Hemispherical Power Asymmetry in the Cosmic Microwave Background by Gibbs Sampling

Dag Sverre Seljebotn

Master's degree in Computational Science

Faculty of Mathematics and Natural Sciences, University of Oslo

September 2010

Abstract

The current cosmological concordance model states that the fluctuations in the Cosmic Microwave Background (CMB) should be Gaussian and isotropic. However, many studies claim to have found small deviations from this theory. One such deviation is the apparent existence of hemispherical power asymmetry: One hemisphere of the CMB appears to contain stronger fluctuations than the other. As of yet, it is not clear whether this is a statistical fluke, a systematic effect, or a genuine violation of the cosmological principle.

Current studies are either limited to studying structures on large scales due to the poor computational scaling of $O({N_{\mathrm{pix}}}^3)$ (Hoftuft et al., 2009, Eriksen et al., 2007, Gordon, 2007), make use of approximate methods (Hanson & Lewis, 2009, Bennett et al., 2010), or focus on non-parametric statistics (Hansen et al., 2009).

A computationally efficient method for fully exact, Bayesian analysis of the hemispherical CMB power asymmetry has been developed in this thesis, based on the CMB Gibbs sampling algorithm (Wandelt et al., 2004, Jewell et al., 2004). With a computational scaling of $O({N_{\mathrm{pix}}}^{3/2})$, the method is able to explore current and future CMB observations at full resolution. Probing for the presence of asymmetry at $\ell \geq 1000$ is fully realistic for the upcoming Planck data. In this thesis, a dipole modulation field model gets particular attention. However, the computational foundation is also laid for exploring more general models than what has previously been possible. Models with arbitrary azimuthally symmetric modulation fields or scale-dependent modulation strength can be fitted to data at the same computational cost.

The complete algorithm has been implemented from scratch in Python and thoroughly tested on simulations. A direct comparison is made with the results of Hoftuft et al. on low resolution data. Some preliminary results of analysis of full resolution WMAP 7-year data are also presented. The preliminary findings are consistent with the earlier studies, indicating the presence of asymmetry on scales up to $\ell = 600$. In particular, the preferred direction is consistent with earlier results.

The code is believed to be ready for a more thorough study of WMAP data, although a few final checks are outlined that should be carried out first. As a direct consequence of reviewing the CMB Gibbs sampling algorithm in detail, a couple of minor flaws were found in the existing CMB Gibbs sampler Commander.

Preface

Doing a Master's project in computational cosmology has been like entering a strange land. The notation is all messed up, scalars come with "units" tacked on to them, and people find Fortran an adequate programming language. Still, after roughly one and a half year, I can wholeheartedly say that I have enjoyed the tour. Cosmology turns out to be field where incredibly diverse statistical and computational techniques are combined into scientific results.

This transition would never have happened without Hans Kristian Eriksen. We talked about a project in the third year of my B. Sc., at a time when I was thoroughly demotivated. The promise of a challenging applied project at the end of the tunnel was just the motivation I needed at the time. Thanks for finding just the right project, for filling me in over and over again, for helping me to stay focused, and for putting so much time and effort into all your students in general. Also, thanks for bearing over with my cub-like enthusiasm when I show you wrong in unimportant details, and for ignoring all my snide remarks about Fortran. I will learn to restrain myself some day.

My main side-project and source of procrastination over these years has been scientific Python in general, and Cython in particular. It is safe to say that I have learned more about programming and scientific computing from reading `cython-dev` and `numpy-discuss` than from any course at my university. I want in particular to thank my Cython and Fwrap peers for all they have taught me, and for many interesting and/or pointless discussions: Robert Bradshaw (who has had a knack of putting features into Cython just as I needed them for this thesis), Stefan Behnel, Lisandro Dalcin, and Kurt Smith. Also, thanks to Nathaniel Smith for announcing the Python interface to CHOLMOD on about the same day that I needed it. Without all of you, the code for this thesis might have been written in an inferior language.

Looking back on five years of studies, I want in particular to thank all those I have met in the Natural Sciences group in Oslo Student Christian Fellowship for all the good times.

Finally, thanks a lot to my wife Åshild for all your support. I am really looking forward to spending more time with you and our daughter Astrid, whom I have barely seen in the past couple of weeks. I love you both very much.

Dag Sverre Seljebotn
Blindern, September 30, 2010

Contents

Chapter 1

Overview

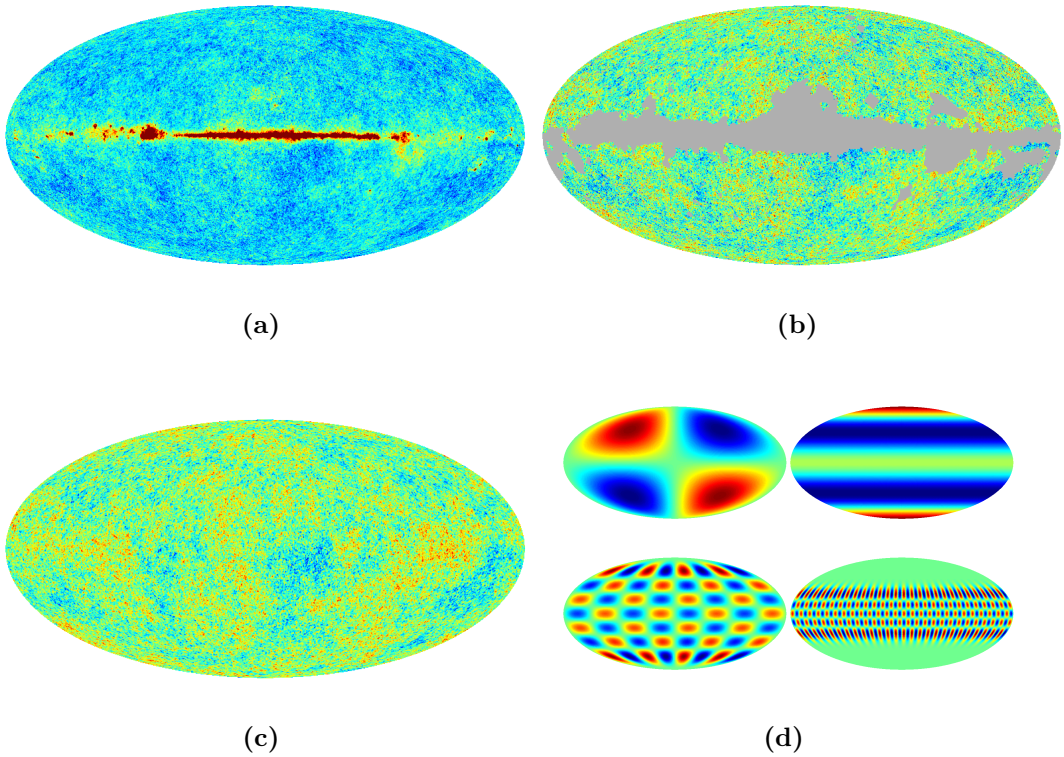
1.1 The Cosmic Microwave Background

Cosmology is the study of our universe on vast scales. In a sense, the universe is the largest physics laboratory one can ever hope for. Modern cosmology relies on both particle physics, quantum mechanics and General Relativity. By modelling the evolution of the universe, from the Big Bang and until today, our current knowledge of physics is put to the test.

In one sense, modern cosmology is an astounding success. With just six parameters, the established concordance model, dubbed the ΛCDM model, is able to fit millions of data points. In another sense, we still understand little. The ΛCDM model ("Dark energy (Λ) and Cold Dark Matter") requires that approximately a quarter of the energy content of the universe is something we know very little about (Cold Dark Matter). Most of the remaining three quarters we have even less of an idea about (dark energy). Less than 5% of the energy in the universe is the ordinary atoms and photons that we can observe (Dodelson, 2003).

What sources of data do these claims rely on? The most important one is the Cosmic Microwave Background (CMB). When looking out in the universe, we look further and further out, and further and further back in time, until we observe photons coming from approximately 300 000 years after the Big Bang, or 13.7 billion years ago. At that point, one can not see any further, because the universe at that time was a very hot dense fog. This fireball today takes the form of a shell around us, 50 billion light years away. It is remarkably uniform. Regardless of the direction in which we point our instruments, we observe a perfect black body spectrum at 2.725 Kelvin. Still, it does contain tiny fluctuations, well under a millikelvin in temperature. These fluctuations have been measured to high accuracy by the Wilkinson Microwave Anisotropy Probe (WMAP) (Jarosik et al., 2010). Much higher resolution data is soon to come from the ongoing Planck experiment (The Planck Collaboration, 2006).

For some time, cosmology was a field with more speculations than data. However, over the last couple of decades there has been an explosion in the



(a) **(b)**

(c) **(d)**

Figure 1.1: Anisotropies in the Cosmic Microwave Background, as observed in the V frequency band by WMAP (Jarosik et al., 2010). **(a)** The CMB is remarkably uniform, and the radiation of our own galaxy in the same frequency band is much stronger. **(b)** For analysis, our galaxy and point sources are masked out. Red is warmer and blue is colder than 2.725 K by about 0.5 mK. **(c)** Constrained realization. Using the methods of chapter 3, one can draw samples from the Bayesian posterior distribution of the underlying CMB signal, given data, instrument properties and an assumed cosmological model. **(d)** The signal is often represented as a sum of spherical harmonic basis functions $Y_{\ell m}(\hat{n})$. Plotted here are the real parts of $Y_{2,1}(\hat{n})$, $Y_{4,4}(\hat{n})$, $Y_{10,6}(\hat{n})$, and $Y_{30,4}(\hat{n})$. Higher ℓ corresponds to more waves.

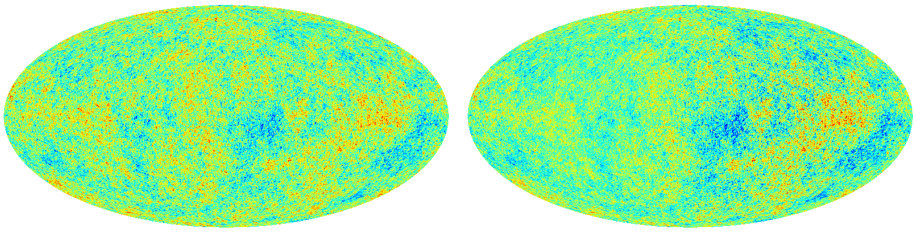


Figure 1.2: The effect of dipole-modulation. On the left is an isotropic signal $f(\hat{n})$, while on the right is $(1 + \alpha\hat{p} \cdot \hat{n})f(\hat{n})$. The effect is to induce stronger fluctuations in one hemisphere and smooth the fluctuations in the opposite hemisphere. In this case, $\alpha = 0.3$, and $\hat{p}$ is to the right on the map.

amounts of available data, such as CMB observations, galaxy surveys, and gravitational lensing observations. It is no longer lack of data that is the bottleneck, but rather the computational challenges.

1.2 Spherical harmonics and the power spectrum

A very central technique in cosmology is the spherical harmonic expansion. It is the analogue to a Fourier transform on the sphere. A field f on the sphere can be expanded into spherical harmonic coefficients $a_{\ell m}$,

$$f(\hat{n}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\hat{n}).$$

What makes this transform so useful is that it separates the signal into different scales. An average of the entire map is represented by a_{00}, the dipole component by $(a_{1-1}, a_{10}, a_{11})$, the $\ell = 2$-coefficients specify a quadrupole (two waves around the equator), and so on. As ℓ gets higher, smaller and smaller scales are characterized.

We now let f above be the perfect CMB signal, and consider the properties of its spherical harmonic coefficients. In an isotropic universe, there should in a statistical sense be nothing special about any particular direction. Therefore, only the scale should matter, and for a given ℓ the $a_{\ell m}$'s should have the same statistical properties for all m. Furthermore, standard cosmological theory predicts that the $a_{\ell m}$'s are Gaussian, and that all the $a_{\ell m}$'s are statistically independent[1]. Therefore the temperature part of the signal is, in a statistical sense, perfectly described by the *power spectrum*

$$C_\ell \equiv \mathrm{Var}(a_{\ell m}).$$

This is where observation and theory gets linked. While our particular universe is assumed to be "random", cosmological theory makes very definite predictions about the exact shape of the power spectrum, i.e., how much variance there should be on each scale (see figure 1.3).

Note that for each ℓ, the CMB signal has $2\ell+1$ data points. Since we only have one universe to observe (and one position to observe it from), this is all we are ever going to get, and it sets an inherent limit to how well it is possible to estimate C_ℓ from data. This is known as cosmic variance. In generalizing to anisotropic models (as this thesis does), one path that is clearly infeasible is to model each $a_{\ell m}$ independently, as we only have one observation.

[1] Apart from the fact that $a_{\ell -m} = (-1)^m a^*_{\ell m}$. However, all coefficients but $a_{\ell 0}$ are complex with independent real and imaginary part, so there are $2\ell + 1$ independent data points per ℓ.

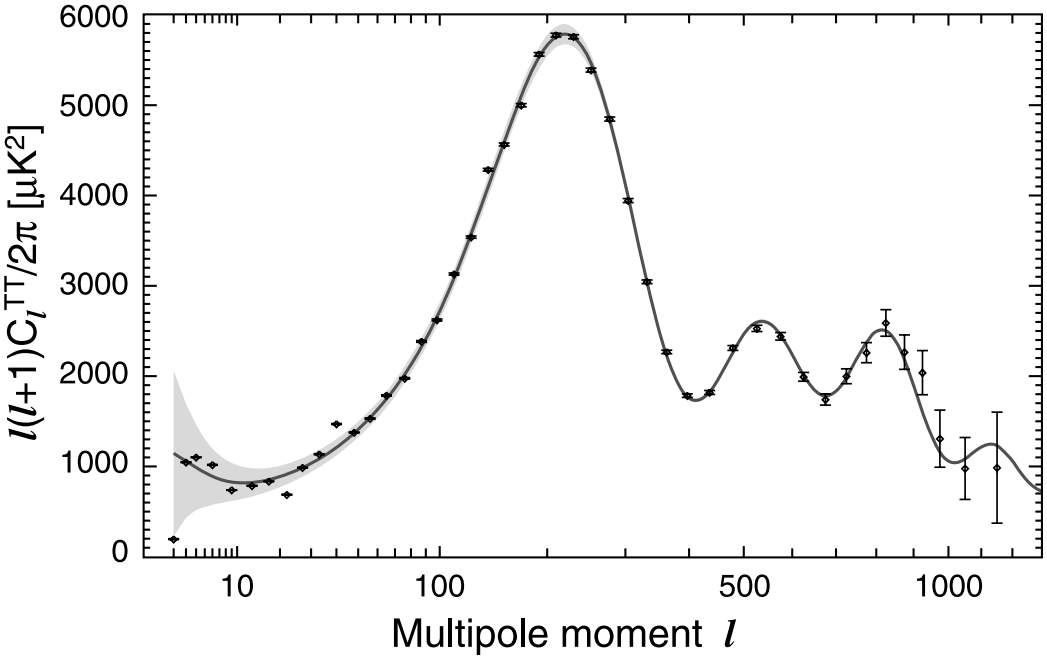


Figure 1.3: The current best fit power spectrum as reported by WMAP. The dots and error bars show the estimated power spectrum from the 7 year WMAP CMB measurements. The solid line is the power spectrum predicted by the best fit ΛCDM model. The gray band represents cosmic variance. The power spectrum is roughly proportional to ℓ^{-2}, so it is conventional to scale the power spectrum by $\ell(\ell+1)/2\pi$ when plotting. Image courtesy of Larson et al. (2010).

The power spectrum is used to fit the parameters of our universe to great accuracy. Theoretical cosmological models each predict a power spectrum, and stands or falls with how well that power spectrum match observation. For instance, Ω_m, the proportion of energy in the universe made up of matter (including dark matter), shifts the first peak up or down, while Ω_b, the proportion of energy that is ordinary matter, scales the second peak relative to the first and third peak (Dodelson, 2003).

1.3 Hemispherical power asymmetry

Is the universe isotropic? That is, does it have the same statistical properties in all directions? This is one of the fundamental assumptions of cosmology, and so far there has been no decisive reason to believe otherwise. With increasing amounts of data do however come an ability to ask more detailed questions. Is there, perhaps, a tiny amount of anisotropy present that can still not be attributed to chance or observational errors?

The question of cosmological anisotropy is currently an active field of research. One of the most intriguing questions is the one that concerns this thesis, namely that the CMB fluctuations appear to be stronger in one hemi-

sphere than the other. In the following, we briefly review the literature to date concerning this question.

The effect was first reported by Eriksen et al. (2004a), Hansen et al. (2004) and Eriksen et al. (2005), who used several complementary analyses on WMAP 1-year data as well as COBE data. First, the power spectrum was computed locally for many small patches. Second, the sphere was split in half (using many different directions) and a power spectrum estimated for each hemisphere. In both cases, more power (stronger fluctuations) was found in one particular direction on the sky. The results were confirmed to be significant at the 95%-level or higher by Monte Carlo simulations.

These introductory studies were followed by model-based parametric studies that postulated the following phenomenological model. Suppose that a Gaussian and isotropic CMB signal is modulated (multiplied point-wise) by a dipole field (see figure 1.2). The dipole modulation serves to suppress fluctuations on one hemisphere, and amplify them on the other, with a smooth transition in-between. One can then estimate the parameters of this field; an amplitude α and a preferred direction $\hat{p}$. This model was first fit to data by Gordon (2007) and Eriksen et al. (2007), both using the Metropolis algorithm to sample from the Bayesian posterior distribution. The latter found a best fit $\alpha = 0.114$ at the 99% significance level in the WMAP 3-year data, when including multipoles up to $\ell_{\mathrm{mod}} = 40$. The analysis was repeated at higher resolution by Hoftuft et al. (2009), who found the best fit in the WMAP 5-year data (V band) to be $\alpha = 0.08 \pm 0.021$ for $\ell_{\mathrm{mod}} = 64$, and $\alpha = 0.07 \pm 0.019$ for $\ell_{\mathrm{mod}} = 80$. Uncertainties indicate one standard deviation in the Bayesian posterior, and correspond to $3.8\,\sigma$ and $3.7\,\sigma$ detections, respectively. Due to the computational scaling of $O({N_{\mathrm{pix}}}^3) = O({\ell_{\mathrm{mod}}}^6)$, it has been impossible to extend these exact analyses to higher resolutions. Hoftuft et al. (2009) note that their computations required about 50 000 CPU hours, and that increasing ℓ_{mod} further would require quadrupling the number of pixels, at a computational cost of about 3 million CPU hours.

Hanson & Lewis (2009) developed an approximate, quadratic maximum likelihood (QML) estimator in order to fit a set of anisotropic models to WMAP 5-year data, including the dipole-modulation model. By applying their estimator to both WMAP data and a set of isotropic simulations, they find the same effect, although at lower significance (see figure 1.5). They note that the significance seem to fluctuate depending on how much data is taken into account, and that the previously studied values of ℓ_{mod} yielded higher significance than some other choices. They also found that the effect diminish at higher scales. The WMAP team (Bennett et al., 2010) repeated this analysis on 7-year WMAP data with similar results. They claim that *all* findings of cosmological anisotropy to date are solely an effect of *a posteriori* bias: If one tries too many weird estimators, some of them are bound to result in spurious significant results.

In contrast to these model-based studies, Hansen et al. (2009) use the

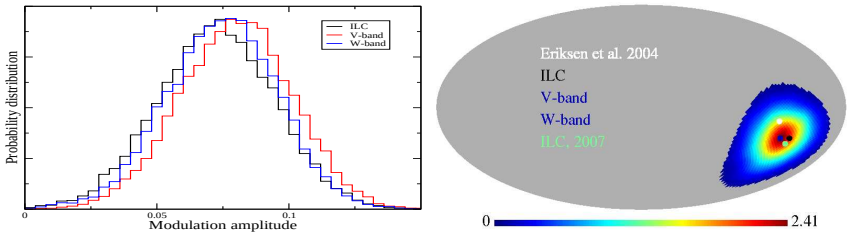


Figure 1.4: Figures from Hoftuft et al. (2009), the most recent exact model-based analysis. **Left:** Posterior distribution $p(\alpha|\mathbf{d})$, including data up to $\ell_{\rm mod} = 64$. **Right:** Estimates of the direction of strongest fluctuations, $\hat{p}$. Other studies on hemispherical power asymmetry all claim directions consistent with these.

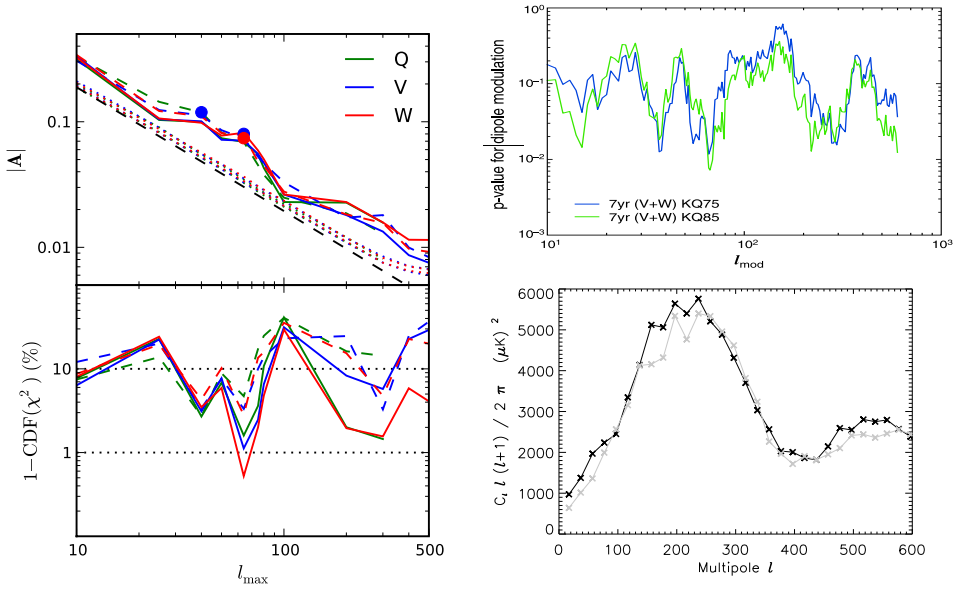


Figure 1.5: Recent results on hemispherical power asymmetry. All figures are taken from the papers cited. **Left:** The results of Hanson & Lewis (2009) of fitting the dipole-modulation field model to the WMAP 5-year data. Their $|\mathbf{A}|$ corresponds to our α, and the different values of $\ell_{\rm max}$ indicate that scales above this value were not included (that is, were assumed to be isotropic). In the panel below, significance based on simulating many isotropic maps is given. **Upper right:** The p-values for dipole modulation as given by Bennett et al. (2010), computed by applying the methods of Hanson & Lewis (2009) on WMAP 7-year data. **Lower right:** The power spectrum estimated separately on a 90° discs in opposing hemispheres. The direction is chosen to maximize power asymmetry (Hansen et al., 2009).

following non-parametric approach. For each of 3072 directions, the (binned) power spectrum is computed for the hemisphere centered in that direction. The result is a particular kind of "power maps" that clearly show more power in one side of the sky than the other, in a direction consistent with the earlier findings. In particular, the maps clearly show power asymmetry as high as $\ell = 502 - 601$, independent of lower ℓ's. As earlier mentioned, a claim made by an isotropic universe model is that the data should be statistically independent between different ℓ's. Therefore, the (spurious) preferred direction found at different ℓ's should have been completely random. To establish the significance of the results, the first order (dipole component) of the produced power maps was used as a statistic, and compared with simulations. The significance is found to vary with range of ℓ's, the mask used, and the parameters of the estimator. In some cases the result is too significant to be reliably estimated with the number of simulations used (p-value less than 0.01), although a number of cases have p-values as high as 0.05–0.10, or higher.

The parametric analyses have to date either been restricted to only include data to some particular $\ell_{\rm mod}$ due to computational cost, or used approximate methods. The non-parametric analysis does give a strong indication that hemispherical power asymmetry is present on higher ℓ's, but the sensitivity of the statistic chosen appears to be unclear. Further study is therefore needed. The aim of this thesis is to develop an algorithm based on the *CMB Gibbs sampler* that can be used to fit the dipole-modulation model to data. The resulting algorithm is significantly faster than the previous exact and model-based analysis used by Hoftuft et al. (2009), and makes it possible to both check the claims of Hanson & Lewis (2009) and Bennett et al. (2010) in detail, and to generalize to a wider class of parametric hemispherical power asymmetry models. The algorithm will also scale well to the full resolution of the Planck experiment. While a fit to WMAP data is important, it is Planck that will provide the ultimate measurements of the CMB. Whatever the findings, the jury is still out until the model is also tested against the Planck data.

1.4 The Gibbs sampling framework

The CMB does not reach us in pristine condition. Many sources of noise contaminate it, and these must be dealt with in order to do CMB analysis. The Gibbs sampling algorithm, introduced to the CMB research community by Jewell et al. (2004) and Wandelt et al. (2004), provides an elegant and efficient way of dealing with all sources of signal contamination. Most applications so far have focused on reliable estimation of the power spectrum, ΛCDM parameters, and foreground components in the case of an anisotropic universe (e.g., Eriksen et al., 2004b, 2008). However, Groeneboom & Eriksen (2009) use the Gibbs sampler to estimate the parameters of a particular anisotropic effect, using an approach that is very similar in nature to the one

we will develop.

A simplified version of the model used for data analysis is

$$\mathbf{d} = \mathbf{s} + \mathbf{n},$$

where $\mathbf{d}$ is the data observed by the instrument, $\mathbf{s}$ is the CMB signal that we are interested in, and $\mathbf{n}$ is the instrumental noise. The full model is presented in chapter 3. For various good reasons, we will consider all of these to be multi-dimensional Gaussian vectors.

The data sets are huge: In the case of WMAP there are about 3.1 million pixels of data, while the ongoing Planck mission will provide about 50 million pixels. Performing computations that scale linearly with available data is not a problem for such data sets. However, many computations needed for statistics are cubic in the size of the data. The full covariance matrix of $\mathbf{d}$ alone would be 10 petabytes if ever computed. Factoring such a matrix, even once, is clearly out of the question. For this reason, brute-force approaches are not suitable. It does however turn out that some basis changes can be made to make the problem tractable. First off, we will assume that the noise, $\mathbf{n}$, is uncorrelated between pixels, so that its covariance matrix $\mathbf{N}$ is diagonal. And, assuming an isotropic universe, each spherical harmonic coefficient of the CMB signal $\mathbf{s}$ is statistically independent of other coefficients with a given variance C_ℓ, producing a diagonal covariance matrix $\mathbf{S}$. For anisotropic universe models, correlations are introduced, although in our case $\mathbf{S}$ will still have a sparse structure.

It must be noted that the covariance matrix $\mathrm{Var}(\mathbf{d}) = \mathbf{S} + \mathbf{N}$ is still dense in either basis. This is where the Gibbs sampling algorithm comes in. Assume that the cosmological model is parametrised through a parameter vector θ, and that we want to draw samples from the Bayesian posterior distribution $p(\theta|\mathbf{d})$. The Gibbs sampling approach is then to sample from the joint posterior distribution $p(\theta, \mathbf{s}|\mathbf{d})$, by alternating between sampling from the conditional posteriors. First, $\theta^{(0)}$ is initialized to some arbitrary starting point, and then one samples

$$\begin{aligned}
\mathbf{s}^{(1)} &\sim p(\mathbf{s}|\theta^{(0)}, \mathbf{d}) \\
\theta^{(1)} &\sim p(\theta|\mathbf{s}^{(1)}, \mathbf{d}) = p(\theta|\mathbf{s}^{(1)}) \\
\mathbf{s}^{(2)} &\sim p(\mathbf{s}|\theta^{(1)}, \mathbf{d}) \\
\theta^{(2)} &\sim p(\theta|\mathbf{s}^{(1)}, \mathbf{d}) = p(\theta|\mathbf{s}^{(2)}) \\
&\vdots
\end{aligned}$$

Some of the first samples must be discarded because of the bias introduced by the starting point ("burn-in"), but eventually the samples do come from the right distribution. While the samples will be correlated, this does not stop us from using them to make inferences about θ.

First, note that if one already knows the CMB signal, the observation adds nothing of value, so that $p(\theta|\mathbf{s},\mathbf{d}) = p(\theta|\mathbf{s})$. Therefore the Gibbs sampling approach decouples the issue of dealing with how observations are made from the cosmological modelling. Second, assuming that the CMB signal is Gaussian, there exists an efficient algorithm for sampling from $p(\mathbf{s}|\theta,\mathbf{d})$. In chapter 3, we will describe how this is in fact a Gaussian distribution with mean $\hat{\mathbf{s}} = (\mathbf{S}^{-1} + \mathbf{N}^{-1})^{-1}\mathbf{N}^{-1}\mathbf{d}$ and covariance matrix $(\mathbf{S}^{-1} + \mathbf{N}^{-1})^{-1}$. By using an iterative linear system solver, and performing basis changes between pixel space and spherical harmonic space, it is computationally feasible to draw samples from this distribution. On WMAP data, drawing a sample from $p(\mathbf{s}|\theta,\mathbf{d})$ about 15 minutes, in parallel on eight 2.66 GHz CPUs.

To fit a model, we also need to sample the parameters θ, given the signal $\mathbf{s}$. In chapter 4 we will derive a nice expression for the covariance $\mathbf{S}(\theta)$ in the dipole-modulation model, and show that it is very sparse, so that it is suitable for computations. The distribution of the parameters given the signal is

$$p(\theta|\mathbf{s}) \propto |\mathbf{S}(\theta)|^{-1/2} e^{-\frac{1}{2}\mathbf{s}^\dagger \mathbf{S}(\theta)^{-1}\mathbf{s}}\, p(\theta),$$

where $p(\theta)$ is our Bayesian prior. This is not a Gaussian, because it is $\mathbf{S}$ that varies with the parameters. A method for drawing independent samples being out of reach, we turn to Monte Carlo Markov Chain (MCMC) methods. Developing this sampler and combining it with the Gibbs sampler is the purpose of chapter 5.

1.5 Implementation and analysis

Exciting algorithms do not translate into cosmological insight without first being turned into debugged code. The goal of this thesis has not merely been to develop methods, but also to implement them. The result is PyCMB, a modular Python package for CMB analysis. The code is implemented from scratch, independently of any earlier cosmological code. We give an overview of the package in chapter 6, and turn to testing it on simulations and apply it to data in chapter 7.

Chapter 2

Cosmology

The focus of this thesis is very much on algorithms and computations, and our treatment of hemispherical power asymmetry is purely phenomenological. We do not intend to dive into the question of how power asymmetry could be explained physically. Still, we include a chapter about the currently accepted cosmological concordance model in order to provide some context, focusing on assumptions particularly relevant to statistical CMB analysis. In this chapter we rely on Dodelson (2003) unless otherwise noted.

2.1 Fitting a power spectrum

The cornerstone of cosmological data analysis is the power spectrum. Figure 1.3 displays the temperature power spectrum. Additional information is present in the polarization of the CMB photons, which can be characterized by polarization power spectra and polarization-temperature correlation spectra. Assuming that we live in an isotropic universe, and that the CMB is Gaussian, these power spectra together contain all the cosmological information in the CMB, since they describe the variance of each CMB spherical harmonic coefficient $a_{\ell m}$, and the correlations between such coefficients in temperature data and polarization data. We will not make use of polarization data in this thesis and focus on the temperature power spectrum alone.

In the case of an anisotropic universe, correlations are induced between the CMB signal coefficients $a_{\ell m}$, so that the power spectrum is no longer sufficient to describe the statistical properties of the CMB signal. A Gaussian signal can always be described by the full covariance matrix $\mathbf{S}$, and different anisotropic models result in different predictions for the structure of $\mathbf{S}$. If the signal is non-Gaussian, one obviously need more parameters than the covariance structure to describe the signal. Both anisotropy and non-Gaussianity are topics of current investigation, although it seems safe to say that after WMAP, the isotropic, Gaussian model is the null hypothesis of the CMB research community.

Given a cosmological model and an associated set of parameters, one is

able to predict a power spectrum and check it against data. The current concordance model ΛCDM is based on six parameters, the more available ones being the density of atoms and electrons (Ω_b), density of all matter including dark matter (Ω_m), and dark energy content (Ω_Λ) (Larson et al., 2010, Dodelson, 2003). A cosmological model is typically fit to data by drawing samples from the posterior distribution, using the following process for each sample:

- Use Monte Carlo Markov Chain (MCMC) sampling to propose jumps in cosmological parameter space. A popular code for this is CosmoMC[1] (Lewis & Bridle, 2002).
- For each proposed position in parameter space, compute the corresponding power spectrum by carrying out the computations we will sketch below. A popular code for this is CAMB[2] (Lewis et al., 2000).
- Finally, compute the likelihood of the computed power spectrum with respect to observed CMB data. The likelihood code depends on the data included. We mention a brute force approach in section 3.2.1. An approach based on samples produced by Gibbs sampling can be found in Rudjord et al. (2009).

2.2 Isotropy, inflation and Gaussianity

The observable universe appears to be very close to isotropic and homogeneous, supporting the the cosmological principle that our position in the universe is not "special" in any way. There appears to be no preference for a particular direction, the average density of matter appears to be the same everywhere, and perturbations to matter appear to have the same statistical properties everywhere. In particular, the perturbations in the CMB look the same regardless of position on the sphere.

How did this come into being? The idea of Big Bang is not that of an explosion localized in space, but rather that all of space itself was once shrunk together, and has since then expanded. Since the photons in the CMB travel at the speed of light, there is (one would think) no way that regions we observe in one part in the CMB can ever have been in causal contact with regions that we observe in the diametrically opposite direction. And if they have never been in causal contact, there has not been a chance for them to reach equilibrium. There is no reason they should look the same.

Inflation is the currently accepted solution to this problem (and some other problems). The idea is that during a very small fraction of a second, the universe went through a period of accelerated expansion, expanding its size by at least a factor of 10^{28}. The consequence is that regions that are not in causal contact today was in causal contact before inflation happened. By

[1] `http://cosmologist.info/cosmomc`

[2] `http://cosmologist.info/camb`

stretching out a tiny volume to enormous proportions, everything inside that volume now look homogeneous.

The radiation and matter density is thought to have been very uniform. However, in order to eventually form the structures that we see today, it must have had small perturbations in it. Another problem that inflation solves is how those initial perturbations are set up. The somewhat poetic explanation is that at the tiny scales prior to inflation, quantum mechanics comes into play. The radiation and matter density fluctuated quantum mechanically around its ground state. These fluctuations were then blown up by inflation and became the seeds of today's galaxy clusters. A consequence of this theory is that the perturbations should be very close to Gaussian, and that inflation ultimately predicts how much fluctuation there should be on different scales. In this context, Gaussianity is not simply a result of the law of large numbers, as in most other settings. Instead, it is thought to arise from fundamental properties of physics. After inflation, causal physics starts to act, structures form and the radiation and matter densities are processed, but for large scales all of these are accurately described as linear transformations, so that Gaussianity is preserved. The projection of density fluctuation from 3D space to the sphere of the CMB is also a linear operation, and so the CMB is Gaussian, or at least very close to it. Naturally, much work has gone into checking both isotropy and Gaussianity in the CMB.

2.3 Evolution: The Einstein-Boltzmann differential equations

After inflation, matter (primarily dark matter) and photons are spread across the universe, with tiny perturbations in the density. Gravity then comes into play, so that perturbations grow larger. Because of the limited speed of light, gravity acts first on small scales, and then on larger and larger scales.

In the very early universe, it is too hot for electrons and protons to combine into hydrogen. All the free electrons interact with photons, creating a "fog" in which light cannot move far. At some point, electrons, protons and neutrons (baryons) are so clumped together that the photon pressure (collisions between photons and free electrons) eventually cause the baryons to push away from each other again. Photon pressure and gravity acting in opposite directions cause oscillations. These oscillations can down the line be observed in the CMB power spectrum. The first peak is the scale on which baryons has had time to compress once, the second peak the scale on which they had time to compress once and then decompress, the third peak had time to compress–decompress–compress, and so on.

Finally, temperature got so low (around 3000 Kelvin) that electrons could bind to proton to form hydrogen. Suddenly, the universe became transparent, and the image froze ("recombination"). Many photons reach us today that

last scattered at this point, and those make up the CMB. While the CMB as a whole is very uniform, the small perturbations in the CMB correspond directly to small perturbations in matter and light back then.

If one postulate a specific cosmological model and a set of cosmological parameter values, one can now carry out the calculations to find out the statistical properties of the constituents of the universe at the time of recombination, and how the photons would travel through space-time to reach us today. The result is a prediction of the CMB power spectrum (see figure 1.3).

We will of course skip the details, but we hope to give the gist of what kind of calculations is needed to do this. Essentially, one sets up differential equations for the quantities that needs to be tracked. To keep things tractable, one first set up a zero order universe where one averages over all locations and only works with a time component, and then work with a first order perturbation of the zero order solution. Any higher order terms are neglected. This procedure is believed to provide reliable answers for scales down to approximately 32 million light years. The components that must be tracked are:

- $\Phi(\vec{x},t), \Psi(\vec{x},t)$ – Curvature and Newtonian potentials. Describes how space-time curves at position $\vec{x}$ at time t, according to General Relativity (this is but one choice of parameters).
- $T(\vec{x},t,\hat{p}) = T_0(t)(1+\Theta(\vec{x},t,\hat{p}))$ – Denotes how many photons are present at a given time and position, having the direction given by $\hat{p}$.
- $n_b(\vec{x},t) = n_b^{(0)}(1+\delta_b(\vec{x},t))$, $v_b(\vec{x},t)$ – Number density of baryons at a given time and position, and their average velocity, respectively.
- $n(\vec{x},t) = n^{(0)}(1+\delta(\vec{x},t))$, $\vec{v}(\vec{x},t)$ – Number density of dark matter at a given time and position, and its average velocity, respectively.
- Neutrinos are included in a similar way.

The Boltzmann equations then describe the collisions between particles, while the Einstein equations describe the behaviour of space-time in the presence of particles. The force of gravity is embedded into the latter and is not treated explicitly. Combining them gives a set of Einstein-Boltzmann equations. For instance, for dark matter we have

$$\frac{\partial\delta}{\partial t} + \frac{1}{a}\sum_{j=1}^{3}\frac{\partial v_j}{\partial x_j} + 3\frac{\partial\Phi}{\partial t} = 0, \tag{2.1}$$

$$\sum_{j=1}^{3}\left(\frac{\partial v_j}{\partial t} + \frac{da/dt}{a}v_j + \frac{1}{a}\frac{\partial\Psi}{\partial x_j}\right) = 0, \tag{2.2}$$

where $a(t)$ is the scale factor (size of the universe relative to today). All of the components described above set up similar partial differential equations, all coupled together.

The next step is to do a Fourier transform from positions $\vec{x}$ to Fourier waves $\vec{k}$. Now, the assumption that the universe is isotropic and homogeneous on large scales comes into play. The properties of the universe should be the same regardless of direction (and phase) of a wave $\vec{k}$. So, rather than studying all of 3D space, we only study Fourier waves in a single direction, characterized by a real scalar wave-number k. Because of the Fourier transform, taking the partial derivatives with respect to x_j is turned into multiplication by ik. We then end up with a much nicer set of ordinary differential equations (ODEs), because we can solve for each k separately. This is a consequence of only expanding the perturbations to first order. Together with a change of variable in the time dimension, the equations above become

$$\dot{\delta} + ikv + 3\dot{\Psi} = 0, \tag{2.3}$$

$$\dot{v} + \frac{\dot{a}}{a}v + ik\Psi = 0. \tag{2.4}$$

Inflation theory sets up the initial conditions for the system. The initial conditions of all variables turn out to eventually be linear in the initial condition for Φ, Φ_{init}. In turn, Φ_{init} is stochastic with zero mean. A particular theory of inflation will determine its variance as a function of k, the primordial power spectrum $P(k)$, where it is assumed that there are no correlations between different k. Because of the linearity, we can at this stage simply let Φ be initialized as 1, and then insert $P(k)$ later.

Now, we solve the set of differential equations numerically. Of particular interest are the photon perturbations Θ. In principle, the quantity $\Theta(\vec{p}) = \Theta(\vec{x} = \text{here}, t = \text{now}, \hat{p})$ is the strength of the CMB fluctuations in direction $\hat{p}$ on our sky. We seek to understand its statistical properties. Once it is Fourier-transformed from $\vec{x}$ to k, it turns out that the only part of $\hat{p}$ that influence Θ is the angle given by $\cos\theta = \hat{p}\cdot\hat{k}$, where $\hat{k}$ is the arbitrary direction of the Fourier waves we choose to track. Furthermore, Θ can be Legendre-transformed in terms of $\cos\theta$, resulting in what is essentially an harmonic transform on the circle. The resulting quantities are $\Theta_\ell(k,t)$, where each integer ℓ indicate scale, just like spherical harmonics ($\ell = 0$ is the monopole, $\ell = 1$ is the dipole, and so on). Now, Θ_ℓ, evaluated today at our current position, does in fact correspond to $a_{\ell,0}$ in the spherical harmonic expansion of the CMB signal. Since all $a_{\ell m}$ have the same statistical properties for each ℓ, the variance of Θ_ℓ will give us the power spectrum,

$$\text{Cov}(a_{\ell m}, a_{\ell' m'}) = \delta_{\ell\ell'}\delta_{mm'}C_\ell = \delta_{\ell\ell'}\delta_{mm'}\text{Var}(\Theta_\ell).$$

Now, by $\Theta_\ell(k)$ we indicate $\Theta_\ell(k,t)$ evaluated today[3]. Keep in mind that $\Theta_\ell(k)$ is proportional to Φ_{init}, so it has zero mean and a variance proportional

[3]In practice, a so-called line-of-sight integration approach is employed (Seljak & Zaldarriaga, 1996), which allows for computing $\Theta_\ell(k)$ today for a large range of ℓ's, while still only tracking ℓ up to around 6 when solving the Einstein-Boltzmann ODE.

to the primordial power spectrum $P(k)$. To retrieve C_ℓ, we take the variance and Fourier transform back to a particular position, putting back $P(k)$ in the process,

$$C_\ell \propto \int \frac{d^3k}{(2\pi)^3} P(k)\Theta_\ell^2(k).$$

In addition to Dodelson (2003), we have relied on Callin (2006), who provide an excellent introduction to the computational aspects of the power spectrum.

Chapter 3

From CMB observation to CMB signal

3.1 About CMB observations

One can not fit a cosmological model to observational data without taking into account how the data was gathered. Let us start with putting down the typical model for the data analysis:

$$\mathbf{d} = \mathbf{PBs} + \mathbf{n} + \sum_i \mathbf{f}_i \tag{3.1}$$

Here $\mathbf{d}$ is the raw observed data, $\mathbf{P}$ is the "pixel window", $\mathbf{B}$ is the "beam", $\mathbf{s}$ is the underlying CMB signal, $\mathbf{n}$ is instrumental noise, and the $\mathbf{f}_i$ are foreground components contaminating the CMB. The signal $\mathbf{s}$ is the quantity of interest here; a cosmological model will predict the statistical properties of $\mathbf{s}$, and checking a model against data means checking how well those statistical properties match the observed data $\mathbf{d}$.

The vectors above represents fields on a sphere (in $\mathbb{R}$), and can be represented in many ways. The most important ones are as a set of pixels on the sky, and as a set of spherical harmonic coefficients. Either way, the derivations below stay the same, as a Gaussian vector is still Gaussian after a linear transformation. For now, we will not be specific about representation, but get back to the details in section 3.3.

Map making and pixel window The output from an observation, whether using a satellite telescope or a ground-based telescope, is a "time stream": A stream of pointing directions and associated temperatures. Actually, in the case of WMAP, the observing instrument is a differential radiometer with *two* pointings at any time, and where only the difference between the two has any meaning. At any rate, some map-making algorithm is run to turn the time streams into pixelized maps, which in the case of WMAP are downloadable

from NASA's LAMBDA[1] service. We will not care about the details of these algorithms. Pixelization on the sphere is not a trivial problem, and there is no canonical way of doing it. The HEALPix[2] pixelization scheme and software package (Górski et al., 2005) has by now become the *de facto* standard in the CMB research community, and is the format that the WMAP data is made available in.

While we treat map-making as a black box, we do need to care about the effect pixelization has on the data. In the analysis, the value of a pixel is treated as a sample from a field, taken in a single infinitely small point in the center of the pixel. During map-making, all samples within the pixel surface contribute to this quantity, so that the pixel represents the average of an area. Therefore, the pixelization causes a certain smoothing effect which must be accounted for, and this is what $\mathbf{P}$ represents above. In general this operation is difficult to compute, but using an approximation it is simply $P_{\ell m,\ell' m'} = p_\ell \delta_{\ell\ell'} \delta_{mm'}$ in spherical harmonic space. HEALPix ships with data files containing such approximate $\mathbf{P}$ for the different resolutions.

Beam Closely related to the pixel window is the instrumental beam. The telescopes never read the temperature in a single point, but observe photons coming from a small region around the pointing direction. The region is essentially a density, where more photons come from the center than the edges.

Each point on the sky is scanned several times, and unless one assumes that the beam is azimuthally symmetric, one must treat each scan of a point separately depending on the orientation of the beam. To make things computationally feasible, analysis of WMAP data typically assumes that the beam is symmetric. In that case, the observed image of the CMB $I(\hat{p})$ is simply a full sky convolution of the the physical CMB with the beam density b,

$$I(\hat{p}) = \int s(\hat{n}) b(\hat{n} \cdot \hat{p}) d\Omega_{\hat{n}}.$$

In spherical harmonic space, this turns out to be simply the linear transform $\mathbf{B}$ above, with $B_{\ell m,\ell' m'} = b_\ell \delta_{\ell\ell'} \delta_{mm'}$. Here b_ℓ is a normalised Legendre transform of the radial profile of the beam density (Page et al., 2003). As the effects of the beam and the pixel window are so similar, we will typically treat them together, defining $\mathbf{A} \equiv \mathbf{PB}$.

Instrumental noise No observation is perfect, there is always some random noise. If all systematics are known, such random noise should however have zero mean, and one can also hope to know its properties. As one observe the same spot for a longer period, the random noise should cancel out and the average of the observation should tend to the real signal. Thus this is the

[1] http://lambda.gsfc.nasa.gov/
[2] http://healpix.jpl.nasa.gov/

kind of noise that decays with observation time. The WMAP 7-year data has less noise than the 1-year data.

We will model the noise as an additional additive Gaussian component per pixel, with zero mean and no correlation between pixels, denoted by $\mathbf{n}$ above. We will of course never assume that we know anything about its value. However, its statistical properties, given by the covariance matrix, $\mathrm{Var}(\mathbf{n}) = \mathbf{N}$, are very important to our analysis. For WMAP, each radiometer has an estimated noise level σ, which is combined with the number of times a pixel has been scanned to find the noise in each pixel i,

$$\sigma_i = \frac{\sigma}{\sqrt{n_i}}, \tag{3.2}$$

where n_i is the number of times WMAP scanned pixel i. The σ_i are known as the RMS map. In Gibbs sampling computer codes, we will prefer to work with $\mathbf{N}^{-1}$, which, because we assume no correlation, is simply $1/\sigma_i^2$ on the diagonal.

Galaxy cut and point sources Our own Milky Way is a powerful source of radiation in the same frequencies as the CMB, and must simply be masked out. Similarly, several small spots on the sky have been identified as "point sources" of radiation, hiding the CMB, and have to be masked out manually to avoid signal contamination (see figure 1.1). We will adopt the masks of the 7-year WMAP analysis (Jarosik et al., 2010, Wright et al., 2009).

In terms of modelling, we simply embed the mask in the statistical properties of $\mathbf{n}$, so that the additive noise in masked pixels is given so large variance that the pixel values are ignored in any analysis. Specifically, we set the diagonal components of $\mathbf{N}^{-1}$ that corresponds to masked pixels to zero. This makes $\mathbf{N}^{-1}$ singular, but $\mathbf{N}^{-1}$ modified in this way is clearly the limit as the pixel noise within the mask goes to infinity, and in the computer codes it only enters through the matrix $\mathbf{N}^{-1} + \mathbf{S}^{-1}$, which is non-singular.

Foregrounds Even after masking out parts of the sky, the CMB is not the only source of radiation in the frequencies we look at. Three sources are especially important: "Free-free" refers to radiation emitted by collisions of free electrons, "thermal dust" is radiation from atoms within gas clouds, and finally "synchrotron" refers to emissions from certain supernova remnants. The level of these will naturally vary between different pixels on the sky. In order to estimate the foregrounds, an important fact is that all the components have different signatures in the radiation spectra. This can be used to estimate foregrounds from WMAP data itself (see figure 3.1). Another approach is to use independent observations. For instance, independent maps of Hα-emission from hydrogen give hints as to where there are free electrons (Gold et al., 2010).

We will not go into details here, but simply trust that the WMAP team has done a good job with the foregrounds and use the foreground cleaned

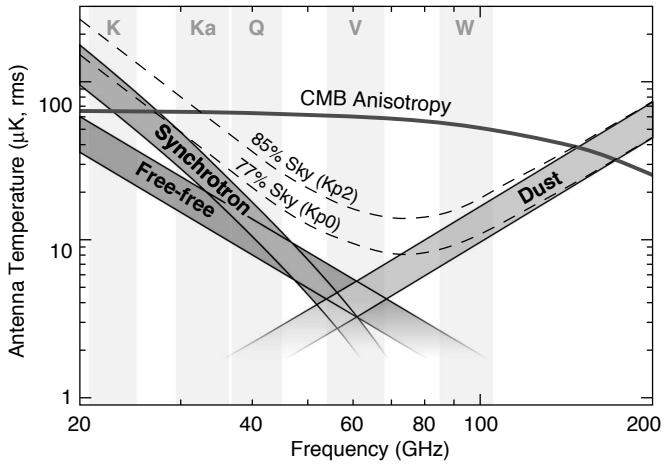


Figure 3.1: The frequency bands of WMAP, and the level of different sources of radiation. Note that the CMB radiation in itself is much stronger than the foregrounds at 2.725 K, but what we are interested in is the fluctuations in the CMB. Image courtesy of LAMBDA/Bennett et al. (2003).

WMAP maps available on LAMBDA (Gold et al., 2010). A consequence is that uncertainties in estimating the foregrounds are not propagated to the final parameter estimates. In our case of an hemispherical power asymmetry model, Hoftuft et al. (2009) found that the estimates parameters are essentially insensitive to foregrounds. We will assume the same in this thesis. However, it would have been possible to do joint foreground and parameter estimation, which would have propagated the uncertainties in the foreground estimates perfectly to uncertainties in the parameter estimates (see Eriksen et al., 2008).

Among the foregrounds are also the monopole and dipole components. First, the monopole (overall average) at 2.725 K is uninteresting for our analysis, and the WMAP observations are in fact insensitive to it. Second, our own point of observation has a distinct movement with respect to the CMB sphere, which because of the Doppler effect creates a strong dipole component that completely drowns out any cosmological information. While the best fit mono- and dipole of the maps are also subtracted from the foreground cleaned maps by the WMAP team, these estimates and their uncertainties are coupled to the rest of the analysis. This coupling has earlier been noted to affect the analysis of the dipole-modulation model by Eriksen et al. (2007). To be on the safe side we should therefore include them in our statistical model and make sure we are insensitive to them, rather than just inserting a single best fit estimate. The mono- and dipole then enters the model as foreground components $\mathbf{f}_i$ above. We defer the details to section 3.4.

3.2 Fitting models to data through Gibbs sampling

3.2.1 The brute-force approach

Given some CMB data $\mathbf{d}$ and its associated properties, how does one fit a cosmological model to the observations? Let us write down the model for the data again:

$$\mathbf{d} = \mathbf{A}\mathbf{s} + \mathbf{n}$$

Here $\mathbf{A}$ contains both pixel window and beam, we assume that foregrounds have been identified and subtracted from $\mathbf{d}$ in a pre-processing step, and we simply ignore the monopole and dipole for the time being out of notational convenience.

Assume now that the signal $\mathbf{s}$ is Gaussian. As it is a perturbation, it should have zero mean, so it is fully characterized by its covariance $\mathbf{S}$, and specifying a cosmological model boils down to specifying some parametrisation of $\mathbf{S}$. For an isotropic model we have $S_{\ell m,\ell' m'} = C_\ell \delta_{\ell\ell'}\delta_{mm'}$, and can let C_ℓ be the parameters. Fitting the model then means estimating an observed power spectrum. Alternatively, one can make further assumptions and use the various cosmological parameters themselves (Ω_m, Ω_b, and so on). By the process described in chapter 2, such parameters can be turned into a power spectrum C_ℓ, and thus $\mathbf{S}$.

At any rate, let θ be some model parameters of choice. From a Bayesian perspective, the recipe is as usual

$$p(\theta|\mathbf{d}) \propto p(\mathbf{d}|\theta)p(\theta),$$

where $p(\theta)$ is our (possibly flat) prior on the parameters. Here $\mathbf{d}$ is a sum of two uncorrelated Gaussians, and is thus Gaussian with parameters

$$\begin{aligned} \mathrm{E}(\mathbf{d}|\theta) &= \mathbf{A}\mathrm{E}(\mathbf{s}) + \mathrm{E}(\mathbf{n}) = 0 \\ \mathrm{Var}(\mathbf{d}|\theta) &= \mathbf{A}\mathrm{Var}(\mathbf{s}|\theta)\mathbf{A}^T + \mathrm{Var}(\mathbf{n}) = \mathbf{A}\mathbf{S}(\theta)\mathbf{A}^T + \mathbf{N} \equiv \mathbf{C}(\theta). \end{aligned}$$

Simple enough in theory, but in practice this approach has a major drawback. While $\mathbf{N}$ is sparse in pixel space, and $\mathbf{S}$ is sparse in spherical harmonic space (at least for the models we will be looking at), their sum is dense in either space. To evaluate the likelihood $p(\mathbf{d}|\theta)$, one would need to Cholesky factor $\mathbf{C}(\theta)$ for each new step in θ. This scales as $O({N_{\mathrm{pix}}}^3) = O({\ell_{\mathrm{max}}}^6)$. This currently stops such computation at $N_{\mathrm{side}} = 32$ or $\ell_{\mathrm{max}} \approx 80$ for most purposes[3].

[3]Cholesky factorization is by no means the only way of computing the exponent of the Gaussian, $\chi^2 \equiv \mathbf{d}^T\mathbf{C}(\theta)^{-1}\mathbf{d}$. For instance, one can use the Conjugate Gradients method described in section 3.5.1 to solve the system. However, in order to properly evaluate the posterior, we need to find $p(\mathbf{d}|\theta)$ as a function of θ (likelihood). This means that we also

3.2.2 Gibbs sampling

We now narrow down and assume that we can make do with the rather common approach of drawing samples from $p(\theta|\mathbf{d})$. That is, we rule out maximum-likelihood type methods for finding confidence regions. The trick is then to draw from the joint posterior of the parameters with the CMB signal, $p(\theta, \mathbf{s}|\mathbf{d})$. The computations will turn out to be considerably cheaper, and the samples of θ will still be from the marginal posterior, $p(\theta|\mathbf{d})$.

Does this really make our job easier? Yes, through the beauty of Gibbs sampling. The algorithm simply states that given a starting point $\theta^{(0)}$, we can iteratively sample from the conditional distributions.

$$\begin{aligned}
\mathbf{s}^{(1)} &\sim p(\mathbf{s}|\theta^{(0)}, \mathbf{d}) \\
\theta^{(1)} &\sim p(\theta|\mathbf{s}^{(1)}, \mathbf{d}) \\
\mathbf{s}^{(2)} &\sim p(\mathbf{s}|\theta^{(1)}, \mathbf{d}) \\
\theta^{(2)} &\sim p(\theta|\mathbf{s}^{(2)}, \mathbf{d}) \\
&\vdots
\end{aligned}$$

Under some conditions, and regardless of the value of $\theta^{(0)}$, the distribution of these samples will converge to the joint posterior distribution, $p(\theta, \mathbf{s}|\mathbf{d})$. The Gibbs sampling algorithm was introduced to the CMB community by Jewell et al. (2004) and Wandelt et al. (2004), who develop a method for efficiently drawing samples from $p(\mathbf{s}|\theta, \mathbf{d})$. In the following we review this algorithm. The crucial point will be that as we draw a sample from a distribution, rather than evaluate a likelihood, the determinant in the expression for the Gaussian density is not needed. Therefore, it scales as $O({\ell_{\mathrm{max}}}^3)$, which is quite an improvement over the $O({\ell_{\mathrm{max}}}^6)$ scaling of the brute-force likelihood evaluation approach.

Sampling from $p(\theta|\mathbf{s}, \mathbf{d})$ is for many models trivial, computationally speaking. We note that $p(\theta|\mathbf{s}, \mathbf{d}) = p(\theta|\mathbf{s})$, i.e., if we already know the CMB signal, the CMB observation adds nothing to our knowledge of the cosmological parameters. We certainly do not expect cosmological parameters to directly affect the WMAP sensors[4].

A neat property of the CMB Gibbs sampler is that one decouples the lower-level issues of of data analysis, such as instrumental properties, from model parameter estimation, which can then be done more efficiently. The remainder of this chapter is dedicated to the former, while we discuss Gibbs sampling in the context of our particular model in chapter 5.

need to efficiently find the determinant, $|\mathbf{C}(\theta)|$. In the case of hypothesis testing, a common approach is to simply treat χ^2 as an estimator with unknown distribution, and then use simulations from a null model to establish the significance of a change in χ^2.

[4]This is simply how we define what we mean by "the CMB signal". The photons emitted at recombination are certainly affected by cosmological parameters on their way to us, but all such effects are considered part of the cosmological model, and embedded in the power spectrum.

3.2.3 Sampling from the CMB signal posterior

To use Gibbs sampling, we need an efficient algorithm for drawing samples from $p(\mathbf{s}|\theta, \mathbf{d})$. Using Bayes' rule,

$$\begin{aligned} p(\mathbf{s}|\theta, \mathbf{d}) &\propto p(\mathbf{d}|\mathbf{s}, \theta)p(\mathbf{s}|\theta) \\ &= p(\mathbf{d}|\mathbf{s})p(\mathbf{s}|\theta). \end{aligned}$$

Note that the "prior" $p(\mathbf{s}|\theta)$ is conditional on the cosmological model, and we will insert a known expression leaving no room for a "prior opinion". Also, the real CMB signal is sufficient for predicting the observed CMB, so $p(\mathbf{d}|\mathbf{s}, \theta) = p(\mathbf{d}|\mathbf{s})$.

Assuming like before that the signal is Gaussian with zero mean and covariance $\mathbf{S}(\theta)$, we can use Bayes' rule to find the posterior distribution of the signal given the observed data. First, note that conditional on $\mathbf{s}$, we have

$$\begin{aligned} \mathrm{E}(\mathbf{d}|\mathbf{s}) &= \mathbf{A}\mathbf{s} + \mathrm{E}(\mathbf{n}) = \mathbf{A}\mathbf{s} \\ \mathrm{Var}(\mathbf{d}|\mathbf{s}) &= \mathrm{Var}(\mathbf{n}) = \mathbf{N}, \end{aligned}$$

so that

$$\begin{aligned} p(\mathbf{s}|\mathbf{d}, \theta) &\propto p(\mathbf{d}|\mathbf{s}, \theta)p(\mathbf{s}|\theta) \\ &\propto e^{-\frac{1}{2}(\mathbf{d}-\mathbf{A}\mathbf{s})^T\mathbf{N}^{-1}(\mathbf{d}-\mathbf{A}\mathbf{s})} e^{-\frac{1}{2}\mathbf{s}^T\mathbf{S}^{-1}\mathbf{s}}. \end{aligned} \tag{3.3}$$

We only need the probability density up to a constant factor, so we can ignore terms in the exponent that do not contain $\mathbf{s}$:

$$\begin{aligned} &(\mathbf{d} - \mathbf{A}\mathbf{s})^T\mathbf{N}^{-1}(\mathbf{d} - \mathbf{A}\mathbf{s}) + \mathbf{s}^T\mathbf{S}^{-1}\mathbf{s} \\ &\quad = \mathbf{s}^T\mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}\mathbf{s} - 2\mathbf{s}^T\mathbf{A}^T\mathbf{N}^{-1}\mathbf{d} + \mathbf{s}^T\mathbf{S}^{-1}\mathbf{s} + \text{const.} \\ &\quad = \mathbf{s}^T(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})\mathbf{s} - 2\mathbf{s}^T\mathbf{A}^T\mathbf{N}^{-1}\mathbf{d} + \text{const.} \end{aligned} \tag{3.4}$$

By only considering the quadratic part for now, it is clear that we can rewrite equation (3.3) as

$$p(\mathbf{s}|\mathbf{d}, \theta) \propto e^{-\frac{1}{2}(\mathbf{s}-\hat{\mathbf{s}})^T(\mathbf{S}^{-1}+\mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})(\mathbf{s}-\hat{\mathbf{s}})}, \tag{3.5}$$

for some expectation $\hat{\mathbf{s}}$ of the posterior. In other words, $p(\mathbf{s}|\mathbf{d}, \theta)$ is Gaussian with covariance[5] $(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})^{-1}$. Equating the parts linear in $\mathbf{s}$ in equations (3.4) and (3.5), we see that $\hat{\mathbf{s}}$ must satisfy

$$-2\mathbf{s}^T(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})\hat{\mathbf{s}} = -2\mathbf{s}^T\mathbf{A}^T\mathbf{N}^{-1}\mathbf{d}.$$

[5]The matrix is invertible: $\mathbf{N}$ (and $\mathbf{N}^{-1}$) may in our setup approach singular matrices, but $\mathbf{S}$ (and $\mathbf{S}^{-1}$) will for all relevant cases be non-singular. All of these matrices are positive (semi)definite, and so $(\mathbf{S}^{-1} + \mathbf{N}^{-1})$ is strictly positive definite and invertible.

Since this must be satisfied for any choice of $\mathbf{s}$, we can simply remove $-2\mathbf{s}^T$ on both sides, and inverting the left-hand side matrix we have

$$\hat{\mathbf{s}} = (\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{N}^{-1}\mathbf{d}.$$

This is known as the "Wiener-filtered map", and represents the single most likely map.

A simple brute-force approach to sampling is now to form and Cholesky decompose the inverse covariance $(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}) = \mathbf{L}\mathbf{L}^T$, draw a vector of standard normal variates $\mathbf{x}$, and let our sample $\mathbf{s} = \mathbf{L}^{-T}\mathbf{x} + \mathbf{L}^{-T}\mathbf{L}^{-1}\mathbf{A}^T\mathbf{N}^{-1}\mathbf{d}$. But again, while $\mathbf{N}$ is sparse in pixel space and $\mathbf{S}$ is sparse in spherical harmonic space, the sum of their inverses is dense in either space, making the approach too expensive for the resolutions we want to look at. However, it is smarter than the default textbook approach of Cholesky decomposing the covariance $(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})^{-1}$.

To get away without forming the full dense matrix, we need another approach. We start with finding the mean $\hat{\mathbf{s}}$. Note that the inverse covariance, $\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}$, is much easier to work with than the covariance itself. Iterative methods seem like a good start, as they make it possible to solve a linear system $\mathcal{A}\mathbf{x} = \mathbf{b}$ simply by repeatedly multiplying with the left hand side matrix $\mathcal{A}$. In this case, the matrix $(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})$ is positive definite, so we can use the Conjugate Gradients (CG) method (which is the subject of section 3.5.1). Since the mean $\hat{\mathbf{s}} = (\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{N}^{-1}\mathbf{d}$, it is clear that we can use CG to efficiently find $\hat{\mathbf{s}}$ by solving the equation

$$(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})\hat{\mathbf{s}} = \mathbf{A}^T\mathbf{N}^{-1}\mathbf{d},$$

using many multiplications rather than a full decomposition. The multiplication can be done by solving for $\mathbf{S}$ and $\mathbf{N}$ separately, so that through some basis changes one can take advantage of the sparse structure of the matrices.

The mean is not enough, what we really want is a sample from the posterior. That is, the mean plus some random fluctuation, so that the samples have the right covariance. Let us try to make use of the work done in the CG search not only to get the right mean, but also to get the right covariance. The only way this could work is by adding a random Gaussian fluctuation map ω to the right hand side:

$$(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})\mathbf{s} = \mathbf{A}^T\mathbf{N}^{-1}\mathbf{d} + \omega.$$

The solution $\mathbf{s}$ is then given by

$$\mathbf{s} = \hat{\mathbf{s}} + (\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})^{-1}\omega,$$

so it is clear that if we let ω have zero mean, $\mathbf{s}$ has the right mean. Note that the covariance of $\mathbf{s}$ is

$$(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})^{-1}\ \mathrm{Var}(\omega)\ (\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})^{-1}.$$

In a sense, twice as much as we want. This problem is not surprising, given that we never found a Cholesky factor or a matrix square root. However, if we can make ω have covariance $\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}$, the problem is solved. And this is much easier, as we can simply add together two independent Gaussian draws with zero mean and covariances $\mathbf{S}^{-1}$ and $\mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}$, respectively. Since $\mathbf{S}$ and $\mathbf{N}$ are sparse in spherical harmonic and pixel space, respectively, these matrices are much easier to factor in order to simulate such draws.

In summary, the algorithm is:

- Draw two vectors of standard normal variates, ω_0 and ω_1.
- In order to draw random vectors, find a factor[6] $\mathbf{F}$ such that $\mathbf{F}\mathbf{F}^T = \mathbf{S}$. In the isotropic case, one simply takes the square root of each diagonal element, $\mathbf{F} = \mathbf{S}^{1/2}$. Since we assume uncorrelated pixel noise, it is easy to factor it as well $\mathbf{N} = \mathbf{N}^{1/2}\mathbf{N}^{1/2}$.
- Using Conjugate Gradients, solve the following equation for $\mathbf{s}$:

$$(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})\mathbf{s} = \mathbf{A}^T\mathbf{N}^{-1}\mathbf{d} + \mathbf{F}^{-T}\omega_0 + \mathbf{A}^T\mathbf{N}^{-1/2}\omega_1. \tag{3.6}$$

By the construction of the algorithm, $\mathbf{s}$ will then be a draw from the posterior $p(\mathbf{s}|\theta, \mathbf{d})$. Finally, it is easy to see that an alternative formulation is

$$(\mathbf{1} + \mathbf{F}^T\mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}\mathbf{F})(\mathbf{F}^{-1}\mathbf{s}) = \mathbf{F}^T\mathbf{A}^T\mathbf{N}^{-1}\mathbf{d} + \omega_0 + \mathbf{F}^T\mathbf{A}^T\mathbf{N}^{-1/2}\omega_1 \tag{3.7}$$

where one solve for $\mathbf{F}^{-1}\mathbf{s}$ first, and then simply multiply with $\mathbf{F}$ to retrieve $\mathbf{s}$. This is the formulation commonly used in the literature. There are further notes on this choice in section 3.5.1.

3.3 Basis changes: Pixels and spherical harmonics

Until now, we have conveniently left the representation of $\mathbf{s}$, $\mathbf{N}^{-1}$, and so on, unspecified. Now the time has come to care. In this section, and this section only, we will be very explicit and denote vectors of spherical harmonic coefficients $\tilde{\mathbf{x}}$, while the corresponding pixel vectors are denoted $\hat{\mathbf{x}}$.

3.3.1 Linear algebra notation for spherical harmonic transforms

The transform from $\tilde{\mathbf{x}}$ to $\hat{\mathbf{x}}$ is

$$\hat{x}_i = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} \tilde{x}_{\ell m} Y_{\ell m}(\hat{n}_i) \tag{3.8}$$

[6] We will denote *symmetric* factors by the notation $\mathbf{S}^{1/2}$, but $\mathbf{F}$ in this case need not be symmetric.

where $\hat{n}_i$ is the position of (the center of) pixel i. In this context, the pixel represents a sample of a field in an infinitely small point, and the only approximation done is by having a finite rather than infinite $\ell_{\max}$. In fact, if the signal is band-limited ($\tilde{x}_{\ell m} = 0$ for all $\ell > \ell_{\max}$), this transformation is exact.

We will write $\mathbf{Y}$ for an N_{pix}-by-$N_{\ell m}$ matrix containing the spherical harmonic coefficients $Y_{i,\ell m} = Y_{\ell m}(\hat{n}_i)$. It is then clear that equation (3.8) can be written

$$\hat{\mathbf{x}} = \mathbf{Y}\tilde{\mathbf{x}}.$$

In HEALPix, the routine `alm2map` is available for this computation. It reformulates the operation in term of discrete Fourier transforms for efficiency.

The opposite transform, going form $\hat{\mathbf{x}}$ to $\tilde{\mathbf{x}}$, is slightly messier, as there is no analogue to the discrete Fourier transform on the sphere. Instead of a sum, we have to approximate an integral over the sphere,

$$\tilde{x}_{\ell m} = \int x(\hat{n}) Y^*_{\ell m}(\hat{n}) d\Omega, \tag{3.9}$$

where $x(\hat{n})$ is the field that is sampled in $\hat{\mathbf{x}}$, and $d\Omega$ represents an area element on the sphere. This is computed by a quadrature,

$$\tilde{x}_{\ell m} = \sum_{i=1}^{N_{\text{pix}}} w_i \hat{x}_i Y^*_{\ell m}(\hat{p}_i) \Delta\Omega_i. \tag{3.10}$$

Here $\Delta\Omega_i$ represents pixel area, which in the case of HEALPix is the same for all pixels, i.e., $\Delta\Omega_i = 4\pi/N_{\text{pix}}$, while the w_i are quadrature weights. HEALPix uses one weight per iso-latitude ring in its `map2alm` routine, in order to be able to use discrete Fourier transforms on each such ring. Equation (3.10) can be written

$$\tilde{\mathbf{x}} = \mathbf{Y}^\dagger \mathbf{W} \hat{\mathbf{x}}$$

with the same $\mathbf{Y}$ as before, and $\mathbf{W}$ a diagonal matrix containing pixel weight and pixel area, $W_{ij} = w_i \Delta\Omega_i \delta_{ij}$.

Because the spherical harmonics are orthogonal, so that

$$\int Y_{\ell m}(\hat{n}) Y^*_{\ell' m'}(\hat{n}) d\Omega = \delta_{\ell\ell'}\delta_{mm'},$$

it is clear that when N_{pix} is high enough, $\mathbf{Y}^\dagger\mathbf{W}\mathbf{Y} \approx \mathbf{1}$. Therefore a set of harmonic coefficients can be adequately represented in pixel space, and a round-trip will be OK. In practice, "high enough" depends on $\ell_{\max}$. Experience show that as long as $\ell_{\max} \leq 2N_{\text{side}}$ one is perfectly safe, and that $\ell_{\max}$ as high as $3N_{\text{side}} - 1$ can work well, although less accurate. These somewhat vague statements are in contrast to Discrete Fourier Transforms, where orthogonality is always exact.

We will not rely on the opposite behaviour, that arbitrary pixel maps can round-trip through spherical harmonic space. The important thing is that $\tilde{\mathbf{x}} = \mathbf{Y}^\dagger\mathbf{W}\hat{\mathbf{x}}$ makes $\tilde{\mathbf{x}}$ contain all information up to some band-limitation scale ℓ_{max}. That is, we treat $\mathbf{Y}^\dagger\mathbf{W}$ as a projection.

3.3.2 Reinterpreting the Gibbs sampling equations

How do the basis changes translate into CMB analysis and Gibbs sampling? Writing down the model again, on a more explicit form, what we have in our observation pixel map is

$$\hat{\mathbf{d}} = \mathbf{Y}\tilde{\mathbf{A}}\tilde{\mathbf{s}} + \hat{\mathbf{n}}.$$

Note that the beam and pixel window $\tilde{\mathbf{A}}$ will become zero at some ℓ (which depends on the size of the beam), and N_{side} can be selected with respect to this to make sure all the information in the smoothed signal $\tilde{\mathbf{A}}\tilde{\mathbf{s}}$ is taken into account. We continue by including $\mathbf{Y}$ in the signal posterior equation (3.3),

$$\begin{aligned} p(\tilde{\mathbf{s}}|\hat{\mathbf{d}},\theta) &\propto p(\hat{\mathbf{d}}|\tilde{\mathbf{s}},\theta)p(\tilde{\mathbf{s}}|\theta) \\ &\propto e^{-\frac{1}{2}(\hat{\mathbf{d}}-\mathbf{Y}\tilde{\mathbf{A}}\tilde{\mathbf{s}})^T\hat{\mathbf{N}}^{-1}(\hat{\mathbf{d}}-\mathbf{Y}\tilde{\mathbf{A}}\tilde{\mathbf{s}})}e^{-\frac{1}{2}\tilde{\mathbf{s}}^\dagger\tilde{\mathbf{S}}^{-1}\tilde{\mathbf{s}}}, \end{aligned}$$

where it should be noted that the first exponential is in pixel space with an N_{pix}-by-N_{pix} covariance matrix, while the second exponential contains an $N_{\ell m}$-by-$N_{\ell m}$ covariance matrix. Gathering terms in $\tilde{\mathbf{s}}$, the exponent becomes

$$\tilde{\mathbf{s}}^\dagger(\tilde{\mathbf{S}}^{-1} + \tilde{\mathbf{A}}^\dagger\mathbf{Y}^\dagger\hat{\mathbf{N}}^{-1}\mathbf{Y}\tilde{\mathbf{A}})\tilde{\mathbf{s}} - 2\tilde{\mathbf{s}}^\dagger\tilde{\mathbf{A}}^\dagger\mathbf{Y}^\dagger\hat{\mathbf{N}}^{-1}\hat{\mathbf{d}} + \text{const.},$$

so that the total Gaussian density is expressed in spherical harmonics. By repeating the derivations of the CMB Gibbs sampler again, but this time include $\mathbf{Y}$, we find that a sample from $p(\tilde{\mathbf{s}}|\hat{\mathbf{d}},\theta)$ is given by solving

$$(\tilde{\mathbf{S}^{-1}} + \tilde{\mathbf{A}}^\dagger\mathbf{Y}^\dagger\hat{\mathbf{N}}^{-1}\mathbf{Y}\tilde{\mathbf{A}})\mathbf{s} = \tilde{\mathbf{A}}^\dagger\mathbf{Y}^\dagger\hat{\mathbf{N}}^{-1}\hat{\mathbf{d}} + \tilde{\mathbf{F}}^{-\dagger}\tilde{\omega}_0 + \tilde{\mathbf{A}}^\dagger\mathbf{Y}^\dagger\hat{\mathbf{N}}^{-1/2}\hat{\omega}_1. \quad (3.11)$$

Note that $\mathbf{Y}^\dagger$ does *not* indicate a transform from pixel space to spherical harmonic space (an integral, which we have denoted $\mathbf{Y}^\dagger\mathbf{W}$), but is rather the conjugate transpose of the transform from spherical harmonic space to pixel space (a sum). A way to understand this is that we solve for a spherical harmonic signal, which, when smoothed by the beam and projected to pixel space, is constrained by the pixel data. Outside of this section, we will mostly drop the $\mathbf{Y}$'s, and also refer to $\mathbf{Y}^\dagger\hat{\mathbf{N}}^{-1}\mathbf{Y}$ as being $\mathbf{N}^{-1}$ represented in spherical harmonic space, as is conventional in the literature.

When using CG to solve equation (3.11), it is vital that multiplications with the left hand side matrix is done without forming the full matrix, but instead by letting the matrices act upon vectors. As noted, we may use `alm2map` from HEALPix to apply $\mathbf{Y}$, but what about $\mathbf{Y}^\dagger$? Fortunately, the `map2alm` routine accepts the set of quadrature weights as a parameter, and by passing in $\Delta\Omega^{-1} = N_{\text{pix}}/4\pi$ as the quadrature weights we achieve the desired effect (see table 3.1).

We note that the Commander CMB Gibbs sampler currently use the `map2alm` routine *with* quadrature weights in this setting. To estimate the consequences of this, consider that, given uncorrelated pixel noise, in $\mathbf{Y}^\dagger\mathbf{W}\hat{\mathbf{N}}^{-1}\mathbf{Y}$ we may take $\mathbf{W}$ as being part of the inverse noise covariance. That, is,

Table 3.1: Comparison of `map2alm` and the transpose of `alm2map`, with and without pixel weights. First, we compute the true $\mathbf{Y}^\dagger$ by repeatedly calling `alm2map` with unit vectors to retrieve each column of $\mathbf{Y}$. Then, we call `map2alm` in a similar way to retrieve an explicit matrix $\mathbf{B}$, column by column, using two approaches: i) Use the default ring weights, but scale the input by $N_{\rm pix}/4\pi$; ii) Set the ring weights uniformly to $N_{\rm pix}/4\pi$ (thus disabling the default quadrature weights). Quoted below is $\|\mathbf{Y}^\dagger - \mathbf{B}\|$ in each case (using the Frobenius norm). HEALPix clearly behave as we would expect.

$N_{\rm side}$	$\ell_{\rm max}$	With weights	Without weights
8	16	$2.2 \cdot 10^{0}$	$4.4 \cdot 10^{-14}$
16	32	$5.5 \cdot 10^{0}$	$2.9 \cdot 10^{-13}$
32	64	$1.1 \cdot 10^{1}$	$4.0 \cdot 10^{-12}$

Commander draws samples with mean (suppressing $\mathbf{A}$ and $\mathbf{Y}$ for notational clarity)

$$\hat{\mathbf{s}} = (\mathbf{S}^{-1} + \mathbf{N}'^{-1})^{-1}\mathbf{N}'^{-1}\mathbf{d},$$

and covariance

$$(\mathbf{S}^{-1} + \mathbf{N}'^{-1})^{-1}(\mathbf{S}^{-1} + \mathbf{N}''^{-1})(\mathbf{S}^{-1} + \mathbf{N}'^{-1})^{-1},$$

where $\mathbf{N}'$ and $\mathbf{N}''$ denote noise covariance with altered RMS maps,

$$\sigma_i' = \frac{\sigma_i}{\sqrt{w_i}}, \quad \sigma_i'' = \frac{\sigma_i}{w_i}.$$

That is, the effect will at least be close to the effect of the HEALPix quadrature weights finding their way into the RMS maps. On the north and south poles the HEALPix quadrature weights w_i are close in absolute value to 1.2, but they fall off very rapidly to approximately 1.0 as one approaches equator. Therefore, the effect may very well be unimportant. A perhaps more important issue is that when generalizing to correlated noise, or marginalizing over foreground templates, including quadrature weights can lead to "$\mathbf{N}^{-1}$" being non-symmetric in spherical harmonic space.

3.3.3 Real spherical harmonics

The Conjugate Gradients (CG) algorithm is only defined for linear systems in $\mathbb{R}$. The above definition of spherical harmonics is therefore impractical when we want to solve equation (3.11) by CG. However, since all our fields $f(\hat{n})$ on the sphere are real, the spherical harmonic expansions contain some redundancy, as we have

$$a_{\ell m} = (-1)^m a^*_{\ell -m}.$$

We can therefore change to a basis where all the coefficients are real coefficients. Such a basis, *real spherical harmonics*, is introduced in detail in

appendix A.2. We will denote the unitary transform from complex coefficients to real coefficients $\mathbf{U}$, so that if $\mathbf{x}^C$ is a vector of complex spherical harmonic coefficients, then the corresponding real vector $\mathbf{x}^R$ is given by $\mathbf{x}^R = \mathbf{U}\mathbf{x}^C$. The reverse transform is $\mathbf{x}^C = \mathbf{U}^\dagger\mathbf{x}^R$, and a matrix $\mathbf{K}^C$ transforms as $\mathbf{K}^R = \mathbf{U}\mathbf{K}^C\mathbf{U}^\dagger$. Because $\mathbf{U}$ is unitary, we can go back and forth between real and complex spherical harmonics without worrying. We will usually suppress any implicit $\mathbf{U}$s that are necessary for the concrete implementation. The exception comes when we construct preconditioners for the CG algorithm.

3.4 The monopole and the dipole

As mentioned, the monopole and dipole component of the data has no significance to us, but are present in the data and must be accounted for. Our final model therefore reads

$$\mathbf{d} = \mathbf{A}\mathbf{s} + \mathbf{n} + \sum_{i=1}^{4} \beta_i \mathbf{t}_i = \mathbf{A}\mathbf{s} + \mathbf{n} + \mathbf{T}\beta,$$

where we have parametrized the foreground component of equation (3.1) by scalar parameters $\beta = [\beta_1 \dots \beta_4]$ and a set of hard-coded template vectors $\mathbf{T} = [\mathbf{t}_1 \dots \mathbf{t}_4]$. We let $\mathbf{t}_1$ be all ones (a monopole), while $\mathbf{t}_2$, $\mathbf{t}_3$ and $\mathbf{t}_4$ should be three dipole basis vectors that together span out the space of possible dipoles. For simple implementation, we model β_i as being independent Gaussians. Furthermore, since any dipole in $\mathbf{s}$ will have been completely drowned out by the Doppler effect, $\mathbf{s}$ has zero variance for these components; $C_0 = C_1 = 0$.

There are now two approaches that both makes our analysis insensitive to the presence of a monopole or dipole in the data. The first is to assign a prior $p(\beta_i)$ and estimate the posterior $p(\beta_i|\mathbf{d})$ jointly in our analysis. This approach especially shines in a more general setting where one also estimate other forms of CMB foregrounds in a joint analysis (Eriksen et al., 2008, Jewell et al., 2004, Wandelt et al., 2004). However, since we use foreground-cleaned maps, and for ease of implementation, we instead opt for marginalizing up front, as described by Wandelt et al. (2004). The idea is then to treat $\mathbf{T}\beta$ as a noise term in the model, and state that

$$\beta_i \sim N(0, \sigma_t^2),$$

where σ_t is taken very large, so that the components in the data corresponding to the templates do not impact the posterior. Now, $p(\mathbf{d}|\mathbf{s})$ still has mean $\mathbf{A}\mathbf{s}$, but the variance is

$$\mathrm{Var}(\mathbf{d}|\mathbf{s}) = \mathrm{Var}(\mathbf{n}) + \mathrm{Var}(\mathbf{T}\beta) = \mathbf{N} + \sigma_t^2\mathbf{T}\mathbf{T}^T.$$

Therefore, the additional term $\sigma_t^2\mathbf{T}\mathbf{T}^T$ enters everywhere we have $\mathbf{N}$ in the previous sections. In terms of representation, $\mathbf{T}\beta$ should be understood to be in pixel space.

The system we must solve by CG now becomes (suppressing $\mathbf{A}$)

$$(\mathbf{S}^{-1} + (\mathbf{N} + \sigma_t^2 \mathbf{T}\mathbf{T}^T)^{-1})\mathbf{s} = (\mathbf{N} + \sigma_t^2 \mathbf{T}\mathbf{T}^T)^{-1}\mathbf{d} + \mathbf{F}^{-T}\omega_0 + \xi, \tag{3.12}$$

where ξ is a Gaussian vector with zero mean and covariance $(\mathbf{N} + \sigma_t^2 \mathbf{T}\mathbf{T}^T)^{-1}$. Computing $(\mathbf{N} + \sigma_t^2 \mathbf{T}\mathbf{T}^T)^{-1}\mathbf{x}$ for an arbitrary vector $\mathbf{x}$ is done efficiently by the Sherman-Morrison-Woodbury formula (Harville, 1997):

$$(\mathbf{N} + \sigma_t^2 \mathbf{T}\mathbf{T}^T)^{-1} = \mathbf{N}^{-1} - \mathbf{N}^{-1}\mathbf{T}\left(\frac{1}{\sigma_t^2}\mathbf{1} + \mathbf{T}^T\mathbf{N}^{-1}\mathbf{T}\right)^{-1}\mathbf{T}^T\mathbf{N}^{-1}. \tag{3.13}$$

The inner matrix on the right hand side is a 4-by-4 matrix and is trivial to solve for, so for a diagonal $\mathbf{N}^{-1}$ the additional computational overhead is negligible. It is now customary to let $\sigma_t \to \infty$. However, in the context of Gibbs sampling, we must also draw samples ξ with covariance $(\mathbf{N}+\sigma_t^2 \mathbf{T}\mathbf{T}^T)^{-1}$. This is similar to a situation we already encountered, and the same trick works. We draw two standard Gaussian vectors $\omega_1 \in \mathbb{R}^{N_{\mathrm{pix}}}$ and $\omega_2 \in \mathbb{R}^4$, and let

$$\xi = (\mathbf{N} + \sigma_t^2 \mathbf{T}\mathbf{T}^T)^{-1}(\mathbf{N}^{-1/2}\omega_1 + \sigma_t \mathbf{T}\omega_2).$$

It is easily verified that ξ has the right covariance. It seems safer for numerical stability to apply the inverse matrix, using equation (3.13), on each vector in turn, before adding them together. Unfortunately, letting $\sigma_t \to \infty$ does not seem to work in this context. We must set it large enough for the monopole and dipole to not affect the analysis, but low enough that we do not get numerical problems. The level of residual monopole and dipole in the data is coupled to the general CMB fluctuation level, and is not larger than $\sim$ 10–100 μK. Letting $\sigma_t = 10\,\mathrm{mK}$ worked well for WMAP data.

3.5 Solving the linear system

3.5.1 The Conjugate Gradients algorithm

As we have seen, in order to draw from the posterior $p(\mathbf{s}|\theta, \mathbf{d})$ we need to efficiently solve the system

$$(\mathbf{S}^1 + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})\mathbf{x} = \mathbf{b}, \tag{3.14}$$

or, alternatively,

$$(\mathbf{1} + \mathbf{F}^T\mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}\mathbf{F})\mathbf{x}' = \mathbf{b}'. \tag{3.15}$$

It is clear that the matrix is dense in both spherical harmonic and pixel space. Therefore, we use an iterative algorithm where we only need to repeatedly multiply vectors with the matrix in question. As we have seen, multiplying with $\mathbf{N}^{-1}$ can be done in $O({\ell_{\mathrm{max}}}^3)$ (which is the cost of `alm2map` and `map2alm`) by doing the operation in pixel space. Applying the beam and pixel window

$\mathbf{A}$ and operations with the factor of the model covariance matrix $\mathbf{F}$ are in our case both going to be linear in the number of coefficients, $O({\ell_{\max}}^2)$.

In this case, our matrices are symmetric and positive definite. Focusing on the matrix of equation (3.15), we have

$$\mathbf{x}^T(\mathbf{1} + \mathbf{F}^T\mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}\mathbf{F})\mathbf{x} = \mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{N}^{-1}\mathbf{y} \geq \mathbf{x}^T\mathbf{x} > 0$$

for all $\mathbf{x} \neq \mathbf{0}$, where we let $\mathbf{y} \equiv \mathbf{A}\mathbf{F}\mathbf{x}$. Here we only care that $\mathbf{y}^T\mathbf{N}^{-1}\mathbf{y} \geq 0$ for all $\mathbf{y}$, which should be clear from our definition of $\mathbf{N}^{-1}$. Even if $\mathbf{N}^{-1}$ may be close to singular, the identity matrix makes sure the total matrix is non-singular and positive definite. In fact, all eigenvalues are ≥ 1.

The iterative method of choice in the case of symmetric, positive definite matrices over $\mathbb{R}$ is the method of Conjugate Gradients (CG). It is very intuitively explained in Shewchuk (1994), and here we will only give a very brief summary. CG is based on solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ iteratively by minimizing the quadratic form

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}.$$

It can be shown that when $\mathbf{A}$ is symmetric and positive definite, the solution to the linear system also minimizes f. Given a starting point $\mathbf{x}_0$ (in our case $\mathbf{x}_0 = \mathbf{0}$), then for each step i, the CG algorithm finds a new $\mathbf{x}_i$ that minimizes f along a search direction which is $\mathbf{A}$-orthogonal to all previous search directions. $\mathbf{A}$-orthogonal means being orthogonal under the norm

$$\|\mathbf{x}\|_{\mathbf{A}} = \mathbf{x}^T\mathbf{A}\mathbf{x}.$$

It turns out that each such step can be found very easily, and one only needs one multiplication of a vector with $\mathbf{A}$ and a small number of $O(N)$ vector arithmetic operations, where N is the number of elements in $\mathbf{x}$ and $\mathbf{b}$.

In many cases CG converges very fast, so that relatively few such steps are needed to attain a good approximate solution. In our setting we will be satisfied, and terminate the algorithm, when the residual $\mathbf{r}_i \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_i$ satisfies $\mathbf{r}_i^T\mathbf{r}_i < \epsilon\mathbf{r}_0^T\mathbf{r}_0$ with $\epsilon = 10^{-6}$. For debugging purposes, we can let ϵ be higher.

The speed of the convergence depends on how well clustered the eigenvalues of the matrix is. In order to speed up convergence it is essential to use a *preconditioner*. Instead of solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ directly, one finds a symmetric, positive definite matrix $\mathbf{M}^{-1} = \mathbf{E}^{-T}\mathbf{E}^{-1}$ which approximates $\mathbf{A}^{-1}$, and then instead solve

$$\mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-T}\mathbf{x} = \mathbf{b}.$$

Amazingly, it turns out that the factor $\mathbf{E}^{-1}$ of the preconditioner matrix $\mathbf{M}^{-1}$ need never be found to use the CG algorithm. All that is needed is its theoretical existence (by positive-definiteness), and the ability to multiply $\mathbf{M}^{-1}$ with a vector in each iteration of the algorithm. When one uses a preconditioner, Shewchuk (1994) advice us to use $\mathbf{r}_i^T\mathbf{M}^{-1}\mathbf{r}_i < \epsilon\mathbf{r}_0^T\mathbf{M}^{-1}\mathbf{r}_0$ as the stopping criterion, which in the CG algorithm has no extra cost. Since $\mathbf{r}^T\mathbf{M}^{-1}\mathbf{r} = \|\mathbf{E}^{-T}\mathbf{r}\|$,

then unless one has a completely worthless preconditioner, the residual will be transformed into a space where each component is on approximately the same scale before checking whether one should terminate.

In the literature,

$$(\mathbf{S}^{-1} + \mathbf{A}^T\mathbf{N}^{-1}\mathbf{A})\mathbf{x} = \mathbf{b}, \tag{3.16}$$

is transformed into the system

$$(\mathbf{1} + \mathbf{F}^T\mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}\mathbf{F})\mathbf{x} = \mathbf{b} \tag{3.17}$$

on grounds of numerical stability (Groeneboom & Eriksen (2009) for anisotropic models) or simpler construction of a preconditioner (Wandelt et al. (2004) for isotropic models). This seems natural as the power spectrum is roughly proportional to ℓ^{-2}, so that the vectors in equation (3.16) span a large range of values.

However, we see that equationx (3.24) corresponds to an additional preconditioning of the system of equation (3.16), using $\mathbf{S}$ as the preconditioner matrix $\mathbf{M}^{-1}$. For an isotropic universe model, the two are almost equivalent when using CG, because $\mathbf{S}^{-1}$ is included in any reasonable preconditioner for equation (3.16), and in the CG algorithm, any vector multiplied with the left hand side matrix is always preconditioned first[7]. However, equation (3.24) may mean slightly less worries about numerical problems when creating the preconditioner itself.

In generalizing to anisotropic models with sparse covariance matrices, the choice between finding some factor $\mathbf{F}$ and applying an inverse $\mathbf{S}^{-1}$ is no longer arbitrary, as the two may have very different behaviours computationally. Numerical stability will however only be an issue in the preconditioner, and is trivially worked around, so that computational efficiency, speed of convergence (including quality of preconditioner), and ease of implementation are our only guides in making the choice between equations (3.16) and (3.24).

3.5.2 The preconditioner

We will here review a simple preconditioner by Eriksen et al. (2004b) for the isotropic case. We then build on this in chapter 4 in order to create a preconditioner for our particular anisotropic model. We note that a more sophisticatated preconditioner is detailed in Smith et al. (2007), but we did not have time to try it.

In an isotropic model, $S_{\ell m,\ell' m'} = C_\ell \delta_{\ell\ell'}\delta_{mm'}$, and a trivial factor of $\mathbf{S}$ is the symmetric factor $\mathbf{S}^{1/2}$, $S^{1/2}_{\ell m,\ell' m'} = \sqrt{C_\ell}\delta_{\ell\ell'}\delta_{mm'}$. Let

$$\mathcal{A} \equiv \mathbf{1} + \mathbf{S}^{1/2}\mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}\mathbf{S}^{1/2}.$$

The upper-left corner of this matrix is shown in figure 3.2 for a particular data set and mask. The lines along (from upper left to lower right) show

[7] One must however use the preconditioned stopping condition, $\mathbf{r}_i^T\mathbf{M}^{-1}\mathbf{r}_i < \epsilon\mathbf{r}_0^T\mathbf{M}^{-1}\mathbf{r}_0$.

strong correlation between different ℓs. The lines across comes from strong correlation between a_{lm} and a_{l-m}. Eriksen et al. (2004b) include similar plots in m-major ordering which show equally difficult patterns, but has the disadvantage that the high values at low ℓs become scattered throughout the entire matrix.

Note that $\mathbf{S}^{1/2}\mathbf{A}^T\mathbf{N}^{-1}\mathbf{A}\mathbf{S}^{1/2}$ is essentially the ratio of signal power (C_ℓ) to noise power. As the noise power becomes larger than signal power on higher ℓs, the components of the matrix approach zero, so that $\mathcal{A}$ is dominated by the identity matrix and approach a diagonal matrix. There is always more correlation present at higher signal-to-noise, that is, at lower ℓ's. This effect only becomes obvious at much higher ℓ's than shown in the plots.

The simple strategy of Eriksen et al. (2004b) is to approximate $\mathcal{A}$ by a dense block for multipoles up to some $\ell_{\rm precond}$, and a diagonal for the rest:

$$\mathbf{M} = \begin{bmatrix} \mathcal{A}_{2:\ell_{\rm precond}} & 0 \\ 0 & \mathrm{diag}(\mathcal{A}_{\ell_{\rm precond}+1:\ell_{\rm max}}) \end{bmatrix}.$$

Below we detail a procedure for computing any element of $\mathbf{N}^{-1}$ in spherical harmonic space explicitly. Since $\mathbf{S}^{1/2}$ and $\mathbf{A}$ are diagonal, it is then trivial to compute corresponding elements of $\mathcal{A}$. Then, we can use $\mathbf{M}^{-1}$ as a preconditioner by doing a Cholesky decomposition of the dense block, which scales as $O(\ell^6_{\rm precond})$, and by trivial inversion of the diagonal block. We choose $\ell_{\rm precond}$ solely as a trade-off between CG convergence speed and the time of doing the Cholesky decomposition, or amount of available memory. Typically $\ell_{\rm precond}$ should be set between 50 and 70.

This preconditioner does not even manage to get the diagonal part right, since

$$\mathrm{diag}(\mathcal{A}^{-1}) \neq \mathrm{diag}(\mathcal{A})^{-1}.$$

However, it suffices for real world needs, using around 15-20 minutes on 8 cores for real world data. Consistent with the notes above, convergence gets worse as more data is gathered and the noise level decreases.

3.5.3 Computing $\mathbf{N}^{-1}$ explicitly in spherical harmonic space

How do we compute $\mathbf{N}^{-1}$ in spherical harmonic space? At least two methods are available for the dense block $\mathbf{N}^{-1}_{2:\ell_{\rm precond}}$, while for computing the diagonal for higher ℓ's we need an explicit expression.

Unit vector hammering Simply construct unit vectors and multiply them with $\mathbf{N}^{-1}$, using the method of going to pixel space and back again. Each unit vector will pick out a single column of $\mathbf{N}^{-1}$. Assuming one only needs the sections of $\mathbf{N}^{-1}$ corresponding to $\ell \leq \ell_{\rm precond}$, one does this $\ell_{\rm precond}$ times, and only include $a_{\ell m}$s up to $\ell_{\rm precond}$ in the transforms.

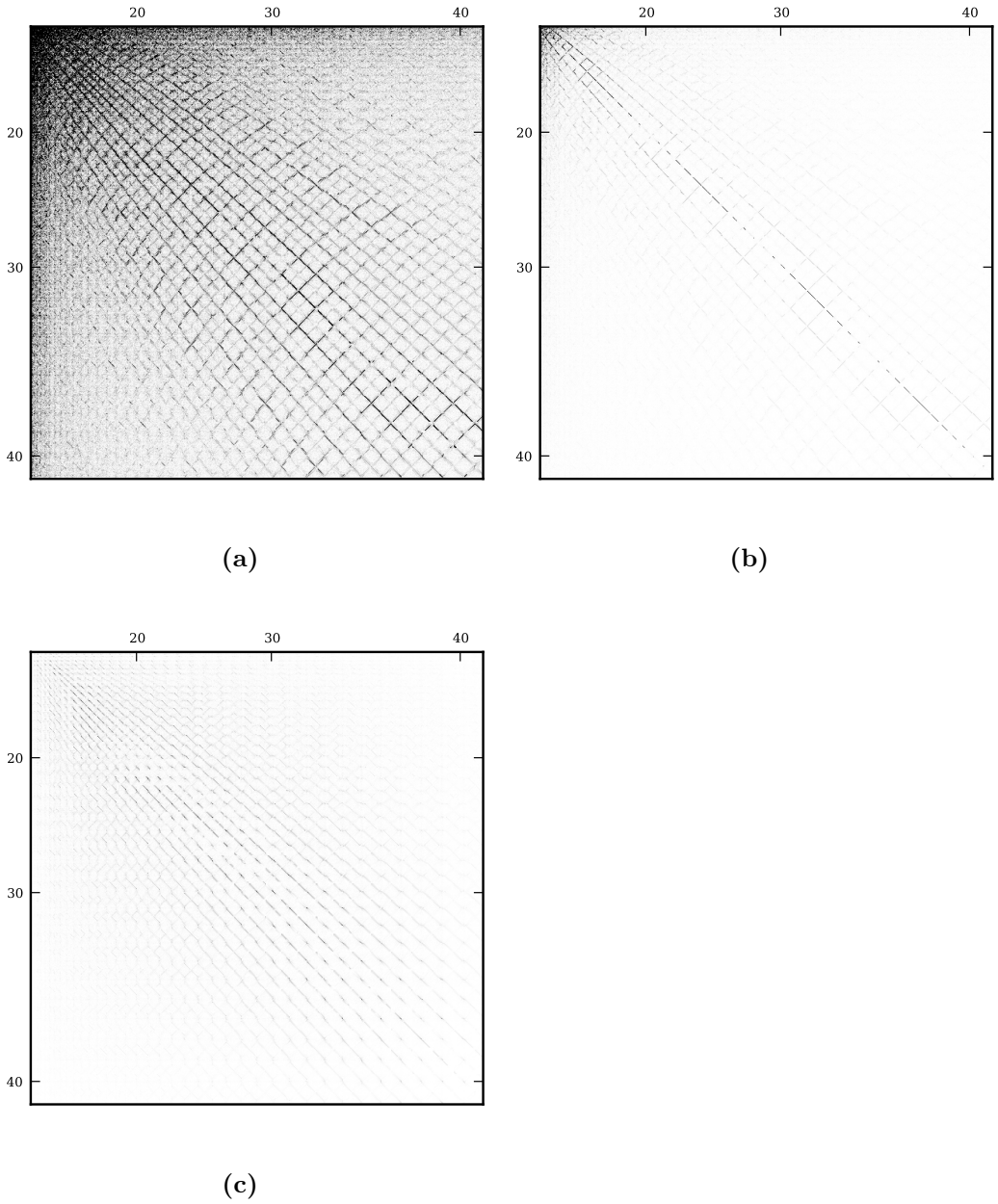


Figure 3.2: **(a)** The upper left of $\mathcal{A}$ ($2 \leq \ell \leq 40$), using the noise properties of the WMAP V1 radiometer and including a mask/galactic cut. Plotted using ℓ-major ordering and in real spherical harmonics (i.e. $\mathbf{U}$ is embedded). Perfect black elements have an absolute value ≥ 10. **(b)** Same as (a), but plotted with a different range so that perfect black elements have an absolute value ≥ 200. **(c)** The inverse of the matrix plotted in (a)-(b). This is $\mathcal{A}^{-1}$ if we pretend that the instrumental beam kills off all signal for $\ell \geq 41$. While a bit artificial, it should capture the main features, in particular in the upper-left quadrant. Perfect black elements have absolute value ≥ 0.1 (which is close to the range of the plot; the maximum elements are close to 0.3). A proper $\mathcal{A}^{-1}$ using the WMAP beam was not plotted for computational reasons.

The advantage of this approach is that it trivially allows for more complicated noise covariances, such as correlated noise or including the monopole and dipole marginalization term $\sigma_t^2 \mathbf{T}\mathbf{T}^T$. However, as the method scales as $O(\ell_{\rm precond}^5)$, it becomes the limiting factor for how high $\ell_{\rm precond}$ can be chosen.

Explicit expression In the case of independent pixel noise, there is a fast explicit expression (e.g., Eriksen et al., 2004b) which we derive here. We start with a derivation in complex spherical harmonics, and then note how it is transformed to real spherical harmonics.

We start with writing out a single component of the matrix $\tilde{\mathbf{N}}^{-1} = \mathbf{Y}^\dagger \hat{\mathbf{N}}^{-1} \mathbf{Y}$. Assuming $\hat{\mathbf{N}}^{-1}$ is diagonal with diagonal elements $\eta(\hat{p}_i) = \sigma_i^{-2}$, then simply writing out the matrix multiplication explicitly gives

$$(\tilde{\mathbf{N}}^{-1})_{\ell m,\ell' m'} = \sum_{i=1}^{N_{\rm pix}} Y^*_{\ell m}(\hat{p}_i)\eta(\hat{p}_i)Y_{\ell' m'}(\hat{p}_i) \tag{3.18}$$

$$\approx \int \frac{N_{\rm pix}}{4\pi}\eta(\hat{p})Y^*_{\ell m}(\hat{p})Y_{\ell' m'}(\hat{p})d\Omega, \tag{3.19}$$

where the first line is a quadrature of the latter integral. Note that equation (3.18) in this context is the exact version since it specifies exactly the arithmetic that will happen in the CG search when multiplying with $\mathbf{N}^{-1}$. However, approximation is not a problem in a preconditioner, and the approximation clearly gets better as $N_{\rm pix}$ grows.

Note that we have pretended that $\eta(\hat{p})$ is some continuous field on the unit sphere from which our $\eta(\hat{p}_i) = \sigma_i^{-2}$ are samples. This is a rather meaningless quantity, since σ_i is the noise of a particular pixel and not a sample from a field on the sphere. However, for the purposes of our computation, we can still transform the inverse noise variance map into spherical harmonics *as if* it was samples from such a field, and then let

$$\eta(\hat{p}) = \sum_{\ell'' m''} \eta_{\ell'' m''} Y_{\ell'' m''}(\hat{p}). \tag{3.20}$$

Thus we get

$$(\tilde{\mathbf{N}}^{-1})_{\ell m,\ell' m'} \approx \frac{N_{\rm pix}}{4\pi} \sum_{\ell'' m''} \eta_{\ell'' m''} \int Y^*_{\ell m}(\hat{p})Y_{\ell' m'}(\hat{p})Y_{\ell'' m''}(\hat{p})d\Omega \tag{3.21}$$

$$= \frac{N_{\rm pix}}{4\pi}(-1)^m \sum_{\ell'' m''} \eta_{\ell'' m''} \sqrt{\frac{(2\ell+1)(2\ell'+1)(2\ell''+1)}{4\pi}} \times \tag{3.22}$$

$$\times \begin{pmatrix} \ell & \ell' & \ell'' \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ell & \ell' & \ell'' \\ -m & m' & m'' \end{pmatrix} \tag{3.23}$$

where the factors on the last line are so-called Wigner 3j-symbols, for which computer code is available (see appendix A.3). We have also used the fact that

$Y_{\ell m}(\hat{p})^* = (-1)^m Y_{\ell,-m}(\hat{p})$. The last Wigner 3j-symbol is 0 unless $m'' = m' - m$, so the sum only needs to happen over ℓ''. It is also 0 unless $\ell'' \leq \ell + \ell'$, so one only has to expand the η-map up to $2\ell_{\max}$.

Above, complex spherical harmonics were implied. To use the result in the CG preconditioner we must make sure to convert vectors back and forth when applying the preconditioner. Since we have

$$\mathbf{M}_R^{-1} = (\mathbf{U}\mathbf{M}_C\mathbf{U}^\dagger)^{-1} = \mathbf{U}\mathbf{M}_C^{-1}\mathbf{U}^\dagger,$$

this is straightforward. Alternatively, one can precompute $\mathbf{N}_R^{-1} = \mathbf{U}\mathbf{N}_C^{-1}\mathbf{U}^\dagger$. The dense block is easily found using the approach outlined in appendix A.2. Computing the diagonal part of $\mathbf{N}_R^{-1}$is slightly non-trivial, but an explicit expression is given in Result 4 in appendix A.2. To compute $(\mathbf{N}_R^{-1})_{\ell m,\ell m}$ and $(\mathbf{N}_R^{-1})_{\ell -m,\ell -m}$, one needs both $(\mathbf{N}_C^{-1})_{\ell m,\ell m}$ and $(\mathbf{N}_C^{-1})_{\ell m,\ell -m}$. That is, both the diagonal and anti-diagonal is needed[8] for each ℓ-block in $\mathbf{N}_C^{-1}$.

3.5.4 Including monopole and dipole marginalization in the preconditioner

In the end, the system we want to solve is

$$(\mathbf{1} + \mathbf{F}^T\mathbf{A}^T(\mathbf{N} + \sigma_f^2\mathbf{T}\mathbf{T}^T)^{-1}\mathbf{A}\mathbf{F})\mathbf{x} = \mathbf{b},$$

so we should include $\sigma_f^2\mathbf{T}\mathbf{T}^T$ in the preconditioner. The simplest approach (and the one we actually implemented) is to use the unit vector hammering approach outlined in the previous section, and ignore the $\sigma_f^2\mathbf{T}\mathbf{T}^T$ term in the diagonal part of the preconditioner. This approach cost less than 10% additional iterations, compared to not including the marginalization term.

Not having implemented it, it is unclear whether a better preconditioner will help, but we outline an approach for completeness, which at any rate should be computionally faster than the unit vector hammering approach for constructing the dense block. Recalling the Sherman-Morrison-Woodbury formula, we have

$$\begin{aligned}(\mathbf{N} + \sigma_t^2\mathbf{T}\mathbf{T}^T)^{-1} &= \mathbf{N}^{-1} - \mathbf{N}^{-1}\mathbf{T}\left(\frac{1}{\sigma_t^2}\mathbf{1} + \mathbf{T}^T\mathbf{N}^{-1}\mathbf{T}\right)^{-1}\mathbf{T}^T\mathbf{N}^{-1} \\ &= \mathbf{N}^{-1} - \mathbf{N}^{-1}\mathbf{T}\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{T}^T\mathbf{N}^{-1},\end{aligned}$$

where we diagonalize the symmetric 4-by-4 matrix $\left(\frac{1}{\sigma_t^2}\mathbf{1} + \mathbf{T}^T\mathbf{N}^{-1}\mathbf{T}\right)^{-1} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ by an eigenvalue decomposition. Having found $\mathbf{D}$ and $\mathbf{V}$ in pixel

[8] The Commander software currently gets this wrong, and applies the diagonal of $\mathbf{N}_C^{-1}$ to vectors of real spherical harmonics in the CG preconditioner. However, the preconditioner turned out to be equally efficient and the CG search terminates in the same number of iterations. Perhaps this is because the difference just happens to be small, or perhaps this is because the $\mathrm{diag}(\mathcal{A}^{-1}) \neq \mathrm{diag}(\mathcal{A})^{-1}$ issue means that one is already approximate, and being a little more approximate doesn't matter.

space, we multiply with $\mathbf{Y}$ to transition to spherical harmonic space,

$$\mathbf{Y}^\dagger(\mathbf{N}+\sigma_t^2\mathbf{T}\mathbf{T}^T)^{-1}\mathbf{Y} = \mathbf{Y}^\dagger\mathbf{N}^{-1}\mathbf{Y} - \mathbf{Y}^\dagger\mathbf{N}^{-1}\mathbf{T}\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{T}^T\mathbf{N}^{-1}\mathbf{Y}.$$

Any component of $\mathbf{Y}^\dagger\mathbf{N}^{-1}\mathbf{Y}$ can be found explicitly using the method outlined in the previous section, while explicit elements of the latter matrix are efficiently computed since $\mathbf{D}$ is a 4-by-4 diagonal matrix:

$$(\mathbf{N}^{-1}\mathbf{T}\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{T}^T\mathbf{N}^{-1})_{\ell m,\ell' m'} = \sum_{i=1}^{4} D_{ii}(\mathbf{Y}^\dagger\mathbf{N}^{-1}\mathbf{T}\mathbf{V})_{\ell m,i}(\mathbf{Y}^\dagger\mathbf{N}^{-1}\mathbf{T}\mathbf{V})^*_{\ell' m',i}.$$

Chapter 4

Modelling hemispherical power asymmetry

4.1 Modulation

Our main topic is investigating the statistical strength of the apparent hemispherical power asymmetry. To do this we use a very simple model: Imagine that the isotropic model is correct, except that the fluctuations are stronger in a preferred direction, weaker in the opposite direction, with a smooth transition in-between. This is modelled by having an isotropic signal being modulated (multiplied point-wise) with a modulation field γ,

$$s(\hat{n}) = \gamma(\hat{n}) s_{\text{iso}}(\hat{n}). \tag{4.1}$$

Here s_{iso} is the isotropic signal, assumed to be Gaussian with a covariance matrix in harmonic space given by $S_{\text{iso},\ell m,\ell' m'} = C_\ell \delta_{\ell\ell'} \delta_{mm'}$. In particular we will focus on dipole modulation,

$$s(\hat{n}) = (1 + \alpha(\hat{p} \cdot \hat{n})) s_{\text{iso}}(\hat{n}), \tag{4.2}$$

where $\hat{p}$ is the preferred direction of the dipole modulation field and $\alpha \in [0, 1]$ is the strength of the field. The standard isotropic model corresponds to $\alpha = 0$, while $\alpha = 1$ means that there are no fluctuations in the point opposite to $\hat{p}$ on the sphere.

We will not propose any physical motivation for setting up this model, but it can be well justified phenomenologically. If there exists hemispherical power asymmetry, it is not unreasonable to model it as some modulation on top of an isotropic signal, since the isotropic model fits so well. Any such modulations can be expanded into spherical harmonics, and, as we will see below, equation (4.2) simply takes the first two multipoles of this expansion. This is very similar to standard linear regression. In most settings the underlying function is not perfectly linear, but one still attempts to pick up any linear trend.

It will be useful to control the range of multipoles where we assume that modulation is happening. This turns out equation (4.1) into

$$s(\hat{n}) = \sum_{\ell} \gamma^{(\ell)}(\hat{n}) s^{(\ell)}_{\text{iso}}(\hat{n}), \tag{4.3}$$

where $s^{(\ell)}_{\text{iso}}$ denotes the isotropic signal with everything but the ℓ-modes filtered out. For the special case of dipole modulation we introduce α_ℓ,

$$s(\hat{n}) = \sum_{\ell} (1 + \alpha_\ell (\hat{p} \cdot \hat{n})) s^{(\ell)}_{\text{iso}}(\hat{n}). \tag{4.4}$$

4.2 The covariance matrix of the modulated signal

Since the modulation equation (4.3) is clearly a linear operation, the modulated signal is still Gaussian with vanishing mean, and can be fully described by a covariance matrix $\mathbf{S}$. To fit the model to data, we need a way to compute this covariance matrix given model parameters.

If one simply wishes to evaluate expressions such as $\mathbf{S}^{1/2}\mathbf{x}$ or $\mathbf{S}^{-1}\mathbf{x}$, it could have been possible to simply use equation (4.3) directly in real space. However, to evaluate the posterior distribution of the α_ℓ and $\hat{p}$ parameters we also need the determinant of $\mathbf{S}$. Therefore, we will in this section develop explicit expressions for $\mathbf{S}$. In harmonic space, the resulting matrix is quite sparse, so it turns out that this approach is also computationally faster than going back and forth between pixel and harmonic space, potentially *much* faster for complicated choices of α_ℓ. This approach also also provides some nice insights into what is really going on when the dipole modulation is applied.

The strategy will be to first find an expression for the modulation itself in harmonic space. Equation (4.3) is linear and can be written

$$\mathbf{s} = \mathbf{M}\mathbf{s}_{\text{iso}} \tag{4.5}$$

for some modulation matrix $\mathbf{M}$. Once $\mathbf{M}$ is found, it is easy to find $\mathbf{S}$; since the signal has zero mean, we have

$$\mathbf{S} = \text{Var}(\mathbf{s}) = \mathbf{M}\,\text{Var}(\mathbf{s}_{\text{iso}})\,\mathbf{M}^\dagger = \mathbf{M}\mathbf{S}_{\text{iso}}\mathbf{M}^\dagger. \tag{4.6}$$

The following two sections contain results for $\mathbf{M}$ that are valid for arbitrary γ, while in the rest of the thesis we use $\mathbf{M}$ to denote a dipole modulation.

4.2.1 The modulation operation in spherical harmonic space

We want to find $\mathbf{M}$, where

$$\mathbf{s} = \mathbf{M}\mathbf{s}_{\text{iso}}.$$

We start with expanding the modulation field for each scale ℓ'' in spherical harmonics, $\gamma^{(\ell'')}(\hat{n}) = \sum_{LM} \gamma^{(\ell'')}_{LM} Y_{LM}(\hat{n})$. Then we consider the relationship in spherical harmonic space:

$$\begin{aligned}
s_{\ell m} &= \int \sum_{\ell''} (1 + \alpha_{\ell''}(\hat{p} \cdot \hat{n})) s^{(\ell'')}_{\text{iso}}(\hat{n}) Y^*_{\ell m}(\hat{n}) d\Omega \\
&= \int \sum_{\ell''} \left(\sum_{LM} \gamma^{(\ell'')}_{LM} Y_{LM}(\hat{n}) \right) \left(\sum_{\ell' m'} s^{(\ell'')}_{\text{iso},\ell' m'} Y_{\ell' m'}(\hat{n}) \right) Y^*_{\ell m}(\hat{n}) d\Omega \\
&= \sum_{\ell''} \sum_{\ell' m'} \sum_{LM} \gamma^{(\ell'')}_{LM} s^{(\ell'')}_{\text{iso},\ell' m'} \int Y^*_{\ell m}(\hat{n}) Y_{\ell' m'}(\hat{n}) Y_{LM}(\hat{n}) d\Omega \\
&= \sum_{\ell' m'} \left(\sum_{LM} (-1)^m \gamma^{(\ell')}_{LM} \int Y_{\ell -m}(\hat{n}) Y_{\ell' m'}(\hat{n}) Y_{LM}(\hat{n}) d\Omega \right) s_{\text{iso},\ell' m'},
\end{aligned}$$

since $s^{(\ell'')}_{\text{iso},\ell' m'} = 0$ when $\ell' \neq \ell''$. Demanding that $s_{\ell m} = \sum_{\ell' m'} M_{\ell m, \ell' m'} s_{\text{iso},\ell' m'}$, it is then clear that

$$M_{\ell m, \ell' m'} = (-1)^m \sum_{LM} \gamma^{(\ell')}_{LM} \int Y_{\ell -m}(\hat{n}) Y_{\ell' m'}(\hat{n}) Y_{LM}(\hat{n}) d\Omega.$$

The integral is a Gaunt integral, described in appendix A.4, so we write

$$M_{\ell m, \ell' m'} = (-1)^m \sum_{LM} \gamma^{(\ell')}_{LM} Y^{\ell, \ell', L}_{-m, m', M}. \tag{4.7}$$

The Gaunt integral vanish unless $|\ell - \ell'| \leq L$, and unless $\ell + \ell' + L$ is even, so bandwidth limited modulation fields where $\gamma_{LM} = 0$ for $L > L_{\text{max}}$ have the potential to form matrices $\mathbf{M}$ that are rather sparse. We return to this for the dipole case below.

4.2.2 Azimuthally symmetric modulation fields

The matrix $\mathbf{M}$ contains couplings in both ℓ and m. Treated as a dense matrix, memory consumption and matrix-vector multiplication scale as $O({\ell_{\text{max}}}^4)$. Treated as a sparse matrix, couplings in both ℓ and m can create patterns that make decompositions less efficient.

In the case of modulation fields that are azimuthally symmetric around some preferred direction $\hat{p}$, the couplings in m can to some degree be worked around. We first fix $\hat{p}$ along the z-axis, and denote the corresponding modulation operation $\mathbf{M}_{\hat{z}}$. In this case, the spherical harmonic expansion γ_{LM} is non-zero only when $M = 0$, so we have

$$M_{\hat{z}, \ell m, \ell' m'} = (-1)^m \sum_{L=0}^{L_{\text{max}}} \gamma^{(\ell')}_{L,0} Y^{\ell, \ell', L}_{-m, m', 0}.$$

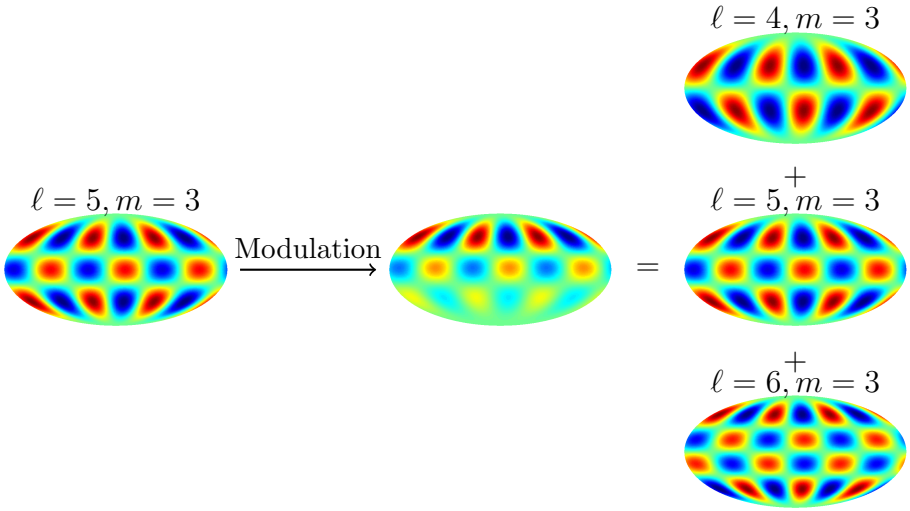


Figure 4.1: Example of the modulation of a single mode. On the left is shown a single spherical harmonic mode, $a_{5,3} = 1 + i$. It is then modulated by the field $f(\hat{n}) = (1 + 0.9(\hat{n} \cdot \hat{z}))$. The result is effectively a combination of the original mode plus two new modes which build up the modulation. Note how the extra modes strengthen each other on the northern hemisphere and cancel each other on the southern. If the modulation dipole had been oriented in some other direction, additional modes for $m \pm 1$ would have been required as well to build up the modulation.

Because $Y^{\ell,\ell',L}_{-m,m',0}$ vanish whenever $m + m' \neq 0$, we see that $M_{\hat{z},\ell m,\ell' m'}$ vanish whenever $m \neq m'$. That is, in m-major ordering, $\mathbf{M}_{\hat{z}}$ is block-diagonal with no couplings between different m's. Since $Y^{\ell,\ell',L}_{-m,m',0}$ also vanish whenever $|\ell - \ell'| > L$, and $\gamma^{(\ell')}_{L,0} = 0$ for $L > L_{\max}$, we have that $\mathbf{M}_{\hat{z}}$ is also band-diagonal with bandwidth $L_{\max}$. So, memory consumption and matrix-vector multiplication scale as $\min(O({\ell_{\max}}^3), O(L_{\max}{\ell_{\max}}^2))$.

It is possible to rotate a spherical map solely by operating on its spherical harmonic coefficients, using the Wigner **D**-matrix (see appendix A.5). We denote the rotation such that a map gets its z-axis rotated to $\hat{p}$ as $\mathbf{R}$. This rotation is not unique, since after rotating a map in this fashion one can freely rotate around the $\hat{p}$ axis, but this extra freedom does not matter for our purposes. $\mathbf{R}$ is a unitary matrix, that is, $\mathbf{R}^{-1} = \mathbf{R}^\dagger$.

Clearly, rotating a map so that the modulation direction $\hat{p}$ is oriented along the $\hat{z}$ axis, applying the modulation in that coordinate system, and rotating the final map back, has the same effect as applying the modulation directly. Therefore, $\mathbf{M} = \mathbf{R}\mathbf{M}_{\hat{z}}\mathbf{R}^\dagger$, and we can make use of $\mathbf{M}_{\hat{z}}$ also in the general case. Since applying $\mathbf{R}$ scales as $O({\ell_{\max}}^3)$, this can lead to savings in many situations.

4.2.3 Dipole modulation fields

We now specialize to dipole modulation fields. From here on and out,

$$\gamma^{(\ell')}(\hat{n}) = 1 + \alpha_{\ell'}\hat{p}\cdot\hat{n},$$

and $\mathbf{M}$ denotes the corresponding linear modulation operation. According to Result 1 in appendix A.1, $\gamma^{(\ell')}_{LM} = 0$ for $L > 1$, fixing $L_{\max} = 1$. Since the Gaunt integral $Y^{\ell,\ell',L}_{-m,m',M}$ vanish unless $|\ell - \ell'| \leq L \leq 1$, and unless $\ell + \ell' + L$ is even, we find that equation (4.7) can be written

$$\begin{aligned} M_{\ell m,\ell' m'} =& (-1)^m \sum_{LM} \gamma^{(\ell')}_{LM} Y^{\ell,\ell',L}_{-m,m',M} \\ =& (-1)^m \, \gamma^{(\ell')}_{|\ell-\ell'|,m-m'} Y^{\ell,\ell',|\ell-\ell'|}_{-m,m',m-m'}. \end{aligned} \tag{4.8}$$

On the diagonal, things simplify further: Since $Y^{\ell,\ell,0}_{-m,m,0} = (-1)^m\sqrt{1/4\pi}$ by equation (A.23), and we know that $\gamma^{(\ell)}_{00} = \sqrt{4\pi}$, we have

$$M_{\ell m,\ell m} = (-1)^{m+m}\frac{\sqrt{4\pi}}{\sqrt{4\pi}} = 1. \tag{4.9}$$

In summary,

$$M_{\ell m,\ell' m'} = \begin{cases} 1 & \text{if } \ell = \ell', m = m' \\ (-1)^m \gamma^{(\ell')}_{1,m-m'} Y^{\ell,\ell',1}_{-m,m',m-m'} & \text{if } |m - m'| \leq |\ell - \ell'| = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

Thus $\mathbf{M}$ turns out to be quite sparse and only a few modes are needed to build up a single dipole modulated mode (see figure 4.1). Finally, we take a look at the symmetry properties. On the off-diagonal elements we have

$$\begin{aligned} M^*_{\ell' m',\ell m} &= (-1)^{m'} \gamma^{(\ell)*}_{1,m'-m} Y^{\ell,\ell',1}_{-m',m,m'-m} \\ &= (-1)^{m'}(-1)^{m'-m} \gamma^{(\ell)}_{1,m-m'} Y^{\ell',\ell,1}_{-m',m,m'-m} \\ &= (-1)^m \gamma^{(\ell)}_{1,m-m'} Y^{\ell,\ell',1}_{-m,m',m-m'}, \end{aligned} \tag{4.11}$$

by first using that $a_{\ell m} = (-1)^m a^*_{\ell -m}$, and then a symmetry property of the Gaunt integral. Since the ℓ-dependency of $\gamma^{(\ell)}_{1,m}$ comes from a term α_ℓ, $\alpha_{\ell'}\gamma^{(\ell)}_{1,m} = \alpha_\ell \gamma^{(\ell')}_{1,m}$, and we get

$$\alpha_{\ell'} M^*_{\ell' m',\ell m} = \alpha_\ell M_{\ell m,\ell' m'}. \tag{4.12}$$

So depending on the choice of α_ℓ, $\mathbf{M}$ can have Hermitian sub-blocks or be entirely Hermitian. We will discuss the choice of α_ℓ further in section 4.2.5.

Finally, as the dipole field is obviously azimuthally symmetric, we consider $\mathbf{M}_{\hat{z}}$ with a dipole modulation field. Using Result 1 in appendix A.1 again, we see that when fixing the preferred direction $\hat{p} = \hat{z}$ we have

$$\gamma^{(\ell')}_{00} = \sqrt{4\pi}, \qquad \gamma^{(\ell')}_{10} = \sqrt{4\pi/3}\alpha_{\ell'}, \qquad \gamma^{(\ell')}_{11} = \gamma^{(\ell')}_{1-1} = 0.$$

So,

$$\begin{aligned} M_{\hat{z},\ell m,\ell' m} &= (-1)^m \gamma^{(\ell')}_{1,0} Y^{\ell,\ell',1}_{-m,m,0} \\ &= \gamma^{(\ell')}_{1,0} \sqrt{\frac{3(\ell^*+m+1)(\ell^*-m+1)}{4\pi(2\ell^*+1)(2\ell^*+3)}} \\ &= \alpha_{\ell'} \sqrt{\frac{(\ell^*+m+1)(\ell^*-m+1)}{(2\ell^*+1)(2\ell^*+3)}} \end{aligned}$$

where $\ell^* = \min(\ell, \ell')$. The explicit expression for $Y^{\ell,\ell',1}_{-m,m,0}$ is computed in appendix A.4, although nothing would be lost by computing $Y^{\ell,\ell+1,1}_{-m,m,0}$ with the help of computer codes for the Wigner 3j symbol instead.

In summary, when $\hat{p} = \hat{z}$ and with $\ell^* = \min(\ell, \ell')$, we have

$$M_{\hat{z},\ell m,\ell' m'} = \begin{cases} 1 & \text{when } \ell = \ell', m = m' \text{ (by eq. (4.9))} \\ \alpha_{\ell'} \sqrt{\frac{(\ell^*+m+1)(\ell^*-m+1)}{(2\ell^*+1)(2\ell^*+3)}} & \text{when } |\ell - \ell'| = 1, m = m' \\ 0 & \text{otherwise.} \end{cases} \tag{4.13}$$

It is now easy to see that there is no general trend in ℓ, so that the magnitude of the signature should be roughly constant as ℓ increases.

Note that $\mathbf{M}_{\hat{z}}$ is now tri-diagonal in m-major ordering. We can therefore use highly efficient LAPACK routines for tri-diagonal matrices. It is also very fast to construct, since the diagonal is all ones, and the off-diagonals can simply be scaled by α_ℓ from instance to instance.

4.2.4 In real spherical harmonics

So far we have worked exclusively with complex spherical harmonics, but sometimes we want to use real spherical harmonics instead (see appendix A.2). An advantage with letting $\hat{p} = \hat{z}$ is that the matrix is the same whether written in real or complex spherical harmonics. From equation (4.13) it is clear that $M_{\hat{z},\ell m,\ell' m'} = \delta_{mm'} M_{\hat{z},\ell m,\ell' m'}$ and that $M_{\hat{z},\ell m,\ell' m} = M_{\hat{z},\ell -m,\ell' -m}$. So $\mathbf{M}_{\hat{z}}$ satisfies the conditions put on $\mathbf{J}$ in Result 2 in appendix A.2, and we have

$$\mathbf{U}\mathbf{M}_{\hat{z}}\mathbf{U}^\dagger = \mathbf{M}_{\hat{z}}. \tag{4.14}$$

Thus $\mathbf{M}_{\hat{z}}$ has the same representation for both complex and real spherical harmonics. We also have $\mathbf{U}\mathbf{S}_{\text{iso}}\mathbf{U}^\dagger = \mathbf{S}_{\text{iso}}$ by this result. Therefore, $\mathbf{S}_{\hat{z}}$ is also the same in real and complex spherical harmonics, since

$$\mathbf{U}\mathbf{S}_{\hat{z}}\mathbf{U}^\dagger = \mathbf{U}\mathbf{M}_{\hat{z}}\mathbf{U}^\dagger\mathbf{U}\mathbf{S}_{\text{iso}}\mathbf{U}^\dagger\mathbf{U}\mathbf{M}_{\hat{z}}^\dagger\mathbf{U}^\dagger = \mathbf{M}_{\hat{z}}\mathbf{S}_{\text{iso}}\mathbf{M}_{\hat{z}}^\dagger = \mathbf{S}_{\hat{z}}.$$

While useful now and then for speed and simplicity, it doesn't apply in the situation where it is really needed: Constructing preconditioners for the Conjugate Gradients algorithm. One then needs an explicit expression for each

element of the final matrix, so one cannot factor out the rotation. However, $\mathbf{U}$ only contains $O({\ell_{\max}}^2)$ non-zero elements, so it works well enough to construct a given matrix in complex spherical harmonics and then carry out the matrix multiplication.

4.2.5 Consequences of choice of α_ℓ

In practice, we only consider the signal in harmonic space over a finite subset of ℓs (usually $\ell_{\min} \leq \ell \leq \ell_{\max}$, where $\ell_{\min} = 2$ and $\ell_{\max}$ is set by band-limitation). $\mathbf{M}$ is then not in general a square matrix. For instance, if $a_{\ell m}$ are coefficients with covariance matrix $\mathbf{M}\mathbf{S}_{\text{iso}}\mathbf{M}^\dagger$, then the modulated $a_{2,m}$ will get some power from the isotropic $a_{1,m}$ component, and so if we let $\ell_{\min} = 2$ then $\mathbf{S}_{\text{iso}}$ must also contain information about C_1. The same effect applies around $\ell_{\max}$ (but here the beam kills off the signal so that it doesn't matter either way).

There are several choices here, and because we are working with a purely phenomenological model the choice is somewhat arbitrary. Our choice is to do what is simplest in terms of implementation: We simply decide to let $\alpha_{\ell_{\min}-1} = \alpha_{\ell_{\max}+1} = 0$. One could however make other choices here. The reason this choice is so convenient is that we recover a square $\mathbf{M}$.

4.2.6 Other properties of M

Assuming a square $\mathbf{M}$, we will now show sufficient conditions for $\mathbf{M}$ to be invertible, and, when it is Hermitian (constant α_ℓ) that it is positive definite. This is convenient because it is then straightforward to solve linear systems in $\mathbf{S}$ through the decomposition $(\mathbf{M}\mathbf{S}_{\text{iso}}^{1/2})(\mathbf{M}\mathbf{S}_{\text{iso}}^{1/2})^\dagger$ (other methods of solving the system is discussed in section 4.3).

Since $\mathbf{R}$ is unitary, we have that $\mathbf{M}$ is non-singular (positive definite) if and only if $\mathbf{M}_{\hat{z}}$ is non-singular (positive definite). The strategy is to find conditions under which $\mathbf{M}_{\hat{z}}^\dagger$ is diagonally dominant, that is, that we have $|M_{\ell m,\ell m}| > \sum_{\ell' m',\ell\neq\ell',m\neq m'} |M_{\ell' m',\ell m}|$ (note that we will work with column-wise diagonal dominance). It is well known (e.g. Harville, 1997, pp. 279) that such matrices are non-singular. Also, if the diagonal elements are all positive and the matrix is Hermitian, then the matrix is positive definite. Note that these are sufficient, but not necessary, conditions.

The diagonal of $\mathbf{M}_{\hat{z}}$ is all ones. With $\ell^* = \min(\ell, \ell')$, the off-diagonal elements are

$$\alpha_\ell \sqrt{\frac{(\ell^* + m + 1)(\ell^* - m + 1)}{(2\ell^* + 1)(2\ell^* + 3)}}$$

when $|\ell - \ell'| = 1$ and $m = m'$, and 0 otherwise. The numerator is maximized at $m = 0$:

$$(\ell^* + m + 1)(\ell^* - m - 1) = (\ell^*)^2 + 2\ell^* - m^2 + 1 \leq (\ell^*)^2 + 2\ell^* + 1 = (\ell^* + 1)^2.$$

The denominator can be written as

$$(2\ell *+1)(2\ell^* + 3) = 4(\ell^* + 1)^2 - 1,$$

so

$$\sqrt{\frac{(\ell^* + m + 1)(\ell^* - m + 1)}{(2\ell^* + 1)(2\ell^* + 3)}} \leq \frac{\alpha_{\ell'}}{2}\sqrt{\frac{(\ell^* + 1)^2}{(\ell^* + 1)^2 - \frac{1}{4}}}.$$

This is clearly maximized at $\ell^* = 0$, so for the non-zero off-diagonal $M_{\hat{z},\ell m,\ell' m'}$ we have $M_{\hat{z},\ell m,\ell' m'} \leq \alpha_{\ell'}\sqrt{1/3} \approx 0.58\alpha_\ell$. Columns with only one off-diagonal element ($\ell' \in \{\ell_{\min}, \ell_{\max}\}$) always satisfy diagonal dominance. For inner ℓ's, a column consists of one super-diagonal element with $\ell = \ell' - 1$ (so $\ell^* = \ell' - 1$), the unity diagonal element, and one sub-diagonal element with $\ell = \ell' + 1$ (so $\ell^* = \ell'$). Therefore we have diagonal dominance in such columns if

$$\frac{\alpha_{\ell'}}{2}\left(\sqrt{\frac{\ell'^2}{\ell'^2 - \frac{1}{4}}} + \sqrt{\frac{(\ell' - 1)^2}{(\ell' - 1)^2 - \frac{1}{4}}}\right) < 1$$

Since the square root decrease monotonically with ℓ',

$$\frac{\alpha_{\ell'}}{2}\left(\sqrt{\frac{\ell'^2}{\ell'^2 - \frac{1}{4}}} + \sqrt{\frac{(\ell' + 1)^2}{(\ell' + 1)^2 - \frac{1}{4}}}\right) < \alpha_{\ell'}\sqrt{\frac{(\ell' + 1)^2}{(\ell' + 1)^2 - \frac{1}{4}}} < \alpha_{\ell'}\sqrt{\frac{4}{3}}.$$

So a sufficient condition for $\mathbf{M}_{\hat{z}}$ to be diagonally dominant is that $\alpha_{\ell'} \leq \sqrt{3/4} \approx 0.87$. Under the same condition, $\mathbf{M}$ is invertible and, when Hermitian, positive definite. For our purposes, α_ℓ will never come anywhere close to this value, and so we are satisfied.

4.3 Computations with S

What is the best way of doing computations involving $\mathbf{S}$? In order to solve linear systems $\mathbf{S}^{-1}\mathbf{x} = \mathbf{b}$ or find the determinant $|\mathbf{S}|$ we need to somehow factor $\mathbf{S}$. Since $\mathbf{S}$ is by construction Hermitian and positive definite, we can find some factor $\mathbf{F}$ such that $\mathbf{S} = \mathbf{F}\mathbf{F}^\dagger$. There are many possibilities, and our choice is determined solely by computation speed.

4.3.1 Factors of S

One way of producing a factor is through Cholesky factorization, using code such as CHOLMOD (see appendix A.6):

$$\mathbf{P}\mathbf{S}\mathbf{P}^\dagger = \mathbf{L}\mathbf{L}^\dagger.$$

Here $\mathbf{P}$ is a permutation matrix and $\mathbf{L}$ is lower-triangular. Cholesky factors are not sparse in general, and the amount of fill-in (new non-zero elements created) is heavily affected by the permutation of rows and columns chosen.

Sparse Cholesky libraries will try to find a good one. For a general preferred direction $\hat{p}$, CHOLMOD found permutations that yielded factors with about 12-13 times as many non-zero elements as $\mathbf{S}$ itself for $\ell_{\text{mod}} \sim 100$. This ratio does however increase with ℓ_{mod}.

This is however not the only choice. We have already seen that $\mathbf{S}$ can be written in a variety of ways:

$$\mathbf{S} = \mathbf{R}\mathbf{S}_{\hat{z}}\mathbf{R}^\dagger = \mathbf{M}\mathbf{S}_{\text{iso}}\mathbf{M}^\dagger = \mathbf{R}\mathbf{M}_{\hat{z}}\mathbf{S}_{\text{iso}}\mathbf{M}_{\hat{z}}^\dagger\mathbf{R}^\dagger,$$

where the latter follows from $\mathbf{R}\mathbf{S}_{\text{iso}}\mathbf{R}^\dagger = \mathbf{S}_{\text{iso}}$ and $\mathbf{R}^\dagger\mathbf{R} = \mathbf{1}$. These ways of writing $\mathbf{S}$ each give rise to a natural factor $\mathbf{F}$,

$$\mathbf{P}^\dagger\mathbf{L}, \quad \mathbf{R}\mathbf{L}_{\hat{z}}, \quad \mathbf{M}\mathbf{S}_{\text{iso}}^{1/2}, \quad \text{and} \quad \mathbf{R}\mathbf{M}_{\hat{z}}\mathbf{S}_{\text{iso}}^{1/2}. \tag{4.15}$$

We treat them in order:

$\mathbf{P}^\dagger\mathbf{L}\mathbf{P}$ – See above. As mentioned, the computational complexity of this decomposition is difficult to assess up front (except that it lies somewhere between $O({\ell_{\text{max}}}^2)$ and $O({\ell_{\text{max}}}^6)$...) and benchmarks will be our only guide.

$\mathbf{R}\mathbf{L}_{\hat{z}}\mathbf{R}^\dagger$ – Applying a rotation $\mathbf{R}$ or its inverse scales as $O({\ell_{\text{max}}}^3)$. $\mathbf{S}_{\hat{z}}$ is penta-diagonal (in m-major ordering) because it is the product of two tri-diagonal matrices ($\mathbf{M}_{\hat{z}}\mathbf{S}_{\text{iso}}\mathbf{M}_{\hat{z}}^\dagger$), so its Cholesky factor $\mathbf{L}_{\hat{z}}$ has two sub-diagonals and all operations scale as $O({\ell_{\text{max}}}^2)$. Computationally, $\mathbf{L}_{\hat{z}}^{-1}\mathbf{x}$ seems to be very close to $(\mathbf{M}_{\hat{z}}\mathbf{S}_{\text{iso}}^{1/2})^{-1}\mathbf{x}$: The former requires one solve with the Cholesky factor of a penta-diagonal matrix, while the latter requires a solve for each of the two LU factors of a tri-diagonal matrix. The latter option will certainly be faster to construct and for multiplication. We therefore eliminate $\mathbf{R}\mathbf{L}_{\hat{z}}$ from the discussion in favor of $\mathbf{R}\mathbf{M}_{\hat{z}}\mathbf{S}_{\text{iso}}^{1/2}$. However, this factor may be an attractive alternative in a more general case where $\mathbf{M}$ is non-square (that is, $\alpha_1 \neq 0$).

$\mathbf{M}\mathbf{S}_{\text{iso}}^{1/2}$ – Solving with this factor is only viable if $\mathbf{M}$ is square, which we assume for this thesis (see section 4.2.5). Computationally, $\mathbf{M}$ must be treated as a generic sparse matrix – it is sparser than $\mathbf{S}$, but still contain couplings both in ℓ and m. A sparse LU factorization, such as the ones provided by UMFPACK, is needed to find the determinant or solve linear systems (unless $\ell_{\text{mod}} = \ell_{\text{max}}$, in that case Cholesky could be used).

$\mathbf{R}\mathbf{M}_{\hat{z}}\mathbf{S}_{\text{iso}}^{1/2}$ – $\mathbf{M}_{\hat{z}}$ is tri-diagonal, and so the tri-diagonal LU-factorization routines in LAPACK can be used for solving and finding the determinant. Any computation with this factor is essentially bound by the speed of applying $\mathbf{R}$, while the other operations are "free".

Figure 4.3 contains the necessary benchmarks.

4.3.2 Best choice of factor for $p(\theta|\mathbf{s}, \mathbf{d})$

Looking ahead to the next chapter, we need to repeatedly evaluate the likelihood

$$p(\mathbf{s}|\alpha, \hat{p}, \theta_{\text{iso}}) \propto |\mathbf{F}\mathbf{F}^\dagger|^{-1/2} e^{-\mathbf{s}^\dagger(\mathbf{F}\mathbf{F}^\dagger)^{-1}\mathbf{s}/2} = |\mathbf{F}|^{-1} e^{-\|\mathbf{F}^{-1}\mathbf{s}\|/2}, \tag{4.16}$$

where $\|\cdot\|$ is the usual Euclidean norm. Many changes will happen to the parameters between each change of $\mathbf{s}$, and each time the parameters change we need to do a decomposition in order to find $\mathbf{F}^{-1}\mathbf{s}$ and $|\mathbf{F}|$.

Consulting figure 4.3, the best option is clearly $\mathbf{R}\mathbf{M}_{\hat{z}}\mathbf{S}_{\text{iso}}^{1/2}$. Since $\mathbf{R}$ is unitary, $|\mathbf{R}\mathbf{M}_{\hat{z}}\mathbf{S}_{\text{iso}}^{1/2}| = |\mathbf{M}_{\hat{z}}\mathbf{S}_{\text{iso}}^{1/2}|$, so we have

$$p(\mathbf{s}|\alpha, \hat{p}, \theta_{\text{iso}}) \propto |\mathbf{M}_{\hat{z}}|^{-1} |\mathbf{S}_{\text{iso}}|^{-1/2} e^{-\|\mathbf{S}_{\text{iso}}^{-1/2}\mathbf{M}_{\hat{z}}^{-1}\mathbf{R}^\dagger\mathbf{s}\|/2}. \tag{4.17}$$

It is clear that by using this factorization, the operations are conveniently ordered by computational complexity:

- When changing $\hat{p}$, $\mathbf{R}^\dagger\mathbf{s}$ must be recomputed, which scales as $O({\ell_{\text{max}}}^3)$.
- When changing α, one needs to factor $\mathbf{M}_{\hat{z}}$ and solve $\mathbf{M}_{\hat{z}}^{-1}\mathbf{x}$; both scale as $O({\ell_{\text{max}}}^2)$.
- When θ_{iso} changes, one simply needs to multiply a vector with the diagonal $\mathbf{S}_{\text{iso}}^{-1/2}$ matrix ($O({\ell_{\text{max}}}^2)$ with a very low prefactor).

It is possible to exploit this by making steps in α more often than $\hat{p}$. When fixing α and $\hat{p}$ one essentially has the signal likelihood in the isotropic case, inserted the demodulated signal. Therefore, any sampling scheme used to sample θ_{iso} within an isotropic model can also be used in the dipole-modulated model without any additional computational overhead.

4.3.3 Drawing samples from $p(\mathbf{s}|\theta, \mathbf{d})$

In chapter 3, we saw that one can sample from $p(\mathbf{s}|\theta, \mathbf{d})$, given an arbitrary $\mathbf{S} = \mathbf{F}\mathbf{F}^T$, by solving either

$$(\mathbf{S}^{-1} + \mathbf{N}^{-1})\mathbf{x} = \mathbf{b},$$

or

$$(\mathbf{1} + \mathbf{F}^T\mathbf{N}^{-1}\mathbf{F})\mathbf{x}' = \mathbf{b}'$$

by Conjugate Gradients. Here and in the rest of this section we suppress the beam $\mathbf{A}$ as well as the foreground marginalization term $\sigma_t^2\mathbf{T}\mathbf{T}^T$, as they can easily be put back in.

We must multiply vectors repeatedly with the left-hand side hundreds of times for the same set of parameters. In this case, the second version of the

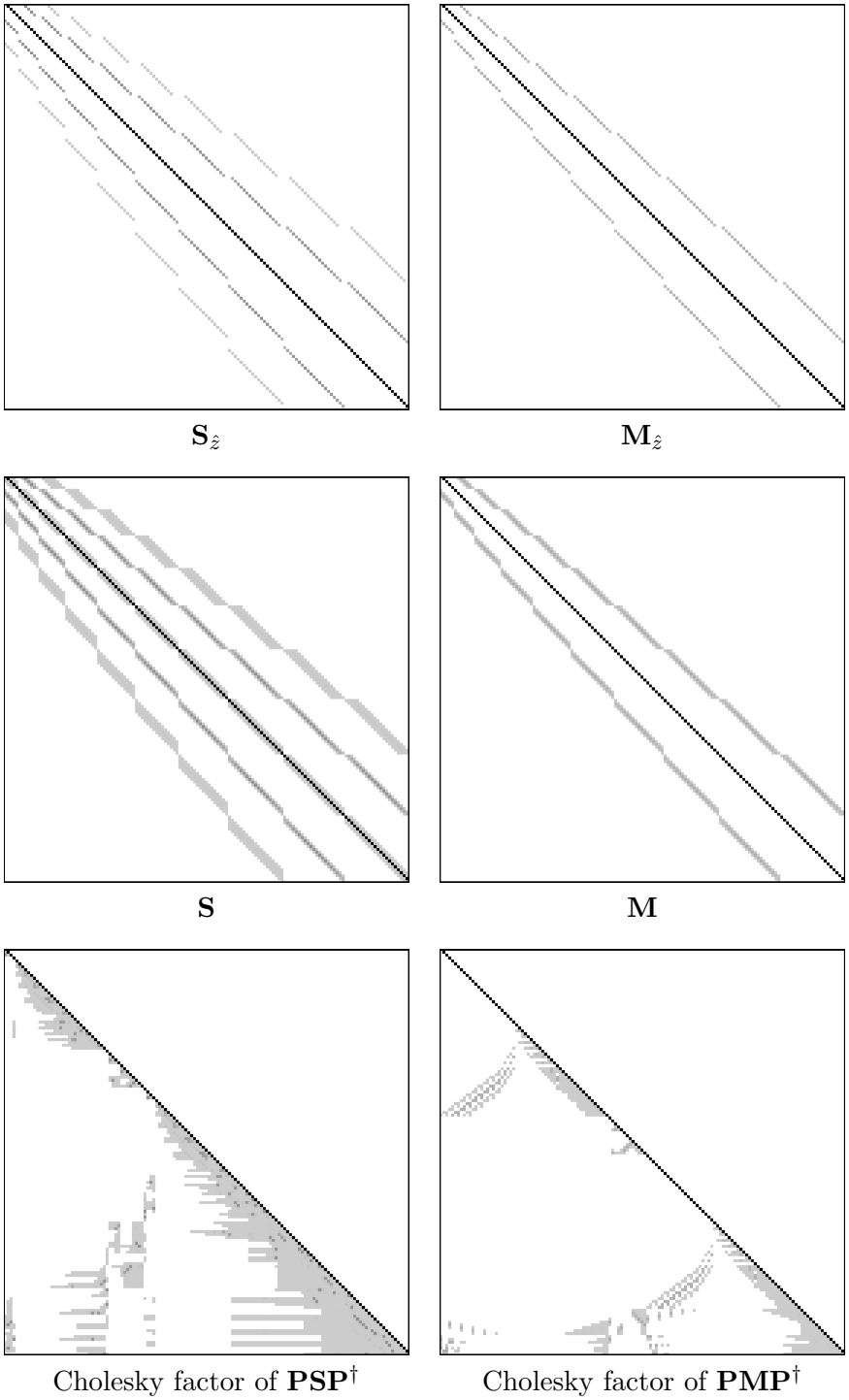
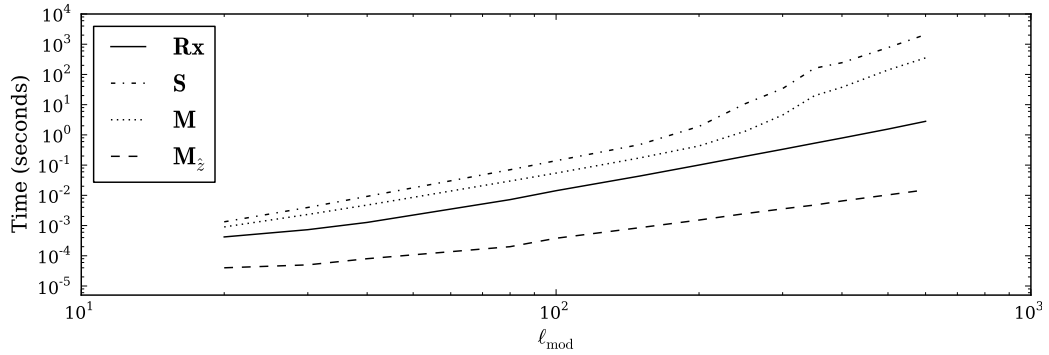
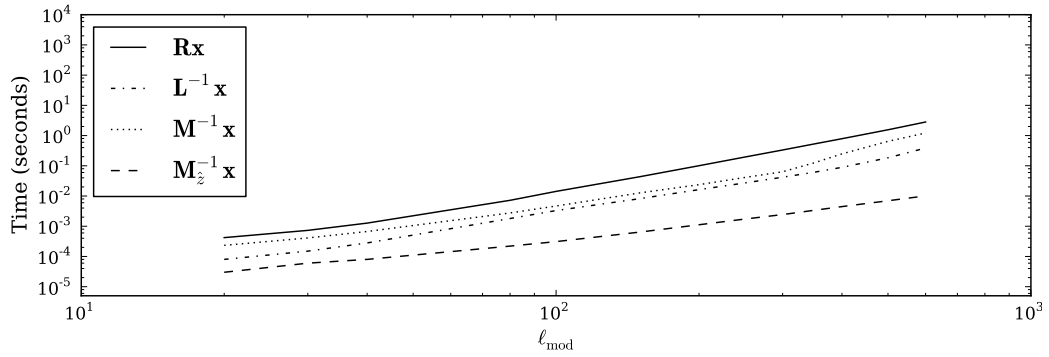


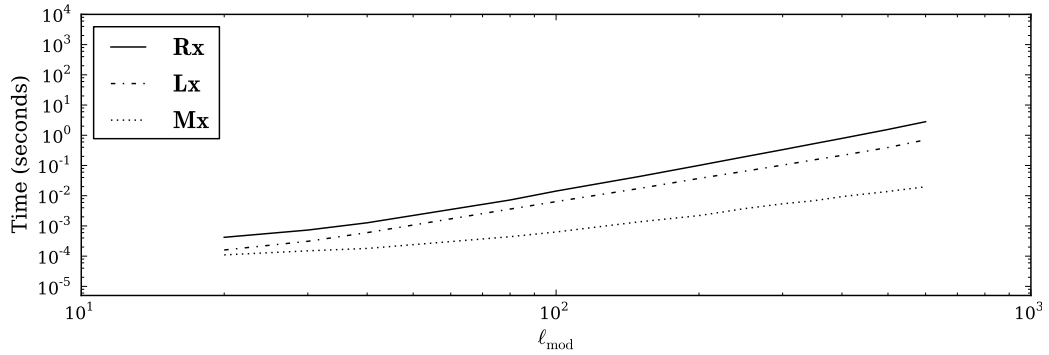
Figure 4.2: Sparsity of matrices; $\ell = 2..11$, $C_\ell = 1$, $\alpha_\ell = 0.3$. Plotted are the absolute values of the matrices in complex spherical harmonic space (in real spherical harmonics there are couplings "across" between (ℓ, m) and $(\ell, -m)$ as well). Numerical values go from gray to black; white represents *symbolic* zero. In m-major ordering, $\mathbf{S}_{\hat{z}}$ and $\mathbf{M}_{\hat{z}}$ (and their Cholesky factors) are penta-diagonal and tri-diagonal, respectively. Using a Cholesky factor of $\mathbf{M}$ is slightly unrealistic (for general α_ℓ a LU factorization is needed). The permutation matrix $\mathbf{P}$ represents the default permutation chosen by CHOLMOD in each case.



(a) Decomposition



(b) Solve



(c) Multiplication

Figure 4.3: Benchmarks of matrix operations (single core, Intel® Xeon E5530, 2.40GHz, 8 KB cache). The $O({\ell_{\mathrm{mod}}}^2)$ construction times of original matrices are not included, neither is sparse matrix analysis time (since a matrix pattern can be pre-analysed). The cost of applying a rotation is included in all plots for comparison. **(a)** $\mathbf{M}_{\hat{z}}$ is decomposed using LAPACK tri-diagonal LU (`dgttrf`), the other matrices with CHOLMOD supernodal Cholesky (this is being kind to $\mathbf{M}$, for which LU is needed in the general case). **(b)** The $\mathbf{R}\mathbf{M}_{\hat{z}}\mathbf{S}_{\mathrm{iso}}^{1/2}$ factor is the most expensive one; the other factors are paid back at this stage for the costly decomposition time. **(c)** The cost of $\mathbf{R}\mathbf{M}_{\hat{z}}\mathbf{x}$ is essentially that of the rotation, and so $\mathbf{M}_{\hat{z}}$ is not included.

system together with the factor $\mathbf{M}\mathbf{S}_{\text{iso}}^{1/2}$ clearly wins out as the fastest alternative. It has the fastest multiplication time and requires no decomposition up front. Our system of choice is thus

$$\mathcal{A}\mathbf{x} = (\mathbf{1} + \mathbf{S}_{\text{iso}}^{1/2}\mathbf{M}^T\mathbf{N}^{-1}\mathbf{M}\mathbf{S}_{\text{iso}}^{1/2})\mathbf{x} = \mathbf{b}.$$

We here treat $\mathbf{M}$ as a matrix in real spherical harmonics, recalling that $\mathbf{M}_R = \mathbf{U}\mathbf{M}_C\mathbf{U}^\dagger$.

Since the matrix has changed, we need a new preconditioner. We use the principle of section 3.5.2 of one dense block for $2, \ldots, \ell_{\text{precond}}$, and a diagonal block for $\ell_{\text{precond}} + 1, \ldots, \ell_{\max}$. The dense block is constructed directly by computing $\mathbf{N}^{-1}$ as earlier noted, and use routines for multiplying a sparse matrix with a dense matrix. For the diagonal block we use the approximation

$$\text{diag}(\mathcal{A}) \approx \mathbf{1} + \mathbf{S}_{\text{iso}}^{1/2}\text{diag}(\mathbf{M})\text{diag}(\mathbf{N}^{-1})\text{diag}(\mathbf{M})\mathbf{S}_{\text{iso}}^{1/2} = \mathbf{1} + \mathbf{S}_{\text{iso}}\text{diag}(\mathbf{N}^{-1}).$$

Since we recover the isotropic case as α gets smaller, and $\alpha < 0.15$ for our purposes, this works reasonably well. Depending on α, the number of iterations required increase by up to 60%, with a more typical number being 10%–20%,

This is clearly an area where things could be improved, and we outline a possible approach. We opt for constructing the preconditioner in complex spherical harmonics, and convert vectors when applying it, to simplify the sparsity pattern. We then need

$$(\mathbf{M}\mathbf{N}^{-1}\mathbf{M}^\dagger)_{\ell m,\ell m} = \sum_{LM}\sum_{L'M'}(\mathbf{M})_{\ell m,LM}(\mathbf{M})^*_{\ell m,L'M'}(\mathbf{N}^{-1})_{LM,L'M'},$$

so only the elements of $\mathbf{N}^{-1}$ such that $(\mathbf{M})_{\ell m,LM}$ and $(\mathbf{M})^*_{\ell m,L'M'}$ are both nonzero are needed. In the case of the dipole modulation field this restricts the number of elements we need to compute to a few bands, scaling as $O({\ell_{\max}}^2)$. In particular, we must have $|L - L'| \leq 2$ and $|M - M'| \leq 2$. One can then implement the above sum directly, or compute the given subset of $\mathbf{N}^{-1}$ as a sparse matrix, use generic sparse matrix multiplication routines, and extract the diagonal. We note, however, that this approach does not work with more general modulation fields.

4.4 Modelling the isotropic power spectrum

Our model assumes that we know the real, underlying isotropic power spectrum. Knowing this power spectrum for sure is not easy, since if we break isotropy, then the assumptions underlying today's best fit power spectra are violated. However, in practice this might not matter much. The dipole modulation model was justified in the first place as a small correction to the isotropic model, and we can similarly justify using the best fit ΛCDM power spectrum as our starting point.

4.4.1 Effects of dipole modulation on the isotropic power spectrum

Assuming that the dipole-modulated model is correct, estimating a power spectrum under an isotropic model will lead to a bias due to model misspecification. What does this bias look like?

We will ignore both data analysis artifacts (such as noise and mask) and how the bias couple to the ΛCDM model, and work with the estimator given a perfect CMB signal $a_{\ell m}$,

$$\sigma_\ell = \frac{1}{2\ell+1} \sum_{m=-\ell}^{\ell} a_{\ell m} a_{\ell m}^*.$$

Now, we take the expectation, i.e. average over all possible universes:

$$\mathrm{E}(\sigma_\ell) = \sigma_\ell = \frac{1}{2\ell+1} \sum_{m=-\ell}^{\ell} \mathrm{E}(a_{\ell m} a_{\ell m}^*) = \frac{1}{2\ell+1} \sum_{m=-\ell}^{\ell} S_{\ell m,\ell m}$$

In an isotropic universe we naturally recover C_ℓ, but in the case of model misspecification we end up averaging $S_{\ell m,\ell m}$ over m, without being justified in doing so. In figure 4.4 we see both the expected bias, and σ_ℓ computed for a specific realization. We see that the expected bias in σ_ℓ is negligible for low α, and that it corresponds exactly to a scaling factor (not plotted). However, the effect on individual realized σ_ℓ's is quite noticeable even with $\alpha = 0.1$.

4.4.2 Choosing a parametrisation

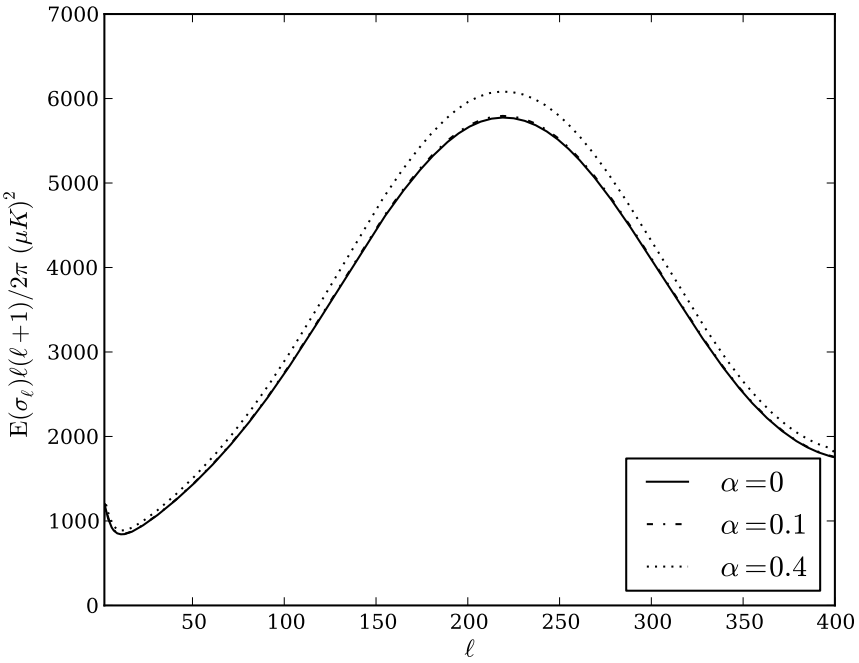
With the results above in mind, it seems reasonable to start with a best fit ΛCDM model, as long as the scale is left as a free parameter. We will adopt the approach taken earlier by e.g. Hoftuft et al. (2009), and let

$$C_\ell = \begin{cases} q \left(\frac{\ell}{\ell_0}\right)^n C_\ell^{\mathrm{fid}} & \text{for } 2 \leq \ell \leq \ell_{\mathrm{mod}} + 1 \\ C_\ell^{\mathrm{fid}} & \text{otherwise.} \end{cases}$$

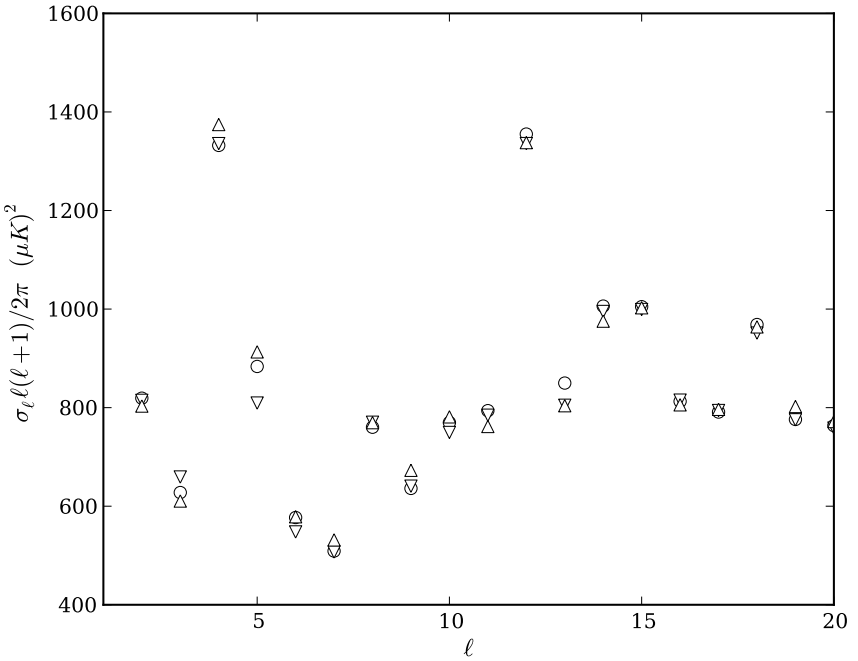
Here, C_ℓ^{fid} is the best-fit ΛCDM power spectrum from the WMAP 7-year data (Larson et al., 2010), q an amplitude, and n a tilt. The tilt pivot ℓ_0 is chosen freely to minimize the correlation between q and n in the posterior distribution. The mean $(2+\ell_{\mathrm{mod}}+1)/2$ appears to be a good choice. We will adopt flat priors on q and n[1]

It would also be possible to leave the C_ℓ as free variables, in order to produce an observed demodulated power spectrum. We comment further on this in chapter 8. As we can see in figure 4.4, such demodulated quantities would be different from current estimates of the observed power spectrum, assuming our estimate of α is significantly different from zero.

[1]It should be noted that n is not the scalar perturbation spectral index n_s that is used to parametrize the primordial power spectrum $P(k)$, although it will have a similar effect.



(a)



(b)

Figure 4.4: The effect of dipole modulation on the isotropic power spectrum estimator σ_ℓ. **(a)** The expected value change with α, although for the levels we are interested in, with α less than 0.1, the effect is very slight. The effect is (naturally) invariant with preferred direction. **(b)** A sample was taken and σ_ℓ estimated with no modulation (circles) and a dipole modulation with $\alpha = 0.1$ using two different preferred directions (triangles).

Chapter 5

Fitting the hemispherical power asymmetry model

The goal of the this chapter is to provide a method for fitting the model of chapter 4 to data, building on the Gibbs framework presented in chapter 3. We then need a method to sample from the posterior distribution of the model parameters θ. If one knows the real CMB signal, the data is of little use, so the posterior distribution is

$$p(\theta|\mathbf{s}, \mathbf{d}) = p(\theta|\mathbf{s}).$$

In our case, θ will consist of the dipole modulation field parameters α and $\hat{p}$, as well as the parameters of the underlying isotropic model, which in our case are an amplitude q and a tilt n. By the usual argument,

$$p(\theta|\mathbf{s}) \propto p(\mathbf{s}|\theta)p(\theta) = |\mathbf{S}|^{-1/2}e^{-\mathbf{s}^\dagger\mathbf{S}^{-1}\mathbf{s}/2}p(\theta),$$

for some prior $p(\theta)$. Regardless of choice of prior, this distribution is very much non-Gaussian, as it is $\mathbf{S}$ that varies with the parameters. We now need a method to sample from this distribution. As all we have is an unnormalized density, and we do not know (and can not easily find) the normalizing prefactor, we must turn to Markov Chain Monte Carlo (MCMC) methods. These give samples that are correlated.

Interpreting the original Gibbs algorithm literally, we need a fresh, *uncorrelated* sample from $p(\theta|\mathbf{s})$ at each step. One strategy would be to, at each step, run an MCMC chain for a few hundred iterations until we were sure we have a single converged and independent sample. However, this is clearly wasteful, and MCMC sampling offers other possibilities.

5.1 MCMC theory

We include a brief caricature of Markov chain theory to benefit the further discussion. We rely on Robert & Casella (2004) and Chib & Greenberg (1995) throughout this section.

5.1.1 Markov chains and MCMC

Consider a chain of random variables $x^{(0)}, x^{(1)}, \ldots, x^{(t)}, \ldots$; where all variables are in a common space $\mathbb{R}^n$. Such a chain is a *time homogeneous Markov chain* if the probability distribution of each step in the chain only depends on the previous step (the Markov property), and does not depend on the step index t (time homogeneous). That is, for all t, we have

$$p(x^{(t+1)}|x^{(0)}, \ldots, x^{(t)}) = p(x^{(t+1)}|x^{(t)}) = K(x^{(t)}, x^{(t+1)}).$$

We here introduce the *transition kernel* K: For each $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$, $K(x, y)$ gives the probability of the Markov chain jumping to y, given that the current position is x. $K(x, \cdot)$ is a probability density[1], so that $\int_{\mathbb{R}^n} K(x, y)dy = 1$ for all x.

Together with the distribution of the starting $x^{(0)}$, K fully characterises the Markov chain. As an example, to know the distribution of $x^{(3)}$, we simply take into account all possible $x^{(1)}$ and $x^{(2)}$ the chain could go through in its way from $x^{(0)}$ to $x^{(3)}$,

$$\begin{aligned} p(x^{(3)}|x^{(0)}) &= \int dx^{(1)} \int dx^{(2)}\ p(x^{(1)}, x^{(2)}, x^{(3)}|x^{(0)}) \\ &= \int dx^{(1)} \int dx^{(2)}\ p(x^{(3)}|x^{(2)}, x^{(1)}, x^{(0)}) p(x^{(2)}|x^{(1)}, x^{(0)}) p(x^{(1)}|x^{(0)}) \\ &= \int dx^{(1)} \int dx^{(2)}\ K(x^{(0)}, x^{(1)}) K(x^{(1)}, x^{(2)}) K(x^{(2)}, x^{(3)}). \end{aligned}$$

Certain classes of Markov chains may be shown to have a *stationary distribution*, denoted π, with the property that if we know that the distribution of $x^{(t)}$ is π, then we also have that the marginal distribution of $x^{(t+1)}$ is π. In other words, for any $A \subset \mathbb{R}^n$, π and K satisfies

$$\int_A \pi(y)dy = \int_A \int_{\mathbb{R}^n} \pi(x)K(x, y)dxdy.$$

Note that π is the *marginal* density. If we condition on $x^{(t)}$, the distribution of $x^{(t+1)}$ is not π, but $K(x^{(t)}, \cdot)$. The stationary distribution is unique for a given Markov chain if it exists. Assuming that a chain has a stationary distribution, then, under some conditions, the chain will converge to it independent of $x^{(0)}$. That is, as $t \to \infty$, $x^{(t)}$ is approximately a sample from π, no matter what $x^{(0)}$ was. Fulfilling the following two conditions is sufficient for convergence:

[1] This is rather informal, but we hope to avoid measure theory in this exposition. It should be understood that $K(x, \cdot)$ is not a proper function and may have "strange" features; in much the same way that the Dirac δ-function is often used without the measure theoretic framework. For instance, one may have $\int_{x-\epsilon}^{x+\epsilon} K(x, y)dy \geq 1/2$ no matter how small ϵ gets, in the case of a chain that only moves from its current position with probability 1/2.

- *(Harris) recurrence* – For every set $A \in \mathbb{R}^n$ such that $\int_A \pi(x)dx > 0$, the chain will (when run for infinitely long) visit A an infinite number of times. Informally, recurrence means that there are no parts of the distribution that the chain stays away from, which would obviously be a problem for convergence.

- *Aperiodicity* – The number of the steps required to get from $A \subset \mathbb{R}^n$ to $B \subset \mathbb{R}^n$ should not be required to be a multiple of some integer for any A, B.

The motivation for Markov chain theory was originally in modelling and studying existing random processes, studying questions such as whether a given process converged to a stationary distribution or not. In MCMC simulation, Markov theory is turned around. Rather than trying to find the stationary distribution, we construct a Markov chain K which has the stationary distribution π that we want to sample from. Then, we simulate a concrete realization of the chain by drawing samples; first $x^{(1)}$ given $x^{(0)}$, then $x^{(2)}$ given $x^{(1)}$, and so on. After a period of *burn-in*, the bias of the starting point $x^{(0)}$ disappears and all the samples come from the target distribution. The Metropolis-Hastings algorithm gives a recipe for constructing chains where it is trivial to sample $x^{(t+1)} \sim K(x^{(t)}, \cdot)$ in each step, while the chain itself will converge to a stationary density π that can be highly non-trivial.

While the samples will be strongly correlated, we can still use them to make inferences. Correlation just means that the number of samples we need for a given level of precision is larger than if they were independent.

5.1.2 Block-at-a-time sampling

As noted by Hastings (1970), it is possible to combine multiple MCMC chains to sample from a joint distribution. This is the cornerstone of Gibbs samplers, and it is also valid for more general samplers.

Let the vector $x \in \mathbb{R}^n$ be distributed according to a target density f, and let it consist of p blocks, $x = [x_1\ x_2\ \ldots\ x_p]^T$, with each block having some conditional density, $x_i \sim f_i(\cdot|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_p)$. Furthermore, assume that we have a corresponding set of Markov chain kernels, $K_1, K_2, \ldots, K_p$, where each kernel K_i leaves each block but x_i stationary, and makes a step in x_i in such a way that the conditional density f_i is the stationary distribution. Then, it turns out that composition of kernels,

$$K = K_1 \circ K_2 \circ \cdots \circ K_p,$$

has f as its stationary distribution. Here $\circ$ denotes that we first make a step with K_1, changing x_1, then a step with K_2, changing x_2, and so on. It must still be checked that the chain converges, but only the full kernel K needs to fulfill the recurrence and aperiodicity conditions, not the individual K_i's.

This principle has profound consequences, because it makes it possible to break the problem of sampling from a joint density into many smaller problems of sampling from conditionals. In fact, the principle is even more general. There would be nothing wrong with repeating the same kernel multiple times in a row, or have kernels correspond to partially overlapping blocks, or select one of the kernels at random for each step using a fixed random rule. In the case of repeating the same kernel multiple times, then for the theory to go through (time-homogeneous) we can only observe the chain after each full round of kernels $K_1 \circ K_2 \circ \cdots \circ K_p$ is applied. However, we can choose to also observe the (strongly related) Markov chains $K_2 \circ \cdots \circ K_p \circ K_1$, $K_3 \circ \cdots \circ K_p \circ K_1 \circ K_2$, and so on, and thus using all the samples is not a problem.

Finding a conditional density given a joint density is straightforward from basic probability theory. Assume that $p(x, y)$ is a joint probability density. Then, for all y such that $p(y) > 0$, we have

$$p(x|y) = \frac{p(x,y)}{p(y)} \propto p(x,y),$$

so $p(x|y)$ is proportional to the joint density with y kept fixed. The Metropolis-Hastings algorithm never jump to a y such that $p(y) = 0$.

5.1.3 The Metropolis-Hastings algorithm

We turn to how to construct an MCMC chain, and start with a very general case, the Metropolis-Hastings sampler (Hastings, 1970). Chib & Greenberg (1995) give an excellent introduction to this algorithm. We will also rely on Robert & Casella (2004).

Consider that we want to sample from some probability distribution $f(x)$, but all we know is the expression for computing the density up to a constant factor. The Metropolis-Hastings algorithm then gives us a recipe to construct a Markov chain which converges to our target density as its stationary distribution. The idea is to supply a proposal density q, which the Metropolis-Hastings "corrects" so that the samples come the target density f instead. The method of correction is by simply staying put, and use the current sample *once more*, if the proposed sample was not usable.

First, one chooses some starting point $x^{(0)}$. Then, given that one has a sample $x^{(t)}$, one draws $x^{(t+1)}$ with the following method:

i) Draw a proposal x^* from some proposal density $q(\cdot|x^{(t)})$.

ii) Compute

$$\rho = \min\left\{\frac{f(x^*)}{f(x^{(t)})}\frac{q(x^{(t)}|x^*)}{q(x^*|x^{(t)})}, 1\right\}. \tag{5.1}$$

Then accept the proposal, that is, assign $x^{(t+1)}$ the value of x^*, with probability ρ. Otherwise, let $x^{(t+1)}$ take the value $x^{(t)}$.

We see that all that is needed is the ability to sample from $q(\cdot|x^{(t)})$, and the ability to evaluate $f(x^*)/q(x^*|x^{(t)})$ up to a constant factor. The fact that constant prefactors for f are not needed is incredibly useful in most practical settings.

There is relatively large freedom in how q is selected. Some proposals leads to algorithms that converge quickly, some to algorithms that converge in theory but in practice would require billions of years of CPU time, and some to algorithms that don't converge even theoretically. The conditions for theoretical convergence were mentioned above. The Metropolis-Hastings algorithm ensures that the chain has a stationary distribution, but one must still check in every case that the proposal density q allows getting back to every part of the support of f (recurrence) and that it doesn't enforce some cyclic pattern of exploration (aperiodicity). The conditions for practical convergence are related. One must simply move quickly enough around in the entire support, and not get stuck in one part of the density for long periods at the time.

Assuming that the chain manages to converge, a measure of how well we are doing in the exploration ("good mixing") is the chain auto-correlation. Low auto-correlation translates into more effective independent samples, and quicker convergence in parameter estimates. The *auto-correlation function* tells us how correlated $x^{(t)}$ and $x^{(t+k)}$ are for a given lag k. This in turn tells us how far apart we would have to pick samples in order to have them approximately independent (however, when making parameter estimates, it is always better to include all the samples). We will simply estimate the auto-correlation from the chain itself,

$$\mathrm{ACF}(k) = \frac{1}{(n-k)\widehat{\sigma}^2} \sum_{t=1}^{n-k} (x^{(t)} - \widehat{\mu})(x^{(t+k)} - \widehat{\mu}),$$

where $\widehat{\mu}$ and $\widehat{\sigma}^2$ are estimates of the mean and variance of $x^{(t)}$ using the entire chain. Our guides for choosing the proposal density q are then: a) The chain must explore the entire distribution, b) $\mathrm{ACF}(k)$ should fall off as quickly as possible. Of course, computational efficiency enters as well. We accept more correlation if it means that we can produce many times more samples in the same time to make up for it.

5.1.4 Random walk steps

In the Metropolis-Hastings algorithm, it is possible to take our proposal density $q(\cdot|x^{(t)})$ as some density around our current point. For instance, if $x \in \mathbb{R}$, then we might let $q(\cdot|x^{(t)})$ be a Gaussian density with mean $x^{(t)}$ and some standard deviation σ. If the proposal density is *symmetric*, so that $q(x^*|x^{(t)}) = q(x^{(t)}|x^*)$, then the *Hastings factor* $q(x^{(t)}|x^*)/q(x^*|x^{(t)})$ disappears in equation (5.1), and we recover the original *Metropolis algorithm*.

This is typically a good default choice when there are no other obvious choices, and will manage to explore most distributions with connected support. Traditional MCMC lore states that the proposal density should typically be tuned so that the acceptance rate is between 0.2-0.5, although as high as 0.8 may work too. A good starting point is to estimate the standard deviation of the target density, and then scale it up or down until the acceptance is right. It is also a good idea to make the proposal density have roughly the same correlation as the target density. That is, the proposal density should be "slanted" in roughly the same direction.

The random walk Metropolis algorithm is vulnerable to local maxima that are weakly connected to the rest of the distribution. Any Metropolis-Hastings method only "sees where it has been", which in the case of random walk means not very far. Since the normalizing factor is never used, it is impossible to know in the context of a single chain whether one is exploring a local isolated region, or the entire support of the posterior. The practical solution to this problem is to run multiple chains with different starting points, and be satisfied if they all find the same distribution.

Continuous automatic tuning can lead to convergence to the wrong distribution (Robert & Casella, 2004, pp. 299), unless one uses specific results for this. We will be doing all tuning up front, independent of any data set, although tuning during burn-in would also be possible.

5.1.5 Gibbs steps

We have already touched upon the Gibbs sampling algorithm. The idea is to first sample x_1 conditional on x_2, then x_2 conditional on x_1, and so on. The conditional distributions are ofter easier to sample from than the joint distribution.

The Gibbs algorithm can be described as an instance of the block-at-the-time sampling where each conditional MCMC kernel simply consists of sampling directly from the conditional distribution. Such MCMC kernels do in fact correspond directly to the Metropolis-Hastings algorithm by using the actual (conditional) target density as the proposal density q, so that ρ in equation (5.1) always becomes 1 and the proposal is always accepted.

If a Gibbs step is available for sampling from a conditional density, say, $f(x_1|x_2, x_3)$, then a single Gibbs step is always superior to a single random walk Metropolis step. Still, random walk Metropolis can sometimes be useful even when Gibbs steps are available. If there is a computational difference, many cheap Metropolis steps could approximate a costly Gibbs step. Also, if we can sample directly from $f(x_1|x_2, x_3)$ and $f(x_2|x_1, x_3)$, but not from $f(x_1, x_2|x_3)$, then a Metropolis step on the latter can outperform Gibbs steps if x_1 and x_2 are correlated.

Gibbs sampling can also be vulnerable to the presence of local maxima, depending on how the maxima are lined up with respect to the axes that the variables are sampled along.

5.2 Fitting our model through MCMC

5.2.1 Combining Gibbs steps and Metropolis-Hastings steps

The Gibbs sampling framework of chapter 3 now gives us the main recipe. For each iteration, first make a random realization of a CMB signal, constrained by observation and a model, and then sample possible model parameters given this signal. By the remarks above on block sampling, we are allowed to make an MCMC step in the model parameters conditional not only on the CMB signal, but also on the model parameters of the previous iteration. We can also choose to repeatedly sample model parameters T times given the same signal. Assuming we have a Metropolis-Hastings proposal density q for the model parameters, we end up with the following algorithm:

$\mathbf{s}^{(k)}$– Gibbs step, sample from $p(\mathbf{s}|\theta^{(k-1,T)}, \mathbf{d})$
$\theta^{(k,1)}$– Propose from $q(\cdot|\mathbf{s}^{(k)}, \theta^{(k-1,T)})$, then accept or reject
$\theta^{(k,2)}$– Propose from $q(\cdot|\mathbf{s}^{(k)}, \theta^{(k,1)})$, then accept or reject
...repeat $T-2$ more times...
$\mathbf{s}^{(k+1)}$– Gibbs step, sample from $p(\mathbf{s}|\theta^{(k,T)}, \mathbf{d})$
...and so on...

The main reason we want to sample the model parameters many times in each loop is the comparatively huge cost of performing the CG search in order to sample $\mathbf{s}$. Having sampled $\mathbf{s}$, it makes sense to get as much information out of it as we can. Another reason is that our scheme for sampling θ is unlikely to be perfect (that is, a Gibbs step, independent of previous values of θ). By repeating the sampling step we can mostly remedy this. Setting T involves a trade-off between reducing chain correlation and computational cost. If sampling θ is cheap, setting T high does not hurt. For the sampler we describe below, $T = 40$ worked well. Note that even a choice of $T = 1$ is theoretically valid, it would just lead to longer correlation lengths.

This approach is also taken in Groeneboom & Eriksen (2009). It is also similar to the use of a Blackwell-Rao estimator for the CMB power spectrum discussed in Rudjord et al. (2009) and Wandelt et al. (2004), although in that case the samples $p(C_\ell|\mathbf{s})$ *are* perfect, and getting additional samples can be delayed to a post-processing step without affecting the correlation lengths.

5.2.2 Separating data analysis and model estimation

We have earlier noted that $p(\theta|\mathbf{s}, \mathbf{d}) = p(\theta|\mathbf{s})$, so that data analysis and model estimation decouple in the Gibbs sampler. It is time to make a proper visit to this statement.

The algorithm above can be viewed as a block-at-the-time Metropolis-Hastings sampler for the density

$$p(\theta, \mathbf{s}|\mathbf{d}) \propto p(\mathbf{d}|\theta, \mathbf{s})p(\mathbf{s}|\theta)p(\theta) = p(\mathbf{d}|\mathbf{s})p(\mathbf{s}|\theta)p(\theta),$$

where each step is either in $\mathbf{s}$ or in θ, and we recall that $p(\mathbf{d}|\theta, \mathbf{s}) = p(\mathbf{d}|\mathbf{s})$.

To have our algorithm scale as $O({\ell_{\text{max}}}^3)$, we are never allowed to evaluate $p(\mathbf{d}|\mathbf{s})$, which would scale as $O({N_{\text{pix}}}^3) = O({\ell_{\text{max}}}^6)$. This is not a problem, because when sampling the signal we do not need to evaluate the likelihood, and when sampling the parameters we can view $p(\mathbf{d}|\mathbf{s})$ as a constant prefactor. However, this means that the total parameter probability (or the log-likelihood, often denoted $\log \mathcal{L}$) is unavailable. One often plotted quantity in MCMC settings is the posterior log-probability, which should increase during burn-in and then have small fluctuations around the maximum plateau. Without computing $p(\mathbf{d}|\mathbf{s})$, the normalization of our posterior is changed with every new signal, making such plots meaningless to us. Of course, when N_{side} is small, we can overcome this by brute force.

An important detail for implementation is that after a Gibbs step in $\mathbf{s}^{(k)}$, we must reevaluate $p(\theta = \theta^{(k-1,T)}|\mathbf{s} = \mathbf{s}^{(k)})$ before any new steps are taken, so that ρ is correctly computed in the Metropolis-Hastings algorithm. While likelihood evaluation is not necessary in pure Gibbs samplers, one must in hybrid Gibbs-Metropolis samplers evaluate the likelihood also after the Gibbs steps.

5.2.3 Sampling model parameters

It is time to attack the model posterior conditional on the signal,

$$p(\alpha, \hat{p}, q, n|\mathbf{s}) \propto p(\mathbf{s}|\alpha, \hat{p}, q, n)p(\alpha, \hat{p}, q, n) = |\mathbf{S}|^{-1/2}e^{-\mathbf{s}^\dagger\mathbf{S}^{-1}\mathbf{s}/2}p(\alpha, \hat{p}, q, n).$$

We adopt independent and flat priors on all parameters, $p(\alpha, \hat{p}, q, n) \propto 1$. It should be understood that $\hat{p}$ is bounded on the sphere, and that we must have $q > 0$ and $\alpha \geq 0$.

The dipole parameters α and $\hat{p}$ take effect up to $\ell_{\text{mod}} + 1$. After that point, $\mathbf{M}$ is an identity matrix. Furthermore, we let q and n only affect the power spectrum up to $\ell_{\text{mod}} + 1$ (or ℓ_{max}, whichever is lower). Then, $\mathbf{S} = \mathbf{M}\mathbf{S}_{\text{iso}}\mathbf{M}^\dagger$ is block-diagonal with one block for $2, \ldots, \ell_{\text{mod}} + 1$ and one block for $\ell_{\text{mod}} + 2, \ldots, \ell_{\text{max}}$, where the latter block does not depend on any parameters. We therefore only need to consider coefficients for $2, \ldots, \ell_{\text{mod}}+1$, and consider the likelihood for the higher multipoles as part of the normalizing constant (except if we want to compute the full posterior probability, as noted in the previous section). Building on the benchmarks of section 4.3.2, we write this as

$$p(\alpha, \hat{p}, q, n|\mathbf{s}) \propto q^{-\frac{N}{2}}|\mathbf{M}_{\hat{z}}(\alpha)|^{-1}|\mathbf{S}_{\text{iso}}(n)|^{-\frac{1}{2}}e^{-\frac{1}{2q}\|\mathbf{S}_{\text{iso}}(n)^{-\frac{1}{2}}\mathbf{M}_{\hat{z}}(\alpha)^{-1}\mathbf{R}(\hat{\mathbf{p}})^\dagger\mathbf{s}\|}. \quad (5.2)$$

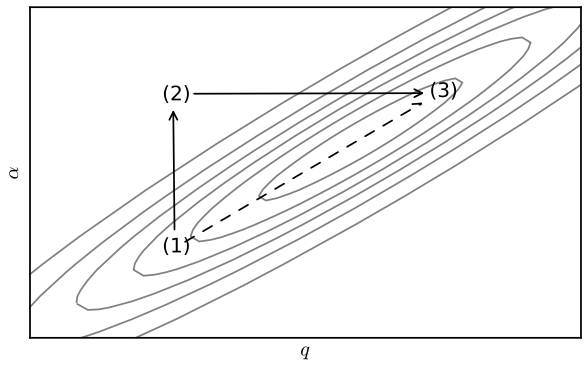


Figure 5.1: Illustration of a problem with conditional sampling. If $p(q, \alpha|\hat{p}, n, \mathbf{s})$ turns out to be a distribution with strong correlation, a conditional step in α alone from (1) to (2) would almost surely be rejected, preventing the probable move of (q, α) to (3). This forces all random walk steps to be small, requiring many steps to explore the distribution. When stepping both variables simultaneously, much larger jumps are allowed (dashed line).

Note that we have factored out the power spectrum amplitude q, so that $\mathbf{S}_{\text{iso}}$ only depends on the tilt n. We have also defined N as the number of spherical harmonic coefficients included in the system, $N = (\ell_{\text{mod}} + 2)^2 - 2^2$ (assuming $\ell_{\text{mod}} < \ell_{\text{max}}$).

We choose to explore the posterior by moving one of the four parameters at the time, conditional on the three others. One reason is that, as noted earlier, the necessary computations fully decouple. There is no computational gain from changing parameters simultaneously. Changing $\hat{p}$ scales as $O({\ell_{\text{max}}}^3)$, changing α or n scales as $O({\ell_{\text{max}}}^2)$, and a change of q is done in constant time. By sampling the parameters one by one, we avoid a situation where proposing an unlikely value of n leads to simultaneously rejecting a likely value of $\hat{p}$ that was costly to compute. We also become free to repeat the technique of doing the cheaper steps more often than the costlier steps, although this turned out to not be necessary.

Another reason for making steps one parameter at the time is the possibility to create highly adaptive proposal rules, *without* sacrificing the simplicity of the random walk Metropolis algorithm. For instance, one can imagine that $p(\hat{p}|\alpha, \dots)$ becomes more dispersed when conditioning on lower α's, and that one wants to adapt the proposal rule correspondingly. In the end, this was not necessary either, although it may become necessary in the future if the sampler is to be used on isotropic simulations.

The disadvantage of not making joint moves would become clear in the presence of strong correlations in the posterior. In that case, stepping one component at the time would lead to unnecessarily bad proposals (see figure 5.1). Fortunately, as long as ℓ_0 is well chosen, the parameters seem free of strong correlations in the posterior.

As our proposal density for α and n, we simply adopt symmetric Gaussian random walk steps, where the standard deviations σ_α and σ_n are tuned as described in the following section. For q we currently do the same, but we observe that

$$p(q|\alpha, \hat{p}, n) \propto q^{-\frac{N}{2}} e^{-\frac{1}{2q}\mathbf{s}^T\hat{\mathbf{S}}^{-1}\mathbf{s}},$$

where $\hat{\mathbf{S}}$ indicate the signal covariance evaluated with $q = 1$. We recognize $p(q|\alpha, \hat{p}, n, \mathbf{s})$ as an Inverse-gamma distribution with shape parameter $N/2-1$ and scale parameter $\mathbf{s}^T\hat{\mathbf{S}}^{-1}\mathbf{s}/2$ (Gelman et al., 2004). Thus, it would have been possible to use a Gibbs step and draw q using standard routines[2].

To sample $\hat{p}$, we use random walk Metropolis with a uniform proposal density on a cap on the sphere[3], centered in the current position and with angular radius $\sigma_{\hat{p}}$. In order to tune $\sigma_{\hat{p}}$, we need some way to measure the dispersion of $p(\hat{p}|\alpha, q, n, \mathbf{s})$. Standard deviation does not work well with co-latitude and longitude coordinates, as the results would depend on the position on the sphere, and because of the wraparound. Instead, given a set of samples $\{\hat{p}^{(t)}\}_{t=1}^T$, we use the following approach. First, we find the mean direction $\bar{p}$, by averaging $\{\hat{p}^{(t)}\}_{t=1}^T$ as vectors, and normalizing the result to the unit sphere. Then, we measure dispersion in radial distances with respect to this mean,

$$\mathrm{sd}(\hat{p}) \equiv \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\psi^{(t)})^2},$$

where $\psi^{(t)}$ is the angular distance of a sample to $\bar{p}$. Continuing along these lines, it is also useful to view $\hat{p}$ in two coordinates in a way that is independent of the current position on the sphere. For this purpose we establish a coordinate system where $\bar{p}$ is aligned with the z-axis, and $(\psi^{(t)}, \phi^{(t)})$ are the co-latitude and longitude coordinates of each sample with respect to $\bar{p}$ (and an arbitrary orientation). Finally, $\tilde{p}^{(t)}$ are the samples projected to the 2D Euclidean plane,

$$\tilde{p}_0^{(t)} = \psi^{(t)}\cos\phi^{(t)}, \qquad \tilde{p}_1^{(t)} = \psi^{(t)}\sin\phi^{(t)}.$$

The $\tilde{p}^{(t)}$'s are then used for chain diagnostic plots, such as figure 5.2.

5.3 Tuning and performance

A common approach in MCMC is to tune the proposal density for a given data set $\mathbf{d}$ by using test runs prior to the real run. Motivated by the fact

[2]If a routine to sample from Inverse-gamma is not available, one can sample $1/q$ from a Gamma distribution instead.

[3]The exact computation is to a) draw a variate U uniformly from $[0, 1]$, b) let the proposal co-latitude $\theta^* = \cos^{-1}(1 - (1 - \cos\sigma_{\hat{p}})U)$, c) draw the proposal longitude ϕ^* uniformly from $[0, 2\pi]$, d) rotate the resulting point by the coordinates of $\hat{p}^{(t)}$ using Euler matrices.

that what we really should target is convergence to $p(\theta|\mathbf{s} = \mathbf{s}^{(k)})$ within each Gibbs cycle, not convergence to the final marginal posterior marginal $p(\theta|\mathbf{d})$, we opt for performing all tuning *a priori* on simulated signals[4].

The proposal densities for α, $\hat{p}$ and n needs tuning with respect to their respective conditional posterior. After some guesswork and test runs we conclude that the posterior dispersions are strongly dependent on ℓ_{mod}, as expected, but that any other dependencies, such as $p(\hat{p}|\alpha, q, n, \mathbf{s})$ becoming wider at lower α, are small enough to be neglected. To find a rule for predicting the dispersion, we first simulate an ensemble of signals $\mathbf{s}^{(k)}$ with fixed parameters $\alpha_0 = 0.075$, $\hat{p}_0 = (0, 0)$, $q_0 = 1$, and $n_0 = 0$. Then, over a range of ℓ_{mod} values, we estimate the posterior dispersion for each signal, e.g., $\text{sd}(\alpha|\hat{p} = \hat{p}_0, q = q_0, n = n_0, \mathbf{s} = \mathbf{s}^{(k)})$. The dispersion varies quite a bit depending on the signal used. Finally, we fit a power law to these dispersion samples,

$$\text{E}_{\mathbf{s}}(\text{sd}(\alpha|\hat{p} = \hat{p}_0, q = q_0, n = n_0, \mathbf{s})) = a_\alpha {\ell_{\text{mod}}}^{b_\alpha}.$$

The proposal dispersion σ_α was then tuned relative to these estimates,

$$\sigma_\alpha = \lambda_\alpha a_\alpha {\ell_{\text{mod}}}^{b_\alpha},$$

where λ_α was manually tuned for an acceptance rate in the range 0.2–0.7. This process is then repeated for σ_n and $\sigma_{\hat{p}}$. In practice, we have to bootstrap the process by doing some guesswork for a small set of ℓ_{mod} (running long chains to make up for bad tuning), then make our estimates, and then repeat for a wider range of ℓ_{mod}. After tuning for ℓ_{mod} up to 100, the resulting rule results in successful proposal distributions for ℓ_{mod} as high as 500, so the power law is a successful fit. It also runs well on signals simulated with $\alpha_0 = 0$. This is attributed to the fact that almost all signals has some spurious preferred direction $\hat{p}$, so that even if α is estimated very low, $p(\hat{p}|\mathbf{s})$ still does not get too wide. Still, if probing an observation without a dipole modulation signature at high resolution, recalibration could be needed.

An example chain conditional on a single signal is given in figure 5.2. Figure 5.3 shows the results of running a the full chain including the Gibbs steps in $\mathbf{s}$. There are necessarily longer correlation lengths when including the Gibbs step. The effect is to average over multiple signal posteriors to incorporate the uncertainty due to mask and instrumental noise, and each model parameter sample taken conditional on the same signal are therefore strongly correlated with respect to the total posterior. However, what we really care about is correlation per CPU time, not correlation per sample. Therefore, the chains must be said to behave excellently, with almost independent samples between one signal and the next. The chain also converge very quickly, essentially within a couple of CG searches regardless of starting position. Chains at higher ℓ_{mod} take longer to to converge.

[4]This is not to say that tuning for data would be unfruitful, because each signal sample $\mathbf{s}$ in a real run is still constrained by data. Still, the approach of tuning against simulated signals worked very well; a convenient side-effect of the Gibbs sampler.

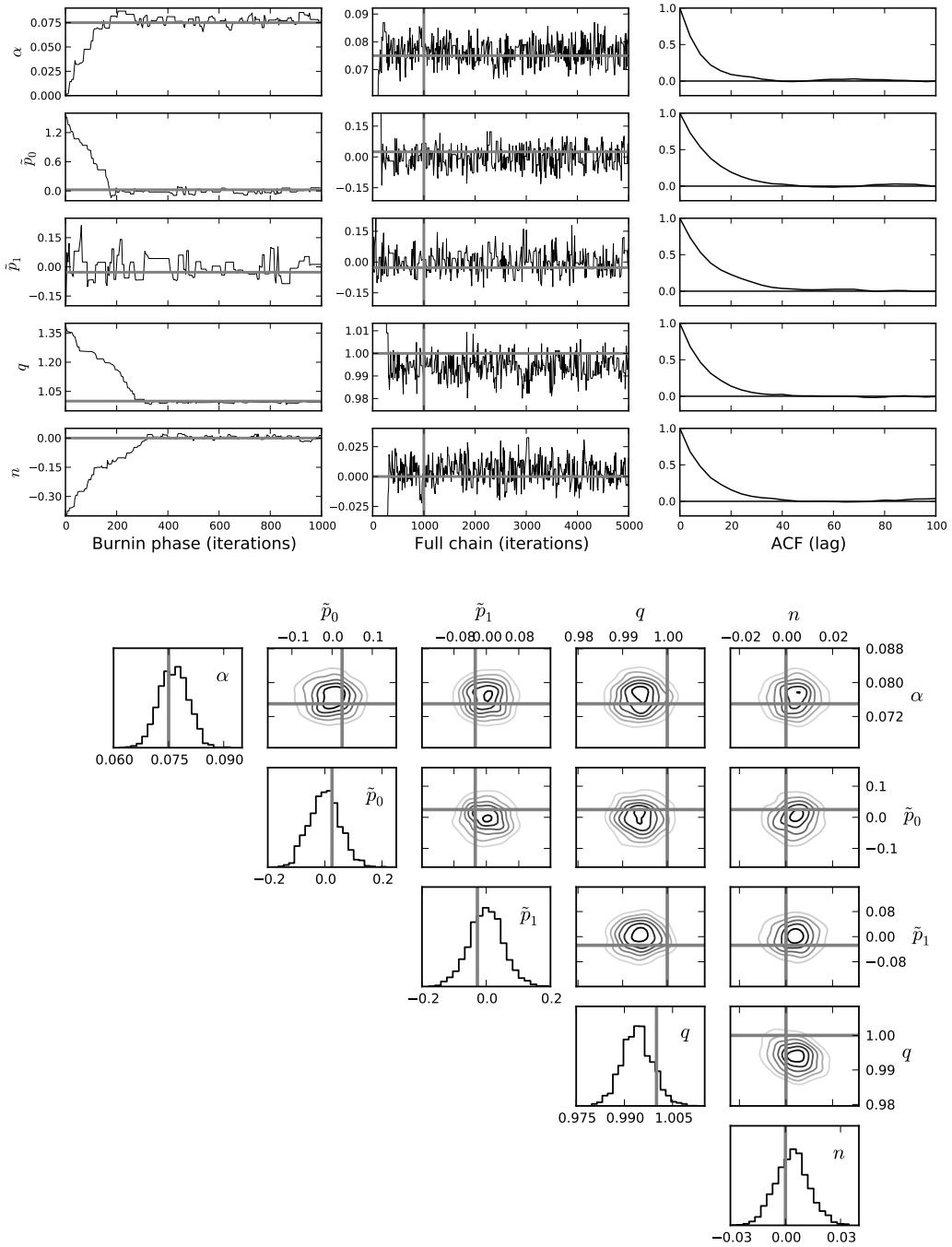


Figure 5.2: MCMC sampling of $p(\alpha, \hat{p}, q, n|\mathbf{s})$ for a *fixed, simulated signal*, $\ell_{\mathrm{mod}} = 300$. The iteration counter increments with every MCMC step, so each parameter change every fourth step (in particular, the ACF lag should be divided by four). The input parameters used for sampling $\mathbf{s}$ are given by gray bands. The left panels above contain a small portion of the full chain, marked by a vertical gray band in the full chain.

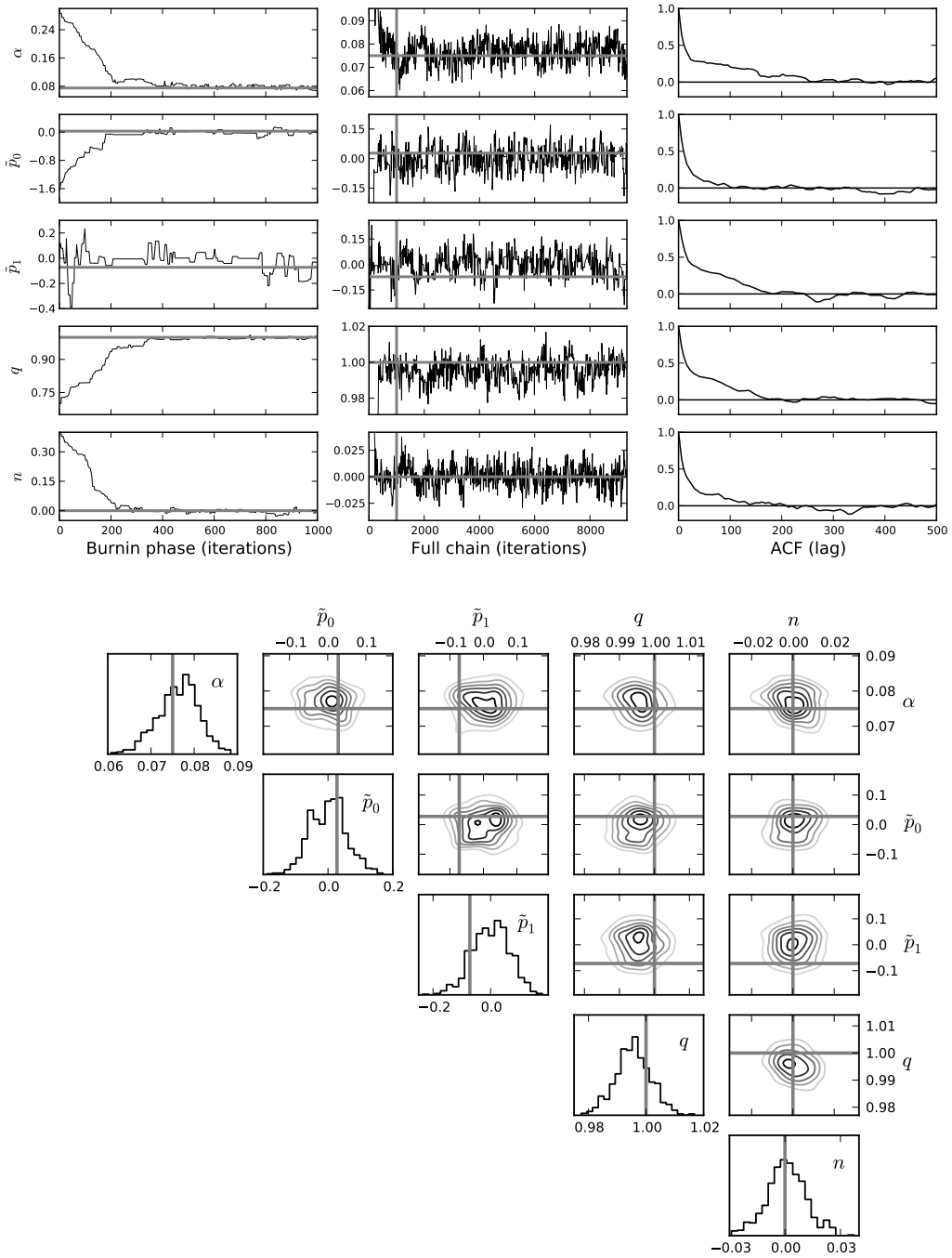


Figure 5.3: Full sampling including Gibbs steps from $p(\alpha, \hat{p}, q, n|\mathbf{d})$. The same signal is used as in figure 5.2, with added noise and beam smoothing to create an observation with WMAP 7-year characteristics (V-band). We use the KQ85y7 mask (Jarosik et al., 2010). A new signal $\mathbf{s}^{(t)}$ is sampled every 161st iteration. Note the different scales from figure 5.2. The posteriors are much wider as a result of averaging over many possible signals. What counts for computational efficiency is correlation lengths per sampled $\mathbf{s}^{(t)}$, and dividing the ACF lag by 161 we see that the there is little correlation between one signal and the next. The chain mixes about as well as we can hope for.

Chapter 6

The PyCMB package

Implementing the algorithms described, debugging them, and verifying the final code against simulations represents the main effort behind this thesis. The implementation was done from scratch in Python. As is customary for scientific Python code, the focus has not been on a monolithic program for a single model, but rather on sowing the seeds for a collection of reusable features which can be put together (and supplemented) in flexible ways for each new analysis. This explains the name: PyCMB, a package for CMB data analysis in Python. It currently contains about 5000 code lines. About 700 of these are specific to the dipole modulation model[1]. The constrained realization sampler used for the Gibbs step in $\mathbf{s}$ also consists of about 700 code lines. We will not cover every detail of PyCMB here, but just give a quick tour.

We do not rely on any existing cosmological code. In particular, we do not rely on the exact likelihood dipole-modulation model code of Hoftuft et al. (2009), or on the Commander CMB Gibbs sampler (Eriksen et al., 2004b). Commander has many features that PyCMB does not have, in particular joint estimation of foregrounds, estimation of observed CMB power spectra, and inclusion of polarization data.

6.1 Command line front-end and chain files

The MCMC part of PyCMB revolves around chain files. Each chain file contains not only the chain, but everything that is needed for full reproducibility:

- The observation data and instrument properties.
- If based on a simulation, the simulated signal and the true values the simulation was based on.

[1] Although this does not include about about 1500 lines of exploratory Python scripts made to better understand the behaviour of the model, make plots for this thesis, and so on. These are obviously specific to the model in question.

- Any parameters used for MCMC sampling, such as $\ell_{\max}$ and ℓ_{mod}, a basic description of the MCMC strategy ("`signal,alpha,phat,q,n,alpha,phat,q,n,...`"), and so on. An important principle is that while defaults are provided for many parameters, they are all persisted to file, so that tuning of the defaults do not cause inconsistencies down the road.

The idea is that it is better to store too much than too little, because one never knows what will be useful for debugging. Storing the full observations directly in each chain file was convenient for WMAP-sized data sets. For Planck-sized data sets, one might want to store data pointers instead.

Another principle is that it should be possible to execute any command directly against the raw observational data, without having to keep track of any manually preprocessed input. Therefore, a few preprocessing options are included in the command line interface. Initializing a new chain is done using the following command:

```
dipmc init \
    --datapath $WMAP_PATH \
    --data wmap_da_forered_iqumap_r9_7yr_V{1,2}_v4.fits \
    --beam wmap_V{1,2}_ampl_bl_7yr_v4.txt \
    --powerspectrum wmap_lcdm_sz_lens_wmap5_cl_v3.dat \
    --mask wmap_7yr_smoothable_mask.fits \
    --combine-channels average \
    --downgrade 16 \
    --lmod 40 --lmax 44 --name V \
    --fix q=1 -n 1000 -c 10
```

This includes the data from the V1 and V2 WMAP radiometers, downgraded[2] to $N_{\mathrm{side}} = 16$. We also request that q is not sampled, but held fixed at 1, and that we want 10 chains of 1000 Gibbs signal samples each. Alternatively, one could base the chains on a simulation with known true parameters, using the original WMAP resolution and noise properties:

```
dipmc init --simulate \
    --datapath $WMAP_PATH \
    --nobs wmap_da_forered_iqumap_r9_7yr_V{1,2}_v4.fits \
    --sigma 3.319mK 2.955mK \
    --mask wmap_temperature_analysis_mask_r9_7yr_v4.fits \
    <...> \
    --truth alpha=0.1 --truth phat='(1.5,1.5)' \
    --truth q=1 --truth n=0
```

[2]This involves automatically finding a suitable ℓ where we want to have a signal-to-noise ratio of 1, find a corresponding beam and noise level, deconvolve the original data, convolve with the new beam, and add uniform RMS noise. This process of downgrading the data is needed in order to correctly solve the linear system of chapter 3, see e.g. Hoftuft et al. (2009).

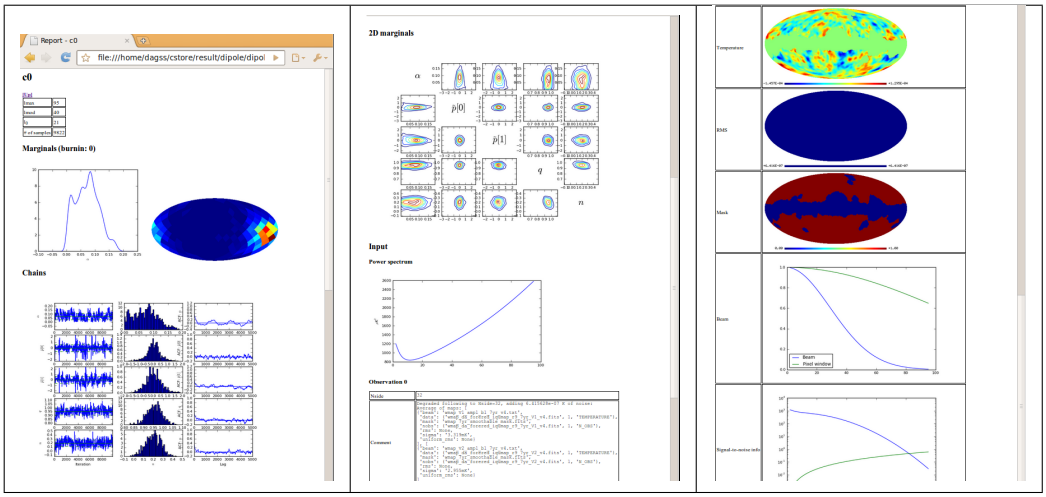


Figure 6.1: Screenshots of an HTML page generated by the `dipmc report` command. It serves to get a quick overview of a chain and the input it was based on.

Both commands create a new directory with a name derived from the arguments (if not explicitly given), containing 10 chain files `c0000.h5` through `c0009.h5`. Each chain is stored in the HDF5 format[3].

Both commands initialize the chain starting point at random, and then immediately terminate. To start sampling, or to resume an aborted sampling process, we type:

```
dipmc run --lprecond=50 <path>
```

This will scan through `<path>` for any unfinished chain files. Lock files are used to make sure no samplers try to work on the same chain. Therefore, worker processes can be launched quickly without having to specify in detail which process works on what chain. The workflow neatly separates parameters of the simulation itself, specified using `init`, from options that only impacts runtime and system resource usage, specified in `run`.

Finally, the following command scans a directory and updates summary reports for all chains, as seen in figure 6.1:

```
dipmc report <path>
```

This is the current limit of the command line interface. Inspection, post-processing and plotting is currently done either by Python scripts or in an interactive Python console. Using the data from non-Python applications should just be a matter of exporting the data using, e.g., the `np.savetxt` command. One can also open the HDF5 files directly, as HDF5 interfaces are

[3]HDF5 is a relatively standard format for self-documenting scientific data. It is similar to FITS (which is more widespread in the cosmological community), but has a lot more features, and was simply more convenient in this case.

available for almost any environment (MATLAB, IDL, Fortran, C, etc.). An example Python session inspecting a chain file:

```
In [1]: import cmb

In [2]: c = cmb.openfile('c0003.h5'); len(c)
Out[2]: 128800

In [3]: c.plot(burnin=30000) # pops up window with chain plots

In [4]: m = c.plot_marginal_phat(Nside=16); m
Out[4]: Pixel sphere map (ring-ordered, Nside=16, Npix=3072)

In [5]: m.map2gif('direction-posterior.gif')

In [6]: log_alpha = np.log(c.get_chain('alpha')); plt.hist(log_alpha)
```

6.2 Overall design

A few key principles were followed out of habit:

- Code should never reach out for what it needs, but instead expect to have it passed in. This is the most straightforward way of ensuring that code does not tangle, and remains easily testable without relying on a particular program flow.

- Test each piece in isolation, by writing a small script that exercise functionality in each piece. As an example, the dipole-modulated covariance matrix $\mathbf{S}$ was computed both from samples and from theory. Unfortunately, the automated test suite (the pillar of many modern software projects) is lacking. This was because plots mostly had to be inspected manually, and we did not spend time to investigate automatic methods for creating tests that, e.g., compare with Monte Carlo results.

Python is a flexible language that has support for many programming paradigms, whether procedural, object-oriented, or functional. Following community customs and experience, we decided to let the package consist of four layers. From the bottom and up:

Sub-packages Fully reusable, isolated functionality that can eventually make their way into separate projects:

- `lightmc` – Functionality for storing arbitrary MCMC chains to file and various plotting functions (PyMC was tried and found unsuitable).

- `healpix4py` – A Python interface to the Fortran implementation of HEALPix.

Basic algorithms layer These are mostly procedural, and as isolated as possible. Examples include computing the diagonal of $\mathbf{N}^{-1}$ in spherical harmonic space, rotating a set of vectors using Euler matrices, repacking a sparse matrix from complex to real spherical harmonics, Conjugate Gradients, and so on. A few of these had to be implemented in Cython for speed (see Seljebotn, 2009). An example of using this layer:

```
M = compute_M(lmin, lmax, px, py, pz, alpha_l)
Ninv_diagonal = Ninv_to_real_harmonic_diagonal(lmin, lmax,
                    1 / (rms_array * rms_array))
```

The idea is that it is easy to get an object-oriented interface wrong, so that it gets in the way instead of speeding up development. This can happen either by being too generic or too specific. By keeping the object-oriented layer as thin as possible on top of procedural building blocks, this risk is diminished. It is fully possible to build something directly on the basic procedural algorithms, and only add the functionality in the object-oriented layer when one is sure about the design.

Object-oriented layer Assists in creating code that is generic enough to avoid "code entanglement", i.e., code that does not contain special cases for unrelated issues. An example of using this layer:

```
from cmb import *
# First, sample a signal constrained by V1 and V2 data
with working_directory('$WMAP_PATH'):
    V1 = CmbObservation(
             data='wmap_da_forered_iqumap_r9_7yr_V1_v4.fits',
             sigma0=3.319e-3, # or '3.3 mK'
             beam='wmap_V1_ampl_bl_7yr_v4.txt',
             mask='wmap_temperature_analysis_mask_r9_7yr_v4.fits')
    V2 = <...>
    V = average_observations([V1, V2])
    model = IsotropicCmbModel('power_spectrum.dat')
sampler = ConstrainedSignalSampler([V], model, eps=1e-5, lprecond=40)
signal = sampler.sample_signal()
# Then, simulate an observation of this signal
# using V2 properties (adds beam smoothing and RMS noise)
simulated_observation = V2.simulate_observation(signal)
# Finally, downgrade it
simulated_observation = downgrade_observation(simulated_observation,
                                              Nside=16, FWHM='9 deg')
```

The corresponding code using the procedural layer would have been much longer and much more specific. We can easily switch out `IsotropicCmbModel` with `DipoleModulationCmbModel`, or another exotic model, without changing anything else.

Command-line layer The layer described in the previous section is very thin, and essentially consists of documenting command line options and simple calls to the PyCMB API. Anything that can be done through the command line should also be easily doable from Python scripts using the package directly, but the time spent on a few command line commands was well repaid when running jobs on various data on a cluster.

6.3 Independence of code base

As mentioned, we do not rely upon Commander, and the Gibbs sampler in the PyCMB package can be seen as an independent implementation. The code of Commander was seriously consulted only on one occasion, and in that case led to a bug being discovered in Commander, not the other way around[4]. We did however compare constrained signal realizations with those produced from Commander. In that case, a third, explicitly computed theoretical result was first found to be in agreement with Commander, and then the theoretical result was used for debugging PyCMB.

The main point of contact between Commander and PyCMB is therefore that H. K. Eriksen is both the supervisor of this thesis project, and the main author of Commander, so that any flawed assumption on his side regarding the algorithms could potentially make its way into both projects. All such assumptions should be documented in the preceding chapters.

This is not meant to imply that PyCMB is free of bugs, rather the reverse. However, it does mean that one can realistically hope for any bugs in one code base to be independent of any bugs in the other.

[4]The bug in question was in the CG preconditioner, and does not impact scientific conclusions in any way.

Chapter 7

Analysis

Having developed a working code for exact Bayesian analysis of a dipole-modulation effect, we now put it to use. We start with analysing 500 simulations at low resolution, in order to get an indication that the code is working as it should. Then, we attempt to reproduce the results of Hoftuft et al. (2009). While the results are similar, they are not in complete agreement. This can possibly be explained by different handling of foregrounds, but should be investigated more closely to make sure it is not a result of bugs in our code. Finally, we present a few results from full-resolution WMAP data. These are present primarily to demonstrate that the code works at high resolutions, and we stop short of a full analysis.

Warning: The last known bug in the code was fixed less than a week prior to the thesis submission deadline. The analysis of WMAP data is present solely to give an illustration of using the method, and the results quoted should not be taken as final. In particular, the chains have not run for as long as they should have.

7.1 Validation by simulation

How can we be sure that our code is correct, or at least be somewhat confident that any bugs do not affect the scientific conclusions? The answer is simulations. We construct many artificial data sets $\mathbf{d}_j$ with known real parameter values. Then, we run our code and recover estimates for the parameters we put in, and finally, compare the claims made the posterior distributions with the true values.

7.1.1 Generating simulated data sets

The process of making simulations builds directly on our model specification in chapters 3 and 4, with one exception: We do the modulation directly in pixel space, which causes the modulation effect to extend to ℓ_{max}, ignoring any ℓ_{mod}. This provides an extra check that the parameter estimation works

when including only a subset of the data in the CMB signal likelihood (as we would expect).

In detail, we:

- Simulate an isotropic signal with the best fit WMAP 7-year power spectrum (Larson et al., 2010).
- Apply the dipole modulation in pixel space (at $N_{\rm side} = 512$) for known α_0 and $\hat{p}_0$. The modulation is applied in pixel space, in order to be independent of the sparse matrices derived in chapter 4.
- Add a monopole and dipole component, with amplitude around $80\,\mu$K, to make sure we are insensitive to their presence. The same monopole and dipole is used for all simulations.
- Smooth the signal with a 9° FWHM Gaussian beam.
- Convert to $N_{\rm side} = 16$, and add $0.56\,\mu$K of RMS noise per pixel, which cause unity signal-to-noise power ratio at $\ell = 40$.
- Use a galactic mask similar to the one used in WMAP analyses (but smoothed and downgraded to $N_{\rm side} = 16$).

Finally, we feed the simulated data set to our code together with the RMS map, beam, mask, and the power spectrum, fixing $\ell_{\rm mod} = 35$ and $\ell_{\rm max} = 40$[1]. In each case, we take care to start the chain in a random point, and discard burn-in afterwards.

7.1.2 Standardized estimators

We use the mode as our estimator of the true value for all parameters. In the case of α, q, and n, we standardize the difference between our estimate and the true value by using the standard deviation of the posterior, e.g., for chain j we have

$$z_{\alpha,j} = \frac{\mathrm{mode}(\alpha_j^{(t)}) - \alpha_0}{\sqrt{\langle(\alpha_j^{(t)} - \langle\alpha_j^{(t)}\rangle)^2\rangle}},$$

with corresponding definitions for z_q and z_n. The preferred direction posterior $p(\hat{p}|\mathbf{d})$ is less trivial, being a 2D distribution on the sphere. In the limit of a small dispersion, the sphere can be treated as a 2D plane. The radial

[1]It may have been a mistake to not let the observation be noise dominated at $\ell_{\rm max}$, however, there is not enough time to redo the simulations. An earlier set of simulations (prior to fixing a critical bug) had $\ell_{\rm max} = 47$, using the same noise level, and it displayed the exact same behaviour. We are therefore rather confident that this didn't affect the results.

profile of an azimuthally symmetric 2D Gaussian with variance σ^2 is *Rayleigh distributed*,

$$p(\psi|\sigma) = \frac{\psi}{\sigma} e^{-\frac{\psi^2}{2\sigma^2}}. \tag{7.1}$$

Motivated by this, we define $\psi^{(t)}$ as the radial distance of each sample $\hat{p}^{(t)}$ to the mode of $p(\hat{p}|\mathbf{d})$, and use

$$z_{\hat{p},j} = \frac{\psi_j^{(t)}}{\sqrt{\frac{1}{2}\langle(\psi_j^{(t)})^2\rangle}},$$

as our statistic, where the denominator comes from the Maximum Likelihood Estimator of the Rayleigh distribution parameter σ. From equation (7.1) we see that we can then compare directly with a Rayleigh distribution with $\sigma = 1$, at least in the limit of a small posterior. However, as our posterior will be rather wide, given the low $\ell_{\rm mod}$, we should not be surprised about systematic deviations from this coming from being on the sphere, and not in the 2D plane.

7.1.3 Results of simulations

We base our results on 500 simulations, with a true $\alpha_0 = 0.1$. Table 7.1 display our check of the Bayesian credibility interval claims made, which are in excellent agreement.

The question of an unbiased point estimate is a bit trickier. The results of our standardized estimates can be found in figure 7.1. We note that z_α appears to be a mixture of two Gaussians. One has large mass and is biased slightly high, while the other has small mass and is biased very low. By manually inspecting the corresponding posterior distributions, we find that this is reflected in two possible shapes of the posteriors. The majority of the posteriors are Gaussian-like, located randomly around the true value. The posteriors in the other group, corresponding to the heavy left tail in figure 7.1, have mode at or near zero and look like Gaussian distributions truncated to only include the right half. That is, they are very skewed, and our statistic z_α may not be the best one. The posterior credibility intervals still has the right coverage, so we accept this as an unavoidable, non-Gaussian feature of the estimator.

The single troublesome feature is the outlier that predicts α high by 4σ. The chain in question does however visit the true value of α on a few occasions, indicating that the left tail of the posterior could be much heavier than that of a Gaussian. As of this writing, the chain is being run for longer in order to check how anomalous the simulation is.

Table 7.1: Check of posterior Bayesian credibility intervals (CI). For each of $n = 500$ simulations, the narrowest possible 68% and 95% credibility intervals are computed from the posterior distribution. Then we simply count how many contain the true value. The uncertainties indicate standard deviation of the binomial distribution, $\sqrt{np(1-p)}/n$.

Variable	68% CI hit rate	95% CI hit rate
α	67% ± 2%	93% ± 1%
q	67% ± 2%	96% ± 1%
n	66% ± 2%	96% ± 1%

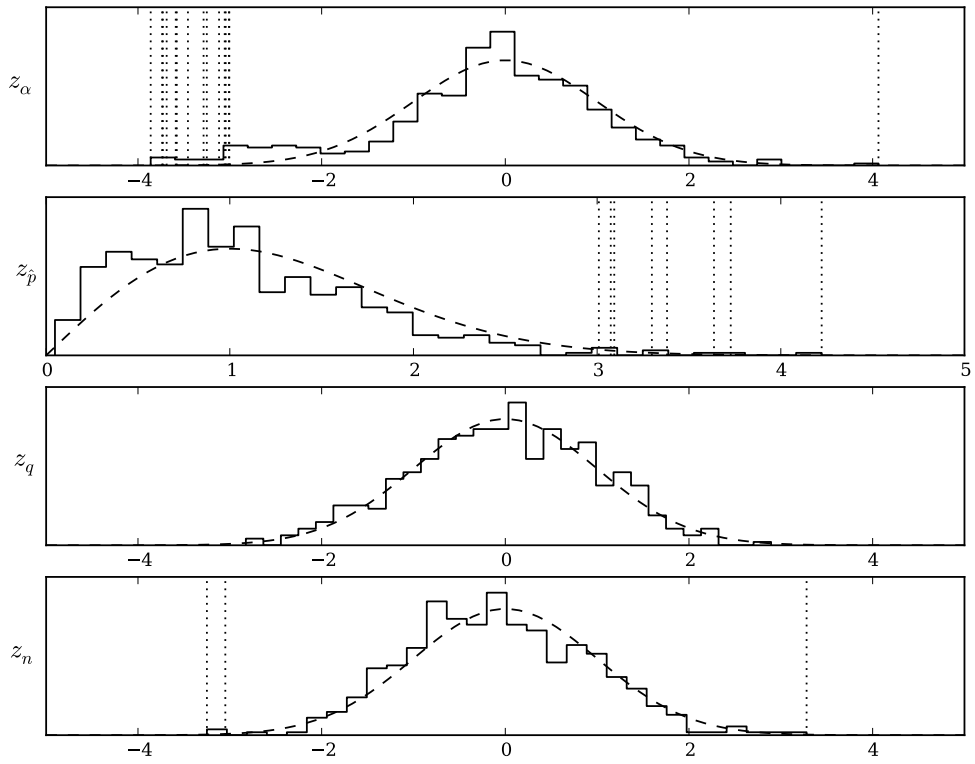


Figure 7.1: Plots of our standardized estimates when running a chain on each of 500 simulated data sets, with the true $\alpha_0 = 0.1$. Outliers (absolute value greater than 3) are marked with dotted vertical lines. No simulation produced an outlier in more than one variable, and plotting outliers against each other show no pattern. Over-plotted distributions are a standard Gaussian for α, q and n, and a standard Rayleigh for $\hat{p}$.

7.2 Analysis of downgraded data

Before running the code on full resolution WMAP data, we attempt to reproduce the results of Hoftuft et al. (2009). For easy comparison we use the same input, which is based on WMAP 5-year data, but downgraded to $N_{\rm side} = 32$. The map was smoothed with a 4.5° FWHM Gaussian beam, $1\,\mu K$ of RMS noise was added, and the KQ85 mask was directly downgraded by excluding pixels where more than half of the corresponding $N_{\rm side} = 512$-pixels were missing. We focus on the map based on the V band maps.

The results can be seen in figure 7.2 and table 7.2. While we certainly reproduce the main features of Hoftuft et al., there is some discrepancy, and the detection is not as strong. Since the results of Hoftuft et al. show a stronger detection for $\ell_{\rm mod} = 64$ than $\ell_{\rm mod} = 80$, we have also included an analysis where we assume that only multipoles $\ell = 65, \ldots, 80$ are modulated[2]. The resulting posterior is very much consistent with the others, showing weak evidence of modulation, but a consistent preferred direction, so that the effect is to pull the $\ell_{\rm mod} = 80$-posterior somewhat downwards.

The method of Hoftuft et al. is based on brute-force likelihood evaluation in pixel space of

$$\mathbf{C} = \mathbf{S} + \mathbf{N} + \mathbf{F},$$

where $\mathbf{F}$ is similar to our $\sigma_t^2 \mathbf{T}\mathbf{T}^T$. However, in addition to marginalizing over the monopole and the dipole, they marginalize over foreground templates, based on what has been subtracted in the foreground cleaned maps by the WMAP team. This difference in the treatment of the data could explain the discrepancy. This is supported by the fact that our results are much more in agreement with Hoftuft et al.'s estimates from the foreground-cleaned Internal Linear Combination (ILC) map, as well as the estimates when using the expanded KQ85e mask.

Evaluation in pixel space is based on converting the covariance matrix $\mathbf{S}$ to a pixel space basis. The dipole modulation was performed in pixel space, that is, $\mathbf{S} = \mathbf{M}\mathbf{Y}\mathbf{S}_{\rm iso}\mathbf{Y}^\dagger\mathbf{M}^\dagger$, where $\mathbf{M}$ is a diagonal pixel space matrix corresponding to the dipole modulation, $\mathbf{S}_{\rm iso}$ is a diagonal spherical harmonic space matrix, and $\mathbf{Y}$ is the spherical harmonic transform. A natural question now, without a representation of $\mathbf{M}$ in spherical harmonic space, is how to introduce the beam. We have learned that Hoftuft et al. let $\mathbf{S} = \mathbf{M}\mathbf{Y}\mathbf{A}\mathbf{S}_{\rm iso}\mathbf{A}\mathbf{Y}^\dagger\mathbf{M}^\dagger$. That is, their full model reads

$$\mathbf{d} = \mathbf{M}\mathbf{Y}\mathbf{A}\mathbf{s} + \mathbf{n} + \mathbf{f},$$

where the modulation is applied *after* beam smoothing. Since the beam in question is the 4.5° FWHM Gaussian beam used in the map downgrading process, it is unclear what the rationale would be for this model, other than

[2]This was achieved simply by setting α_ℓ accordingly, see chapter 4.

Table 7.2: Estimates of α at $N_{\rm side} = 32$.

			Present analysis	Hoftuft et al. (2009)
Band	Mask	$\ell_{\rm mod}$	α	α
V	KQ85	64	0.066 ± 0.022 (2.9σ)	0.080 ± 0.021 (3.8σ)
V	KQ85	80	0.057 ± 0.020 (2.9σ)	0.070 ± 0.019 (3.7σ)
ILC	KQ85	64		0.072 ± 0.022 (3.3σ)
V	KQ85e	64		0.067 ± 0.025 (2.7σ)

that it is much faster to compute when the modulation is represented in pixel space.

Since we have formulated $\mathbf{M}$ in spherical harmonic space, it is easy to try out the effect of this model. To emulate the behaviour of their code, we simply use $C_\ell^{\rm fid} b_\ell^2 p_\ell^2$ as our "power spectrum", and p_ℓ^{-1} as our "beam", where b_ℓ is the 4.5° FWHM Gaussian beam and p_ℓ the pixel window transfer function for $N_{\rm side} = 32$. We also fix the power spectrum parameters, $q = 1$ and $n = 0$. The result is shown as the dashed blue line in figure 7.2, which is in good agreement with the solid blue line. We therefore conclude that the difference in beam handling is unlikely to affect the results.

In summary, it seems likely that a difference in handling foregrounds is the reason for the discrepancy. To become sure of this, we should either run the code of Hoftuft et al. again without marginalization over foreground templates, or include the foreground templates in our code. The difference should be explained before publishing any final results, to make sure it is not caused by any bugs in the code.

7.3 Analysis of full resolution data

Time does not allow for a thorough analysis, but in table 7.3 we give some preliminary results from analysis of full resolution WMAP 7-year data. For each of the frequency bands V and W, we take a simple non-weighted average of the foreground cleaned maps. There are two maps from the V band and four maps from the W band, each map corresponding to a distinct radiometer. The reason for the naive averaging procedure is that it makes it trivial to also average the beams. We apply the KQ85y7 mask (Jarosik et al., 2010), shown in figure 1.1. In each case, the result include two chains started from different positions. Each chain was manually inspected and a reasonable amount of burn-in discarded by eye, but we give no guarantee that they have indeed converged.

The results so far appear to be consistent with both Hansen et al. (2009) and Hanson & Lewis (2009), in that the amplitude diminish, but that the significance[3] show no clear trend when including more data. In particular,

[3]We use this term loosely, meaning "the narrowness of the posterior distribution". The

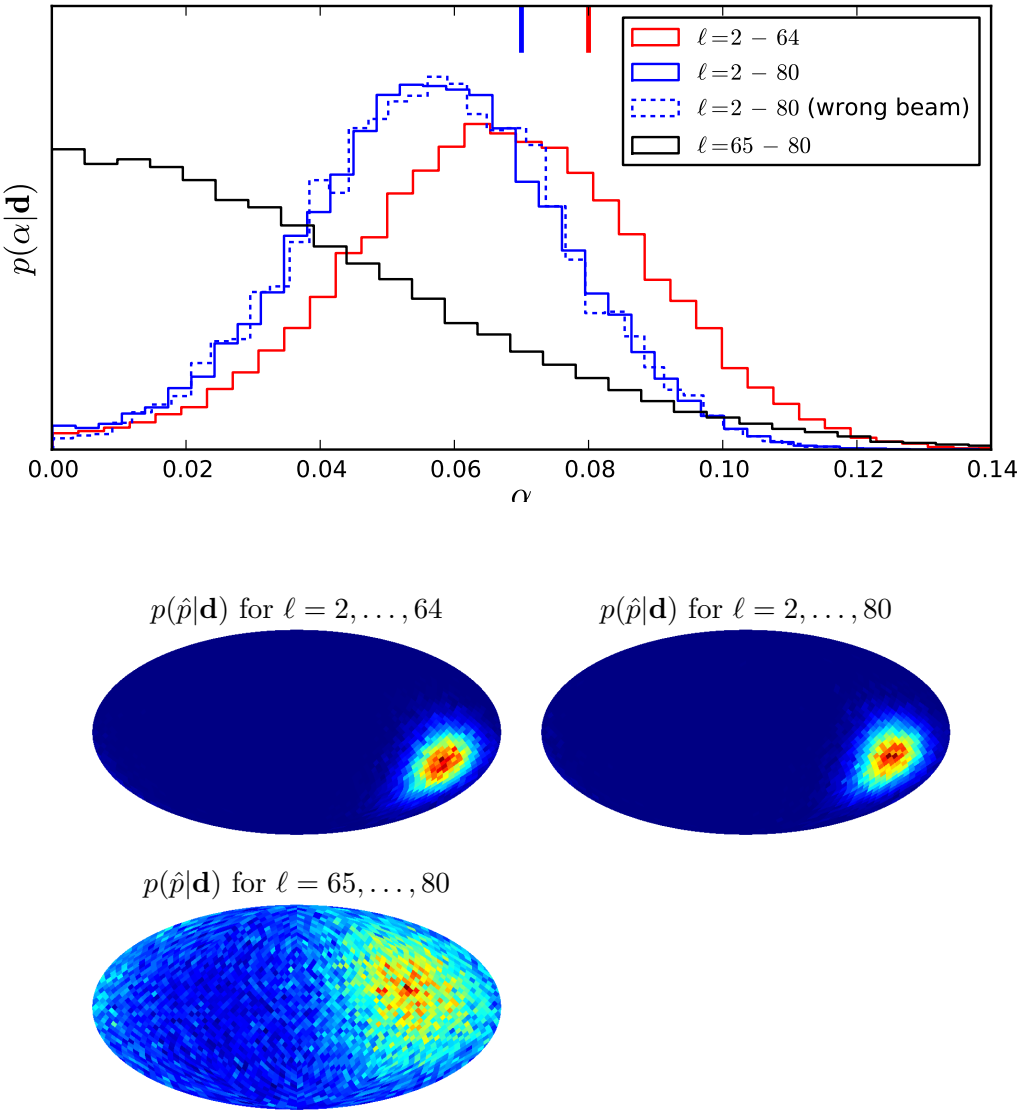
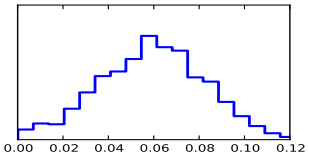
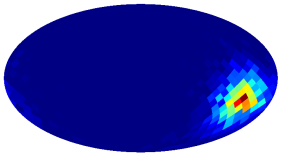
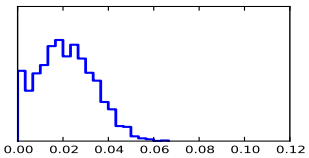
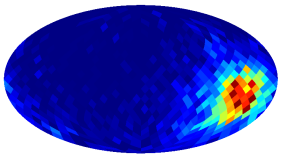
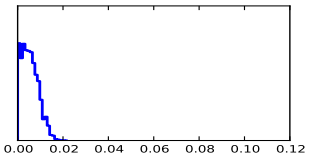
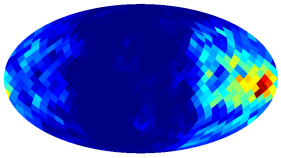
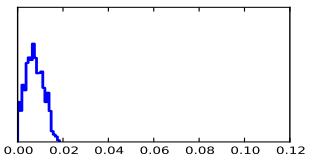
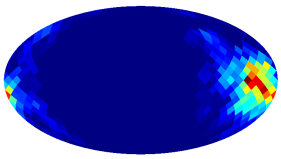
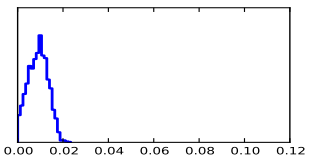
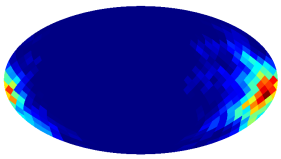
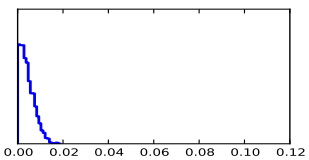
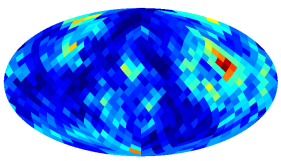
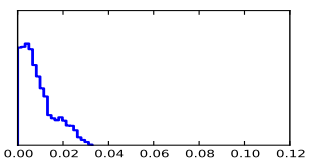
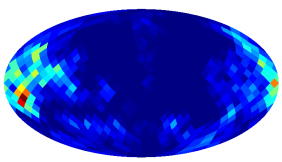


Figure 7.2: Results of repeating the analysis of Hoftuft et al. (2009) in the V-band, using the same input files. In addition, we include a new analysis where only multipoles $\ell = 65$–80 are assumed to be modulated. The corresponding posterior modes found by Hoftuft et al. are indicated by vertical lines on the top axis. See figure 1.4 for the full posterior found by Hoftuft et al. for $\ell_{\mathrm{mod}} = 64$. The power spectrum parameters q and n are not plotted, but are within 1.1σ of the null model.

Table 7.3: Results of WMAP 7-year, $N_{\rm side} = 512$ analysis. The ranges indicate the multipoles ℓ that are assumed to be modulated. Multipoles up to $\ell_{\rm max}$ are included when drawing constrained samples $\mathbf{s}$, with a CG tolerance of $\epsilon = 10^{-5}$. A new signal is sampled every 161st MCMC iteration, and the quantity denoted α-ACF is the auto-correlation in α, 161 MCMC steps apart.

		$p(\alpha\|\mathbf{d})$	$p(\hat{p}\|\mathbf{d})$
Data	V band, 2–64		
α	0.065 ± 0.023 (2.8σ)		
q	0.988 ± 0.025 (0.5σ)		
n	0.058 ± 0.049 (1.2σ)		
$\ell_{\rm max}$	300		
# of $\mathbf{s}$	40 (α-ACF=0.01)		
Data	V band, 2–120		
α	0.018 ± 0.012 (1.5σ)		
q	0.994 ± 0.014 (0.4σ)		
n	0.033 ± 0.026 (1.3σ)		
$\ell_{\rm max}$	500		
# of $\mathbf{s}$	30 (α-ACF=0.02)		
Data	V band, 2–400		
α	0.001 ± 0.004 (0.1σ)		
q	1.003 ± 0.004 (0.7σ)		
n	0.019 ± 0.008 (2.2σ)		
$\ell_{\rm max}$	850		
# of $\mathbf{s}$	54 (α-ACF=0.18)		
Data	V band, 2–600		
α	0.007 ± 0.004 (1.8σ)		
q	1.014 ± 0.004 (3.6σ)		
n	0.022 ± 0.006 (3.5σ)		
$\ell_{\rm max}$	850		
# of $\mathbf{s}$	39 (α-ACF=0.36)		
Data	W band, 2–600		
α	0.010 ± 0.004 (2.3σ)		
q	1.021 ± 0.004 (4.6σ)		
n	0.021 ± 0.007 (2.8σ)		
$\ell_{\rm max}$	850		
# of $\mathbf{s}$	61 (α-ACF=0.44)		
Data	V band, 200–400		
α	0.000 ± 0.003 (0.1σ)		
q	1.010 ± 0.005 (1.9σ)		
n	0.019 ± 0.009 (2.2σ)		
$\ell_{\rm max}$	850		
# of $\mathbf{s}$	63 (α-ACF=0.06)		
Data	V band, 401–600		
α	0.002 ± 0.007 (0.3σ)		
q	1.028 ± 0.006 (4.8σ)		
n	0.024 ± 0.007 (3.6σ)		
$\ell_{\rm max}$	850		
# of $\mathbf{s}$	59 (α-ACF=0.64)		

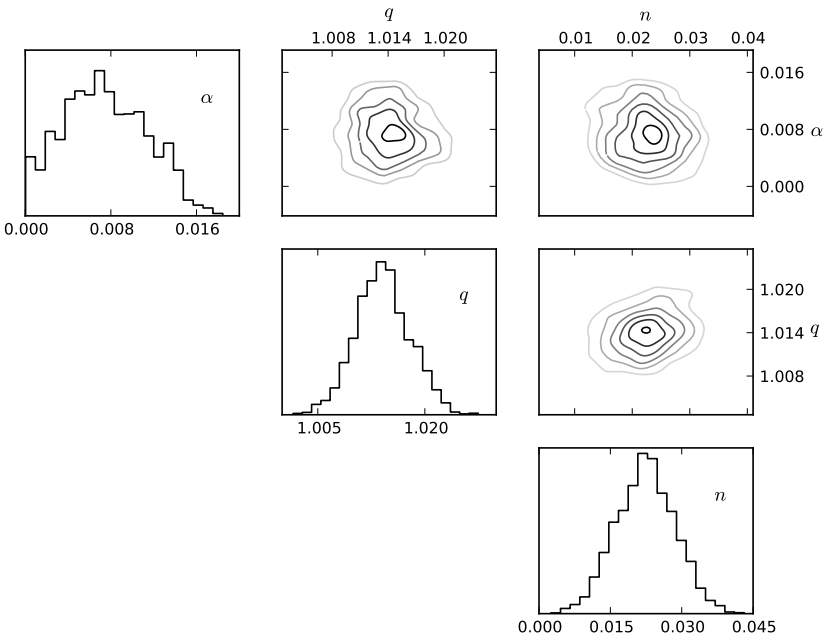


Figure 7.3: Posterior marginals for V band, ℓ-range 2–600 (corresponding to fourth row of table 7.3).

the significance does not approach zero.

We note that at smaller scales, q and n deviate from the null model. This is something that should be looked into in closer detail. Based on the discussion in section 4.4.1, it is clear that there is some theoretical coupling between α and the the power spectrum, but on these scales, α is very low. Also, the posteriors show no correlation in the parameters, which one could expect if this was the reason (see figure 7.3). A more likely explanation at these scales is that unresolved point sources affect the power spectrum.

For comparison, one should try not only to run each chain longer, but also to run them with higher numerical accuracy. The tolerance of the CG search, ϵ, is for the results presented set as high as 10^{-5}, which may be a possible source of inaccuracy. Also, any finite $\ell_{\max}$ represents an approximation that can cause problems when inverting $\mathcal{A} = (\mathbf{1} + \mathbf{F}^T\mathbf{N}^{-1}\mathbf{F})$. This is mostly a concern for the analyses where we let $\ell_{\max} = 300$ and 500. At $\ell_{\max} = 850$, the WMAP data is noise dominated, $\mathcal{A}$ becomes closer to a diagonal matrix, and truncation at $\ell_{\max}$ is less likely to be a problem. In any case, a couple of chains with $\ell_{\max} = 1000$ and $\epsilon = 10^{-7}$ for comparison should shed some light on this issue.

The reason there are no results beyond $\ell_{\mathrm{mod}} = 600$ is that those chains did not converge in the time available, due to a badly tuned proposal distribution

σ values quoted should never be translated into p-values under Gaussian assumptions, as it is evident that the left tail is heavy in all cases.

for $\hat{p}$. Some more effort must therefore be spent to improve the tuning as ℓ is pushed upwards. The last row in table 7.3 suffer from the same problem, because the proposal rules do not at the moment take the lower limit into account.

7.4 Running time

It has been noted many times that our method has a computational scaling of $O({\ell_{\text{max}}}^3) = O({N_{\text{pix}}}^{3/2})$, since for practical applications, $O({\ell_{\text{max}}}^2) = O(N_{\text{pix}})$. Still, decent theoretical scaling is of no use unless the prefactor is reasonable.

The computational behaviour of our method is very well understood. No sparse decompositions are required, which could have been a possible source of unpredictable behaviour. The only routines that scale as $O({\ell_{\text{max}}}^3)$ are found in HEALPix; `alm2map`, `map2alm` and `rotate_alm`.

For each round of parameter sampling, `rotate_alm` is called 40 times. From profiling the code at a moderate $\ell_{\text{mod}} = 100$, we see that `rotate_alm` already claims about 80% of the running time in this step. The exact cost of a `rotate_alm` call can be seen in figure 4.3. It is comparable to that of the spherical harmonic transforms, and since a CG search requires over a thousand of those, the parameter sampling phase can be neglected.

Profiling a CG search at $\ell_{\text{max}} = 850$, $N_{\text{side}} = 512$, we see that `alm2map` occupies 40% of the time and `map2alm` 30% of the time. The remaining 30% is spent in applying the preconditioner matrix, scaling as $O(\ell^4_{\text{precond}})$, or applying $\mathbf{N}^{-1}$, scaling as $O(N_{\text{pix}})$. The most important part is how many CG iterations are required. With $\ell_{\text{precond}} = 60$ and CG tolerance parameter $\epsilon = 10^{-5}$, the above parameters, and the data from the V frequency band, the number of iterations varied between 600 and 700. One iteration is measured at 1.8 seconds wall time when running in parallel on 8 cores[4].

The final multiplier is the number of Gibbs samples we desire. This is something that can only be reliably answered after doing more data analysis. An educated guess is that 5 chains, each run for 100 iterations with 30 iterations discarded as burn-in, will provide a good representation of a given posterior distribution. Using 8 cores for each chain, this means we have to wait a wall time of 35 hours, or 1400 CPU hours in total.

The performance for low resolutions is also of interest, especially for Monte Carlo simulations. The $N_{\text{side}} = 16$ experiment earlier in this chapter required 6 seconds per sample on a single core, or about 1000 CPU hours for 1000 simulations, each run for 600 Gibbs samples. For $N_{\text{side}} = 32$, $\ell_{\text{max}} = 64$, each sample took about 270 CPU seconds.

[4]Intel Xeon E5430, 2.66GHz, 6 KB cache. Intel Fortran was used for compiling HEALPix. The built-in Fast Fourier Transform was used. We rely on using HEALPix in OpenMP mode for parallelization.

Chapter 8

Conclusions & prospects

8.1 Improved methods and new code

Based on the CMB Gibbs sampling framework of Jewell et al. (2004) and Wandelt et al. (2004), a method has been developed for efficient estimation of hemispherical power asymmetry in the CMB, using a class of parametric phenomenological models where an isotropic, Gaussian CMB signal is assumed to be modulated. While we have focused in particular on a dipolar modulation field, the computational foundation is also laid for many generalizations that we detail below.

The existing exact method for the dipole modulation model (Hoftuft et al., 2009, Eriksen et al., 2007, Gordon, 2007) scales as $O({N_{\rm pix}}^3)$, while the method presented here scales as $O({\ell_{\rm max}}^3) = O({N_{\rm pix}}^{3/2})$. The constant overhead is low enough for most practical purposes, with an estimated 1500–2000 CPU hours for an analysis of WMAP data at full resolution. The method should scale well to the resolution provided by the Planck experiment, as all the routines that scale as $O({\ell_{\rm max}}^3)$ are well understood. In particular, sparse linear algebra is kept at the $O({\ell_{\rm max}}^2)$ level and will not cause any surprises when scaling up.

Further optimizations are likely to improve on the running time. In particular, there is potential in the CG preconditioner. There also seems to be some potential in spending more effort on tuning the MCMC kernel for higher resolutions and *a posteriori* tuning for the signals sampled when constrained by WMAP data. Repeating the analysis of Hoftuft et al. (2009) at $N_{\rm side} = 32$ required less 75 CPU hours for a single data set, and an analysis at resolution $N_{\rm side} = 16$ only require a couple of CPU hours. Monte Carlo simulations are therefore possible at these resolutions.

The method discussed has been implemented from scratch in Python, together with many tools for automating simulation, downgrading observations, plotting, etc.. The resulting code is believed to be in working order, although in chapter 7 we have noted a few things that remains to be checked before any results are published.

Writing code from scratch is a great learning device. The drawback is

that we are left with a Gibbs sampler that is not able to do joint CMB model and foreground estimation, which is something Commander is capable of (Eriksen et al., 2008). While foregrounds may be shown to be unimportant by other methods, it would clearly be more flexible to have the option of joint foreground estimation. There are three ways to make this happen. First, one can re-implement the foreground sampling in PyCMB. Or, one can instead implement the dipole modulation model in Commander, which requires using a library for sparse linear algebra from Fortran. Finally, the quickest approach may be to call the constrained signal and foreground sampler of Commander from Python. Note that merely marginalizing over foreground templates, in order to guard against over-subtraction of foregrounds in the foreground cleaned maps, is something that is in place in PyCMB, although it was not utilized (beyond monopole and dipole templates) in our analysis of WMAP data.

8.2 Is the universe isotropic?

The focus in this thesis has very much been on the computational aspects and development of new methods, with less focus on the cosmological question and the analysis of real data. While we present some new results on WMAP 7-year data in chapter 7, it remains to apply the method to data in a more systematic fashion, including:

- Estimation over more multipole ranges, allowing direct comparisons with both Hanson & Lewis (2009) and Hansen et al. (2009).
- Include more frequency bands.
- Check sensitivity to foreground contamination, such as analysing with different masks, and repeat the analysis on raw, non-foreground cleaned data (as done by Hanson & Lewis, 2009).

And, *in particular*, each chain needs to run quite a bit longer, and be checked for convergence, using, e.g., the Kolmogorov-Smirnov test between chains (Robert & Casella, 2004).

Still, an informed guess is that the asymmetry becomes less pronounced at higher $\ell's$, in agreement with Hanson & Lewis (2009), but that there is asymmetry present at all scales with a consistent preferred direction, in agreement with Hansen et al. (2009). Like both of those analyses, we see that the significance of the result vary depending on what data is included. The 400–600 range contains 201,201 data points, while the 2–65 range contains 4,221 data points. Considering that the significance, at best, stays constant, it is clear that the dipole modulation model with constant modulation amplitude is not a very good fit.

Bennett et al. (2010) claim that the reason for this is that the effect is a statistical fluke, driven by *a posteriori* bias. However, under the assumption of

isotropy (and the absence of related systematic effects), the preferred direction should be entirely uncorrelated between one set of scales and another. It seems that Bennett et al. suggests that if one just search enough ranges, one can find any preferred anisotropy direction. If the effect is a fluke, then it should be possible to, e.g., find a multipole range that prefers the direction $(0°, 0°)$, just by looking for long enough.

Both our trial runs in chapter 7 and the study of Hansen et al. (2009) strongly suggest that this is not the case (although there are certainly more sets left to probe). It seems much more likely that something is indeed going on, but that the dipole modulation model in its current form is suboptimal in capturing it. We already noted that the phenomenological postulation of the dipole modulation model can be compared with linear regression, in that it is designed for picking up rough trends for what is likely to be a much more complicated phenomenon. The natural approach at this stage is to generalize the model, and we discuss a few such generalizations below. Whatever this effect turns out to be in the end, it is likely to be much more interesting than a statistical fluke, whether the end of the story is new physics or "only" a better model for foregrounds.

Finally, some words on model selection, a topic Bennett et al. (2010) give particular attention, in noting that current exact methods are too slow to allow for Monte Carlo simulations:

> *Comparing these methods, we find that the Hanson & Lewis (2009) optimal quadratic estimator has significant advantages [...] statistical significance can be assessed straightforwardly by comparing the estimator with an ensemble of Monte Carlo simulations. In particular, maximum likelihood analyses [...] are not a sufficient substitute for true Monte Carlo simulations, which directly give the probability for a simulation to be as anomalous as the data.*

However, this neglects another important aspect. In any model selection setting, there will be infinitely many possible estimators, but not all will have the same power in rejecting a null hypothesis. It is not surprising in itself that an approximate method, subject to approximation errors, finds lower significance and has less rejection power. What must be considered is the precision of the estimator and its power to invalidate the null hypothesis. Hanson & Lewis (2009) are much more cautious with respect to their estimator;

> *[...] in the limit of weak anisotropy this QML estimator [...] is optimal in the minimum-variance sense. In practice, "weak" means non-detection, and so this form of quadratic estimator is excellent for testing statistical isotropy, but needs to be treated with care if a significant detection is made.*

In this setting, what matters is the numerics, not whether or not the cosmological research community consider the result significant. With p-values

lower than 0.01 in some cases, it seems that the QML approach must be further validated before one can trust it to have optimal power in rejecting the isotropic null hypothesis.

Since the Bayesian posterior with a flat prior is identical to the likelihood function, the method developed in this thesis can be used to repeat the Monte Carlo simulations that Bennett et al. crave at large scales (approximately $\ell_{\rm mod} \leq 80$). This is the region where the anisotropy appears to be strongest, and where the QML estimator is therefore most likely to perform poorly. This would also provide valuable insight into the accuracy of the QML estimator. Monte Carlo p-values with an exact estimator for smaller scales seems to still be out of reach. The Bayesian posterior distribution $p(\alpha|\mathbf{d})$, while giving a good subjective indication, does not quantitatively answer the question of model selection. Instead, the Bayesian approach would be to postulate two possible models, M_0 and M_1, and compute their respective posterior probabilities (or their *odds ratio*). We note that there exists algorithms, such as *Reversible Jump MCMC* (Robert & Casella, 2004), that could possibly allow for such exact Bayesian model selection also at small angular scales.

8.3 Generalizations

A phenomenological model is introduced not as a physical hypothesis, but simply as another way of looking at the data. It seems that the dipole-modulation model may pick up something beyond a statistical fluke, but that it is far from a perfect fit, and highly dependent on the data included. Rather than scrutinizing the current dipole-modulation model even further, it seems more interesting to consider generalizations. Two such generalizations seem to stand out in particular. First, one could consider a wider class of arbitrary azimuthally symmetric modulation fields. Second, one could consider a scale dependent modulation amplitude. Both of these fit within the framework of chapter 4, and should have the same computational scaling. A realistic estimate is that these more general models can be fit in 3–4 times the CPU time that the dipole modulation model requires.

8.3.1 Azimuthally symmetric modulation fields

Is the consistent direction of (real or spurious) hemispherical power asymmetry due to localized features? Is it a patch on the northern hemisphere that has less power than the average, or a patch on the southern hemisphere that has more power than the average? Or is it indeed a symmetric hemispherical effect?

Hansen et al. (2009) already probed into this question by computing power in 45°, 90° and 180° discs. We here outline a parametric modulation approach to do the same. In the dipole modulation model we take γ as $\gamma(\hat{p}\cdot\hat{n}) = 1 + \alpha\hat{p}\cdot\hat{n}$, while to probe for the amount of locality we could instead parametrize

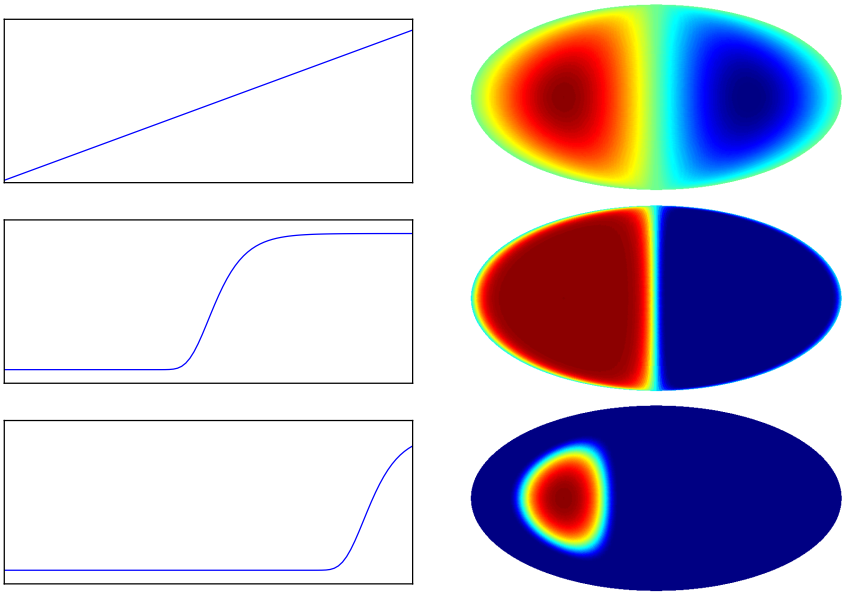


Figure 8.1: Sigmoid power asymmetry. The function $\gamma(t)$ on the left, and the corresponding modulation field $\gamma(\hat{p}\cdot\hat{n})$ on the right. These three examples are based on the three-parameter Gompertz function, $\gamma(\hat{p}\cdot\hat{n}) \propto a\exp(-b\exp(-c\hat{p}\cdot\hat{n})) - 1$. **Top:** As c becomes small, we approach the dipole modulation field.

$\gamma(\hat{p}\cdot\hat{n})$ as some sigmoid function. Figure 8.1 shows some examples using the Gompertz function, which introduce two more parameters to indicate locality and steepness of gradient. Another example that could work well is the cumulative Gaussian distribution. In any case, it seems likely that it would be necessary to put strong priors on the shape parameters. This would merely serve to further specify the constraints on the modulation field, and the amplitude would still be left with an uninformative prior.

Even if the posterior end up having many modes, and the model priors end up including many iteratively applied *a posteriori* choices, the process of fitting such a model to data should give much insight into why we see the effects that we do with the dipole modulation model. In particular, we could get an indication about whether the the posterior is driven by local or global features.

Computationally, recall from chapter 4 that when writing $\mathbf{M} = \mathbf{R}\mathbf{M}_{\hat{z}}\mathbf{R}^\dagger$, then $\mathbf{M}_{\hat{z}}$ is block-diagonal in ℓ, so likelihood evaluation would still scale as $O({\ell_{\mathrm{max}}}^3)$. However, as $\mathbf{M}_{\hat{z}}$ would contain up to $O({\ell_{\mathrm{max}}}^3)$ elements, each CG iteration is likely to take 3–4 times as long, due to two multiplications with $\mathbf{M}_{\hat{z}}$ and two rotations $\mathbf{R}$ in each iteration.

8.3.2 Scale-dependent modulation amplitude

It would be interesting to explore the apparent scale-dependency of α, which seems to diminish with higher ℓ_{mod}. A "modulation spectrum" α_ℓ should

provide many hints about this. Motivated by what we see in the data, we may, e.g, model α_ℓ as a power law in two parameters. Another approach is to follow Hansen et al. (2009) in inspecting various multipole ranges in isolation. It would not be necessary to do a separate analysis for each bin. Instead one would simply model α_ℓ as a piece-wise constant function and estimate α_ℓ for each bin jointly.

Another approach, provided a sampling algorithm can be found, is to simply model each α_ℓ as free parameters and instead require a certain degree of smoothness, e.g.,

$$\mathrm{Cov}(\alpha_\ell, \alpha_{\ell+1}) = \tau,$$

where τ is either fixed, or itself a random variable with a possibly informative prior distribution, corresponding to a form of splining by hierarchical Bayesian modelling.

If the approach of modelling free α_ℓ is successful, that is, the α_ℓ's are well enough constrained by data with a fairly loose prior, then we should have excellent hints as to how to proceed in further exploring power asymmetry. If the universe is truly isotropic, there should be little or no system, with many α_ℓ's being close to zero because the preferred direction would be a bad fit. That is, one could imagine seeing a few very significant outlier scales that essentially fix the preferred direction $\hat{p}$, and that other scales either agree with this direction, or have their α_ℓ's forced to zero.

Finally, a remark on sampling free α_ℓ's. It is likely to be trickier to sample from $p(\alpha_\ell|\hat{p}, \theta_{\mathrm{iso}}, \mathbf{s})$ than the other models, given the large number of correlated parameters. Evaluating $p(\alpha_\ell|\hat{p}, \theta_{\mathrm{iso}}, \mathbf{s})$ scales as $O(L_{\mathrm{max}}{\ell_{\mathrm{max}}}^2)$ where L_{max} is the band-limit of the modulation field, so taking many, many steps per sampled signal is feasible to make up for a mediocre sampler. Even making a step one ℓ at the time, with a few hundred passes per sampled signal, would still be no more expensive than our typical CG search. Tuning the covariance of a multivariate normal proposal density over all α_ℓ's should also be tried for comparison. For a middle road between the two, one could sample independent α_ℓ in blocks. Finally, in the special case of a dipolar modulation field we mention, as a rather far-out and untested idea, that one could attempt to use the formulas for tri-diagonal matrices given by Usmani (1994) to find expressions for $\mathbf{M}_{\hat{z}}^{-1}$ and $|\mathbf{M}_{\hat{z}}|$, say, up to first order in α_ℓ for a single ℓ given the others. Failing that, the formulas of Usmani could still provide some insights into the structure of the correlation between the α_ℓ's in the posterior. We stress that these are merely loose ideas that may well turn out to be unfruitful.

8.3.3 Physically motivated models

Any model that is physically motivated should look at least slightly different from the phenomenological models we have proposed, since the anisotropic couplings would tend to be introduced in real space/Fourier space, "in $\vec{k}$",

and not on the sphere, "in (ℓ, m)". Still, as long as the signal is Gaussian, one can find covariance matrices for the models also on the sphere. One example is Groeneboom & Eriksen (2009), who fit the "Ackerman-Carroll-Wise" model using an algorithm very similar to ours, but with a physically motivated model. As long as the model predicts a sparse covariance matrix, Gibbs sampling should be a promising approach.

Groeneboom & Eriksen (2009) have noted that their approach is only viable up to $\ell_{\text{mod}} \sim 800$, using large amounts of CPU time. It is reasonable to expect similar computational constraints on other physically motivated models, which may not factor as nicely as the modulation field model. Some ideas are:

- One can attempt to factor the covariance matrix analytically as $\mathbf{S} = \mathbf{M}\mathbf{S}'\mathbf{M}^\dagger$, where $\mathbf{S}'$ is sparser than $\mathbf{S}$, and where either $\mathbf{M}$ is is sparse or $\mathbf{M}$, $\mathbf{M}^\dagger$ and $\mathbf{M}^{-1}$ can be efficiently applied to a vector. As an example, perhaps it is possible to give $\mathbf{S}'$ as a primordial covariance in $\vec{k}$, $P(\vec{k}, \vec{k}')$, and let $\mathbf{M}$ contain the Einstein-Boltzmann transfer functions and the integral required to project from the Fourier representation to the spherical harmonics.

- Much research has gone into routines for finding good permutations of sparse matrices prior to factorization, which should be used instead of a naive direct Cholesky factorization. There are also libraries for computing sparse factors in parallel (see appendix A.6) which could help push ℓ_{mod} somewhat.

- One should attempt to increase sparsity somewhat by introducing a rotation matrix $\mathbf{R}$. If at all possible, only have couplings in ℓ, not in m, so that band-diagonal LAPACK routines can be used.

Appendix A

Toolbox

A.1 Complex spherical harmonics

For spherical harmonics we rely on the conventions and properties found in Press et al. (2007). A complex field on the sphere f can be expanded in spherical harmonics,

$$f(\hat{n}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\hat{n}), \tag{A.1}$$

with $\hat{n}$ a unit vector in Euclidian 3D space. We will also refer to points on the sphere by colatitude θ and longitude ϕ. The spherical harmonics, $Y_{\ell m}$, are given by

$$Y_{\ell m}(\theta, \phi) = \sqrt{\frac{(2\ell+1)}{4\pi} \frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(\cos\theta) e^{im\phi} \tag{A.2}$$

when $|m| \leq \ell$, and 0 otherwise. Here P_ℓ^m are the associated Legendre polynomials,

$$P_\ell^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m} P_\ell(x),$$

where P_ℓ are the ordinary Legendre polynomials. A useful symmetry is that $Y_{\ell m} = (-1)^m Y^*_{\ell -m}$.

The spherical harmonics are orthogonal on the sphere surface,

$$\int Y_{\ell m}(\hat{n}) Y^*_{\ell' m'}(\hat{n}) d\Omega = \delta_{\ell\ell'} \delta_{mm'}, \tag{A.3}$$

where we use $d\Omega$ to indicate integration over the sphere surface. By multiplying equation (A.1) with $Y^*_{\ell' m'}(\hat{n})$ on both sides and integrating, we find the inverse transform. Since

$$\int f(\hat{n}) Y^*_{\ell' m'}(\hat{n}) d\Omega = \int \sum_{\ell m} a_{\ell m} Y_{\ell m}(\hat{n}) Y^*_{\ell m}(\hat{n}) d\Omega = a_{\ell m} \delta_{\ell\ell'} \delta_{mm'},$$

we have

$$a_{\ell m} = \int f(\hat{n}) Y^*_{\ell' m'}(\hat{n}) d\Omega. \tag{A.4}$$

It is useful to note that fields that are azimuthally symmetric with respect to the z-axis are only expanded in terms of $m = 0$-modes. Further notes on computing with spherical harmonics are given in 3.3.

Result 1. *Let $\hat{p} = (x', y', z')$. Then, the dipole modulation field $\gamma(\hat{n}) = 1 + \alpha(\hat{p} \cdot \hat{n})$ is represented in spherical harmonics as*

$$\gamma_{00} = \sqrt{4\pi}, \qquad \gamma_{1-1} = \sqrt{\frac{2\pi}{3}}\alpha(x' + iy'),$$
$$\gamma_{10} = \sqrt{\frac{4\pi}{3}}\alpha z', \qquad \gamma_{11} = -\sqrt{\frac{2\pi}{3}}\alpha(x' - iy'),$$

while for $L > 1$ we have $\gamma_{LM} = 0$.

Proof. The first few spherical harmonics $Y_{\ell m}(\hat{n})$ with $\hat{n} = (x, y, z)$ are (Edmonds, 1957):

$$Y_{00} = \sqrt{1/4\pi} \qquad Y_{1-1} = \frac{1}{2}\sqrt{3/2\pi}(x - iy)$$
$$Y_{10} = \sqrt{3/4\pi}z \qquad Y_{11} = -\frac{1}{2}\sqrt{3/2\pi}(x + iy).$$

Solving for the coordinates, we have

$$1 = \sqrt{4\pi}Y_{00} \qquad x = \sqrt{2\pi/3}(Y_{1-1} - Y_{11})$$
$$z = \sqrt{4\pi/3}Y_{10} \qquad y = i\sqrt{2\pi/3}(Y_{1-1} + Y_{11}).$$

So,

$$\begin{aligned}
\gamma_{LM} &= \int (1 + \alpha(xx' + yy' + zz')Y^*_{LM} d\Omega \\
&= \int \sqrt{4\pi} Y_{00} Y^*_{LM} d\Omega + \\
&\quad \alpha\sqrt{\frac{2\pi}{3}} \int \left(x'(Y_{1-1} - Y_{11}) + iy'(Y_{1-1} + Y_{11}) + \sqrt{2}z' Y_{10} \right) Y^*_{LM} d\Omega \\
&= \int \sqrt{4\pi} Y_{00} Y^*_{LM} d\Omega + \\
&\quad \alpha\sqrt{\frac{2\pi}{3}} \int \left((x' + iy')Y_{1-1} - (x' - iy')Y_{11} + \sqrt{2}z' Y_{10} \right) Y^*_{LM} d\Omega.
\end{aligned}$$

Since the spherical harmonics are orthogonal, $\int Y_{\ell m} Y^*_{\ell' m'} d\Omega = \delta_{\ell\ell'}\delta m m'$, each choice of L, M picks out one of the terms, and the result follows. □

A.2 Spherical harmonics of real fields

The spherical harmonic transform results in complex $a_{\ell m}$, even when the field to be expanded in spherical harmonics is real. However, it is possible to use a different convention, real spherical harmonics, where each $a_{\ell m}$ is a real number. This is in particular important because it allows convenient use of the Conjugate Gradients algorithm, which (at least in the form commonly given) assumes that the linear system is in $\mathbb{R}$.

First, observe that if $f(\hat{p}) \in \mathbb{R}$, that is, $f(\hat{p}) = f(\hat{p})^*$, then

$$f(\hat{p}) = \sum_{\ell m} a_{\ell m} Y_{\ell m}(\hat{p}) = \sum_{\ell m} a^*_{\ell m} Y^*_{\ell m}(\hat{p}) \tag{A.5}$$

$$= \sum_{\ell m} a^*_{\ell m} (-1)^m Y_{\ell -m}(\hat{p}) = \sum_{\ell m} (-1)^m a^*_{\ell -m} Y_{\ell m}(\hat{p}), \tag{A.6}$$

where we use the identity $Y^*_{\ell m}(\hat{p}) = (-1)^m Y_{\ell -m}(\hat{p})$. Since this must hold for *any* field f (in particular, it must hold e.g. for a uniform field, any perfect monopole, any perfect dipole, and so on), we must have

$$a_{\ell m} = (-1)^m a^*_{\ell -m}. \tag{A.7}$$

Therefore, it is enough to store the $\ell + 1$ complex coefficients for each ℓ where $m \geq 0$. Also note that a_{00} is real.

Alternatively, one can reorder the data for each ℓ into a set of $2\ell + 1$ real coefficients. Let $\mathbf{a}^C$, $a^C_{\ell m}$ denote complex coefficients and $\mathbf{a}^R$, $a^R_{\ell m}$ the corresponding real coefficients. We then let

$$a^C_{\ell m} = \begin{cases} a^R_{\ell m} & \text{for } m = 0 \\ (a^R_{\ell m} + i a^R_{\ell -m})/\sqrt{2} & \text{for } m > 0, \end{cases} \tag{A.8}$$

and for $m < 0$ we must have $a^C_{\ell m} = (-1)^m (a^C_{\ell -m})^*$. This choice is made because it leads to the transformation being an unitary linear operation: The relationship can be expressed as a linear operator, $\mathbf{U}$, which we define by $\mathbf{a}^R = \mathbf{U}\mathbf{a}^C$ and $\mathbf{a}^C = \mathbf{U}^\dagger \mathbf{a}^R$. From this fact follows the inverse transform,

$$\begin{aligned} a^R_{\ell,0} &= a^C_{\ell,0} \\ a^R_{\ell m} &= \sqrt{2}\mathrm{Re}(a^C_{\ell m}) && \text{for } m > 0 \\ a^R_{\ell m} &= \sqrt{2}\mathrm{Im}(a^C_{\ell -m}) && \text{for } m < 0. \end{aligned} \tag{A.9}$$

The matrix $\mathbf{U}$ is block-diagonal with one block for each ℓ, and for e.g. $\ell = 2$ the corresponding block in $\mathbf{U}$ is

$$\frac{1}{\sqrt{2}} \begin{bmatrix} i & 0 & 0 & 0 & -i \\ 0 & -i & 0 & -i & 0 \\ 0 & 0 & \sqrt{2} & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{A.10}$$

The pattern repeats for higher ℓs. Since the pattern is the same for every ℓ, we write $U_{\ell m,\ell' m'} = \delta_{\ell\ell'} u_{mm'}$, with

$$u_{mm'} = (1/\sqrt{2}) \begin{cases} \sqrt{2} & m = m' = 0 \\ 1 & m > 0, m' = m \\ (-1)^{m'} & m > 0, m' = -m \\ -i & m < 0, m' = -m \\ (-1)^{m'} i & m < 0, m' = m \\ 0 & \text{otherwise.} \end{cases} \tag{A.11}$$

Transforming a vector of complex spherical harmonic coefficients to its real counterpart and vice versa is trivial by using equations (A.8) and (A.9). The same can be done to dense matrices: First apply, e.g., equation (A.9) to all the columns, and then to all the resulting rows. For sparse matrices, one simply form $\mathbf{U}$ explicitly as a sparse matrix, then use generic routines for sparse matrix multiplication to compute $\mathbf{UKU}^\dagger$, and then discard the imaginary part (which is zero up to numerical errors).

Result 2. *Let* $\mathbf{J}$ *be a matrix such that* $J_{\ell m,\ell' m'} = \delta_{mm'} J_{\ell m,\ell' m'} = J_{\ell -m,\ell' -m}$. *Then* $\mathbf{J}$ *is the same in complex and spherical harmonics, that is,* $\mathbf{UJU}^\dagger = \mathbf{J}$. *Furthermore,* $\mathbf{U}$ *is a unitary matrix.*

Proof. When $m \neq 0$, we see (by computing each case) that

$$\begin{aligned} u_{mm} u^*_{mm} = u_{m-m} u^*_{m-m} &= \frac{1}{2} \\ u_{mm} u^*_{m-m} + u_{mm} u^*_{-mm} &= 0 \\ u_{mm} u^*_{-mm} + u_{m-m} u^*_{-m-m} &= 0 \end{aligned}$$

So,

$$\begin{aligned} (\mathbf{UJU}^\dagger)_{\ell m,\ell' m'} &= \sum_{LM} \sum_{L'M'} \delta_{\ell L} u_{mM} \delta_{LL'} J_{LM,L'M'} \delta_{\ell' L'} u^*_{m'M'} \\ &= \sum_M J_{\ell M,\ell' M} u_{mM} u^*_{m'M}. \end{aligned}$$

When $m = 0$, $u_{0M} u^*_{m'M} = \delta_{0M}$, so the sum is $J_{\ell m,\ell' m}$. When $m = m' \neq 0$, we have

$$\begin{aligned} \sum_M J_{\ell M,\ell' M} u_{mM} u^*_{mM} &= J_{\ell m,\ell' m} u_{mm} u^*_{mm} + J_{\ell -m,\ell' -m} u_{m-m} u^*_{m-m} \\ &= J_{\ell m,\ell' m} (u_{mm} u^*_{mm} + u_{m-m} u^*_{m-m}) = J_{\ell m,\ell' m}, \end{aligned}$$

when $m = -m'$, we have

$$\begin{aligned} \sum_M J_{\ell M,\ell' M} u_{mM} u^*_{-mM} &= J_{\ell m,\ell' m} u_{mm} u^*_{-mm} + J_{\ell -m,\ell' -m} u_{m-m} u^*_{-m-m} \\ &= J_{\ell m,\ell' m} (u_{mm} u^*_{-mm} + u_{m-m} u^*_{-m-m}) = 0, \end{aligned}$$

and finally, in other cases, $u_{mM}u^*_{m'M} = 0$ for all M. So,

$$\mathbf{UJU}^\dagger = \mathbf{J}. \tag{A.12}$$

Since the identity matrix satisfies our requirements on $\mathbf{J}$, $\mathbf{U1U}^\dagger = \mathbf{1}$ which shows that $\mathbf{U}$ is unitary. □

Note that the restriction on $\mathbf{J}$ above is satisfied by the isotropic signal covariance $\mathbf{S}_{\text{iso}}$, transfer matrices of symmetric beams, and the dipole-modulation matrix with preferred direction along the z-axis, $\mathbf{M}_{\hat{z}}$.

Result 3. *Assume that* $\mathbf{K}$ *is a matrix in complex spherical harmonics which maps real fields to real fields; that is, if* $\mathbf{x}$ *is the expansion of a field in* $\mathbb{R}$, $\mathbf{Kx}$ *is also the expansion of a field in* $\mathbb{R}$. *Then* $(-1)^{m'}K_{\ell m,\ell' m'} = (-1)^m K^*_{\ell -m,\ell' -m'}$. *In particular,* $K_{\ell m,\ell m} = K^*_{\ell -m,\ell -m}$.

Proof. Let $\mathbf{y} = \mathbf{Kx}$. We know that $x_{\ell m} = (-1)^m x_{\ell -m}$, so

$$y_{\ell m} = \sum_{\ell' m'} K_{\ell m,\ell' m'} x_{\ell' m'} = \sum_{\ell' m'} K_{\ell m,\ell' m'} (-1)^{m'} x^*_{\ell' -m'}.$$

We also know that $y_{\ell m} = (-1)^m y_{\ell -m}$, so

$$y_{\ell m} = \sum_{\ell' m'} K_{\ell m,\ell' m'} x_{\ell' m'} = (-1)^m \sum_{\ell' m'} K^*_{\ell -m,\ell -m'} x^*_{\ell -m'}.$$

Since this must be valid for any $\mathbf{x}$ satisfying our assumption, the result follows. □

Result 4. *If* $\mathbf{K}$ *is an Hermitian, complex spherical harmonic matrix that maps real fields to real fields (such as a covariance matrix for real fields), then the diagonal elements of the corresponding real spherical harmonic matrix is given by*

$$(\mathbf{UKU}^\dagger)_{\ell m,\ell m} = \begin{cases} K_{\ell m,\ell m} & \text{when } m = 0 \\ K_{\ell m,\ell m} + (-1)^m \mathrm{Re}(K_{\ell -m,\ell m}) & \text{when } m > 0 \\ K_{\ell m,\ell m} - (-1)^m \mathrm{Re}(K_{\ell -m,\ell m}) & \text{when } m < 0. \end{cases}$$

Proof. We have

$$\begin{aligned} (\mathbf{UKU}^\dagger)_{\ell m,\ell m} &= \sum_{\ell' m'} \sum_{\ell'' m''} \delta_{\ell\ell'} u_{mm'} K_{\ell' m',\ell'' m''} \delta_{\ell\ell''} u^*_{m'' m} \\ &= \sum_{m'} \sum_{m''} K_{\ell m',\ell m''} u_{mm'} u^*_{mm''}. \end{aligned}$$

When $m = 0$, this is simply $K_{\ell m,\ell m}$. When $m \neq 0$, we have $u_{mm}u^*_{mm} = u_{m-m}u^*_{m-m} = 1/2$ and $u_{mm}u^*_{m-m} = (-1)^{[m<0]+m}/2$, and so

$$\begin{aligned} (\mathbf{UKU}^\dagger)_{\ell m,\ell m} =& K_{\ell m,\ell m} u_{mm} u^*_{mm} + K_{\ell -m,\ell -m} u_{m-m} u^*_{m-m} + \\ & K_{\ell m,\ell -m} u_{mm} u^*_{m-m} + K_{\ell -m,\ell m} u_{m-m} u^*_{mm} \\ =& K_{\ell m,\ell m}(u_{mm} u^*_{mm} + u_{m-m} u^*_{m-m}) \\ & K_{\ell -m,\ell m} u_{m-m} u^*_{mm} + (K_{\ell -m,\ell m} u_{m-m} u^*_{mm})^* \\ =& K_{\ell m,\ell m} + (-1)^{[m<0]+m} \mathrm{Re}(K_{\ell -m,\ell m}) \end{aligned}$$

by using Result 3 and the fact that $z + z^* = \mathrm{Re}(z)/2$. □

A.3 Wigner 3j symbols

The Wigner 3j symbols are used by us primarily to compute Gaunt integrals (see below). We start by list some basic properties, which can be found in Edmonds (1957) and/or Rasch & Yu (2003). The Wigner 3j symbol is denoted

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$$

For our purposes all coefficients are assumed to be integers. Explicit expressions are not suitable for numerical computation, instead recurrence relations are used. Still, an explicit expression is

$$\begin{aligned} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} =& \Delta(j_1, j_2, j_3)\delta_{m_1+m_2+m_3,0}(-1)^{j_1-j_2-m_3}\sqrt{(j_1+m_1)!(j_1-m_1)!} \\ & \times \sqrt{(j_2+m_2)!(j_2-m_2)!(j_3+m_3)!(j_3-m_3)!} \\ & \times \sum_{k=k_{\min}}^{k_{\max}} \frac{(-1)^k}{k!(j_1+j_2-j_3-k)!(j_1-m_1-k)!(j_2+m_2-k)!} \\ & \times \frac{1}{(j_3-j_2+m_1+k)!(j_3-j_1-m_2+k)!}. \end{aligned} \tag{A.13}$$

Here, $\Delta(j_1, j_2, j_3) = 0$ if the triangle inequality is not satisfied ($|j_a - j_b| \leq j_c \leq j_a + j_b$ for all a, b, c). Otherwise it is

$$\Delta(j_1, j_2, j_3) = \sqrt{\frac{(j_1+j_2-j_3)!(j_1-j_2+j_3)!(-j_1+j_2+j_3)!}{(j_1+j_2+j_3+1)!}}.$$

The sum over k runs over indices such that none of the arguments to the factorials are negative, which corresponds to

$$\begin{aligned} k_{\min} &= \max(-j_3+j_2-m_1, -j_3+j_1+m_2, 0) \\ k_{\max} &= \min(j_1+j_2-j_3, j_1-m_1, j_2+m_2) \end{aligned}$$

The 3j symbols vanish under a number of circumstances, including:

- Whenever the triangle inequality mentioned above is not satisfied
- Whenever $|m_i| > j_i$
- Whenever $m_1 + m_2 + m_3 \neq 0$

Changing the signs of all the ms gives a phase:

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = (-1)^{j_1+j_2+j_3} \begin{pmatrix} j_1 & j_2 & j_3 \\ -m_1 & -m_2 & -m_3 \end{pmatrix}. \tag{A.14}$$

This also means that when all ms are zero, the 3j symbol vanish whenever $j_1 + j_2 + j_3$ is odd, since

$$\begin{pmatrix} j & j' & j'' \\ 0 & 0 & 0 \end{pmatrix} = (-1)^{j+j'+j''} \begin{pmatrix} j & j' & j'' \\ 0 & 0 & 0 \end{pmatrix}. \tag{A.15}$$

Any odd permutation of columns gives the same factor:

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = (-1)^{j_1+j_2+j_3} \begin{pmatrix} j_1 & j_3 & j_2 \\ m_1 & m_3 & m_2 \end{pmatrix} = \dots \tag{A.16}$$

For computation, we use the Fortran routines `drc3jj` and `drc3jm` from the SLATEC library (http://netlib.org). It is based on a recurrence scheme (Schulten & Gordon, 1976). In addition, it was often convenient to get exact answers rather than floating point (for experimentation for small arguments), in which case the `wigner_3j` function in Sage (http://www.sagemath.org) was useful (Rasch & Yu, 2003).

A.4 The Gaunt integral

A very important integral in this thesis is the *Gaunt integral*;

$$Y^{\ell\ell'\ell''}_{mm'm''} \equiv \int Y_{\ell m} Y_{\ell' m'} Y_{\ell'' m''} d\Omega. \tag{A.17}$$

It turns out that this can be written in terms of Wigner 3j symbols (Edmonds, 1957);

$$Y^{\ell\ell'\ell''}_{mm'm''} = \sqrt{\frac{(2\ell+1)(2\ell'+1)(2\ell''+1)}{4\pi}} \begin{pmatrix} \ell & \ell' & \ell'' \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ell & \ell' & \ell'' \\ m & m' & m'' \end{pmatrix}. \tag{A.18}$$

This makes it easy to compute the integral fast and exact without any numerical quadrature. Also, it allows us to conveniently study all the symmetry and vanishing properties of the integral.

The properties we use can easily be derived from the properties of the 3j symbol: It vanishes when $\ell + \ell' + \ell''$ is odd, and it is invariant under any permutation of columns and when changing signs of the m-coefficients:

$$Y^{\ell\ell'\ell''}_{mm'm''} = Y^{\ell\ell'\ell''}_{-m-m'-m''}. \tag{A.19}$$

Two important Gaunt integrals for our purposes are $Y^{\ell,\ell,0}_{-m,m,0}$ and $Y^{\ell,\ell+1,1}_{-m,m,0}$; here we find explicit expressions for them. The first one is easy:

$$\begin{aligned} Y^{\ell,\ell,0}_{-m,m,0} &= \int Y_{\ell -m} Y_{\ell m} Y_{00} d\Omega && \text{(A.20)} \\ &= (-1)^m \int Y^*_{\ell m} Y_{\ell m} \frac{1}{\sqrt{4\pi}} d\Omega && \text{(A.21)} \\ &= (-1)^m \frac{1}{\sqrt{4\pi}}, && \text{(A.22)} \end{aligned}$$

by the orthogonality property of the spherical harmonics. This can also be seen from a property of the Wigner 3j symbol mentioned in Edmonds (1957):

$$\begin{pmatrix} j & j & 0 \\ m & -m & 0 \end{pmatrix} = \frac{(-1)^{j-m}}{\sqrt{2j+1}}. \tag{A.23}$$

We then turn to $Y^{\ell,\ell+1,1}_{-m,m,0}$, which can be found by computing

$$\begin{pmatrix} \ell & \ell+1 & 1 \\ -m & m & 0 \end{pmatrix}.$$

We use equation (A.13). First,

$$\begin{aligned} k_{\min} =& \max(-1+\ell+1-(-m), -1+\ell+m) = \ell+m \\ k_{\max} =& \min(\ell+\ell+1-1, \ell-(-m), \ell+1+m) = \ell+m \end{aligned}$$

so equation (A.13) only sums over one term with $k = \ell + m$, resulting in

$$\begin{aligned} \begin{pmatrix} \ell & \ell+1 & 1 \\ -m & m & 0 \end{pmatrix} =& \Delta(\ell,\ell+1,0)(-1)^{\ell+m+1} \\ & \times \sqrt{(\ell-m)!(\ell+m)!(\ell+1+m)!(\ell+1-m)!} \\ & \times \frac{1}{(\ell+m)!(\ell+\ell+1-1-(\ell+m))!} \\ & \times \frac{1}{(\ell+m-(\ell+m))!(\ell+1+m-(\ell+m))!} \\ & \times \frac{1}{(1-\ell-1-m+(\ell+m))!(1-\ell-m+(\ell+m))!} \\ =& \Delta(\ell,\ell+1,0)(-1)^{\ell+m+1} \\ & \times \frac{\sqrt{(\ell-m)!(\ell+m)!(\ell+1+m)!(\ell+1-m)!}}{(\ell+m)!(\ell-m)!} \\ =& \Delta(\ell,\ell+1,0)(-1)^{\ell+m+1}\sqrt{\frac{(\ell+1+m)!(\ell+1-m)!}{(\ell+m)!(\ell-m)!}} \\ =& \Delta(\ell,\ell+1,0)(-1)^{\ell+m+1}\sqrt{(\ell+m+1)(\ell-m+1)}, \end{aligned}$$

$$\begin{aligned} \begin{pmatrix} \ell & \ell+1 & 1 \\ -m & m & 0 \end{pmatrix} =& (-1)^{\ell+m+1}\sqrt{\frac{(2\ell)!(0)!(2)!}{(2\ell+3)!}}\sqrt{(\ell+m+1)(\ell-m+1)} \\ =& (-1)^{\ell+m+1}\sqrt{\frac{2(\ell+m+1)(\ell-m+1)}{(2\ell+3)(2\ell+2)(2\ell+1)}} \\ =& (-1)^{\ell+m+1}\sqrt{\frac{(\ell+m+1)(\ell-m+1)}{(\ell+1)(2\ell+3)(2\ell+1)}}. \end{aligned}$$

So,

$$\begin{aligned} Y^{\ell,\ell+1,1}_{-m,m,0} =& \sqrt{\frac{3(2\ell+1)(2\ell+3)}{4\pi}} \begin{pmatrix} \ell & \ell+1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ell & \ell+1 & 1 \\ -m & m & 0 \end{pmatrix} \\ =& (-1)^{\ell+m+1}(-1)^{\ell+1}\sqrt{\frac{3(2\ell+1)(2\ell+3)}{4\pi}} \\ & \times \sqrt{\frac{(\ell+1)(\ell+1)}{(\ell+1)(2\ell+3)(2\ell+1)}}\sqrt{\frac{(\ell+m+1)(\ell-m+1)}{(\ell+1)(2\ell+3)(2\ell+1)}} \\ =& (-1)^m\sqrt{\frac{3(\ell+m+1)(\ell-m+1)}{4\pi(2\ell+1)(2\ell+3)}} \end{aligned} \tag{A.24}$$

Using the symmetry properties (permutation of columns and changing the signs of the ms), this is useful for all $Y^{\ell,\ell',1}_{-m,m,0}$ where $|\ell-\ell'|=1$; one simply inserts $\min(\ell,\ell')$ in the above expression.

A.5 The Wigner D-matrix

A rotation of a field f on the sphere can be described using three coordinates: First, rotate an angle ψ around the z-axis, then an angle θ around the y-axis, and finally an angle ϕ around the z-axis. In real space, a rotation of a vector can be carried out by using an Euler matrix $\mathbf{E}(\psi,\phi,\theta)$, and the rotated field is simply $g(\hat{n}) = f(\mathbf{E}^{-1}\hat{n})$. However, when we are dealing with a pixelized map, doing the rotation in real space would be inconvenient and error-prone. Fortunately, there is a method to transform the spherical harmonic coefficients of $f(\hat{n})$ directly into the spherical harmonic coefficients of $f(\mathbf{E}^{-1}\hat{n})$.

Suppose we want to find $a'_{\ell m}$ such that

$$\sum_{\ell m} a'_{\ell m} Y_{\ell m}(\hat{n}) = \sum_{\ell m} a_{\ell m} Y_{\ell m}(\mathbf{E}(\psi,\theta,\phi)^{-1}\hat{n}).$$

Then we can simply use the *Wigner* $\mathbf{D}$*-matrix* (e.g. Edmonds (1957) and Risbo (1996)):

$$\mathbf{a}' = \mathbf{D}(\psi,\phi,\theta)\mathbf{a}.$$

The matrix $\mathbf{D}$ is unitary. It is also block-diagonal in ℓ, that is, each scale is rotated seperately. An ℓ-block can be computed recursively from the $\ell-1$-block, so the time required for computing and multiplying $\mathbf{D}$ with a vector scales as $O({\ell_{\max}}^3)$ while the memory requirements scales as $O({\ell_{\max}}^2)$ (if one does the computation of $\mathbf{D}$ and the matrix-vector multiplication jointly). HEALPix contains a routine `rotate_alm` which does exactly what we need, based upon an algorithm by Risbo (1996).

A.6 Sparse linear algebra

We rely on the ability to multiply dense vectors with sparse matrices, which is done with the help of the `scipy.sparse` Python module. Matrices are constructed in the COOrdinate format, which is simply three arrays for row index, column index, and element value, respectively, and then converted to the more efficient Compressed Sparse Column (CSC) or Compressed Sparse Row (CSR) formats.

Finding Cholesky or LU factors is a trickier business. In the end, this turned out to not be necesarry because of the introduction of a rotation $\mathbf{R}$. Still, the ability to find sparse factors is completely crucial in the experimentation phase, and is likely to become important for other sparse-$\mathbf{S}$ models.

The choice of software is made easier by the fact that we need the determinant to evaluate the likelihood. Sparse linear algebra packages appear to mainly be written with equation solving in mind, and often do not allow for finding the determinant (e.g., the Intel Math Kernel Library[1] and the open source SuperLU[2]). Since sophisticated distributed storage schemes are used for the factors, introducing such functionality appears to not be completely trivial.

An honorable exception is the GPL-licensed SparseSuite[3] by Tim Davis et al., consisting of, e.g, UMFPACK for LU decompositions and CHOLMOD for Cholesky decompositions. We have relied heavily on SparseSuite in our work. Nathaniel Smith recently wrote a nice Python interface to CHOLMOD[4].

A disadvantage of CHOLMOD is that it is not parallelized, although if one really needs to, a more generic LU decomposition can be used instead. This would tend to double the total computational cost, but could improve wall time. Both UMFPACK and SuperLU are parellelized (in-process). For scaling up to bigger problems, one has, e.g., the open source MUMPS[5], a package for sparse linear algebra on a cluster using MPI. Unfortunately, on a quick reading we did not find any routines to extract the determinant. We can not see that introducing such routines should be fundamentally impossible, although perhaps a daunting implementation challenge.

[1] http://software.intel.com/en-us/intel-mkl/
[2] http://crd.lbl.gov/ xiaoye/SuperLU/
[3] http://www.cise.ufl.edu/research/sparse/SuiteSparse
[4] http://code.google.com/p/scikits-sparse
[5] http://graal.ens-lyon.fr/MUMPS

References

Bennett, C. L. et al.: 2010, "Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Are There Cosmic Microwave Background Anomalies?", *Astrophys. J.(submitted)*, arXiv:1001.4758

Bennett, C. L. et al.: 2003, "First-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Foreground Emission", *Astrophys. J. Suppl. S.* **148**, 97–117, arXiv:astro-ph/0302208

Callin, P.: 2006, "How to calculate the CMB spectrum", *ArXiv Astrophysics e-prints*, arXiv:astro-ph/0606683

Chib, S., & Greenberg, E.: 1995, "Understanding the Metropolis-Hastings Algorithm", *The American Statistician* **49(4)**, pp. 327–335

Dodelson, S.: 2003, *Modern Cosmology*, Academic Press/Elsevier

Edmonds, A. R.: 1957, *Angular Momentum in Quantum Mechanics*, Princeton University Press

Eriksen, H. K., Banday, A. J., Górski, K. M., Hansen, F. K., & Lilje, P. B.: 2007, "Hemispherical Power Asymmetry in the Third-Year Wilkinson Microwave Anisotropy Probe Sky Maps", *Astrophys. J. Lett.* **660**, L81–L84, arXiv:astro-ph/0701089

Eriksen, H. K., Banday, A. J., Górski, K. M., & Lilje, P. B.: 2005, "The N-Point Correlation Functions of the First-Year Wilkinson Microwave Anisotropy Probe Sky Maps", *Astrophys. J.* **622**, 58–71, arXiv:astro-ph/0407271

Eriksen, H. K., Hansen, F. K., Banday, A. J., Górski, K. M., & Lilje, P. B.: 2004a, "Asymmetries in the Cosmic Microwave Background Anisotropy Field", *Astrophys. J.* **605**, 14–20, arXiv:astro-ph/0307507

Eriksen, H. K. et al.: 2008, "Joint Bayesian Component Separation and CMB Power Spectrum Estimation", *Astrophys. J.* **676**, 10–32, arXiv:0709.1058

Eriksen, H. K. et al.: 2004b, "Power Spectrum Estimation from High-Resolution Maps by Gibbs Sampling", *Astrophys. J. Suppl. S.* **155**, 227–241, arXiv:astro-ph/0407028

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B.: 2004, *Bayesian Data Analysis (2nd ed.)*, Chapman & Hall/CRC

Gold, B. et al.: 2010, "Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Galactic Foreground Emission",, arXiv:1001.4555

Gordon, C.: 2007, "Broken Isotropy from a Linear Modulation of the Primordial Perturbations", *Astrophys. J.* **656**, 636–640, arXiv:astro-ph/0607423

Górski, K. M. et al.: 2005, "HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere", *Astrophys. J.* **622**, 759–771, arXiv:astro-ph/0409513

Groeneboom, N. E., & Eriksen, H. K.: 2009, "Bayesian Analysis of Sparse Anisotropic Universe Models and Application to the Five-Year WMAP Data", *Astrophys. J.* **690**, 1807–1819, arXiv:0807.2242

Hansen, F. K., Banday, A. J., & Górski, K. M.: 2004, "Testing the cosmological principle of isotropy: local power-spectrum estimates of the WMAP data", *Mon. Not. R. Astron. Soc.* **354**, 641–665, arXiv:astro-ph/0404206

Hansen, F. K., Banday, A. J., Górski, K. M., Eriksen, H. K., & Lilje, P. B.: 2009, "Power Asymmetry in Cosmic Microwave Background Fluctuations from Full Sky to Sub-Degree Scales: Is the Universe Isotropic?", *Astrophys. J.* **704**, 1448–1458, arXiv:0812.3795

Hanson, D., & Lewis, A.: 2009, "Estimators for CMB statistical anisotropy", *Phys. Rev. D* **80(6)**, 063004–+, arXiv:0908.0963

Harville, D. A.: 1997, *Matrix Algebra From a Statistician's Perspective*, Springer

Hastings, W. K.: 1970, "Monte Carlo sampling methods using Markov chains and their applications", *Biometrika* **57(1)**, 97–109

Hoftuft, J. et al.: 2009, "Increasing Evidence for Hemispherical Power Asymmetry in the Five-Year WMAP Data", *Astrophys. J.* **699**, 985–989, arXiv:0903.1229

Jarosik, N. et al.: 2010, "Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Sky Maps, Systematic Errors, and Basic Results", *Astrophys. J.(submitted)*, arXiv:1001.4744

Jewell, J., Levin, S., & Anderson, C. H.: 2004, "Application of Monte Carlo Algorithms to the Bayesian Analysis of the Cosmic Microwave Background", *Astrophys. J.* **609**, 1–14, arXiv:astro-ph/0209560

Larson, D. et al.: 2010, "Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Power Spectra and WMAP-Derived Parameters", *Astrophys. J.(submitted)*, arXiv:1001.4635

Lewis, A., & Bridle, S.: 2002, "Cosmological parameters from CMB and other data: a Monte- Carlo approach", *Phys. Rev.* **D66**, 103511, arXiv:astro-ph/0205436

Lewis, A., Challinor, A., & Lasenby, A.: 2000, "Efficient Computation of CMB anisotropies in closed FRW models", *Astrophys. J.* **538**, 473–476, arXiv:astro-ph/9911177

Page, L. et al.: 2003, "First-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Beam Profiles and Window Functions", *Astrophys. J. Suppl. S.* **148**, 39–50, arXiv:astro-ph/0302214

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P.: 2007, *Numerical Recipes: The Art of Scientific Computing (3rd ed.)*, Cambridge University Press

Rasch, J., & Yu, A. C. H.: 2003, "Efficient Storage Scheme for Precalculated Wigner 3j, 6j and Gaunt Coefficients", *SIAM Journal on Scientific Computing* **25(4)**, 1416–1428

Risbo, T.: 1996, "Fourier transform summation of Legendre series and D-functions", *Journal of Geodesy* **70**, 383–396

Robert, C. P., & Casella, G.: 2004, *Monte Carlo Statistical Methods*, Springer

Rudjord, Ø. et al.: 2009, "Cosmic Microwave Background Likelihood Approximation by a Gaussianized Blackwell-Rao Estimator", *Astrophys. J.* **692**, 1669–1677, arXiv:0809.4624

Schulten, K., & Gordon, R. G.: 1976, "Recursive evaluation of 3j and 6j coefficients", *Computer Physics Communications* **11(2)**, 269 – 278

Seljak, U., & Zaldarriaga, M.: 1996, "A Line of Sight Approach to Cosmic Microwave Background Anisotropies", *Astrophys. J.* **469**, 437–444, arXiv:astro-ph/9603033

Seljebotn, D. S.: 2009, "Fast numerical computations with Cython", *Proceedings of the 8th Annual Python in Science Conference*

Shewchuk, J. R.: 1994, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*, `http://www.cs.cmu.edu/~jrs/jrspapers.html`

Smith, K. M., Zahn, O., & Doré, O.: 2007, "Detection of gravitational lensing in the cosmic microwave background", *Phys. Rev. D* **76(4)**, 043510–+, arXiv:0705.3980

The Planck Collaboration: 2006, "The Scientific Programme of Planck", *ArXiv Astrophysics e-prints*, arXiv:astro-ph/0604069

Usmani, R.: 1994, "Inversion of Jacobi's tridiagonal matrix", *Computers & Mathematics with Applications* **27(8)**, 59 – 66

Wandelt, B. D., Larson, D. L., & Lakshminarayanan, A.: 2004, "Global, exact cosmic microwave background data analysis using Gibbs sampling", *Phys. Rev. D* **70(8)**, 083511

Wright, E. L. et al.: 2009, "Five-Year Wilkinson Microwave Anisotropy Probe Observations: Source Catalog", *Astrophys. J. Suppl. S.* **180**, 283–295, arXiv:0803.0577