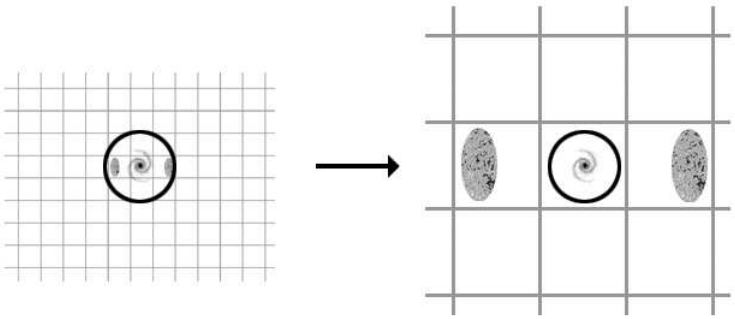

Figure 2.4: Inflation in real space.

The result of the rapid expansion would be that spatial distances blew up, while the Hubble radius remained constant. Regions that previously had time to get into equilibrium, would now be moved outside the Hubble radius and no longer be causally connected, as seen in figure 2.4. The separate points

would continue to evolve according to the same physical laws however, and this explains the smoothness we observe at present when these points reenter our Hubble sphere. In figure 2.5 we see the effect in comoving coordinates.

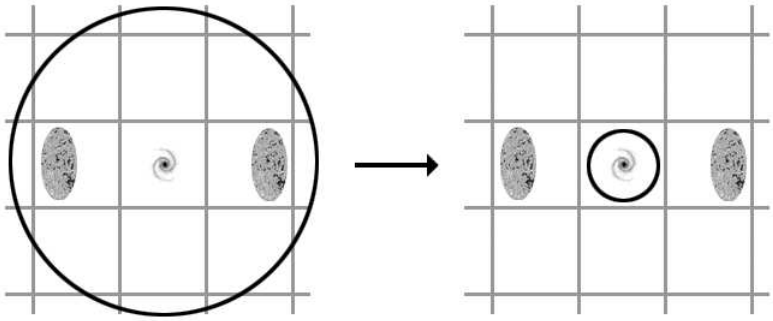

Figure 2.5: Inflation in comoving coordinates.

Here the distances between different points never changes, but inflation has the effect that the Hubble radius is shrinked down drastically. As the Hubble sphere slowly grows again after inflation, these points reenter our Hubble sphere and we can observe the smoothnes that was caused by inflation.

There is an added bonus in these inflationary models. Since the observable universe existed on a sub-atomary scale before inflation, it was also governed by quantum mechanics. In quantum mechanics you cannot have completely determined states, there would exist small fluctuations in position and movement, and therefore in density. During inflation these fluctuations were blown up to cosmological scales. Over time the fluctuations developed into regions of lesser or greater density, and at the time of recombination these variations caused varying gravitational potentials that the radiation forming the CMB had to move out of. This we observe as fluctuations in the temperature of the CMB today. The small over- and underdensities that existed at recombination has since developed into galaxies and the large scale structures we see in the universe today.

With a single mechanism we have explained both the smoothness of the CMB and the small fluctuations we observe. Furthermore, using known physical laws, we are able to determine how these fluctuations develop, and this makes it possible to determine how the cosmological parameters influence the observed CMB.

A problem with this inflationary model is that we know no real mechanism that could have caused it. It is however the most successful explanation we have, and can be used to establish initial conditions when we want to describe the developent of the fluctuations in the CMB.

2.1.4 Anisotropy

Inflation produces inhomogenities in the cosmic fluid of the universe, which eventually lead to the formation of galaxies. Today we are able to observe these inhomogenities as anisotropies in the CMB, small fluctuations in temperature depending on which direction we look. We need to be able to describe how the fluctuations develop into the observed CMB, and a way to do this is described by Dodelson in his book Modern Cosmology [3]. I will make a short overview of the derivation here.

To begin with, we modify the line-element given by eq. 2.6. We assume a flat universe and include small perturbations to the Newtonian potential, Ψ, and the spatial curvature, Φ. These perturbations are set up in the inflationary epoch. In cartesian coordinates we get the metric tensor:

$$\begin{aligned} g_{00}(\vec{x},t) &= -1-2\Psi(\vec{x},t) \\ g_{0i}(\vec{x},t) &= 0 \\ g_{ij}(\vec{x},t) &= a^2\delta_{ij}(1+2\Phi(\vec{x},t)) \end{aligned} \tag{2.20}$$

The metric tensor is an alternative description of the geometry of spacetime. This is the Friedmann-Robertson-Walker metric for a universe with no spatial curvature, describing a homogenous and isotropic universe governed by the scale factor a, in addition to the small perturbations.

Einstein's field equations tells us how the spacetime geometry determines the movement of it's content. At the same time, all energy content influences how the spacetime geometry changes. To be able to use first-order perturbation theory and determine the effect of the perturbations, we need to know a great deal about all the particles that constitute the universe. This means knowing how radiation, electrons, protons, neutrinos and dark matter behaves. To do this we have to develop the Boltzmann equation for each of these components. The Boltzmann equation is used to describe the number density and momentum for a particle given the rate of producing and eliminating that specie of particle.

In the case of photons we expect the Bose-Einstein distribution, with a small perturbation.

$$f(\vec{x}, p, \hat{p}, t) = \left[\exp\left\{\frac{p}{T(t)[1+\Theta(\vec{x},\hat{p},t)]}\right\}\right]^{-1} \tag{2.21}$$

Here Θ represent the temperature fluctuations, and they are dependent on direction of momentum as well as position and time.

For studies of the anisotropies in the CMB it is convenient to expand the fluctuations Θ into multipoles Θ_ℓ. A multipole is defined by the following transformation:

$$\Theta_\ell \equiv \frac{1}{(-i)^l}\int_{-1}^{1}\frac{d\mu}{2}\mathcal{P}_\ell(\mu)\Theta(\mu) \tag{2.22}$$

This is a Legendre transform, where $\mathcal{P}$ is the Legendre polynomial, and μ is the angle between the direction of the photon and the direction in which the temperature is changing. Θ_0 is known as the monopole, Θ_1 is the dipole, Θ_2 the quadrupole and so on.

I will not get into the details of the derivations, they may be found in Dodelson's book [3]. In short, we have to determine the Boltzmann equation for each particle, and use the perturbed metric on the Einstein equations to determine the relation between the perturbed potentials and these components. It is then shown how we can develop a set of differential equations connecting all of the variables involved. Stealing from Dodelson we get the following:

$$\dot{\Theta} + ik\mu\Theta = -\dot{\Phi} - ik\mu\Psi - \dot{\tau}\left[\Theta_0 - \Theta + \mu v_b - \frac{1}{2}\mathcal{P}_2(\mu)\Pi\right] \tag{2.23}$$

$$\Pi = \Theta_2 + \Theta_{P2} + \Theta_{P0} \tag{2.24}$$

$$\dot{\Theta}_P + ik\mu\Theta_P = -\dot{\tau}\left[-\Theta_P + \frac{1}{2}(1-\mathcal{P}_2(\mu))\Pi\right] \tag{2.25}$$

$$\dot{\delta} + ikv = -3\dot{\Phi} \tag{2.26}$$

$$\dot{v} + \frac{\dot{a}}{a}v = -ik\Psi \tag{2.27}$$

$$\dot{\delta}_b + ikv_b = -3\dot{\Psi} \tag{2.28}$$

$$\dot{v}_b + \frac{\dot{a}}{a}v_b = -ik\Psi + \frac{\dot{\tau}}{R}[v_b + 3i\Theta_1] \tag{2.29}$$

$$\dot{\mathcal{N}} + ik\mu\mathcal{N} = -\dot{\Phi} - ik\mu\Psi \tag{2.30}$$

Using the perturbed metric on the Einstein equations 2.5 leads to

$$k^2\Phi + 3\frac{\dot{a}}{a}\left(\dot{\Phi} - \Psi\frac{\dot{a}}{a}\right) = 4\pi G a^2[\rho_m\delta_m + 4\rho_r\Theta_{r,0}] \tag{2.31}$$

$$k^2(\Phi + \Psi) = -32\pi G a^2\rho_r\Theta_{r,2} \tag{2.32}$$

In these equations all time derivatives use conformal time, defined as $\eta \equiv \int_0^t \frac{cdt'}{a(t')}$. In addition, they have been Fourier transformed. The signal is split into different Fourier modes, each with a wavenumber k.

There are a lot of variables involved here. We are dealing with density perturbations and velocities for both dark matter and baryonic matter (δ and δ_b), various multipoles in the temperature perturbations, including polarization (Θ_0, Θ_1, Θ_2, Θ_{P0} and Θ_{P2}), and perturbations in the neutrino radiation ($\mathcal{N}$). I will make no attempt to deal with everything here. The point is that through these equations we are able to relate many cosmological parameters to each other, to the perturbations Ψ and Φ in the metric, and most importantly for our purpose, to the perturbations in photon distribution, via the parameter Θ.

Using assumptions about the earliest times after the Big Bang, it is possible to relate all the variables to the potential Φ. We may then use assumptions about the earliest times and ideas from inflation to establish initial conditions for Φ. As an example of this we can take a closer look at eq. 2.23.

We consider times so early that for any k-mode we are interested in, $k\eta \ll 1$. The first term in eq. 2.23 is of order Θ/η, while the second term is of order $k\Theta$. The first term must then be much larger than the second , and we can use the same argument for all terms multiplied by k. At this early time we also assume a uniform sky, meaning we only get contributions from the monopole. We also assume the optical depth to vary slowly, leading to $\dot{\tau} \ll 1$. The equation can then be written as

$$\dot{\Theta}_0 + \dot{\Phi} = 0$$

In similar ways we relate all the cosmological parameter to perturbations in the gravitational potential. Using quantum mechanical assumptions with the theory of inflation we may then set up initial conditions for Φ and the equations become solvable.

Before recombination the photons were tightly coupled with the free electrons and protons, and this makes it possible to approximate their interactions as

a fluid. In this approximation the only non-negligible moments of Θ are the monopole Θ_0 and dipole Θ_1. This means we only need to consider the anisotropies for these two moments before recombination.

In the end, we are able to calculate the perturbations in any multipole as

$$\begin{aligned}\Theta_\ell(k,\eta_0) \simeq &[\Theta_0(k,\eta_*) + \Psi(k,\eta_*)] j_\ell[k(\eta_0-\eta)] \\ &+ 3\Theta_1(k,\eta_*)\left(j_{\ell-1}[k(\eta_0-\eta_*)] - \frac{(\ell+1)j_l[k(\eta_0-\eta_*)]}{k(\eta_0-\eta_*)}\right) \\ &+ \int_0^{\eta_0} d\eta e^{-\tau}\left[\dot{\Psi}(k,\eta) - \dot{\Phi}(k,\eta)\right] j_\ell[k(\eta_0-\eta)] \end{aligned} \tag{2.33}$$

The term j_ℓ is the Bessel function. Here we note that any multipole today is essentially given by the monopole and dipole at recombination.

2.1.5 Power spectrum

The anisotropic solution gives us the answer to what we expect the temperature fluctuations to be in terms of multipoles Θ_ℓ in the Legendre transformed functions. What we need to do is relate this to actual measurements on the sky.

We observe the temperature fluctuations in the CMB today from the center of a sphere, where the wall of the sphere forms the last scattering surface. The temperature field can be written as

$$T(t_0,\theta,\phi) = T(t_0)[1 + \delta T(t_0,\theta,\phi)] \tag{2.34}$$

To be able to handle this we use spherical harmonics to transform the variations. The spherical harmonics will be described in section 2.2.1, but I will use some of its properties here. When we transform the temperature fluctuations as we see them today to spherical harmonic space we get

$$\delta T(\theta,\phi) = \sum_{\ell=1}^{\infty}\sum_{m=-l}^{l} a_{\ell m} Y_{\ell m}(\theta,\phi) \tag{2.35}$$

Here the coefficients $a_{\ell m}$ describes the amplitudes of the variations in an orthogonal basis, described by $Y_{\ell m}$. By using the orthogonality property of the spherical harmonics,

$$\int d\Omega Y_{\ell m}(\theta,\phi) Y^*_{\ell' m'}(\theta,\phi) = \delta_{\ell\ell'}\delta_{mm'} \tag{2.36}$$

we are able able to invert the expansion. This is done by multiplying both sides by $Y^*_{\ell m}(\theta, \phi)$ and integrating, and leads to

$$a_{\ell m} = \int d\Omega Y^*_{\ell m}(\theta, \phi)\delta T(\theta, \phi) \tag{2.37}$$

Now we are able to express the variations in terms of spherical harmonic coefficients, $a_{\ell m}$. These coefficients characterize the fluctuations in the CMB, and if the underlying temperature fluctuations are Gaussian, the $a_{\ell m}$ will also have a Gaussian distribution. The mean of the coefficients will then be zero, but they will have a nonzero variance. In spherical harmoinc space this gives a very simple relation:

$$\langle a_{\ell m} a^*_{\ell' m'} \rangle = \delta_{\ell \ell'} \delta_{m m'} C_\ell \tag{2.38}$$

The variance of $a_{\ell m}$ is given by C_ℓ, this is known as the power spectrum.

By using the spherical harmonic transform on the analytical expression for Θ_ℓ given in eq. 2.33 we will again get the power spectrum, and in this way we can relate the theoretical power spectrum to the one we get from observations.

A typical power spectrum is shown in figure 1.2, and we see some characteristic features. The cosmological parameter influence the exact shape of the power spectrum. Baryon density has an effect on the heights on the different peaks in the figure, while the cosmological constant influences at what multipoles we find the various peaks. This makes studies of the power spectrum an excellent tool for estimating and finding bounds for the cosmological parameters we are interested in.

We saw the definition of the power spectrum in eq. 2.38, and when using sums it can be recovered as:

$$\begin{aligned} C_l &= \langle a_{\ell m} a^*_{\ell m} \rangle \\ &= \frac{1}{n} \sum_m^n a_{\ell m} a^*_{\ell m} \\ &= \sum_{m=-\ell}^{\ell} \frac{a_{\ell m} a^*_{\ell m}}{2\ell + 1} \end{aligned} \tag{2.39}$$

The last line holds since we sum over all m from $m = -\ell$ to $m = \ell$, which is a total of $2\ell + 1$. This expression will be useful later when we want to recover the actual power spectrum in a random realization of $a_{\ell m}$'s.

The power spectrum is usually scaled as $C_\ell \ell(\ell+1)$. By using eq. 2.33 on large scales (aka small multipoles) we are able to make some simplifications. These scales were not affected by any causal physics, and only the monopole contributes to the anisotropy for these scales. When we find the analytical power spectrum using these simplifications it can be shown that the power spectrum scales as

$$C_\ell \sim \frac{1}{\ell(\ell+1)} \tag{2.40}$$

This is known as the Sachs-Wolffe effect.

2.2 Making maps of the CMB

We now turn to the observations of the CMB. Simply put, the CMB is a picture of the variations in the density of the universe at the moment of recombination. What we want to do is to make quantitative statements about this signal, and that means finding the power spectrum. To be able to do this, we need some convenient tools, and we need to know something about observational limits. The following is based on [6] and [9].

2.2.1 Spherical harmonic transform

We have already seen the spherical harmonic transform in action. Its usefulness comes from the fact that the signals we are dealing with are projected on a spherical surface. We are after all observing the CMB in every direction on the sky, and we need a mathematical tool to decompose this signal to a convenient form. The solution is a generalization of the Fourier transform able to handle a two-dimensional signal on a spherical surface.

When we do Fourier transforms, we get in return coefficients characterizing the original function in a basis consisting of sines and cosines. Similarly, in spherical harmonics, we get a set of complex coefficients, $a_{\ell m}$'s on a set of basis vectors $Y_{\ell m}(\theta, \phi)$. Here θ and ϕ denotes the direction on the sky in spherical coordinates. In a sense, you then get the original image decomposed into stationary waves on the sphere. The ℓ describes the size of of the wave, while the m describes the angle.

The $Y_{\ell m}(\theta, \phi)$ is an orthonormal set, and the following applies:

$$\int d\Omega Y_{\ell m}(\theta, \phi) Y^*_{\ell' m'}(\theta, \phi) = \delta_{\ell \ell'} \delta_{m m'} \tag{2.41}$$

To get the spherical harmonic coefficients in this basis we solve the integral

$$a_{\ell m} = \int \mathrm{d}\Omega f(\theta, \phi) Y_{\ell m}(\theta, \phi) \tag{2.42}$$

To get back the original function, we can do the reverse operation. For our use it is not realistic to use integrals. Since any real observations will be limited in the number of sample points it can make on the sky, what we need is a limited set of $a_{\ell m}$'s. We convert the integrals to sums, and the function can then be recovered as a sum

$$f(\theta, \phi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} a_{\ell m} Y^*_{\ell m}(\theta, \phi) \tag{2.43}$$

A useful identity in spherical harmonics is:

$$\sum_{m=-l}^{\ell} Y^*_{\ell m}(\theta, \phi) Y_{\ell m}(\theta', \phi') = \frac{2\ell + 1}{4\pi} P_\ell(\cos \Delta\theta) \tag{2.44}$$

where $\Delta\theta$ is the angle between (θ, ϕ) and (θ', ϕ'), and P_ℓ is the Legendre polynomial of degree ℓ.

2.2.2 Correlation

The signal we observe in the CBM is a random signal, but the physics behind it ensures some distinct features, such as a characteristic power spectrum. For any given multipole we expect the fluctuations to evolve in a given way from recombination up to present. This means the way the variation looks at one place in the sky will be similar to the way they look in a different direction. We could even have predicted this by examining figure 1.1. Looking at the image it is easy to notice large scale sturctures in the signal. Some areas are overall warmer and others are overall colder than the average.

In theory the amount of information we could get from the sky would be infinite. Real observations are limited however, both by optical limitations

and the storage medium. The area we wish to observe has to be divided into small units, known as pixels. This gives us a finite amount of data to handle, and a basis for our analysis. In any given pixel we measure a value, and from all the pixels together we then get a data-vector which can be described as:

$$\mathbf{m} = \delta\mathbf{t} + \mathbf{n} + \mathbf{g} + \mathbf{f} \tag{2.45}$$

Here $\delta\mathbf{t}$ is the signal from CMB, $\mathbf{n}$ is noise from the dectector, $\mathbf{g}$ is systematic error, and $\mathbf{f}$ is foreground radiation from our galaxy and other sources that lies between us and the last scattering surface. The area where this foregroud radiation contaminates the most is shown as grey areas in figure 1.1. The last two factors in the data will be ignored for the rest of this thesis. We assume we are able to remove all foreground, and noise in one pixel will not influence noise in another, thereby getting rid of systematic error.

We now have a pixelized map of the CMB, and it is finally time to do some statistical analysis. To measure what effect variations on different scales have on each pixel, we have to find the covariance matrix. This is defined as

$$C_{ij} = \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle \tag{2.46}$$

where x_i is the measured quantity we are looking at in pixel number i. For the CMB we are measuring the variations in the temperature, and when we remove the mean temperature and the effect from our galaxy's proper motion, the expression simplifies to $C_{ij} = \langle x_i x_j \rangle$.

The covariance matrix describes the connection any given pixel has to another. To get a clearer picture of what the covariance actually describes, it is convenient to normalize it. This produces a new matrix which is called the correlation matrix. The correlation matrix is given as $C_{ij} = \sigma_{ij}/\sigma_i\sigma_j$, where $\sigma_{ij} = \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle$ is the standard deviation.

The correlation is a number with absolute value between 0 and 1, and describes to what degree the two measures depended on each other. When examining multiple sets of data, a high correlation would mean a high probability of the two measure points to have connected values. A correlation of 1 means that one measurement would be completely dependent on the other, while a correlation of 0 means that the two measured values had no connection with each other at all. A pixel compared with itself will always have a correlation of 1, but the correlation with other pixels depends on the underlying principles forming the image.

As we can see in eq. 2.45 our signal consists of various components, and this might lead to complications. Even with a mixed signal consisting only of noise

and the true signal, we would mathematically still expect cross-correlations between the different components.

$$\begin{aligned}\left\langle \mathbf{m}\mathbf{m}^T \right\rangle &= \left\langle (\delta\mathbf{t}+\mathbf{n})(\delta\mathbf{t}+\mathbf{n})^T \right\rangle \\ &= \left\langle \delta\mathbf{t}\delta\mathbf{t}^T + \mathbf{n}\mathbf{n}^T + \delta\mathbf{t}\mathbf{n}^T + \mathbf{n}\delta\mathbf{t}^T \right\rangle\end{aligned}$$

This is simplified by the fact that the correlation is a measure of the real connection between different parts of the signal. It would not make sense to expect any correlations between the CMB sent out over 13 billion years ago, and the noise in our telescope. Therefore we may safely assume that the full covariance matrix consists of a sum of completely separate covariance matrixes, one for each component.

For the CMB, when we ignore foreground and systematic errors, we then have

$$\mathbf{C} = \mathbf{S} + \mathbf{N} \tag{2.47}$$

where $\mathbf{S} = \left\langle \delta\mathbf{t}\delta\mathbf{t}^T \right\rangle$ and $\mathbf{N} = \left\langle \mathbf{n}\mathbf{n}^T \right\rangle$.

The covariance of the CMB will depend on what configuration space we use to represent the signal. If we use pixel space representation we do not expect any correlation between the noise in different pixels as long as we do not have any systematic errors in our observational equipment. In the pixel basis the noise matrix must then be diagonal, $N_{ij} = \sigma_i^2\delta_{ij}$, where σ_i is the rms noise in pixel i.

If we want to convert our data to spherical harmonic space the noise correlation will become more complex, but here the signal correlations is much simpler. We have already seen the result ineq.2.38 , $S_{\ell m} = \delta_{\ell\ell'}\delta_{mm'}C_\ell$. In other words, the covariance matrix for the signal in spherical harmonic space is diagonal and consists of the elements in the power spectrum. For pixel space the full covariance matrix will consist of these elements:

$$\mathbf{C} = \begin{pmatrix} S_{11} & \cdots & S_{1j} \\ \vdots & \ddots & \\ S_{i1} & & S_{ij} \end{pmatrix} + \delta_{ij}\sigma_i^2$$

while in the spherical harmonic space it will look like:

$$\mathbf{C} = \delta_{\ell\ell'}\delta_{mm'}C_\ell + \begin{pmatrix} N_{(10)(10)} & \cdots & N_{(10)(\ell m)} \\ \vdots & \ddots & \\ N_{(\ell m)(10)} & & N_{(\ell m)(\ell m)} \end{pmatrix}$$

The covariance matrices are constructed from a large sampling values, and this poses a problem. All we have is a single observation, the CMB of our own universe. This is just a single sample of $\delta\mathbf{t}$ drawn from a Gaussian probability distribution during inflation, and we simply do not have enough data to construct a covariance matrix. This does after all require an average, which we cannot find with only one sample. There is a way to handle this statistically however.

For the Gaussian probability distribution we know the following:

$$P(x) = \frac{e^{-\frac{(x-a)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \tag{2.48}$$

Here x is our variable, a is the mean value and σ is the standard deviation, and it gives us the probability P for getting the value x.

If we wanted to find the value of x with the highest probability, we could try several things. We could test out different values and see what happened, or we could take the derivative and set it equal to zero and solve for x. But in this simple equation the answer is easy to see. The function must be at it's maximum when $x = a$.

For a case with multiple variables, like the CMB, the multidimensional distribution of N elements is generalized to

$$P(\mathbf{m}) = \frac{e^{-\frac{1}{2}(\mathbf{m}-\mathbf{a})^T\mathbf{C}^{-1}(\mathbf{m}-\mathbf{a})}}{\sqrt{(2\pi)^N \det\mathbf{C}}} \tag{2.49}$$

Here the vector $\mathbf{m}$ is the data we are testing for, the vector $\mathbf{a}$ is the average for each element, and $\mathbf{C}$ is the covariance matrix. What we now have is the probability of getting a particular vector $\mathbf{m}$ if we have a given covariance matrix $\mathbf{C}$.

In the case of CMB we may ignore the vector $\mathbf{a}$, since we have an average of zero, and the equation can be written as

$$\mathcal{L}(C_\ell|\mathbf{m}) = \frac{\exp\left(-\frac{1}{2}\mathbf{m}^T\mathbf{C}^{-1}\mathbf{m}\right)}{(2\pi)^{N/2}\left(\det\mathbf{C}\right)^{1/2}} \tag{2.50}$$

This function is known as the likelihood function, and gives the probability that a given experiment would give the data it did ($\mathbf{m}$) given a chosen theory (C_ℓ). The theory in our case is in the form of the power spectrum, and is hidden inside the covariance matrix.

As in the one-dimensional case we would like to maximize our probability, in other words finding the power spectrum most likely to produce the set of data we have. And now we are closing in on one of the major difficulties of CMB analysis.

The amount of information we can extract from an image of the CMB is related to the pixel-resolution we are working with. The covariance matrix is also related to the pixel-resolution; a data-set of n pixels would give a covariance matrix consisting of n^2 elements. In addition we note that in the likelihood function we need the inverse of the covariance matrix as well. To construct this takes an operation $\mathcal{O}(n^3)$ [10], so depending on how much information we want to extract, the computational demands increase rapidly.

Beginning in the next chapter, we will look at a specific method for maximizing the likelihood function based on the Newton-Rhapson approximation, and try to simplify the the complexity of the covariance matrix by using needlets.

2.2.3 The HEALPix-package

An especially useful tool we will make use of is the HEALPix-package [5]. This is a set of tools that consists of programs, libraries and extensions to IDL which enables us to quickly and efficiently transform back and forth between pixel space and spherical harmonic space. In addition it contains a specific pixelization scheme, ideally suited for dividing the spherical sky into discreete units. The parameter used to describe the resolution of any image in this scheme is the notion of N_{side}.

Looking at figure 2.6 we see the lowest possible resolution available in HEALPix. The sphere is divided into 12 pixels that cover the same amount of area. This resolution is called $N_{\text{side}} = 1$, meaning that each side on each pixel is undivided.

To increase the resolution we divide each of the pixels in a specific way. Dividing up each pixel into four equal areas we get the resolution $N_{\text{side}} = 2$. Each side has now been divided by two, hence the name N_{side}. To get the next resolution we divide each of these smaller pixels into four equal areas, getting $N_{\text{side}} = 4$ The choices we have for nside is then $\{1, 2, 4, 8, 16, \cdots\}$, and the number of pixels for a given N_{side} is given as

$$\text{pixels} = 12 \cdot N_{\text{side}}^2 \tag{2.51}$$

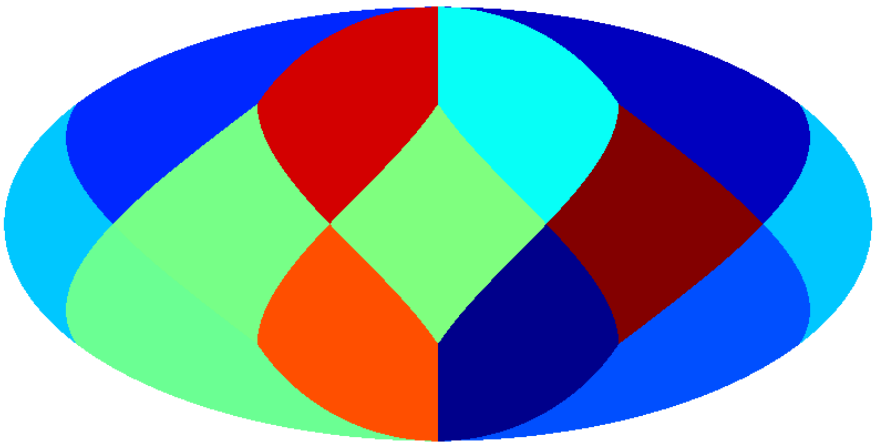

Figure 2.6: Mollweide projection of a map with $N_{\text{side}} = 1$

Figure 2.51 shows the effect of increasing the resolution to $N_{\text{side}} = 2$ and $N_{\text{side}} = 4$.

The HEALPix package gives us access to functions making the transformation back and forth between pixel space and spherical harmonic space fast and efficient. Additionally it gives us the ability to make simulated universes from a given power spectrum. Remember, the power spectrum is describing the variance for the coefficients in spherical harmonic space, and this enables us to make random $a_{\ell m}$'s with a Gaussian distribution. This is perfect for testing out our algorithms later on.

2.2.4 Some observational limitations

Going from a presumably limitless amount of information in the real world, to our limited pixelation of the sphere brings a few complications to the analysis of the power spectrum.

First off, the coefficients in the power spectrum are functions of ℓ. The ℓ's correspond roughly to an angular scale, and for higher ℓ's you get finer details. The relation to angular scale is given approximately as π/ℓ. The

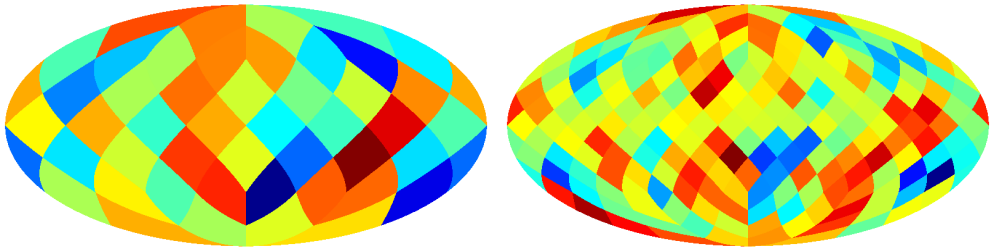

Figure 2.7: The effect of increasing resolution to $N_{\text{side}} = 2$ and $N_{\text{side}} = 4$

quadrupole ($\ell = 2$) will then describe variations on about a quarter of the sphere.

Assuming for a moment that each of the pixels are quadratic, we may find an approximate angular size of each pixel by the following. The total angular area of a sphere is 4π, so dividing by the number of pixels we get the angular area per pixel as

$$\frac{4\pi}{12\mathrm{N_{side}}^2} = \frac{\pi}{3\mathrm{N_{side}}^2}$$

With the assumption that each pixel is quadratic the angular length of a side should then be

$$\Delta\theta \approx \sqrt{\frac{\pi}{3\mathrm{N_{side}}^2}}$$

As noted, the multipole is related to an angular scale as π/ℓ. Combining this with our expression for the angular size of a pixel, we get

$$\begin{aligned}\frac{\pi}{\ell} &> \Delta\theta \\ &> \sqrt{\frac{\pi}{3N_{\text{side}}^2}}\end{aligned}$$

The smallest possible difference we would be able to measure would then be the distance between two pixels, or equivalently the length of the side of one pixel. Any variations occuring on smaller scales would correspond to higher ℓ's, and would be lost inside the pixel. If we now approximate $\pi \approx 3$, we get

$$\ell < 3N_{\text{side}}$$

In practice it is safe to use

$$\ell_{\text{max}} = 2N_{\text{side}} \tag{2.52}$$

Looking for a moment at figure 1.2 we see a typical power spectrum with it's most prominent features. This power spectrum ranges from $\ell = 2$ to about $\ell = 1000$, and to capture this amount of detail we would want to use $N_{\text{side}} = 512$. This in turn leads to about $3 \cdot 10^6$ pixels, and a covariance matrix consisting of $9 \cdot 10^{12}$ elements.

There are further complications arising from pixelisation. When making a measurement we have a finite size pixel representing the data in a specific direction. But we do not get the exact measurement of an infinitely small area in that exact direction. Instead we get an average of all the information covered by that pixel. The effect is an amount of degradation of the information, depending on the size of the pixel. This can be accounted for however, by using a *pixel-window* function w_ℓ. The pixel-window function is a function of ℓ and leads to a gradual weakening of the power spectrum for higher ℓ's. The effect is given as

$$C_\ell(\text{observed}) = C_\ell w_\ell^2$$

When looking at a real signal we would like to remove this effect. To handle it we reverse the operation, dividing the observed power spectrum by w_ℓ. This effect will be dependent on the resolution of our image, and in the case of the HEALpix pixelization scheme the effect has been precalculated for different N_{side}.

There is one more effect we need to account for. The optical nature of observations are such that we don't get measurements from just the exact direction we are looking at. We will always get an amount of stray light from nearby points as well. Since this will somewhat obscure the data we really want from the measured direction, it has to be taken into consideration as well. This effect is known as the beam, and to counter it we use the *beam function.*

The exact effect of the beam is determined by the design of the telescope, but in most cases it can be modeled by a Gaussian function. In spherical harmonic space a Gaussian beam function can be written as

$$b_\ell = e^{-\ell(\ell+1)\sigma^2}$$

where σ is a measure of the angular width of the beam given in radians. The angular width is known for the instrument we are using. It is usually given as Full Width at Half Maximum (FWHM) in degrees. The relation between σ and FWHM is then given as

$$\sigma = FWHM/\sqrt{8 \ln 2}$$

In the case of WMAP we know FWHM=7 arc minutes, and have to convert this to degrees.

Again the effect on the power spectrum is

$$C_\ell(\text{observed}) = C_\ell b_\ell^2$$

When we need to account for both effects we finally have

$$C_\ell(\text{observed}) = C_\ell b_\ell^2 w_\ell^2 \tag{2.53}$$

To get the real power spectrum from observed data we of course reverse the process, getting $C_\ell = C_\ell(observed)/b_\ell^2 w_\ell^2$.

Later on when we will do simulations of the power spectrum, it is necessary to include these effects. We want the algorithm to work on real data, and it would be nice to know it works with the pixel-window and beam effects included as well. In addition we will have to cut off our signal somewhere, and if we do this abruptly it may cause unwanted effects in the analysis later. The beam and pixel-window effects serve as a way to naturally taper off the real signal and removing any abrupt cut-offs we might otherwise have had.

Chapter 3

Method

The power spectrum is a very useful tool for estimating the cosmological parameters. We have already seen how we are able to start from these parameters, using the theory to make models and give predictions about what the power spectrum should look like.

What we want to do now is doing it the other way round. We have an image of the CMB, and we want to estimate which set of parameters C_ℓ is the best fit to the observations we have. To be able to do this we need to establish an algorithm we can use. Starting from the likelihood function we will make an iterative method that is able to make precise estimates of the power spectrum.

In addition we will make use of a special kind of transformation, the needlet transformation. The hope is that needlet transformations will simplify some of the computationally heavy problems that arise in our algorithm.

3.1 Oh-Spergel-Hinshaw algorithm

This algorithm was initially described by Oh, Spergel and Hinshaw [9], and in the following I will make a detailed outline of their derivation of the algorithm.

3.1.1 Developing the algorithm

We have already seen the likelihood function,

$$\mathcal{L}(C_\ell|\mathbf{m}) = \frac{\exp\left(-\frac{1}{2}\mathbf{m}^T\mathbf{C}^{-1}\mathbf{m}\right)}{(2\pi)^{N/2}\left(\det\mathbf{C}\right)^{1/2}}$$

and it is common to rewrite it as

$$f = -2\ln\mathcal{L}$$

This makes the entire function a lot easier to handle, and if we should ever need the real value it is easy to exponentiate our results. In this treatment however, we don't ever need to recover the likelihood. What we are after is simply trying to find what parameters C_ℓ gives a maximum, and all we are interested in is finding when we hit this maximum, not the exact value of $\mathcal{L}$.

To find this extremal point, we go about the usual way, derivating the function and finding when it is equal to zero. In the case of the likelihood function we may safely assume we find a maximum by this method, since the function itself by definition will have to go to zero at very unlikely configurations, and the maximum likelihood will have to be finite. We are not guaranteed to find the global maximum, but we will assume that our underlying model does not have any local maxima that will throw us off.

The parameters of interest to us are the C_ℓ's, so to determine when the likelihood function is maximized with respect to these parameters, we find the derivative of the function and sets it equal to zero.

$$\begin{aligned}\frac{\partial f}{\partial C_\ell} &= \frac{\partial}{\partial C_\ell}\left(-2\ln\left(\exp\left(-\frac{1}{2}\mathbf{m}^T\mathbf{C}^{-1}\mathbf{m}\right)\right) + 2\ln\left(\det\mathbf{C}\right)^{1/2}\right) \\ &= \mathbf{m}^T\frac{\partial\,\mathbf{C}^{-1}}{\partial C_\ell}\mathbf{m} + \frac{\partial}{\partial C_\ell}\ln\left(\det\mathbf{C}\right) \qquad (3.1)\end{aligned}$$

Here we note that the only term dependent on C_ℓ is the matrix $\mathbf{C}$ and it's inverse, $\mathbf{C}^{-1}$. The real work lies in deciding how this matrix have to be constructed to maximize the likelihood.

To find the derivative of the first part of this equation we need to be able to derivate the inverse of the matrix $\mathbf{C}$. We know from elementary matrix operation that we get the identity matrix as

$$\mathbf{C}^{-1}\,\mathbf{C} = \mathbf{I} \qquad (3.2)$$

Since this is a constant, it follows that $\frac{\partial}{\partial c_l}\mathbf{I} = 0$. Let's se what happens when we do the derivative of $\mathbf{C}^{-1}\,\mathbf{C}$. Using the product rule for the derivative we get the following:

$$\begin{aligned}
\frac{\partial}{\partial C_\ell}\left(\mathbf{C}^{-1}\,\mathbf{C}\right) &= 0 \\
\frac{\partial\,\mathbf{C}^{-1}}{\partial C_\ell}\mathbf{C} &= -\mathbf{C}^{-1}\frac{\partial\,\mathbf{C}}{\partial C_\ell} \\
\frac{\partial\,\mathbf{C}^{-1}}{\partial C_\ell}\mathbf{C}\,\mathbf{C}^{-1} &= -\mathbf{C}^{-1}\frac{\partial\,\mathbf{C}}{\partial C_\ell}\mathbf{C}^{-1} \\
\frac{\partial\,\mathbf{C}^{-1}}{\partial C_\ell} &= -\mathbf{C}^{-1}\mathbf{P}^{l}\mathbf{C}^{-1}
\end{aligned} \tag{3.3}$$

In the third line we have multiplied both sides by $\mathbf{C}^{-1}$, and in the last line we have defined $\mathbf{P}^l \equiv \frac{\partial\,\mathbf{C}}{\partial C_\ell}$. This result is very helpful in that we no longer need to find the derivative of the inverse of the matrix, it has instead become a matrix product.

To solve the second half of equation 3.1 we use Jacobi's formula[12], stating:

$$\frac{d}{dx}\left[\det\left(\mathbf{A}\right)\right] = \mathrm{Tr}\left(\mathrm{adj}(\mathbf{A})\frac{d\,\mathbf{A}}{dx}\right) \tag{3.4}$$

Again using the product rule, we find:

$$\frac{\partial}{\partial C_\ell}\left[\ln\left(\det\mathbf{C}\right)\right] = \frac{1}{\det\mathbf{C}}\,\mathrm{Tr}\left(\mathrm{adj}(\mathbf{C})\mathbf{P}^{\ell}\right) \tag{3.5}$$

Since the determinant is a scalar, we may move it inside the trace. In addition a basic theorem from linear algebra states the identity $\mathbf{C}^{-1} = \frac{1}{\det\mathbf{C}}\,\mathrm{adj}(\mathbf{C})$, and this gives us

$$\frac{\partial}{\partial C_\ell}\left[\ln\left(\det\mathbf{C}\right)\right] = \mathrm{Tr}\left(\mathbf{C}^{-1}\,\mathbf{P}^{\ell}\right) \tag{3.6}$$

This gets rid of the complicated task of calculating the determinant of $\mathbf{C}$, at the cost of finding the trace of a matrix, a much simpler operation.

The full differentiated likelihood function now reads

$$\frac{\partial\,f}{\partial C_\ell} = -\mathbf{m}^T\mathbf{C}^{-1}\mathbf{P}^{\ell}\mathbf{C}^{-1}\mathbf{m} + \mathrm{Tr}\left(\mathbf{C}^{-1}\,\mathbf{P}^{\ell}\right) \tag{3.7}$$

We will soon see that it is useful to have the second derivative of our likelihood as well. More specifically we want to calculate $\frac{1}{2}\frac{\partial^2 f}{\partial C_\ell \partial C_{\ell'}}$. We get

$$\begin{aligned}\frac{1}{2}\frac{\partial^2 f}{\partial C_\ell \partial C_{\ell'}} =& \frac{1}{2}\frac{\partial}{\partial C_\ell}\left(-\mathbf{m^T C^{-1} P^{\ell'} C^{-1} m} + \mathrm{Tr}(\mathbf{C^{-1}P^{\ell'}})\right) \\ =& \frac{1}{2}\Bigg(- \mathbf{m^T}\frac{\partial \mathbf{C^{-1}}}{\partial C_\ell}\mathbf{P^{\ell'}C^{-1}m} - \mathbf{m^T C^{-1} P^{\ell'}}\frac{\partial \mathbf{C^{-1}}}{\partial C_\ell}\mathbf{m} \\ & + \mathrm{Tr}\left(\frac{\partial \mathbf{C^{-1}}}{\partial C_\ell}\mathbf{P^{\ell'}}\right)\Bigg)\end{aligned}$$

We have already seen the solution to $\frac{\partial \mathbf{C^{-1}}}{\partial C_\ell}$ from equation 3.3. It is now simply a matter of substitution every time it occurs in our equations. Continuing the calculations, we get:

$$\begin{aligned}\frac{1}{2}\frac{\partial^2 f}{\partial C_\ell \partial C_{\ell'}} =& \frac{1}{2}\Bigg(- \mathbf{m^T}(-\mathbf{C^{-1}P^\ell C^{-1}})\mathbf{P^{\ell'}C^{-1}m} \\ & - \mathbf{m^T C^{-1} P^{\ell'}}(-\mathbf{C}-\mathbf{1}\mathbf{P^\ell C^{-1}})\mathbf{m} \\ & + \mathrm{Tr}((-\mathbf{C^{-1}P^\ell C^{-1}})\mathbf{P^{\ell'}})\Bigg) \\ =& \frac{1}{2}\Bigg(\mathbf{m^T C^{-1} P^\ell C^{-1} P^{\ell'} C^{-1} m} + \mathbf{m^T C^{-1} P^{\ell'} C^{-1} P^\ell C^{-1} m} \\ & - \mathrm{Tr}(\mathbf{C^{-1}P^\ell C^{-1} P^{\ell'}})\Bigg)\end{aligned}$$

We expect our likelihood function to be well behaved, in such a way that $\frac{\partial^2 f}{\partial C_\ell \partial C_{\ell'}} = \frac{\partial^2 f}{\partial C_{\ell'} \partial C_\ell}$. Since this is what determines the index of $\mathbf{P}$, the order of ℓ and ℓ' does not matter. We then just change the order in one of the two first terms and may add the two together. This finally gives

$$\frac{1}{2}\frac{\partial^2 f}{\partial C_\ell \partial C_{\ell'}} = \mathbf{m^T C^{-1} P^\ell C^{-1} P^{\ell'} C^{-1} m} - \frac{1}{2}\mathrm{Tr}(\mathbf{C^{-1}P^\ell C^{-1} P^{\ell'}}) \tag{3.8}$$

It is also useful to know the expectation value of equation 3.8. Using the expression we just calculated we get:

$$\begin{aligned}\left\langle \frac{1}{2}\frac{\partial^2 f}{\partial C_\ell \partial C_{\ell'}} \right\rangle =& \left\langle \mathbf{m^T C^{-1} P^\ell C^{-1} P^{\ell'} C^{-1} m} - \frac{1}{2}\mathrm{Tr}(\mathbf{C^{-1}P^\ell C^{-1} P^{\ell'}}) \right\rangle \\ =& \left\langle \mathbf{m^T C^{-1} P^\ell C^{-1} P^{\ell'} C^{-1} m} \right\rangle \\ & - \frac{1}{2}\mathrm{Tr}(\mathbf{C^{-1}P^\ell C^{-1} P^{\ell'}})\end{aligned} \tag{3.9}$$

Since the only varying quantity in the expression is the vector $\mathbf{m}$, we can move the brackets indicating the mean to only include the term involving $\mathbf{m}$. To be able to see what happens inside the brackets, it is useful to look closer at the first term in component form.

$$\left\langle m_i^T C_{ij}^{-1} P_{jk}^{\ell} C_{km}^{-1} P_{mn}^{\ell'} C_{no}^{-1} m_o \right\rangle = C_{ij}^{-1} P_{jk}^{\ell} C_{km}^{-1} P_{mn}^{\ell'} C_{no}^{-1} \left\langle m_o m_i^T \right\rangle$$

The definition of the covariance matrix is $\mathbf{C} = \left\langle \mathbf{m}\mathbf{m}^{\mathbf{T}} \right\rangle$, so on component form this means we get $\left\langle m_o m_i^T \right\rangle = C_{oi}$. Using the identity matrix on component form, $C_{no}^{-1} C_{oi} = \delta_{ni}$, we get

$$\begin{aligned} C_{ij}^{-1} P_{jk}^{\ell} C_{km}^{-1} P_{mn}^{\ell'} C_{no}^{-1} C_{oi} &= C_{ij}^{-1} P_{jk}^{\ell} C_{km}^{-1} P_{mn}^{\ell'} \delta_{ni} \\ &= C_{ij}^{-1} \underbrace{\underbrace{\underbrace{P_{jk}^{\ell} \underbrace{C_{km}^{-1} \underbrace{P_{mn}^{\ell'} \delta_{ni}}_{\text{mi}}}_{\text{ki}}}_{\text{ji}}}_{\text{ii}}} \end{aligned}$$

What happens here is that the delta function picks out which components survive, and we are in the end left with a quantity that evidently is only the sum of the diagonal elements ii. But this is simply the trace of the matrix, so we can simplify to

$$\left\langle \mathbf{m^T C^{-1} P^{\ell} C^{-1} P^{\ell'} C^{-1} m} \right\rangle = \mathrm{Tr}(\mathbf{C^{-1} P^{\ell} C^{-1} P^{\ell'}})$$

and continuing from equation 3.9 we now get

$$\begin{aligned} \left\langle \frac{1}{2} \frac{\partial^2 f}{\partial C_\ell \partial C_{\ell'}} \right\rangle &= \mathrm{Tr}(\mathbf{C^{-1} P^{\ell} C^{-1} P^{\ell'}}) - \frac{1}{2} \mathrm{Tr}(\mathbf{C^{-1} P^{\ell} C^{-1} P^{\ell'}}) \\ &= \frac{1}{2} \mathrm{Tr}(\mathbf{C^{-1} P^{\ell} C^{-1} P^{\ell'}}) \\ &\equiv \mathbf{F}_{\ell\ell'} \end{aligned} \tag{3.10}$$

This quantity is defined as the Fisher matrix, $\mathbf{F}_{\ell\ell'}$. The Fisher matrix is a very interesting quantity in that it gives some important statistical limits to the equation. The *Cramer-Rao inequality* states that no method can measure the parameters with errors smaller than what is given by the Fisher matrix [3].

To develop the algorithm for estimating the power spectrum, we now find the Taylor expansion of our original likelihood function f, to second order

$$f = \overline{f} + \sum_l \left. \frac{\partial f}{\partial C_\ell} \right|_{\overline{C}_\ell} (C_\ell - \overline{C}_\ell) + \sum_{\ell,\ell'} \frac{1}{2} \left. \frac{\partial^2 f}{\partial C_\ell \partial C_{\ell'}} \right|_{\overline{C}_\ell} (C_\ell - \overline{C}_\ell)(C_{\ell'} - \overline{C}_{\ell'}) \tag{3.11}$$

To maximize the likelihood, we differentiate this expression, sets it equal to zero, and throws away any terms higher than second order. To see what is going on with this equation, we may use a test-spectrum, $C_\ell^{(0)}$. The derivative of the likelihood function in the neighborhood of $C_\ell^{(0)}$ can then be written as

$$\left.\frac{\partial f}{\partial C'_\ell}\right|_{\overline{C}_\ell} + \sum_\ell \left.\frac{\partial^2 f}{\partial C_\ell \partial C_{\ell'}}\right|_{\overline{C}_\ell} (C_\ell - C_\ell^{(0)}) = 0 \tag{3.12}$$

We will assume the second derivative can be approximated with it's expectation value given by eq. 3.10. This leads to:

$$\frac{\partial f}{\partial C_{\ell'}} + 2\sum_\ell \mathbf{F}_{\ell\ell'}(C_\ell - C_\ell^{(0)}) = 0 \tag{3.13}$$

For any particular multipole ℓ we now get

$$\begin{aligned}(C_\ell - C_\ell^{(0)}) &= -\frac{1}{2}\sum_{\ell'} \mathbf{F}^{-1}_{\ell\ell'}\frac{\partial f}{\partial C_{\ell'}} \\ C_\ell &= C_\ell^{(0)} - \frac{1}{2}\sum_{\ell'} \mathbf{F}^{-1}_{\ell\ell'}\frac{\partial f}{\partial C_{\ell'}}\end{aligned} \tag{3.14}$$

This is our final iterative method for estimating the power spectrum,

$$\begin{aligned}C_\ell^{(n+1)} &= C_\ell^{(n)} - \frac{1}{2}\sum_{\ell\ell'} \mathbf{F}^{-1}_{\ell\ell'}\left.\frac{\partial f}{\partial C_{\ell'}}\right|_{C_{\ell'}^{(n)}} \\ &= C_\ell^{(n)} - \frac{1}{2}\sum_{\ell'} \mathbf{F}^{-1}_{\ell\ell'}\left(-\mathbf{m^T C^{-1} P^{\ell'} C^{-1} m} + \mathrm{tr}(\mathbf{C^{-1}P^{\ell'}})\right)\end{aligned} \tag{3.15}$$

where

$$P^\ell \equiv \frac{\partial \mathbf{C}}{\partial C_\ell}$$

For this algorithm to give the correct result, the covariance matrix and it's inverse have to be exact. For the Fisher matrix however the demands are not as stringent. An approxmiate Fisher matrix will do to make the algorithm converge.

There are a few challenges facing us in this algorithm. Our data set is very large, leading to even bigger covariance matrices, and they need to be inverted. However, there are tools to help overcome these problems. We will look at them next.

3.1.2 Conjugate Gradient method

To be able to calculate the term $\mathbf{C^{-1}m}$ we need to invert the covariance matrix. An efficient technique for doing this is using the Conjugate Gradient method.

The conjugate gradient method is an iterative method of solving the linear equation

$$\mathbf{Cz} = \mathbf{m}$$

where $\mathbf{z}$ is an unknown vector, while the vector $\mathbf{m}$ and matrix $\mathbf{C}$ are known.

A good explanation of this method is given by [11], and I will not go into the details here. The method works by creating a new basis for the matrix involved, and making the set of linear equations into a problem of minimizing the derivative of the equations. It starts from any chosen point and searches along one of the basis vectors for minimum points. The remaining error after one of these searches is known as the residual, and the method is guaranteed to look for any minima along one basis vector only once. In this way there are a maximum number of steps the algorithm has to take to get an answer, and since it also makes the biggest steps initially it is possible to terminate the search when the residual becomes small. This often leads to the method having to do considerably fewer iterations than the maximum required. The method converges fast and is very memory efficient.

To improve the convergence of the method, it is useful to include a preconditioner $\mathbf{M}$ in the algorithm. The equation we try to solve will then have the form

$$\mathbf{M^{-1}Cz} = \mathbf{M^{-1}m}$$

The preconditioner is a matrix that approximates $\mathbf{C}$ but is easier to invert. In the original OSH-algorithm much of the work consists of trying to construct a sufficiently good preconditioner. When we use needlet transformations we will later see we get a useful preconditioner almost for free.

The Conjugate Gradient method puts some demand on the matrix involved. For the method to be efficient, the matrix $\mathbf{C}$ should be sparse. In addition it is required to be symmetric, and positive-definite. Our covariance matrix is obviously symmetric, but we will have to test the conjugate gradient method with a needlet transformed matrix to determine if it is sparse or positive-definite.

3.1.3 Calculating trace and Fisher matrix

The Conjugate Gradient method takes care of the inverse covariance matrix in the first term, but $\mathbf{C}^{-1}$ appears in the trace-term and the Fisher matrix as well, and here it is of no direct use. There is a workaround however.

Starting from any given power spectrum we are able to generate a random universe. In this universe we are able to calculate the result

$$\mathbf{q} = \mathbf{m^T C^{-1} P^{\ell'} C^{-1} m}$$

which is simply the term we have already calculated with the help of the CG-method.

We may now use the fact [9] that the trace can be calculated as

$$\mathrm{tr}(\mathbf{C^{-1}P^{\ell}}) = \left\langle \mathbf{m^T C^{-1} P^{\ell'} C^{-1} m} \right\rangle = \langle \mathbf{q} \rangle$$

To get this we then use Monte Carlo-simulations to generate random universes given our previously guessed power spectrum, and this should give us the trace after a sufficient number of realizations.

In addition we can use the same set of simulations to calculate the Fisher matrix, given that:

$$\mathbf{F} = \frac{1}{4} \left\langle (\mathbf{q} - \langle \mathbf{q} \rangle)(\mathbf{q} - \langle \mathbf{q} \rangle)^{\mathbf{T}} \right\rangle$$

In this way we have been able to use the Conjugate gradient method when calculating the trace and Fisher matrix as well.

The Monte Carlo simulations required to obtain the trace and Fisher matrix in this way can make the algorithm quite time consuming. To counter this somewhat it is suggested in the original paper to use simplified inverse matrixes directly for the first few iterations of the algorithm, and only use the full Monte Carlo simulations when the largest steps in the algorithm have been done and better precision is required.

3.2 Needlets

In recent years wavelets have become a popular tool in many different fields of signal processing. Various types of wavelets have been developed for different

needs. The one we will look at here has been made for analysis of the CMB and is known as *spherical needlet transformation.*

In the OSH-algorithm we have to work with very large matrices, and it is hoped that the special properties of the needlet transformation will make it possible to construct much simpler matrices.

3.2.1 The sound analogy

Needlets are a kind of wavelet, a special mathematical transformation designed to pick out a specific range of the original signal, and a good analogy can be found in the treatment of a sound signal.

Sound is basically waves in air, and easily recorded by a microphone, thereby making an electrical or digital signal that varies through time. We may Fourier-transform this signal, getting it on a new form that contains information about different frequencies in the original sound.

When processing the transformed sound we may be interested in picking out only a small range of frequencies in the signal, for example only the frequencies below 100 Hz. This is what would be called filtering the signal in sound processing, in this case we are dealing with a low-pass filter.

The fluctuations in the CMB are really an image of the fluctuations in the cosmic fluid at recombination. The spherical harmonic transformation of this signal is then analogous to determining what frequencies appear in the signal. Using needlets on the transformed signal can be compared to filtering out a specific range of frequencies, picking out only a certain range of multipoles.

3.2.2 Definition of Needlets

Needlets are a special type of wavelets constructed to be used on the two-dimensional sphere, and this makes it especially useful when studying the CMB. They are mathematically described by [1], and the use combined with the CMB is described by [8]. They have excellent localization properties in both pixel-space and spherical harmonic space, and can be seen as a kind of midway point between the two representations.

When making needlet coefficients we are basically taking a limited range of multipoles from the original signal and turning them into a new map. This map is known as a j-scale, and the original map can be decomposed into many different scales, each with it's own N_{side} resolution. The detailed constructin is as follows:

$$\beta_{jk} = \sqrt{\lambda_{jk}} \sum_{\ell} g\left(\frac{\ell}{B^j}\right) \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\xi_{jk})$$

Here the ξ represents points on the sphere, and λ is a weighting of these points. When using the points defined by the pixelization scheme in HEALPix, the weighting function can be set to 1. The needlet coefficient β is a real number, and we get one for each scale j and pixel k.

The most important part of the construction is the function $g(\cdot)$, which acts as a window function that picks out the multipoles we want to include in each j-scale. It is a function of the parameters ℓ and B, where B is used to control the range of ℓ's to be included in each scale j. It can be considered a parameter that controls the bandwidth of the scales. Each j-scale will then contain information related to the ℓ's from B^{j-1} to B^{j+1}. In this way we are able to choose the resolution and range of ℓ's we want to look at.

Figure 3.1 shows the window functions for scales $j = 3$, $j = 4$ and $j = 5$, with $B = 2.0$. The needle-like shape of the window function is what has given the needlets their name. We see here some notable features. Each scale has an overlap with it's neighbouring scale, but no other scales. Each scale has a range in multipole from B^{j-1} to B^{j+1}. As an example, scale $j = 5$ has a range from multipole $\ell = 16$ to $\ell = 64$. The endpoints are set to zero, so they do not contribute to the scale. We should note that when the B has been chosen, the number of multipoles included in each scale will increase exponentially for each scale.

In the implementation with HEALPix the needlet coefficients are given as

$$\beta_{jk} = \sum_{\ell} g_{\ell j} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(k) \tag{3.16}$$

The only thing we need to provide is the window function $g_{\ell j}$. As we can see from figure 3.1 this seems to be a fairly simple function, going from 0 to 1. All that needs to be done is some scaling along the multipole axis to include the relevant ℓ's. A detailed instruction of how to construct the

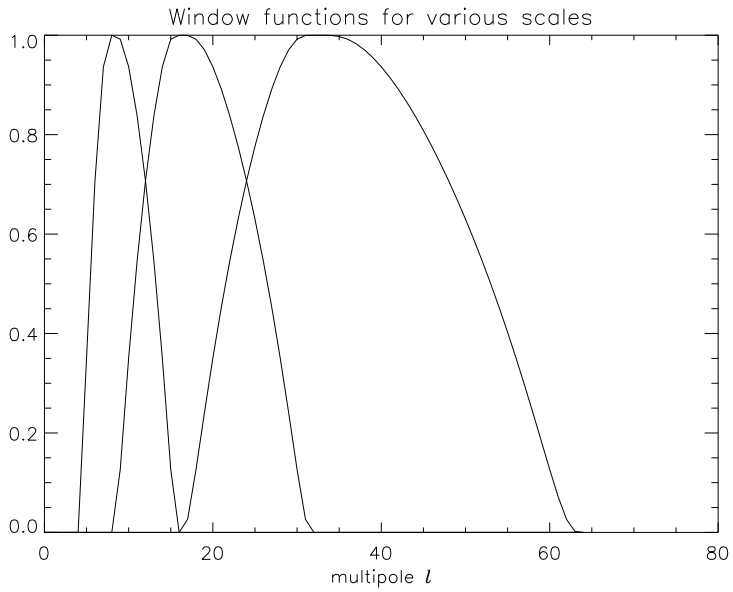

Figure 3.1: Window functions $g_{\ell j}$ for $B = 2.0$. Here we see the windowfunctions for scales $j = 3$, $j = 4$ and $j = 5$

window function is given by [8], and in this thesis I have made use of a Fortran function provided by my supervisor.

When used on the spherical harmonic transformed signal, the needlet transform does not give an orthogonal representation of the original signal, but a so-called tight frame. This kind of semi-orthogonality we write as:

$$\langle \beta_{jk} \beta_{j'k'} \rangle = \delta_{kk'} \delta_{j|j'-1|} \tag{3.17}$$

As we can see from this expression, the β-coefficients are orthogonal in k, while the scales do have overlap with the neighbouring scales.

It is this feature that motivates our attempt at using needlets in the first place. If we can make use of the needlet transformation on our CMB and get a set of coefficients that obey this simple relation, the construction and inversion of the covariance matrix will become very easy. This will help keeping the size of $\mathbf{C}$ down, even if the amount of data increases, and the inversion process will also be simplified. We will explore in some detail how feasible this is in reality when combining needlets with the OSH-algorithm.

3.2.3 Needlets and HEALPix

We should now be able to construct the necessary window functions $g_{\ell j}$, and by combining them with the built-in functions in the HEALPix-package we may generate new maps of the needlet-coefficients in the desired ranges. Each individual map then represent one scale and can be given it's own N_{side}-resolution.

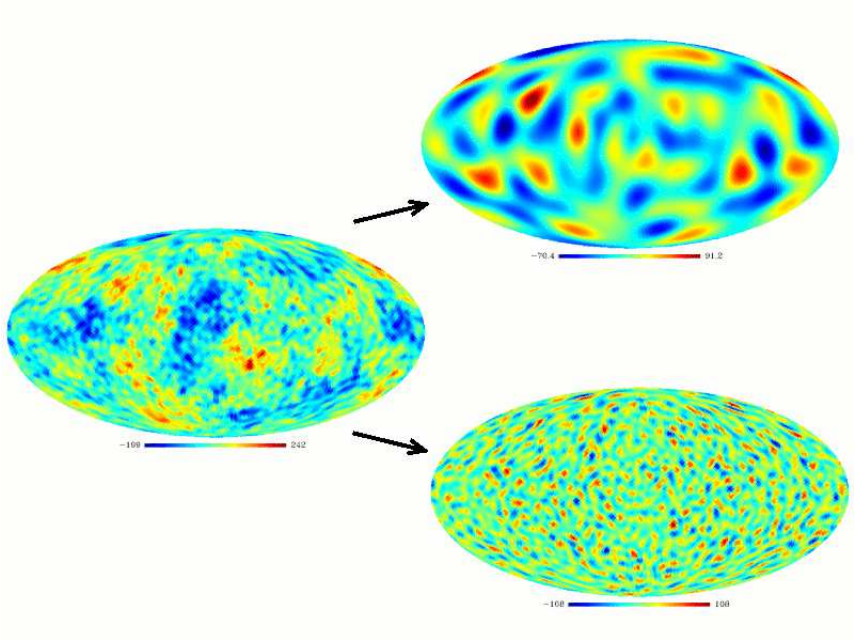

Figure 3.2: Visual result of needlet-transformations.

The visual result of the needlet-transformation is shown in figure 3.2. Here a simulated CMB-map is transformed into maps on scales $j = 3$ and $j = 5$, each with nside=32. Now the map for $j = 3$ contains information about multipoles 4 to 16, while the map for $j = 5$ contains information about multipoles 16 to 64.

3.2.4 Covariance matrix with needlets

A very important quantity in the study of the CMB is the covariance matrix, and in the case of needlets we are able to generate it analytically. The

covariance matrix for needlet coefficients is constructed in the usual way, and since they are real numbers we may write the correlation matrix as

$$C_{jk,j'k'} = \langle \beta_{jk} \beta_{j'k'} \rangle = \left\langle \beta_{jk} \beta^*_{j'k'} \right\rangle$$

where the $*$ indicates the complex conjugated. This is possible since the needlet coefficients are real numbers. We may now insert equation 3.16 and solve.

$$\begin{aligned}
\left\langle \beta_{jk} \beta^*_{j'k'} \right\rangle &= \left\langle \sum_\ell g_{\ell j} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(k) \sum_{\ell'} g_{\ell' j'} \sum_{m'=-\ell'}^{\ell'} a^*_{\ell' m'} Y^*_{\ell' m'}(k') \right\rangle \\
&= \left\langle \sum_\ell g_{\ell j} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(k) \sum_{\ell'} g_{\ell' j'} \sum_{m'=-\ell'}^{\ell'} a_{\ell' m'} Y^*_{\ell' m'}(k') \right\rangle \\
&= \left\langle \sum_\ell \sum_{\ell'} \sum_{m=-\ell}^{\ell} \sum_{m'=-\ell'}^{\ell'} g_{\ell j} a_{\ell m} Y_{\ell m}(k) g_{\ell' j'} a_{\ell' m'} Y^*_{\ell' m'}(k') \right\rangle \\
&= \sum_\ell \sum_{\ell'} \sum_{m=-\ell}^{\ell} \sum_{m'=-\ell'}^{\ell'} \langle a_{\ell m} a_{\ell' m'} \rangle \, g_{\ell j} Y_{\ell m}(k) g_{\ell' j'} Y^*_{\ell' m'}(k') \\
&= \sum_\ell \sum_{\ell'} \sum_{m=-\ell}^{\ell} \sum_{m'=-\ell'}^{\ell'} \delta_{\ell \ell'} \delta_{m m'} C_\ell g_{\ell j} Y_{\ell m}(k) g_{\ell' j'} Y^*_{\ell' m'}(k') \\
&= \sum_\ell C_\ell g_{\ell j} g_{\ell j'} \sum_{m=-\ell}^{\ell} Y_{\ell m}(k) Y^*_{\ell m}(k') \\
&= \sum_\ell C_\ell g_{\ell j} g_{\ell j'} \frac{2\ell + 1}{4\pi} P_\ell(\cos \Delta k) \qquad (3.18)
\end{aligned}$$

Here P_ℓ is the Legendre polynomial and Δk is the angular distance between k and k'. The second line holds because $a_{\ell -m} = (-1)^m a^*_{\ell m}$, and since we sum over all m the minus signs gets elliminated. The fourth line holds since the only varying quantity are the $a_{\ell m}$ coefficients, and furhter we expect the relation $\langle a_{\ell m} a_{\ell' m'} \rangle = \delta_{\ell \ell'} \delta_{m m'} C_\ell$. Finally the last line uses the identity from eq. 2.44.

Before we go ahead and make use of this result we should examine if it holds. The alternative to constructing the covariance matrix analytically is doing it with Monte Carlo simulations. By making simulations we may test how well the analytical expression fares compared to a set of randomly generated universes.

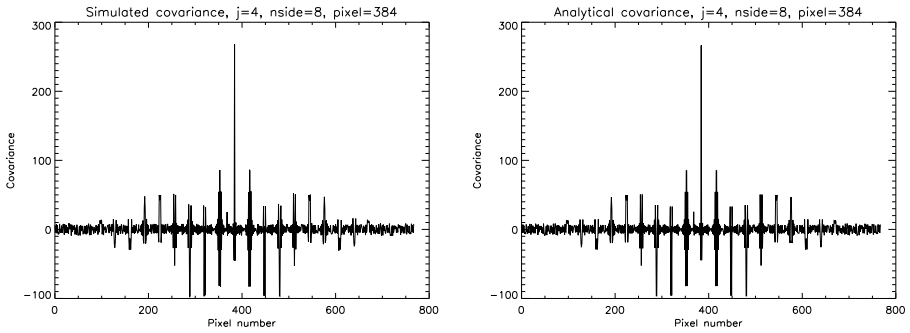

Figure 3.3: Left: Simulated covariance for a single pixel. Right: Analytical covariance for the same pixel.

Figure 3.3 shows the covariance for a pixel using a Monte Carlo simulated covariance matrix on the left, and the same results from an analytically constructed covariance matrix on the right. The central peak of each plot is the variance of the pixel. The simulation has been done with 100.000 realizations, and apart from some small amount of noise there is hardly any difference between the two. In fact we would be hard pressed to notice any differece between the two plots at all.

Looking at figure 3.4 we see the absolute error between these two matrixes plotted in black. In this case the absolute error of the two plots has a maximum magnitude of 6. Compared to the maximum value of the covariance matrix this is a relative error of about 2%. The relative difference is in most cases small, but breaks down in places, since the covariance approaches zero for several pixels.

When we try to compare with even more realizations we get even better results. Looking at figure 3.4 we also have the absolute error when comparing the analytical result with a simulation of 10 million realizations. As we can see from the figure the absolute error has decreased drastically, and it seems safe to assume that even more simulations would bring the analytical and simulated covariance matrices even closer together. The analytical expression given by eq. 3.18 seems to be holding up.

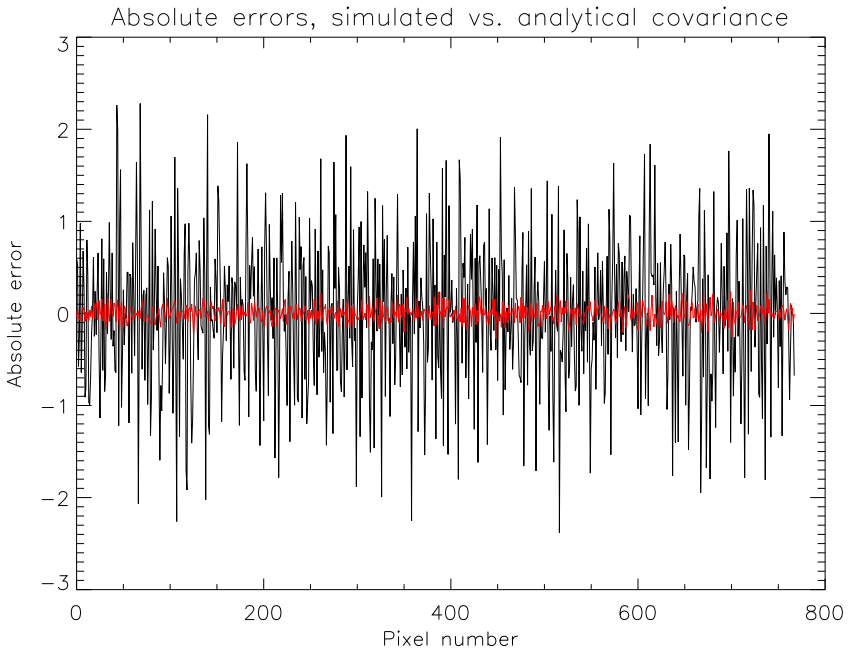

Figure 3.4: Comparison of absolute errors for analytical versus simulated covariance. Black line indicates a simulation with 100.000 realizations, red line a simulation with 10 million realizations.

3.2.5 Noise correlation with Needlets

We expect the noise to be uncorrelated with the signal, and this means we may consider the noise-correlation in needlet-space separately, just as in pixel space. If we naively consider the noise-component of the CMB to be isotropic white noise, Gaussian random variables, we expect the power spectrum to be

$$\langle n_{\ell m} n_{\ell' m'} \rangle = \delta_{\ell \ell'} \delta_{m m'} C$$

where C is a constant and independent of both ℓ and m.

Using the same procedure as in the analytical expression for signal correlation, we would then expect the noise correlation to be

$$\langle n_{\ell m} n_{\ell' m'} \rangle = C \sum_{\ell} g_{\ell j} g_{\ell j'} \frac{2\ell + 1}{4\pi} P_{\ell}(\cos \Delta k)$$

where C is a constant describing the power of any multipole in the white noise.

Sadly, for the real CMB-map, this doesn't really hold. Looking at figure 3.5 we see the standard deviation of the noise for WMAP [2]. As we can clearly

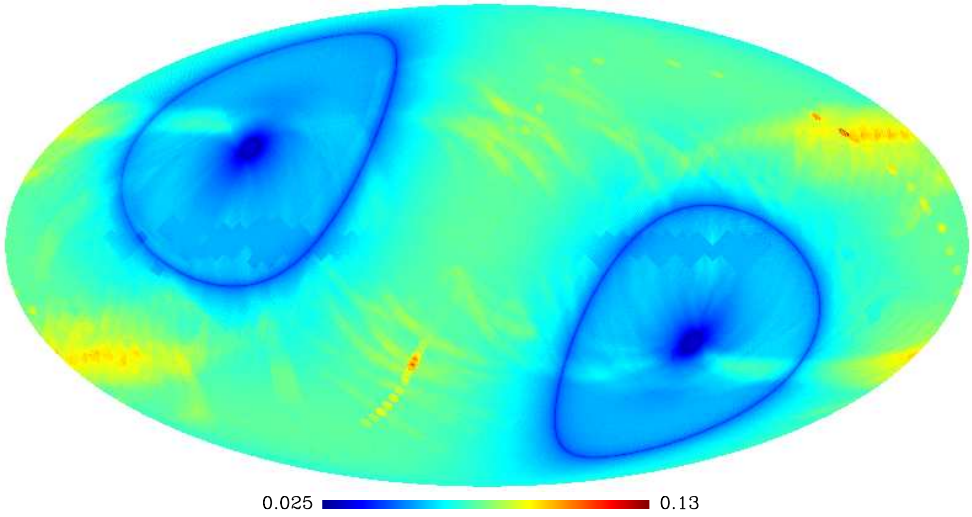

Figure 3.5: Standard deviation of noise σ_i for each pixel i in the WMAP V-band channel. The units are in milliKelvin.

see, the noise is heavily dependent on which direction we are looking at. This is a property of the method of observation and limits in the instruments, and we are unable to circumvent it.

A simulated test confirms this. Looking at figure 3.6 we see the absolute error of a simulation consisting of 10 million realizations versus the analytical expression with a single power coefficient C. Although the absolute error is not large, we do see a difference in the errors when comparing with figure 3.3. There is some kind of systematic error in the left part of the graph in figure 3.6 and the variation it the errors does not seem to be evenly distributed. We do get discrepancies and we therefore have to make proper simulations if we wish to incorporate noise in our analysis.

3.2.6 P^ℓ with Needlets

So far we have an analytical expression for the signal part of the covariance matrix $\mathbf{C}$. In the Oh-Spergel-Hinshaw algorithm we previously defined the

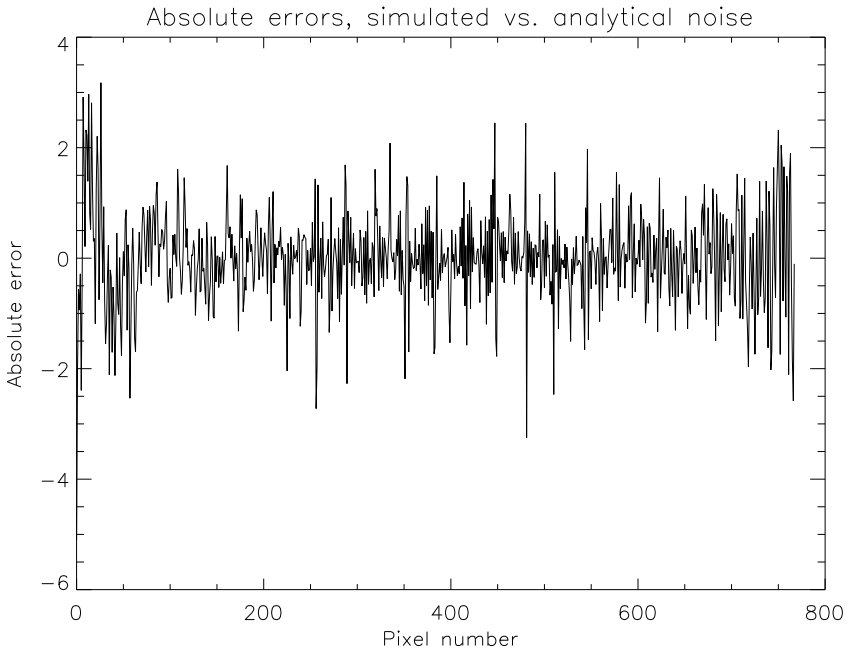

Figure 3.6: Results when comparing analytical and simulated noise covariance using data from WMAP.

quantity $P^\ell = \frac{\partial \mathbf{C}}{\partial C_\ell}$, and this makes it possible to obtain an analytical expression for P^ℓ as well. Since the noise correlation matrix is independent of the power spectrum it has no effect on the partial derivative, we only need the signal part. The full P^ℓ in needlet space then becomes

$$\begin{aligned} P^l &= \frac{\partial \mathbf{C}}{\partial C_{\ell'}} \\ &= \frac{\partial}{\partial C_{\ell'}} \sum_\ell C_\ell g_{\ell j} g_{\ell j'} \frac{2\ell+1}{4\pi} P_\ell(\cos \Delta k) \\ &= \sum_\ell \frac{\partial C_\ell}{\partial C_{\ell'}} g_{\ell j} g_{\ell j'} \frac{2\ell+1}{4\pi} P_\ell(\cos \Delta k) \\ &= \sum_\ell \delta_{\ell\ell'} g_{\ell j} g_{\ell j'} \frac{2\ell+1}{4\pi} P_\ell(\cos \Delta k) \\ &= g_{\ell j} g_{\ell j'} \frac{2\ell+1}{4\pi} P_\ell(\cos \Delta k) \end{aligned} \tag{3.19}$$

We should now be equipped with what we need to implement needlets in the Oh-Spergel-Hinshaw algorithm.

Chapter 4

Results

Having developed the Oh-Spergel-Hinshaw algorithm and examined needlets, we are now able to put the method to the test using needlet transformed signals. We will begin by looking at the general shape of the algorithm and what we can expect to gain from it. We will also test the needlet-transformation to check that the transformation back and forth between pixel space and needlet space works, and also to see if there is any dependence on resolution N_{side}. Then we will finally look at the results when we actually implement needlets in the OSH-algorithm. The hope is that needlet transformations will make it possible to use simpler covariance matrices, and thereby making the algorithm quicker for large amounts of data.

In the rest of this thesis we will work with simulated signals that does not contain noise. This does not correspond to a realistic case, but makes it possible to use purely analytical matrixes in the algorithm, avoiding the need for making time-consuming simulations of the noise correlation.

4.1 Implementing needlets with OSH algorithm

We will begin by examining closely the structure of the OSH-algorithm when implementing needlets, and also test the transformation back and forth between real space and needlet space. We will also make sure the Conjugate Gradient method works as expected.

4.1.1 The structure of the algorithm

As we found in eq. 3.15 the Oh-Spergel-Hinshaw algorithm has the following form:

$$C_\ell^{(n+1)} = C_\ell^{(n)} - \frac{1}{2}\sum_{\ell'} \mathbf{F}^{-1}_{\ell\ell'} \left(-\mathbf{m^T C^{-1} P^{\ell'} C^{-1} m} + \mathrm{tr}(\mathbf{C^{-1} P^{\ell'}}) \right) \tag{4.1}$$

In this algorithm there are several elements involved, and to implement needlets we have to ensure that all elements are properly transformed to needlet space.

Starting with the data vector $\mathbf{m}$,

$$\mathbf{m} = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_N \end{pmatrix}$$

where each pixel has been numbered up to the maximum N determined by the resolution of the map (eq. 2.51).

When transforming to needlet space we have to decide the bandwidth of the needlet window-function (the B-parameter), the N_{side} resolution of each j-scale, and what number of j-scales to include. Each of these choices will affect what information we can gain in our analysis, and we will come back to the details later.

By doing the needlet transformation on our data vector we get

$$\begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_N \end{pmatrix} \rightarrow \left(\begin{array}{c} \beta_{1,1} \\ \beta_{2,1} \\ \vdots \\ \beta_{N_1,1} \\ \hline \beta_{1,2} \\ \beta_{2,2} \\ \vdots \\ \beta_{N_2,2} \\ \hline \vdots \\ \hline \beta_{1,j} \\ \beta_{2,j} \\ \vdots \\ \beta_{N_j,j} \end{array}\right)$$

The result is a set of needlet coefficients β for each scale, and each coefficient is indexed by the scale it belongs to and the pixel number it is given, determined by the resolution. Each scale has a maximum number of pixels given by N_j. Clearly the data vector gets much larger.

This also means that the matrixes involved will have to be larger, and so far it does not look like we are getting a simplification of the method. But we do have to remember that needlets are supposed to obey the relation $\langle \beta_{jk} \beta_{j'k'} \rangle = \delta_{kk'} \delta_{j|j'-1|}$, and this would lead to tridiagonal matrices.

When constraining ourselves to looking at a pure signal wihtout any noise, we have already found an analytical expression for the covariance matrix in eq. 3.18. Figure 4.1 shows the covariance matrix containing four j-scales.

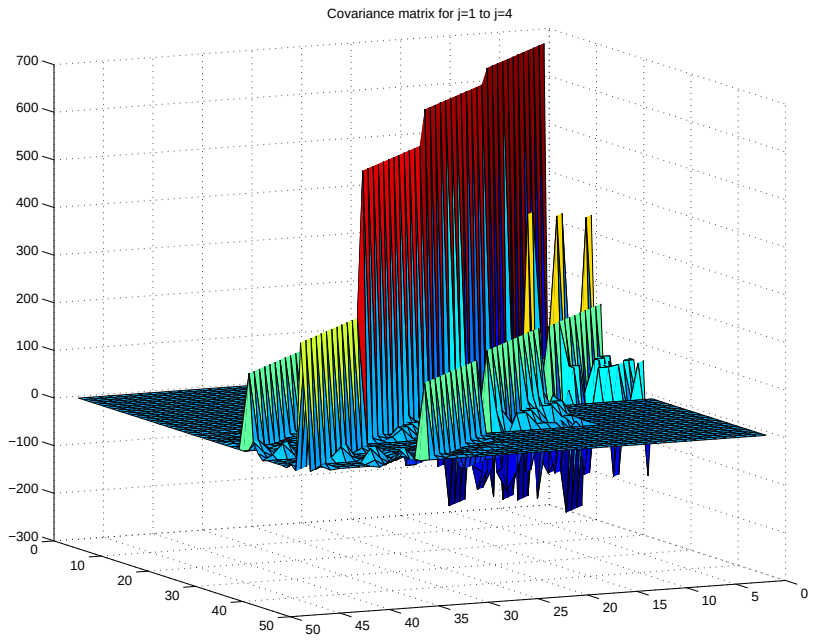

Figure 4.1: Covariance matrix containing scales $j = 1$ to 4.

Each scale has a resolution $N_{\text{side}} = 1$, which leads to each scale containing 12 needlet coefficients. The diagonal in this matrix represents the self-correlation $\delta_{kk'}$, and this is clearly dominant. But particularly for the scale $j = 1$, which is representend by elements 1 to 12, we have a lot of non-zero

values in the off-diagonal elements as well. This is easier to see in figure 4.2 where we are looking at a top-down view of the same matrix.

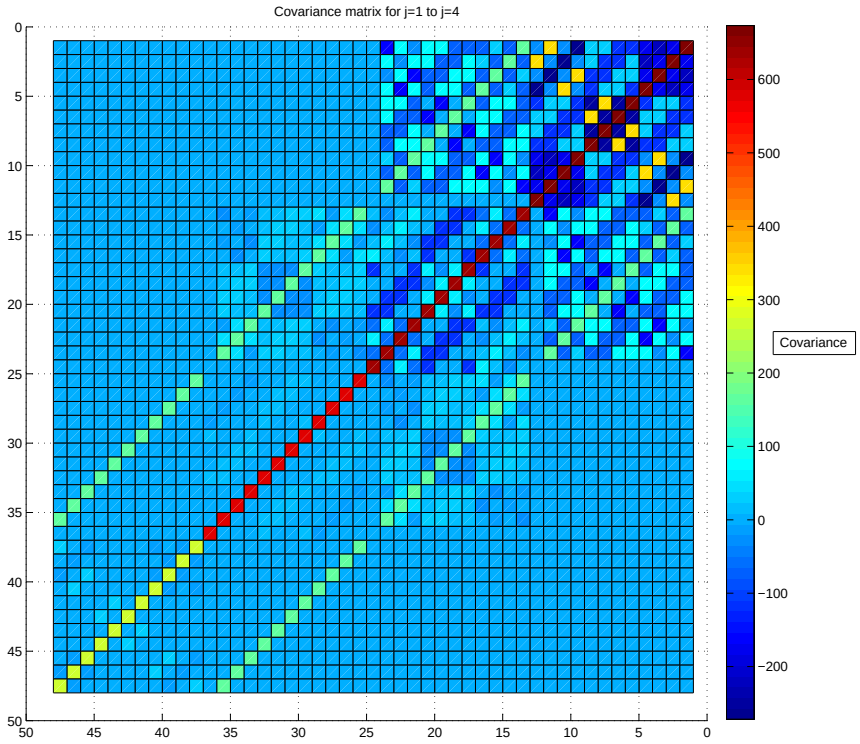

Figure 4.2: A top-down view of the covariance matrix shown in fig. 4.1

Here the color-coding indicates to what degree the pixels correlate, and there are big variations in the correlations between pixels 1 to 24. In the figure we can also see traces of the tridiagonal structure of the matrix, stemming from the term $\delta_{j|j'-1|}$.

The plots of this matrix indicates that the diagonal elements belonging to a particular scale has the same value, and looking at eq. 3.18 we can see that this is true. The equation gives the signal part of the covariance matrix as

$$\mathbf{C} = \sum_{\ell} C_\ell g_{\ell j} g_{\ell j'} \frac{2\ell + 1}{4\pi} P_\ell \cos(\Delta k)$$

The only part of the equation dependent on the pixels are the Legendre polynomials, $P_\ell(\cos \Delta k)$. When we look at the diagonals, we are seeing the

variance of each pixel. In this case $\Delta k = 0$, leading to $P_\ell(\cos 0) = 1$ for every ℓ. The only other dependence we have in the analytical expression is then the sum over ℓ, and this will be the same for every k in a given j-scale. This means that for a matrix where the relation $\delta_{kk'}$ holds, we only need a single number per element C_{jj} to characterize the covariance matrix, given by

$$C_{jj'} = \sum_\ell C_\ell g_{\ell j} g_{\ell j'} \frac{2\ell + 1}{4\pi}$$

The same argument holds for the large off-diagonal elements coming from the term $\delta_{j|j'-1|}$. If we can use this either as an approximation or as the true covariance matrix, we need only a matrix of dimension $j \times j$,

$$\mathbf{C}_{jj'} = \begin{pmatrix} C_{11} & C_{12} & 0 & \cdots & 0 \\ C_{21} & C_{22} & C_{23} & 0 & \vdots \\ 0 & C_{32} & C_{33} & \ddots & 0 \\ \vdots & & \ddots & \ddots & C_{j-1,j} \\ 0 & \cdots & 0 & C_{j,j-1} & C_{jj} \end{pmatrix}$$

and this matrix would be symmetric and tridiagonal. This would lead to a drastic decrease in complexity, particularly when we need to invert the covariance matrix.

From eq. 3.19 it should be clear that the structure of $\mathbf{P}^\ell$ is the same as for the covariance matrix. This leads to a simple tridiagonal matrix in this case as well.

The shape of the fisher matrix $\mathbf{F}_{\ell\ell'}$ is dependent on how we decide to construct our covariance and $\mathbf{P}^\ell$ matrices. Looking at figure 4.3 we see the diagonal elements of the Fisher matrix for two different cases. The continuous line shows the diagonal elements constructed from the full $\mathbf{C}^{-1}$ and $\mathbf{P}^\ell$ matrices, while the dotted line is constructed from the tridiagonal simplified matrices. Here we have again used resolution $N_{\text{side}} = 1$, and using other resolutions will also affect the result. What this means for our analysis is not immediately clear, and we will have to look into the detailed estimates to possibly get a better idea of the effect it has.

In the final algorithm we do not want to use simplified matrices. Remember, the algorithm is only correct with exact covariance matrices. But we may be able to perform the first steps of the algorithm using simplified matrices, and using the full matrices only when precision is required.

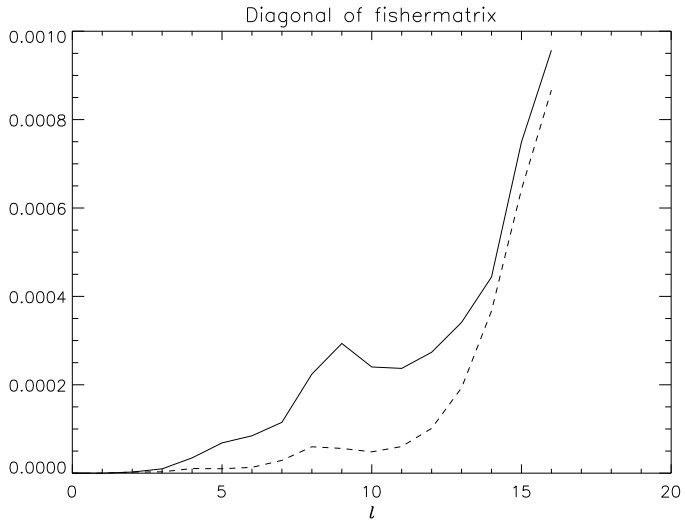

Figure 4.3: Diagonal elements of the Fisher matrix constructed from a full matrix(continuous line), and from a tridiagonal matrix(dotted line)

Let's finally look at one of the simplest possible ways to set up our algorithm. If we want to estimate only $\ell = 2$, it is a simple matter to construct the needlets in such a way that only one scale is needed. Using $B = 2.0$ leads to $\ell = 2$ being described solely by $j = 1$, when we remember that any scale covers B^{j-1} to B^{j+1}. The endpoints are set to zero, so for $j = 2$ the endpoints would be $\ell = 2$ and $\ell = 8$, meaning that this scale dose not contain any infromation about $\ell = 2$. With this choice we get $g_{\ell=2,j=1} = 1.0$, and assuming $\delta_{kk'}$ holds we can express the variance as:

$$\begin{aligned}\mathbf{C} &= \sum_{\ell} C_\ell g_{\ell j} g_{\ell j'} \frac{2\ell+1}{4\pi} P_\ell \cos(\Delta k) \\ &= C_2 \frac{5}{4\pi}\end{aligned}$$

and the inverse may immediately be written as $\mathbf{C}^{-1} = \frac{4\pi}{5C_2}$. From eq. 3.19 we get $\mathbf{P}^2 = \frac{5}{4\pi}$. This leads to $\mathbf{C}^{-1}\mathbf{P}^2 = 1/C_2$. The Fisher matrix given by eq. 3.10 turns out to be $\mathbf{F}_{22} = 1/2C_2^2$ and the inverted Fisher must then be

$\mathbf{F}_{22}^{-1} = 2C_2^2$. When inserting this into our algorithm (eq. 4.1) we get:

$$\begin{aligned} C_2^{(n+1)} &= C_2^{(n)} - \frac{1}{2}\mathbf{F}_{2,2}^{-1}\left(-\mathbf{m^T C^{-1} P^2 C^{-1} m} + \mathrm{tr}(\mathbf{C^{-1}P^2})\right) \\ &= C_2^{(n)} - \frac{1}{2}2C_2^{(n)2}\left(-\frac{4\pi}{5C_2^{(n)3}}\mathbf{m^T m} + \frac{1}{C_2^{(0)}}\right) \\ &= C_2^{(n)}\left(1 - \left(-\frac{4\pi\mathbf{m^T m}}{5C_2^{(n)2}} + 1\right)\right) \\ &= \frac{1}{C_2^{(n)}}\frac{4\pi}{5}\mathbf{m^T m} \end{aligned}$$

The algorithm have to converge at some point, leading to $C_2^{(n+1)} = C_2^{(n)}$, and inserting this we finally get

$$C_2^2 = \frac{4\pi}{5}\mathbf{m^T m} \tag{4.2}$$

We have now ended up with a simple result where C_2 is given by the inner product of the original data vector. We can test this out by generating a random needlet transformed data vector using the above paraeters. Using eq. 2.38 we find the true powerspectrum as $C_2 = 122$. The simple result in eq. 4.2 gives $C_2 = 33$. This is obviously not a very good estimate of the power spectrum, and we may suspect some of the assumptions we made to be wrong. It does however illustrate how much simpler the algorithm becomes by using the needlet transformation with the assumptions. It also gives some insight into how the more complex constructions of the covariance matrix behaves in needlet space.

In the following we will constantly refer to the various elements involved in the algorithm. For simplicity I will from now on use *mc-term* when refering to the $-\mathbf{m^T C^{-1} P^2 C^{-1} m}$-term, and $\mathrm{tr}(\mathbf{C^{-1}P^2})$ will be called the *trace-term*. The *Fisher matrix* should be unambiguous.

4.1.2 Testing the derivatives

As we have seen in the development of the Oh-Spergel-Hinshaw algorithm, the various elements entering into eq. 4.1 corresponds to the first and second derivatives of the original likelihood function. This means we have a way of

testing the validity of the algorithm by doing the derivatives numerically. In the following we have used

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} \tag{4.3}$$

as our numerical derivation.

To test the first derivative, we know we can write the logarithm of the likelihood function as

$$f = -\frac{1}{2}\left(\mathbf{m^T C^{-1} m} + \ln(\det \mathbf{C})\right)$$

There is a constant term as well, but we may ignore it since it disappears when we do the derivative. The first term in the parenthesis leads to the term $-\mathbf{m^T C^{-1} P^{\ell'} C^{-1} m}$ of our algorithm, while the second term gives us the trace-part. To calculate the determinant of $\mathbf{C}$ we need some way to calculate it numerically. In general $\mathbf{C}$ is too large to do this directly, but here we will look at some very small cases in order to test the code. I have used an LU-decomposition algorithm described by [10] to find the determinant.

What we have to do now is to compare the results calculated by each term in the algorithm by the numerical results found by using equation 4.3 on our undifferentiated likelihood function, doing a small variation around a chosen C_ℓ. I have chosen to look at $\ell = 2$ and $\ell = 8$.

We will examine the results for two different versions of the algorithm. Both versions use the Conjugate Gradient method to construct the inverse covariance, and they both use Monte Carlo simulations to determine the trace and Fisher-terms. The algorithm named *Tridiagonal* uses a tridiagonal version of the covariance matrix, as shown in eq. 4.1.1, while *Full* uses the full analytical covariance matrix given by eq. 3.18. I have then used 1000 realizations for the Monte Carlo simulations to calculate the trace and Fisher.

Table 4.1: First derivatives for $\ell = 2$.

	Mc-term		Trace-term	
$\ell = 2$	Numerical	Analytical	Numerical	Analytical
Tridiagonal	$-7.9220 \cdot 10^{-4}$	$-9.2631 \cdot 10^{-4}$	$7.324 \cdot 10^{-3}$	$1.477 \cdot 10^{-4}$
Full	$-3.8617 \cdot 10^{-4}$	$-3.8617 \cdot 10^{-4}$	$4.505 \cdot 10^{-3}$	$3.843 \cdot 10^{-3}$

Looking at table 4.1 we see the results of the first derivative test for $\ell = 2$. In this case we use only the scale $j = 1$, with an $\ell_{\max} = 3$. The resolution used

is $N_{\text{side}} = 1$. The results are stable when we vary the ΔC_ℓ in the numerical derivative, or when changing the precision of the Conjugate Gradient method.

For the algorithm using the full covariance matrix, the terms seems to be in good agreement between analytical and numerical derivatives. In the mc-term there is no difference between the two results for the precision given in table 4.1. The trace term shows somewhat larger discrepancy. The analytical term does not vary much if we use a higher amount of realizations for the Monte Carlo simulations, and it has been tested with 10.000 and 100.000 realizations. The numerical version of the trace term makes use of the natural logarithm of the determinant of $\mathbf{C}$. The algorithm I have used to calculate the determinant returns very large or small results, ranging in order of 10^{22} to 10^{-122}, and when we take the logarithm of these numbers there is reason to suspect we pick up some rounding-off errors on the way. This may be the explanation for the discrepancy between the numerical and analytical result for the results for full matrix.

In the case where we have used a tridiagonal matrix in the algorithm, the discrepancies are much larger. For the trace term the relative error is about 14% when comparing the numerical to the analytical result, and for the trace term the relative error is almost 50%. This seems to indicate that the use of simplified matrices is not valid in this case.

Table 4.2 shows the results for $\ell = 8$. Here we have used scales $j = 1$ to 4, with resolutions $N_{\text{side}} = 1$, 2, 4 and 8, with $\ell_{\text{max}} = 16$. This results in a matrix with dimensions 1020×1020 and this was too big to handle for the algorithm calculating the determinant. Thus we did not get a result for the trace term in the numerical derivatives.

Table 4.2: First derivatives for $\ell = 8$.

	Mc-term		Trace-term	
$\ell = 8$	Numerical	Analytical	Numerical	Analytical
Tridiagonal	-0.8613	-2.9715	-	2.9558
Full	-0.21493	-0.21495	-	0.22086

For the algorithm using a tridiagonal covariance matrix we note that the numerical and analytical results are far apart for the mc-term. The algorithm using the full matrix again shows excellent agreement between the numerical and analytical results. We have no numerical results to compare with the analytical results for the trace term. Note however that the analytical results

are of the same order as the mc-term. For the OSH-algorithm to converge this has to be true, so this is at least an indication that nothing is particularly wrong with the analytical trace-term for the algorithm using the full matrix.

We are able to test the calculation of the Fisher matrix as well. When looking at the definition in equation 3.10 it is obvious that we should use the trace term in the algorithm to numerically calculate the second derivative, and this should correspond with the Fisher matrix in the algorithm.

Table 4.3: Second derivative for $\ell = 2$ and $\ell = 8$.

$\ell = 2$	Numerical	Analytical	$\ell = 8$	Numerical	Analytical
Tridiagonal	$3.244 \cdot 10^{-6}$	$2.638 \cdot 10^{-5}$		$-6.551 \cdot 10^{-3}$	0.2473
Full	$1.504 \cdot 10^{-6}$	$1.462 \cdot 10^{-6}$		$1.458 \cdot 10^{-3}$	$1.364 \cdot 10^{-3}$

Results for numerical and analytical second derivatives for $\ell = 2$ and $\ell = 8$ are given in table 4.3. The scales and resolutions for the two cases are the same as for the first derivative tests.

The tridiagonal case gives results where the numerical and analytical versions wildly disagrees. This is further indication that using the tridiagonal approximation for the covariance does not produce very exact results. When using the full covariance matrix we see much better agreement between numerical and analytical results.

From these derivative tests we see that the use of a full covariance matrix produces good results for the test cases used. This is good, since it is absolutely necessary to use covariance matrices that are exact at some point in our algorithm. The tridiagonal matrix does not work well however. This means that, at least for the parameters tested here, the tridiagonal matrices do not represent the exact covariance matrices. The relation given by eq. 3.17 does not hold for the parameters we have used. As an approximation it may still be useful for the first steps in the algorithm, but when we need the precise calculation of the power spectrum, a full covariance matrix is required.

4.1.3 Testing the Conjugate Gradient method

We have not yet examined if the needlet transformed covariance matrix works with the Conjugate Gradient method. In our algorithm, the CG-method is used to calculate the term $\mathbf{C^{-1}m}$ where $\mathbf{C}$ is the covariance matrix, and $\mathbf{m}$ is the data vector describing the pixels in needlet-space. The result is a new vector given as

$$\mathbf{C^{-1}m} \equiv \mathbf{z}$$

It is then obivous that

$$\mathbf{Cz} = \mathbf{m}$$

and we should be able to compare the result of the Conjugate Gradient operation with our original data, simply by multiplying the resulting vector from the CG-method with the full covariance matrix.

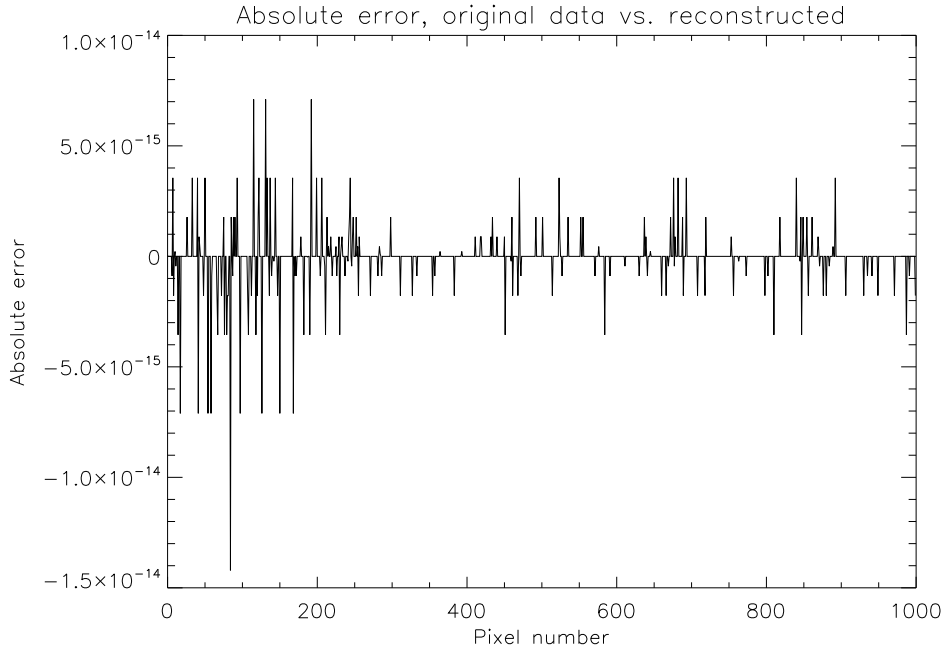

Figure 4.4: Absolute error of original datavector versus data reconstructed from the Conjugate Gradient method.

Figure 4.4 shows the absolute error when comparing the original data with data reconstructed from the result of the Conjugate Gradient method. Here the needlet-transformation is using scales $j = 1$ to 4, with resolutions $N_{\mathrm{side}} =$

1, 2, 4 and 8. When using the Conjugate Gradient method we also need to set a tolerance, and in this case I have used $\epsilon = 0.0001$.

Many of the elements in the data vector approaches zero, so trying to find the relative error does not work very well. But the elements in the vector ar of order 10^0 to 10^1, so compared with the absolute error the results of the comparison are very good. The small absolute errors we see in figur 4.4 can be attributed to small round-off errors.

A few words about the preconditioner. To make the algorithm more efficient we use a preconditioner, a matrix similar to the covariance matrix but easier to invert. Because of the relation given by eq. 3.17 a suggestion for the preconditioner gives itself. We should be able to use only the diagonal or possibly the tridiagonal matrix given by eq. 4.1.1 as a preconditioner. In this thesis I have used only the diagonal information from $\mathbf{C}$.

For the data vector used in figure 4.4 the number of elements included in the vector is 1020. In a worst case scenario this is the number of basis vectors the CG-method would have to search along to get a satisfactory result. In our case the CG-method has converged after about 30 iterations, only 3% of the total number of possible iterations needed. The algorithm may converge even faster if we use the tridiagonal information, but this has not been tested in this thesis.

From section 3.1.2 we remember the demands for a sparse and positive-definite covariance matrix for the Conjugate Gradient method to work. The needlet transformed $\mathbf{C}$ is clearly sparse. Looking back at figure 4.1 and 4.2 we see that the dominating elements in the matrix are the diagonals, with most off-diagonal elements very small. In fact, the construction of the needlet window function $g_{\ell j}$ ensures that all matrix elements not fulfilling the $\delta_{j|j-1|}$ assumption are set equal to zero.

We have not tested directly whether the covariance matrix is positive-definite or not, but since the Conjugate Gradient-method produces good results the matrix fulfulls this requirement. In conclusion it seems like the CG-method works well with needlet transformed $\mathbf{C}$, and should work as intended for our purpose.

4.1.4 Getting maps from needlet-coefficients

The needlet transformation of our signal must retain all information contained in the original map, and we would expect to regain the complete original signal when we do the reverse operation. When using HEALPix the transformation from needlet coefficients back to spherical harmonic space looks like

$$a_{\ell m} = \sum_j g_{\ell j} \beta_{lm} \tag{4.4}$$

where

$$\beta_{lm} = \sum_k \beta_{jk} Y^*_{lm}(k)$$

In practice this means doing a spherical harmonic transformation of each scale, getting spherical harmonic coefficients for the needlet transformed signal. These coefficients then needs to be weighted by the needlet window function and summed over all scales. We are now left with a spherical harmonic realization of the orignial signal and may convert it to a map or extract the power spectrum.

A very important requirement we want to examine is the limit of information we can obtain from an image based on the resolution. We remember from section 2.2.4 the arguments for the maximum multipole in an image as $\ell_{\max} = 2N_{\text{side}}$. We will now look at the resolution dependence when using needlet transformations.

We construct a signal using $\ell_{\max} = 1024$. In pixel representation we will then need to use $N_{\text{side}} = 512$. From this orignial signal we extract the power spectrum using eq. 2.39, and the result is shown as black in the right plot of figure 4.5.

For the needlet transformation of this signal we use bandwidth $B = 2.0$. To include the entire sighal up to $\ell_{\max} = 1024$ we have to use 10 j-scales. For each scale the resolution has been set to $N_{\text{side,j}} = \frac{1}{2}B^{j+1}$, except for scale $j = 10$. The reason for this is that using the above formula for finding the resolution, we should have used $N_{\text{side}} = 1024$. Since the signal only goes to $\ell_{\max} = 1024$ we only need to use $N_{\text{side}} = 512$ for $j = 10$. After having used these parameters for the transformation we use eq. 4.4 to get back the original spherical harmonic coefficients, and extract the power spectrum. The results are plotted in red in figure 4.5.

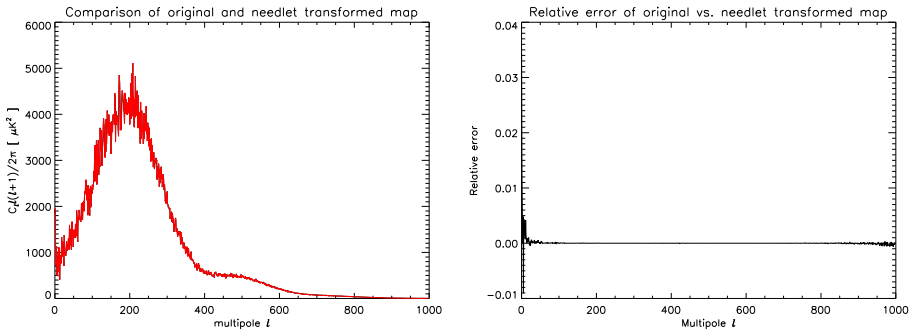

Figure 4.5: Left: Power spectrum for original image(black) and needlet-transformet image(red). Right: Relative error of the transformed vs. original image.

As we see from the figure, both power spectras seems to be in agreement, it is hardly possible to se the plot of the untransformed power spectrum beneath the transformed one. The right plot in figure 4.5 shows the relative error between the two power spectras. As we see the error is small. Some of the larger errors appear for multipoles lower than $\ell = 50$, but even here the difference is less than 1%. For high ℓ's the relative error grows as well, but this is due to round-off error, since there is very little signal left to compare with.

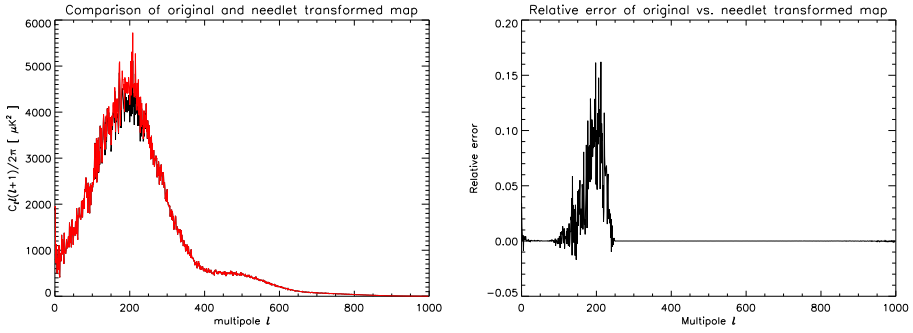

Figure 4.6: Left: Power spectrum for original image(black) and needlet-transformet image(red). Right: Relative error of the transformed vs. original image.

Let's now examine more closely what happens if we change one parameter of the needlet transformation. We use the same setup as above, but for scale

$j = 7$ we change the resolution. In the first test we used $N_{\text{side}} = \frac{1}{2}B^{j+1} = 128$, since this scale includes multipoles from $\ell = 64$ to $\ell = 256$. If we now change the resolution of $j = 7$ to $N_{\text{side}} = 64$ we get the result shown in figure 4.6. Looking at the left plot there is now a clear difference between the original spectrum (in black) and the transformed spectrum (in red). The right plot shows the relative error in this case, and we see a large error in exactly the multipoles spanned by $j = 7$.

From this it seems we have the same constraints to resoultion for the j-scales as for a regular temperature map. There is a limit to how low a resolution we are able to use while still retaining information about the multipoles, and the resolution we have to use is given as

$$N_{\text{side,j}} = \frac{1}{2}B^{j+1} \tag{4.5}$$

4.1.5 Dependence on pixel-resolution

Given the lower boundary in resolutions, the question is now if this resolution is able to fulfill the expectation given by eq. 3.17. In words, we need to examine if the term $\delta_{kk'}$ holds for the resolutions we are required to work with.

Using the analytical expression for the covariance matrix, we may look a bit closer at the angular dependency for $\delta_{kk'}$. For scale $j = 1$ we have a multipole range of $\ell = 1$ to $\ell = 4$ when using $B = 2.0$. Since the factor $g_{\ell j}$ sets the endpoints to zero, the only contributing multipoles for this scale are $\ell = 2$ and $\ell = 3$. The covariance matrix is given by eq. 3.18 as

$$\mathbf{C} = \sum_{\ell} C_\ell g_{\ell j} g_{\ell j'} \frac{2\ell + 1}{4\pi} P_\ell(\cos \Delta k)$$

and when only using the relevant multipoles we get

$$C_{kk'} = C_{\ell=2} g^2_{\ell=2,j=1} \frac{5}{4\pi} P_{\ell=2}(\cos \Delta k) + C_{\ell=3} g^2_{\ell=3,j=1} \frac{7}{4\pi} P_{\ell=3}(\cos \Delta k)$$

We get the values for the power spectrum from the WMAP results [2], giving $C_2 \approx 1279$ and $C_3 \approx 591$. For the needlet parameters used here we get $g_{2,1} = 1.0$ and $g_{3,1} = 0.707$. Inserting these values we get approximately

$$C_{kk'} \approx 511 P_2(\cos \Delta k) + 177 P_3(\cos \Delta k)$$

The covariance matrix has become a weighted sum of two Legendre polynomials. The Legendre polynomials are known functions, and the covariance becomes a weighted sum of these polynomials of relevant order. Looking at

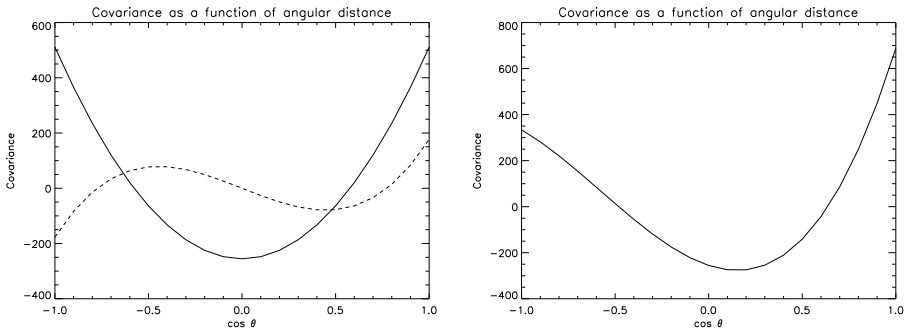

Figure 4.7: Left: The Legendre polynomials P_2(black) and P_3(dotted) plotted with the weighted values for scale $j = 1$. Right: The sum of the two polynomials.

figure 4.7 we see each of the weighted Legendre polynomials (left) with P_2 in black and P_3 as a dotted line. To the right we see the sum of the two. As expected the highest covariance is at $\cos\theta = 1.0$, which corresponds to angular distance zero, or the variance of the pixel. We also notice a fairly high correlation between points at the opposite side of the sphere. In fact, if we had a scale consisting of only $\ell = 2$ (we could get this by setting $B = 1.5$) the correlation on the opposite side of the sphere would be of the same shape as the plot of P_2 on the left side in figure 4.7. The correlation would be as high as the self correlation, and the needlet transformed CMB would be spherically symmetric.

Lets look at the case for scale $j = 2$. The multipole window now ranges from $\ell = 2$ to $\ell = 8$, so only $\ell = 3$ to $\ell = 7$ contribute. By inserting the relevant values for C_ℓ and $g_{\ell j}$, the expression for the covariance matrix now becomes

$$C_{kk'} \approx 165P_3 + 235P_4 + 157P_5 + 71P_6 + 15P_7$$

Again we can look up the expressions for these Legendre polynomials, mutliplying them with the weights, and add them together. The result is plotted in figure 4.8. The correlations on the opposite side of the sphere are now largely suppressed. Closer to our test pixel however, we do get significant negative correlation. Since this scale goes to $\ell = 8$ we should not use a lower resolution than $N_{\text{side}} = 4$. Using healpix we can find the typical distance between two neighbouring pixels at this scale. This turns out to be an angular

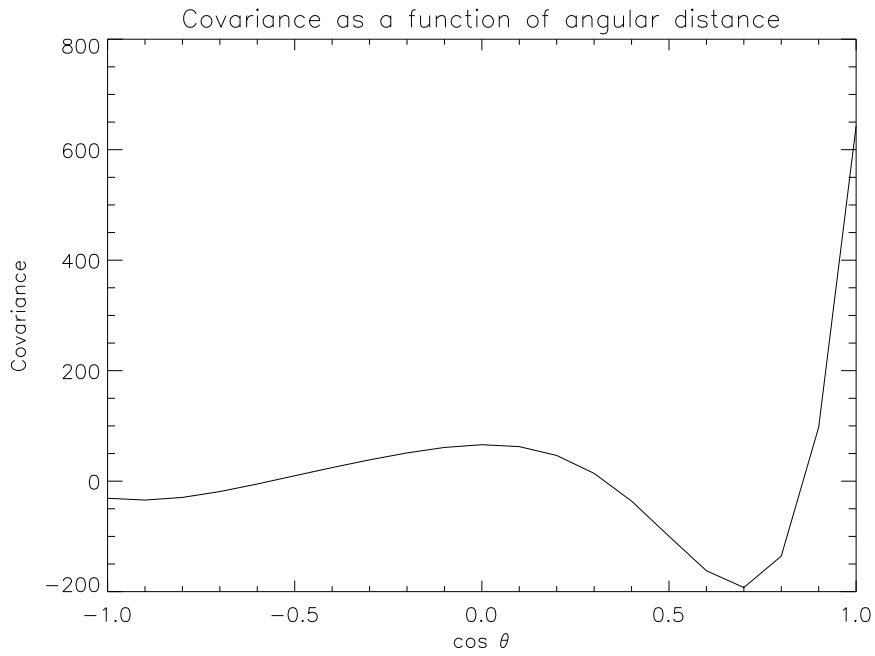

Figure 4.8: Sum of the weighted Legendre polynomials for scale $j = 2$

distance of $\cos\theta \approx 0.1$. The pixels closest to the testpixel would then be found at $\cos\theta \approx 0.9$ in the figure. At this point in the plot, the correlation is still about 30% of its maximum. In addition we have pixels at a distance $\cos\theta \approx 0.7$ which shows even larger, negative correlations.

There are two possibilities for getting a covariance matrix that really fullfills the expected $\delta_{kk}\delta_{j|j-1|}$ relation. We can reduce the resolution of the scales so the angular size of the pixels are larger than the variations close to $\cos\theta = 1.0$. The other way is to increase the bandwidth of each scale to include more multipoles.

For all multipoles the Legendre polynom will give the value 1 for $\cos\theta = 1$, which corresponds to self-correlation. At the opposite end, for $\cos\theta = -1$, the value will alternate between -1 and 1, depending upon whether ℓ is even or odd. The more polynomials we sum together, the more we will suppress all values except at $\cos\theta = 1$. We would presumably end up with a delta-function $\delta(\cos\theta - 1)$ when summing over an infinite number of multipoles. This would in our case correspond to a CMB image of infinite resolution, with a single scale j containing all the multipoles. A finite set of multipoles included in a scale will approximate this.

In the previous section we examined the demand for a minimum N_{side} to retain maximum information of the multipoles. Now we see that to get an exact covariance matrix fullfilling the $\delta_{kk'}$ relation we need to use lower resolutions. These two demands work in opposite, and for the example we used in this section they are incompatible. The general trend, based on earlier simulations [7] seems to be that the relation $\delta_{kk'}$ does not hold for the N_{side} we have to use given eq. 4.5.

The other option is to increase the bandwidth, and we will get back to this when we start using needlets with the Oh-Spergel-Hinshaw algorithm.

4.2 Estimating the power spectrum

We should now have some idea of the limitations when using the Oh-Spergel-Hinshaw algorithm in needlet space. Specifically we have to take into account the minimum resolution we have to use for a given scale.

The algorithm includes the option to use simplified versions of the covariance matrix. The intent is then to use these in the first iterations of the algorithm to make some rough steps in the right direction, then switch to exact matrices when greater precision is required for the final convergence. We will therefore examine how well the algorithm works with different approximations and with full matrices.

In each case we have to decide on the bandwidth of the needlet-window function, adjusted by the parameter B, what scales j we want to include, and what resolution $N_{\text{side,j}}$ we will use for each scale. This in turn decides what multipole ℓ_{max} we can expect to learn something about, given that any j-scale has an ℓ-range going from B^{j-1} to B^{j+1}. The largest ℓ in any given scale determines the minimum N_{side} we can use to make sure that the information for the power spectrum is retained. This is then given as $N_{\text{side,j}} = \frac{1}{2}B^{j+1}$ for any given scale. Whith all this in mind, we are now ready to make use of the algorithm.

In the following we are using HEALPix to generate a random set of spherical harmonic coefficients $a_{\ell m}$ based on the estimates of the power spectrum given by the WMAP results [2]. Any of these specific realizations will not necessarily give the same power spectrum back, they will only give a Gaussian distributed random set. We will therefore use eq. 2.39 to find the actual

power spectrum contained in any specific realization, and compare the estimated results with this. None of the estimates include the noise covariance, but all make use of the pixel window functions and beam as described by section 2.2.4. For the beam we have used the result from WMAP, with FWHM=7 arc minutes.

4.2.1 Diagonal covariance without CG-method

We will begin by examining simplified versions of the covariance matrix. One of the simplest assumptions we can make is

$$\langle \beta_{jk} \beta_{j'k'} \rangle = \delta_{jj'} \delta_{kk'} \tag{4.6}$$

where we are ignoring the correlations between neighboring scales. Unless we are able to find situations where this describes the true covariance matrix, this can only be a usable approximation in the initial iterations of the OSH-algorithm. For a proper estimate we will have to use a complete covariance matrix.

The above assumption leads to a diagonal covariance matrix $\mathbf{C}$, and we do not need any special methods to find the inverse, we may simply calculate it directly by inverting the values on the diagonal of $\mathbf{C}$. In addition the $\mathbf{P}^\ell$-matrix also becomes diagonal, and the calculation of both the first term and the trace in the algorithm becomes the simple operation of multiplying the diagonals of the various matrices together. Let's see where this simplification takes us.

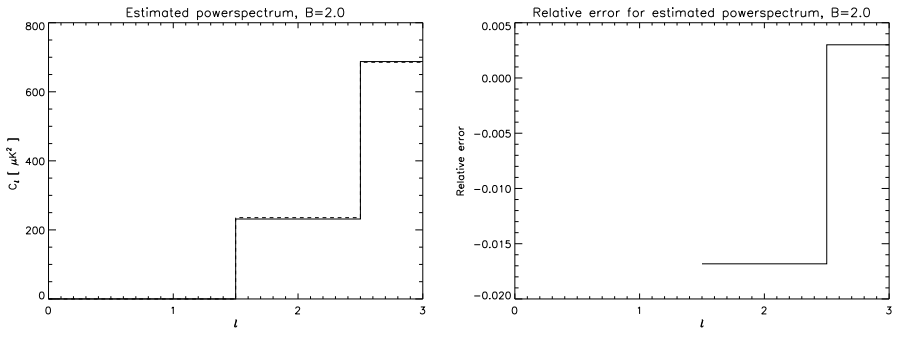

Figure 4.9: Left: Estimated power spectrum with $\ell_{\max} = 3$, $B = 2.0$. Right: Relative error for estimate vs. expected power spectrum.

Figure 4.9 shows the result when using $B = 2.0$, scales $j = 1$ to 2, and $\ell_{\max} = 3$. In this case the algorithm is extremely fast, and the whole estimation process is done in less than a second. The estimates converges quickly. After 3 iterations of the algorithm the estimates for the C_ℓ's has converged to the 11. decimal. The dotted line in the figure, which is hardly visible at all, represents the actual power spectrum given by using equation 2.39, and as we can see the estimates come quite close. The right figure shows the relative error, giving the error of the estimate of the quadrupole at about 1.7% of the actual signal. For the octopole the error is even smaller. These results seem promising, and it would be interesting to see the results when estimating a large number of universes.

By repeating the algorithm and finding the mean, we get the results shown in figure 4.10. Here we see the results of doing 1000 estimates. The results

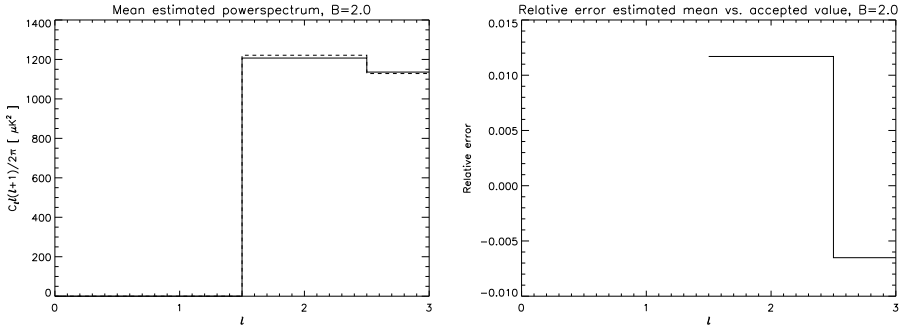

Figure 4.10: Left: Mean estimated power spectrum for $\ell_{\max} = 3$, $B = 2.0$. Right: Relative error for estimates vs. WMAP values.

are scaled and compared with the values from WMAP, and again we see very good agreement. The relative error is largest for the quadrupole, at 1.3%.

So far the algorithm seems to be working perfectly. Let's try to expand the estimates to include $\ell = 4$. Since scales $j = 1$ and 2 includes all we can know about $\ell = 2$ to 4 we do not need to change the setup from the previous estimate.

Table 4.4 shows results when trying to do the estimates up to $\ell_{\max} = 4$. Clearly something went very wrong here. Let's take a closer look at the details of the estimates. The algorithm basically consists of three distinct parts that need to be calculated. We are separately calculating the mc-term

Table 4.4: Expected and estimated values when using lmax=4, B=2.0

ℓ	expected	result
2	235.4	$9.9 \cdot 10^{118}$
3	685.5	$1.4 \cdot 10^{119}$
4	493.1	$5.5 \cdot 10^{118}$

and the trace term of the algorithm, and in addition we need to find the inverse Fisher matrix.

We will first look at the mc-term and the trace term. When using $B = 2.0$ and $\ell_{\max} = 3$ we got good estimates. Table 4.5 shows the development of the

Table 4.5: Development of the terms for $\ell_{\max} = 3$

Iteration	1	2	3
$\ell = 2$, mc-term	$-1.24 \cdot 10^{-2}$	$-8.01 \cdot 10^{-2}$	$-7.92 \cdot 10^{-2}$
$\ell = 2$, tr.term	$3.14 \cdot 10^{-2}$	$7.96 \cdot 10^{-2}$	$7.91 \cdot 10^{-2}$
$\ell = 3$, mc-term	-0.116	-0.134	-0.132
$\ell = 3$, tr.term	0.113	0.133	0.132

terms iteration for iteration for this case. We remember that the algorithm converged after three iterations, and from the table we can see why. The two terms have converged towards each other, cancelling each other out, and this means there will be almost no corrections to the C_ℓ's in further iterations.

Let's see what happens for $\ell_{\max} = 4$. Looking at table 4.6 we see a quite dif-

Table 4.6: Development of the terms for $\ell_{\max} = 4$

Iteration	1	2
$\ell = 2$, 1.term	$-1.24 \cdot 10^{-2}$	$1.57 \cdot 10^{-31}$
$\ell = 2$, 2.term	$3.14 \cdot 10^{-2}$	$-1.11 \cdot 10^{-16}$
$\ell = 3$, 1.term	$-6.40 \cdot 10^{-2}$	$-2.93 \cdot 10^{-31}$
$\ell = 3$, 2.term	$6.33 \cdot 10^{-2}$	$1.52 \cdot 10^{-016}$
$\ell = 4$, 1.term	0.130	$4.45 \cdot 10^{-31}$
$\ell = 4$, 2.term	0.100	$-1.85 \cdot 10^{-16}$

ferent development of the terms compared with the previous case. Although

all the terms start out with values comparable to the $\ell_{\max} = 3$ case, they are driven towards very small values in only one iteration. The results from table 4.4 showed that the estimates grew extremely large, and since the terms at first iteration seems to be comparable to the case for $\ell_{\max} = 3$, we should be suspicious about the inverse Fisher matrix.

Let's compare the Fisher matrices for $\ell_{\max} = 3$ and $\ell_{\max} = 4$ to see if we can explain the very different developments after only one iteration. Starting with the Fisher matrix for $\ell_{\max} = 3$ after one iteration we get:

$$\mathbf{F} = \begin{pmatrix} 3.52 \cdot 10^{-5} & 2.42 \cdot 10^{-5} \\ 2.42 \cdot 10^{-5} & 3.11 \cdot 10^{-4} \end{pmatrix}$$

Inverting this gives us the following matrix:

$$\mathbf{F}^{-1} = \begin{pmatrix} 29944 & -2333 \\ -2333 & 3392 \end{pmatrix}$$

Each of the elements here is of an order of magnitude that closely match the parenthetical terms in the algorithm, leading to small corrections of the initial guess.

In the case of $\ell_{\max} = 4$ we get something quite different, however. Looking first at the Fisher matrix, we have:

$$\mathbf{F}^{-1} = \begin{pmatrix} 3.52 \cdot 10{-}5 & 2.42 \cdot 10^{-5} & 0.0 \\ 2.420 \cdot 10^{-5} & 6.95 \cdot 10^{-5} & 1.32 \cdot 10^{-4} \\ 0.0 & 1.32 \cdot 10^{-4} & 3.33 \cdot 10^{-4} \end{pmatrix}$$

Nothing seems out of the ordinary here, and the terms seems to match the ones we got for $\ell_{\max} = 3$. When we invert however, the results become completely different:

$$\mathbf{F}^{-1} = \begin{pmatrix} -1.2 \cdot 10^{20} & 1.9 \cdot 10^{20} & -8.2 \cdot 10^{19} \\ 1.9 \cdot 10^{20} & -2.9 \cdot 10^{20} & 1.2 \cdot 10^{20} \\ -8.2 \cdot 10^{19} & 1.2 \cdot 10^{20} & -5.4 \cdot 10^{19} \end{pmatrix}$$

Here we see that the elements in the matrix do not match the terms at first iteration at all. Summing the parenthetical terms from table 4.6 together would at most lead to values approximately in the order of 10^{-1} to 10^{-2}. Multiplying this with the elements of the inverse Fisher matrix would lead to very large corrections of the power spectrum, and in the next iteration we would get even more inaccurate estimates. The estimates simply explode to very large powers, just like we saw in table 4.4.

The problem is the invertibility of the matrix. A simple way to test how the inversion went is finding the matrix product of the Fisher matrix and it's inverse, which should give the result $\mathbf{F}\mathbf{F}^{-1} = \mathbf{I}$. In the case for $\ell_{\max} = 3$ we get:

$$\mathbf{F}\mathbf{F}^{-1} = \begin{pmatrix} 1 & 1.11 \cdot 10^{-16} \\ 1.38 \cdot 10^{-17} & 1 \end{pmatrix}$$

The off-diagonal elements are not exactly equal to zero, but they are so small as to not create any problems in the algorithm. In other words, we got an approximate identity matrix.

For $\ell_{\max} = 4$ we get something completely different:

$$\mathbf{F}\mathbf{F}^{-1} = \begin{pmatrix} 1 & 0.69 & 0.00022 \\ 0 & 0.40 & 1.00013 \\ 6.77 \cdot 10^{-21} & -0.00013 & -0.00033 \end{pmatrix}$$

This result is very far from an identity matrix, and since our algorithm depends on the invertibility of the Fisher matrix, we can not expect to get any reliable results in this case. It seems the Fisher matrices in needlet-space are very unreliable.

The suspicion is strengthened by looking at two different attempts at estimation, each using the same data. One was programmed by me in Fortran, the other implementation was provided by my supervisor and programmed in IDL. For the case of using $\ell_{\max} = 3$ with scales $j = 1$ and 2, the two methods produced comparable results:

$$\begin{aligned} \text{Fortran}: & C_2 = 235.46, C_3 = 685.53 \\ \text{IDL}: & C_2 = 235.33, C_3 = 684.78 \end{aligned}$$

with the results of the inverted fisher-matrixes being

$$\text{Fortran}: \mathbf{F}^{-1} = \begin{pmatrix} 60.38 & -26.77 \\ -26.77 & 38.24 \end{pmatrix}, \text{IDL}: \mathbf{F}^{-1} = \begin{pmatrix} 60.26 & -26.71 \\ -26.71 & 38.16 \end{pmatrix}$$

As we can see the two methods produced approximately the same results. The differences comes from variations in the details of the code leading to slightly differend round-off errors.

An interesting thing happens whe we attempt estimating up to $\ell_{\max} = 7$. In this case the IDL code was still able to make an estimate of the power spectrum, while the Fortran code was not.

After one iteration however, the two methods produced almost the same result for the mc-term and the trace term in our algorithm. Even the Fisher matrix agreed up to 7. decimal in the two cases. But in the matrix inversion process, the IDL code produced matrices where each element was in the range of 10^7 to 10^{10}, while the Fortran code gave elements in the range of 10^{15} to 10^{18}. This huge difference in the inverse Fisher matrix is what is causing the problems, and the reason for it seems to be small differences in the rounding off in the different implementations. The inversion of the Fisher matrix is very unstable, and minor differences in the 8. decimal of the elements in the Fisher matrix decides if we are able to invert the matrix or not. The non-invertible Fisher matrix is what has caused our problems.

So far we have used $B = 2.0$ when constructing the needlet coefficients. As we have seen before, the B-parameter controls how many multipoles are included in each scale in the needlet transformations. Let's try setting $B = 1.5$, which leads to a smaller bandwidth and fewer multipoles included in each scale.

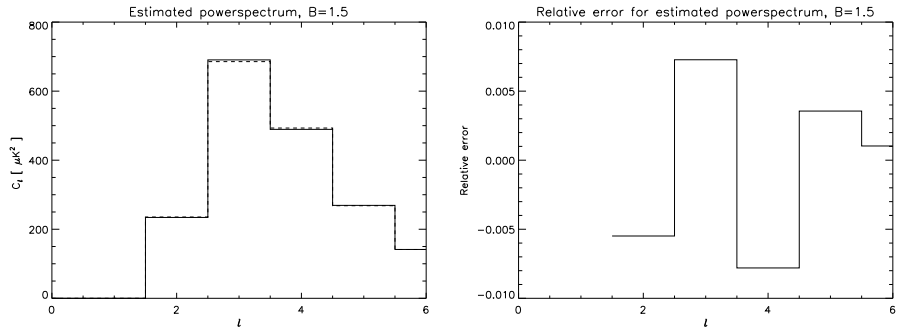

Figure 4.11: Left: Estimated power spectrum for $\ell_{\max} = 6$, $B = 2.0$. Right: Relative errors for estimate vs. expected value.

Figure 4.11 shows the result of trying this with $\ell_{\max} = 6$. In this case we need scales up to $j = 5$ to retain all information about the multipoles. Again the method converges quickly, beeing stable to the 11. decimal after 3 iterations. The relative error is less than 1%, as seen from the right plot. The Fisher matrix is invertible in this case, and producing an identity matrix when we do the matrix operation $\mathbf{F}\mathbf{F}^{-1}$.

In figure 4.12 we see a simulation with 1000 estimates compared with the accepted values for the power spectrum, and again we see good agreement with relative errors less than 3%. So by reducing the bandwith, leading to

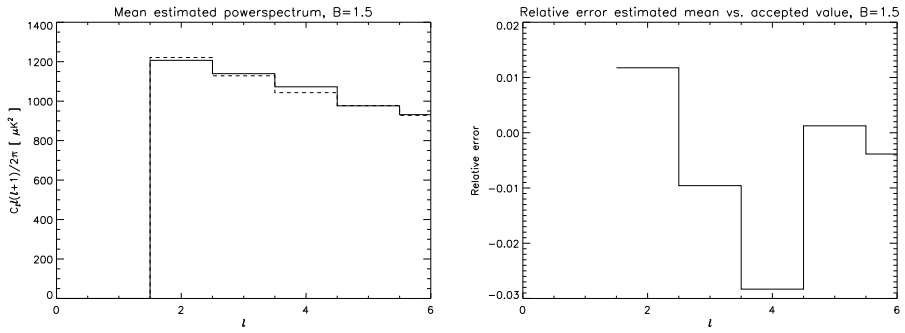

Figure 4.12: Left: Mean estimated power spectrum for $\ell_{\max} = 6$, $B = 2.0$. Right: Relative errors for estimate vs. WMAP values.

Table 4.7: Expected and estimated values when using lmax=7, B=1.5

ℓ	expected	result
2	235.4	234.1
3	685.5	690.4
4	493.1	$-4.4 \cdot 10^{112}$
5	268.1	$-1.4 \cdot 10^{117}$
6	141.3	$-1.8 \cdot 10^{117}$
7	88.7	$-5.1 \cdot 10^{116}$

more scales included, with fewer multipoles in each scale, we are able to geet good estimates for higher $\ell_{\max}$.

But again we hit a ceiling in the stability of the algorithm. Since $B^j \approx 7.5$ when using $B = 1.5$ and $j = 5$, we need only 5 scales if we want to make an estimate with $\ell_{\max} = 7$. In essence, we only have to include one more multipole when running the algorithm. Table 4.7 shows results after ten iteration when trying to estimate up to $\ell_{\max} = 7$, and we see the same problem we had earlier. In this particular case the estimates for C_2 and C_3 are still good, but again the inverted Fisher matrix has pushed the results for the higher multipoles in a very wrong direction. The matrix product $\mathbf{F}\mathbf{F}^{-1}$ does not produce an identity matrix, and the conclusion is that the Fisher matrix was non-invertible in this case.

Presumably it should be possible to get estimates to even higher multipoles by narrowing the bandwidth through the B-parameter. This does go against what we found in section 4.1.5 however. By including fewer multipoles in

each scale we are effectively getting further away from the the assumption made at the beginning of this section, and the diagonal case can not be a true covariance matrix. Even for doing initial estimates the algorithm fails, except for very low multipoles.

4.2.2 Tridiagonal covariance without CG-method

We will make use of the simplified matrix once more, so this will not be a preferred matrix for the final estimates in our algorithm. But just as for the diagonal case we will check if it could be useful in the first few steps.

The original expectation for needlet coefficients was given as $\langle \beta_{jk} \beta_{j'k'} \rangle = \delta_{kk'} \delta_{j|j'-1|}$, and we will try to implement this directly. By assuming no correlation between the needlet-coefficients, $\delta_{kk'}$, we have previously seen that we get a single number characterizing each scale. This means we can essentially construct a covariance matrix $\mathbf{C}$ with dimensionality given by the number of scales j we are using. This matrix should be easy to handle directly, making the calculations of the trace and Fisher matrix quite simple.

When calculating $\mathbf{C^{-1}m}$ we have to be to keep track of which element in $\mathbf{m}$ corresponds to which scale. For a simplified case with two scales, each scale having a resolution of only three pixels, we would have to make the following transformation of the matrix:

$$\left(\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right) \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \hline b_1 \\ b_2 \\ b_3 \end{pmatrix} \Rightarrow \left(\begin{array}{ccc|ccc} C_{11} & \cdot & \cdot & C_{12} & \cdot & \cdot \\ \cdot & C_{11} & \cdot & \cdot & C_{12} & \cdot \\ \cdot & \cdot & C_{11} & \cdot & \cdot & C_{12} \\ \hline C_{21} & \cdot & \cdot & C_{22} & \cdot & \cdot \\ \cdot & C_{21} & \cdot & \cdot & C_{22} & \cdot \\ \cdot & \cdot & C_{21} & \cdot & \cdot & C_{22} \end{array}\right) \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \hline b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

The same applies to $\mathbf{C}^{-1}$ and P^{ℓ}.

We begin by testing with parameters $B = 2.0$ and $\ell_{\max} = 6$. The results are shown in figure 4.13, and they have converged to 8. desimal after 3 iterations. This works fine with highest relative error at approximately 7% for the highest multipole. The Fisher matrix is invertible in this case.

In figure 4.14 we see the mean estimates after 1000 realizations. Again we see good agreement with accepted values for the power spectrum, with the largest relative error a little over 3%.

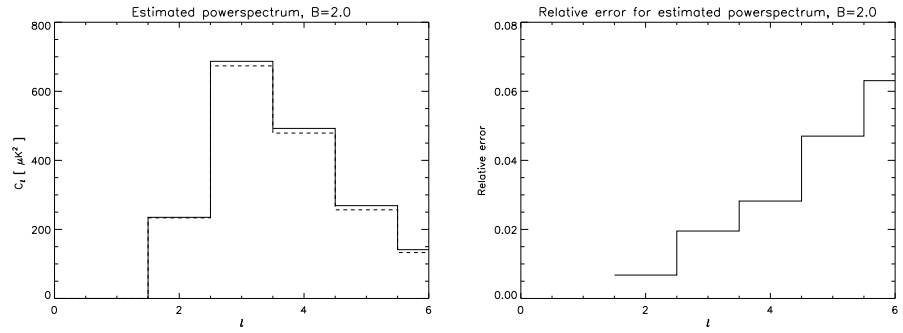

Figure 4.13: Left: Estimated power spectrum for $\ell_{\max} = 6$, $B = 2.0$. Right: Relative error for estimate vs. expected value.

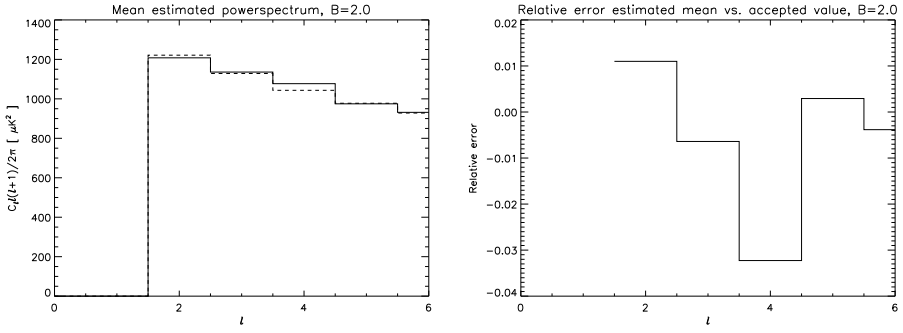

Figure 4.14: Left: Mean estimated power spectrum for $\ell_{\max} = 6$, $B = 2.0$. Right: Relative error for estimate vs. WMAP values.

But once again the algorithm fails when we try to do estimates of higher multipoles. Just as we saw in the diagonal case, the Fisher matrix is no longer invertible, and we get the by now familiar results of exploding estimates that do not converge.

Just as in the case of diagonal covariance matrices, we get slightly better results when we try to decrease the amount of multipoles in each scale. Figure 4.15 shows the result when using $B = 1.5$. In this case we are able to get results up to $\ell_{\max} = 9$. The relative error seems to be growing with larger ℓ, and ending up at about 14% for the highest multipoles, but looking at figure 4.16 we get a much better result. Here we see the mean after 1000 simulations compared with accepted values for the power spectrum. The largest relative error here is just above 3%.

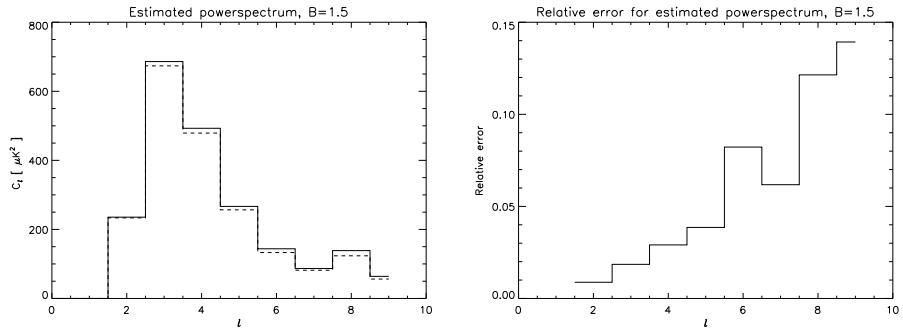

Figure 4.15: Left: Estimated power spectrum for $\ell_{\max} = 9$, $B = 1.5$. Right: Relative error for estimate vs. expected value.

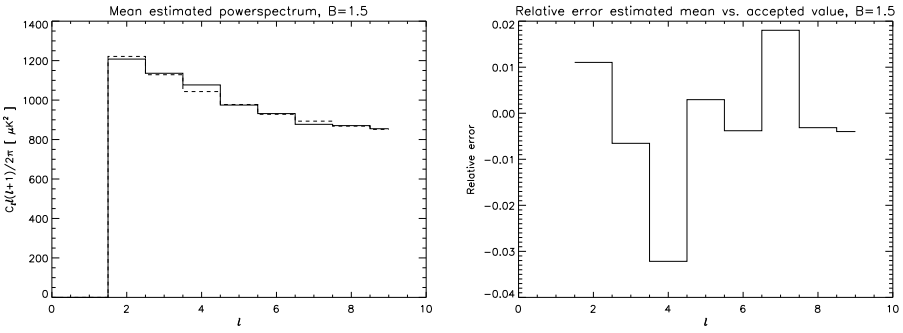

Figure 4.16: Left: Mean estimated power spectrum for $\ell_{\max} = 9$, $B = 1.5$. Right: Relative error for estimate vs. WMAP values.

A test with $\ell_{\max} = 1024$ has been tested to see if the solutions are more unstable when only dealing with a few multipoles, but we get the same problem as before. The estimates after only a few iterations quicly grow to exponential values, just as we saw in table 4.4, and the Fisher matrix is non-invertible.

4.2.3 Full covariance with CG-method

So far we have not made use of the Conjugate Gradient method in our algorithm, and we have not used the full analytical covariance matrix given by eq. 3.18. In this section we will look at the results for using these.

To get correct results when using the OSH-algorithm we will have to use correct covariance matrices, and we will then expect to get a correct result. Using the CG-method we will find an approximate invers covariance matrix, and Monte Carlo simulations will be used to calculate the trace and Fisher terms. The preconditioner used is the diagonal of $\mathbf{C}$.

As before we will use a bandwidth $B = 2.0$. We set $\ell_{\max} = 12$, and have to include 4 j-scales, with resolutions up to $N_{\text{side}} = 8$ for the last scale. The results of the estimation are shown in figure 4.17. The expected values of

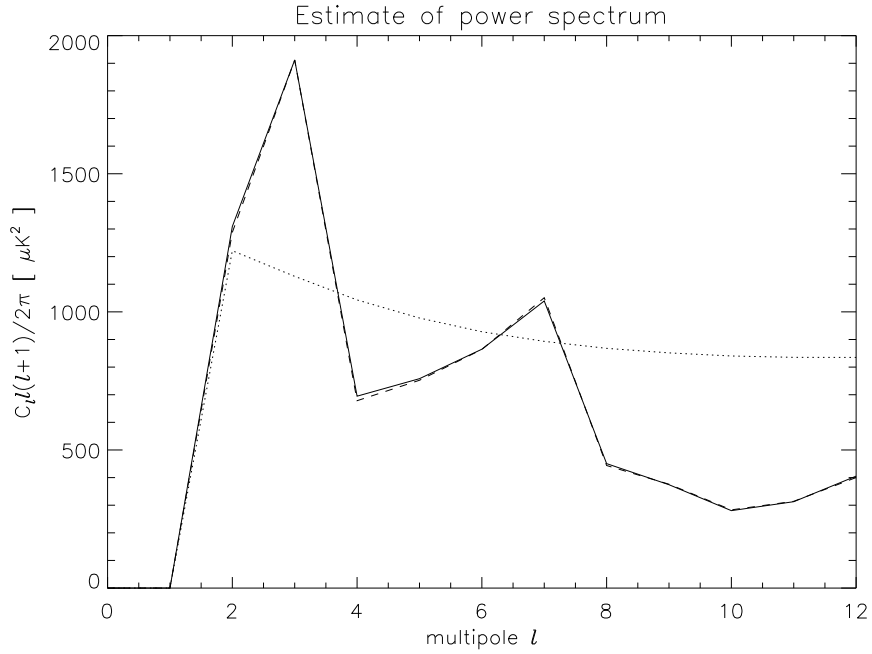

Figure 4.17: Results for $\ell_{\max} = 12$, $B = 2.0$. Full line shows estimate, broken line shows expected value, and dotted line shows initial guess.

the simulation are shown as a broken line, while the estimates are shown as a full line. The dotted line indicates our initial guess in the algorithm, and uses the values of the power spectrum estimated from WMAP.

As we can see from the figure the estimation is a success. The algorithm has converged after 3 to 4 iterations. In figure 4.18 we see the relative error for the estimate compared with the expected values of the power spectrum. As we can see the errors are small, in the range of $1 - 2\%$.

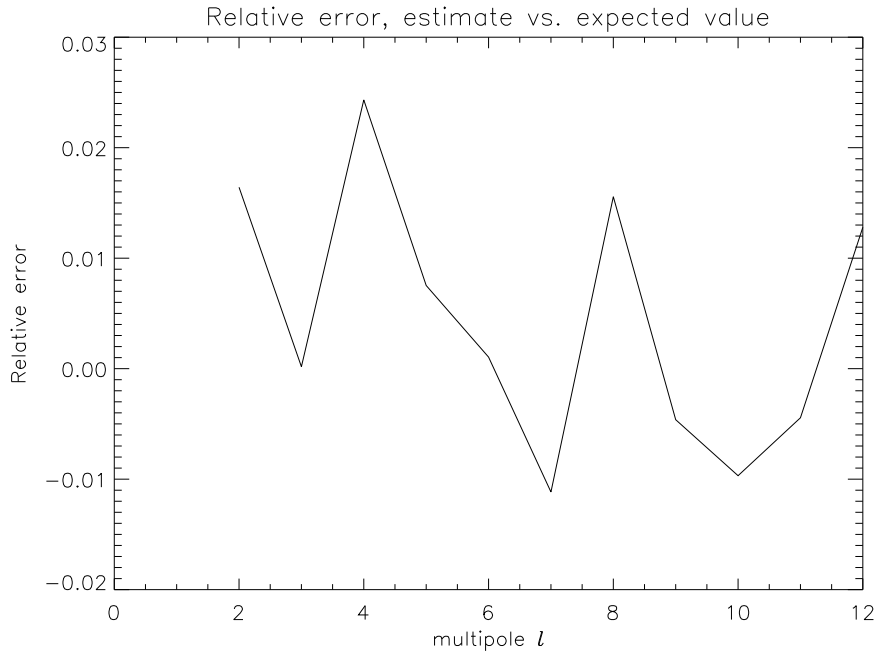

Figure 4.18: Relative error for expected vs.estimated power spectrum.

The results for $\ell_{\max} = 12$ represents the current best results obtained when using needlet transformations. This is not due to failure of the algorithm when trying to do higher estimates, but simply due to time constraints. When using the full matrix, the dimensions grows quickly when we include more multipoles and scales. This makes particularly the Monte Carlo simulations time consuming. It should not pose any problems to implement some sort of parallellization to the algorithm. We could then do many simulations at the same time, and get the results quicker.

4.2.4 Tridiagonal covariance with CG-method

As a final test we want to examine what happens if we use the previously described tridiagonal version of the covariance matrix and combine this with the Conjugate Gradient method and Monte Carlo simulations. Even if this can not be used as a final algorithm it may be useful for the first few iterations of the OSH algorithm.

We use a setup of the needlet transformations with $B = 2.0$, $\ell_{\max} = 4$, with

Table 4.8: Estimated values for $\ell_{\max} = 7, B = 1.5$, tridiagonal matrix.

Iteration	C_2	C_3	C_4
1	1279.2	633.3	317.7
2	1282.4	671.1	310.4
3	1285.5	703.7	303.5
30	1351.3	1021.3	239.1
100	1430.7	1114.7	238.9
Expected	1393.3	1068.4	237.6

Table 4.9: Estimated values for $\ell_{\max} = 7, B = 1.5$, full matrix.

Iteration	C_2	C_3	C_4
1	1446.2	865.5	241.9
2	1416.1	1015.9	244.2
3	1417.3	1097.2	243.7
Expected	1393.3	1068.4	237.6

two scales at $N_{\text{side}} = 2$. Results for a varying amount of iterations is shown in table 4.8. There are a few things to note in these results. First off, every iteration takes a very small step, and we can compare this with the results of the same setup using the full covariance matrix. The results for this is shown in table 4.9. The two algorithms use the same starting guess for the first iteration, and the algorithm using the full covariance matrix has taken much bigger steps toward the expected values of the power spectrum. The tridiagonal algorithm takes much smaller steps, and if we are going to use it to quickly iterate the first few steps in the algorithm it doesn't look like it is going to get us much closer to a better estimate.

The second thing to note from table 4.8 is that it's not clear if the algorithm is converging at all. Although the value for $\ell = 4$ seems to have stabilized after 100 iterations, the other two have continued past the expected values and does not show signs of converging to any value.

In conclusion the tridiagonal covariance matrix combined with CG-method and Monte Carlo simulations does not show great promise in getting quick estimates. It seems we are better of just using the full matrix in the first place.

4.3 Discussion of the results

We have now looked at several different version of combining needlet transformations with the OSH-algorithm. In this algorithm we are required to use exact covariance matrices to get exact results, but are allowed to use simplified versions to make some initial estimates. When using OSH in pixel space or spherical harmonic space, it was not necessary to have an exact Fisher matrix. From our results this seems to be very different in needlet space.

The two algorithms that made use of diagonal or tridiagonal matrices were only viable for very low ℓ. They were then able to make quite accurate estimates of the power spectrum, and were converging quickly. But they both come to a point where their Fisher matrices become non-invertible, and the algorithm fails. It seems there is need for a more exact Fisher matrix in these cases for the algorithm to work. This is very different from the original OSH-algorithm, where the demands on the Fisher matrix were not very strict. If the algorithm does not work at all it can certainly not be used for the first iterations.

We did have some success in reducing the bandwidth of each scale when using the simplified covariance matrixes. However, as we have already seen this directly contradicts the relation given by eq. 3.17, and the simplified covariance matrix will look less like the true covariance the fewer multipoles we include in each scale. In this thesis we have not explored the results of drastically reducing the bandwidth.

When using the Conjugate Gradient method to calculate $\mathbf{C^{-1}m}$ and indirectly finding the trace and Fisher, we had better results. Using the full covariance matrix produced good estimates, and with the limited amount of testing done it did not show any signs of failing. The tridiagonal matrix did not work particularly well with the CG-method however. It did not show clear signs of converging, and in our test the changes per iteration were to small for this method to be usable for first steps in the OSH-algorithm.

Finally we must conclude with the following: Using needlet transformations in combination with the OSH-algorithm for estimating the power spectrum does not make the estimation process faster or simpler. The only method that shows signs of being reliable is by using the full covariance matrix, and using this to get the trace and Fisher matrix through Monte Carlo simulations. Because of the relation between resolution and multipoles, we end up with a larger covariance matrix in needlet space than in pixel space. We will always

have at least one j-scale with the same resolution as the original signal, and depending on the bandwidth we use in our needlet transformations we get many other scales as well. As a consequence of this the size of the full covariance matrix will then increase instead of decrease, the computations will become heavier, and the motivation for using needlet transformations in the first place is lost.

Chapter 5

Conclusion

The formation of the Cosmic Microwave Background radiation has left us an imprint of the density fluctuations in the early universe. By using statistical tools we are able to characterize these fluctuations with a power spectrum, and this is of major importance when trying to establis the values and bounds of the cosmological parameters, enabling us to better understand the content and evolution of the universe.

The power spectrum describes the variations in the CMB on different angular scales, known as multipoles. The better resolution we have in our data, the higher multipoles we are able to get. As the size of the CMB-data increases the pressure to find effective algorithms to extract the power spectrum grows as well.

Oh, Spergel and Hinshaw has developed an algorithm for extracting the power spectrum from the observed CMB. An important characteristic of the CMB in this algorithm is the covariance matrix, a description of the statistical connections between different parts of the signal. With increased resolution of the CMB observations the dimensions of the covariance matrix grows as well, and this eventually leads to computational challenges. In this thesis we have examined the possibility of using needlet transformations to reduce the complexity of the covariance matrix.

Needlet transformations are able to pick out smaller ranges of multipoles and arrange them in a new way. They create a set of needlet coefficients β_{jk} with a pixel number k and a j-scale number, and these contain the information from the included multipoles. The attraction of using needlets lies in the relation

$\langle \beta_{jk} \beta_{j'k'} \rangle = \delta_{kk'} \delta_{j|j'-1|}$. This relation tells us that the needlet coefficients we get from the transformation are orthogonal to each other, and only correlate with a few neighbouring j-scales. If this relation holds for use in the OSH-algorithm, it will allow us to simplify the covariance matrix. This wil make it possible to have very fast algorithms for estimating the power spectrum.

In the thesis we have examined the properties of needlets combined with the OSH-algorithm, and what information we can expect to get under different conditions. Some problems with the approach have been indicated. The situation can be described as shown in figure 5.1. We have seen the the need for a high enough resolution of the CMB image to be able to extract a certain detail in the multipoles. To be able to get tridiagonal covariance

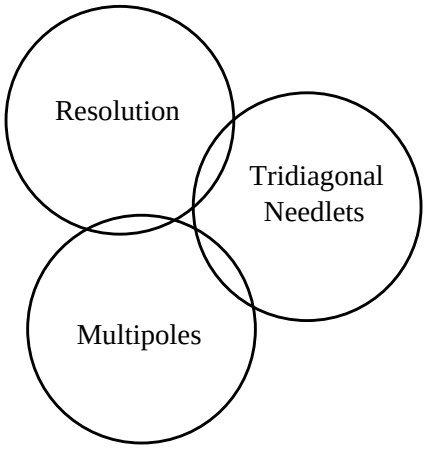

Figure 5.1: Non-overlapping solutions

matrices we have seen two possible solutions; either using lower resolutions for the images, or using more multipoles in each of the j-scales. As indicated on the figure these solutions do not overlap, and because of this we are not able to get good estimates of the power spectrum when we use simplified covariance matrices. The solutions we want for the power spectrum lies in the cross-section between resolution and multipoles, and we are unable to make use of tridiagonal needlets in this case.

The needlet transformations do work with the OSH-algorithm however. We get converging solutions when we make use of more complex needlet transformed covariance matrices. The problem with using this is that we end up with larger matrices than we started out with before the transformations. The estimation becomes more computationally heavy, not less, and this is

not any solution to our initial problem.

When working with this thesis, a lot of work was done in trying to get the algorithm to produce results before realizing the needlets themselves were causing the problems. Therefore, many possibilities have gone untested. What would happen to the invertibility of the Fisher matrix if we included the instrumental noise? How would the estimates have looked if we included the galaxy mask to remove unwanted signal? Is it possible to include a large amount of multipoles in a few j-scales and still use a fairly low resolution, to obtain triangular matrices?

These questions remain unanswered in this thesis, and some of them may provide a solution to simplifying the CMB signal for use with the OSH-algorithm. From the results found in this thesis however, it seems that the use of needlet transformations combined with the OSH-algorithm is not a viable approach to estimating the power spectrum.

Bibliography

[1] P. Baldi, G. Kerkyacharian, D. Marinucci, and D. Picard. Asymptotics for spherical needlets. *Annals of Statistics*, page 1150, 2009.

[2] C. Bennett, R.S. Hill, G. Hishaw, M.R. Nolta, N. Odegard, L. Page, D.N. Spergel, J.L. Weiland, E.L. Wright, M. Halpern, N. Jarosik, A. Kogut, M. Limon, S.S. Meyer, G.S. Tucker, and E. Wollack. First year wilkinson microwave anisotropy probe (wmap) observations: Preliminary maps and basic results. *Astrophys.J.Suppl.148:1*, 2003.

[3] Scott Dodelson. *Modern Cosmology*. Academic Press, 2003.

[4] Oeystein Elgaroey. Ast4220: Cosmology 1. Lecture notes.

[5] Krzysztof M. Gorski, Benjamin D. Wandelt, Frode K. Hansen, and Anthony J. Banday. The healpix primer. *eprint arXiv:astro-ph/9905275*, 1999.

[6] Frode K. Hansen. Healpix tutorial. Primer on Healpix and analysis of CMB.

[7] Tony A. Ingebrigtsen. Testing needlets. Results from summer job 2009 at ITA, UiO.

[8] D. Marinucci, D. Pietrobon, A. Balbi, P. Baldi, P. Cabella, G. Kerkyacharian, P. Natoli, D. Picard, and N. Vittorio. Spherical needlets for cosmic microwave background data analysis. *Monthly Notices of the Royal Astronomical Society*, page 539, 2008.

[9] Siang Peng Oh, David N. Spergel, and Gary Hinshaw. An efficient technique to determine the power spectrum from cosmic microwave background sky maps. *The Astrophysical Journal*, page 551, 1998.

[10] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes*. Cambridge University Press, 2007.

[11] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. School of Computer Science, Carnegie Mellon University, 1994.

[12] Wikipedia. http://en.wikipedia.org/wiki/Jacobi's_formula.