A SELF-CONTAINED GUIDE TO THE CMB GIBBS SAMPLER

NICOLAAS E. GROENEDOONK
(Dated: May 23, 2009)

ABSTRACT

We present a consistent self-contained and pedagogical review of the CMB Gibbs sampler, focusing on computational methods and code design. We provide an easy-to-use CMB Gibbs sampler named SLAVE developed in C++ using object-oriented design. While discussing why the need for a Gibbs sampler is evident and what the Gibbs sampler can be used for in a cosmological context, we review in detail the analytical expressions for the conditional probability densities and discuss the problems of galactic foreground removal and anisotropic noise. Having demonstrated that SLAVE is a working, usable CMB Gibbs sampler, we present the algorithm for white noise level estimation. We then give a short guide on operating SLAVE before introducing the post-processing utilities for obtaining the best-fit power spectrum using the Blackwell-Rao estimator.

Subject headings: cosmic microwave background — cosmology: observations — methods: numerical

1. INTRODUCTION

In recent years, increased resolution in the measurement of the cosmic microwave background (CMB) have driven the need for more accurate data analysis techniques. During the early years of CMB experiments, data was so sparse and noise levels so high that error bars in general overshadowed the observed signal. With the COBE experiment, \cite{Smoot1992} posteriors were mapped out by brute force, and the statistical methods employed were simplistic. This was sufficient, as advanced statistical methods weren’t needed for analyzing crude data. However, all this changed with the Wilkinson Microwave Anisotropy Probe (WMAP) experiment \cite{Bennett2003, Hinshaw2007}. Suddenly, cosmological data became much more detailed, vastly improving our knowledge of the universe, but also introduced new problems. Which parts of the signal were pure CMB, and which were not? The need for knowledge about instrumental noise, point sources, dust emission, synchrotron radiation and other contaminations were required in order to estimate the pure CMB signal from the data. And, how does one properly deal with the sky cut, the contamination from our galaxy? Even harder, how does one maximize the probability that the resulting signal really is the correct CMB signal? A new era of cosmological statistics emerged.

An important event was the introduction of Bayesian statistics in cosmological data analysis. Bayesian statistics differs from the frequentist thought by quantizing ignorance: what one knows and not knows are intrinsic parts of the analysis. The goal of any Bayesian analysis is to go from the prior $P(\theta)$, or what is known about the model, to the posterior $P(\theta|\text{data})$, the probability of a model given data. This is summarized via Bayes’ famous theorem:

$$P(\theta|\text{data}) = \frac{P(\text{data}|\theta)P(\theta)}{P(\text{data})}. \quad (1)$$

The posterior $P(\theta|\text{data})$ tells us something about how well a model $\theta$ fits the data, and is obtained by multiplying the prior $P(\theta)$, our assumption of the model, with the likelihood $P(\text{data}|\theta)$, the probability that the data fits the model.

The need for Bayesian statistics becomes evident when considering that we only have data from one single experiment to analyze. Bayesian statistics merges with frequentist statistics for large number of samples. And, in a cosmological context, we are stuck with only one sample, a sample that we are constantly measuring to higher accuracies. This sample is one realization of the underlying universe model, and we are unable to obtain data from another sample.

In a standard Metropolis-Hastings (MH) Monte Carlo Markov chain-approach (MCMC), one samples from the joint distribution by letting chains of “random walkers” transverse the parameter space. The posterior is obtained by calculating the normalized histogram of all the samples in the chains. The posterior will eventually resemble the underlying joint distribution, or the likelihood surface. This is a simple but not without drawbacks. For one, each MH step is required to test the likelihood value of the chain at the current position in parameter space up against a new proposed position. Many of these steps will be rejected, and this is where the computational costs usually reside. The Gibbs sampler provides something new: one never needs to reject samples, and every move becomes accepted and usable for building the posterior. This is done by assuming that we have prior knowledge of the conditional distributions. These are then sampled from, each in turn yielding accepted steps.

However, the main motivation for introducing the CMB Gibbs sampler is the drastically improvement in scaling. With conventional MCMC methods, one needs to sample from the joint distribution, which results in an $O(n^3)$ operation. For a white noise case, the Gibbs sampler splits the sampling process into independently sampling from the two conditional distributions, which together yields a $O(n^{3.5})$ operation. In other words, the Gibbs sampler enables sampling the high-$t$ regime much more effective than previous MCMC methods.

Electronic address: leuat@irio.co.uk

\1 Institute of Theoretical Astrophysics, University of Oslo, P.O. Box 1029 Blindern, N-0315 Oslo, Norway
The problem of estimating the cosmological signal $s$ from the full signal by Gibbs sampling was first addressed in [Jewell et al. 2004, Wandelt et al. 2004] and [Eriksen et al. 2004]. The ultimate goal of the Gibbs sampler is to estimate the CMB signal $s$ from the data $d$, eliminating noise $n$, convolution $A$, all while including the sky cut. Today, a great number of papers have employed the Gibbs sampler since the introduction of the method [Eriksen et al. 2008a, Dunkley et al. 2008, Cumberbatch et al. 2008, Groeneboom et al. 2008, Groeneboom et al. 2009a, Eriksen et al. 2006, Rudjord et al. 2009, Jewell et al. 2009, Dickinson et al. 2009, Chiu 2005, Dickinson et al. 2004, Larson et al. 2007].

In this paper, we review the basics of the CMB Gibbs sampler, and provide a simple, intuitive non-parallelized CMB Gibbs software bundle named SLAVE. SLAVE is written in C++, and employs object-oriented design in order to simplify mathematical implementation. The OOP design of SLAVE is presented in figure 1. For instance, assuming $A$, $B$ and $C$ are instances of the “real alm” class (they contain a set of real $a_{m\ell}$), operator overloading enables us to directly translate the expression $A = (B+C)^{-1}$ by writing

$$A = (B+C).\text{Invert}();$$

This yields fast code that closely resembles equations, without having optimized too much for parallel computing, multiple data sets and other complexities.

1.1. The Master algorithm

One method of likelihood-estimator for obtaining the best-fit power spectrum for masked CMB data is given by the MASTER algorithm [Hivon et al. 2002]. While Gibbs sampling estimates the full CMB signal $s$, the MASTER method only estimates the power spectrum. This method does not allow for variations in the estimated signal, except for the natural variations from simulating different realizations from the same power spectrum. However, the master algorithm estimates the power spectrum with cost scaling as $O(n^3)$, which is slow for high-$\ell$ operations.

1.2. What do I need the CMB Gibbs sampler for?

Often, people misunderstand the concepts behind the CMB Gibbs sampler, and what the Gibbs sampler can be used for. In this section, we try to explain in simple terms when you should consider employing the CMB Gibbs sampler.

Assume that you have a theoretical universe model $M(\theta)$, where $\theta = \{\theta_i\}$ is a set of cosmological parameters. This model might give rise to some additional gaussian effects in the CMB map, either as fluctuations, altered power, anisotropic contributions, dipoles, ring structures or whatever. You now wish to test whether existing CMB data contains traces of your fabulous new model, and how significant those traces are. Or maybe you are just interested in ruling out the possibility that this model could be observed at all.

In any case, you need to implement some sort of numerical library that generates CMB maps based on your model. These maps will be “pure”, in the sense that you have complete control over its generation process and systematics. Assume that your model has 1 free parameter. You could now loop over the 1-dimensional parameter space and calculate the $\chi^2$ between a pure CMB signal map and the map from your model. This would have to be done for each step in parameter space, before obtaining the minimum. Even better, you could implement a Monte Carlo Markov chain framework, letting random walkers traverse a likelihood surface, yielding posteriors. This would enable support for a larger number of parameters, and is superior to the slow brute force approach.

In real-life however, things are not this simple. Data from any CMB experiment is contaminated by noise and foregrounds, most notably our own galaxy. This means that estimating the signal $s$ from the data is not trivial - one needs to “rebuild”, or make an assumption of what the fluctuations are within the sky cut and noise limits. This implies that it really isn’t possible to obtain “the correct” CMB map, all we can know is that there exist a statistical range of validity where a simulated map agrees with the true CMB signal. Therefore, the consideration that the estimated CMB signal $s$ is a statistical random variable and not a fixed map should be included in the analysis. Hence, if you have implemented the MASTER method mentioned in section 1.1, you should test your model map against a set of realizations from the MASTER-estimated signal power spectrum.

This is where the Gibbs sampler enters the stage. As previously mentioned, the Gibbs sampler will estimate the CMB signal given data, and not only the power spectrum. The Gibbs sampler also ensures that every step in parameter space is always valid, so one never needs to discard samples. And even better, each of these independent steps provide an operation cost for obtaining samples that are much lower than more conventional MCMC methods. In order to test whether your model $m$ fits the data, you therefore include the uncertainty in data by varying the signal. For example:
repeat until convergence

\[ \text{do} \]
\[ s = \text{the CMB signal given the} \]
\[ \text{power spectrum } C_l \]
\[ m = \text{the CMB signal of your model given} \]
\[ \text{the estimated CMB signal } s \]
\[ C_l = \text{the CMB power spectrum given } m \]
\[ \text{save } s, m \text{ and } C_l \]
\[ \text{repeat until convergence} \]

In the end, you calculate the statistical properties of \( s, m \) and \( C_l \). Your model parameters have now been estimated, and the process included the intrinsic uncertainties in the signal. This method is not the most rapid - but it will always yield correct results.

2. THE CMB GIBBS SAMPLER

Throughout this paper, we assume that the data can be expressed as
\[ d = As + n \]
where \( s \) is the CMB signal, \( A \) the instrument beam and \( n \) uncorrelated noise.

The MASTER algorithm estimates the power spectrum \( \langle C_l \rangle \) and the standard deviation \( \Delta C_l \). However, this method is an approximation to a full likelihood that can be expressed as follows:
\[ P(C_l|d) \propto e^{-\frac{1}{2}d^T(S+N)^{-1}d} \]
where \( S \) and \( N \) are the signal and noise covariance matrices, respectively. While it is fully possible to use Markov chain Monte Carlo methods to sample from this distribution, the calculation of the \( (S+N)^{-1} \)-matrix scales as \( n^4 \), where \( n \) is the size of the \( n \times n \) matrix. This is therefore an extremely slow operation, and is not feasible for large \( d \)-s. If we demand that we sample the sky signal \( s \) as well, the joint distribution becomes \( P(C_l, s|d) \). This might seem unnecessary complicated, as one most of the time doesn’t need the signal \( s \). But when feeding this distribution through the Gibbs sampler - that is, calculating the conditional distributions \( P(C_l|s,d) \) and \( P(s|C_l,d) \), we find that sampling from both are computationally faster than sampling from the full distribution in equation 3. The derivations of the conditional distributions are presented in section 8.

2.1. Review of the Metropolis-Hastings algorithm

The Gibbs sampler is a special case of the Metropolis-Hastings algorithm. We therefore review the basics of Monte Carlo Markov (MCMC) chain methods. The Metropolis-Hastings algorithm is an MCMC method for sampling directly from a probability distribution. This is done by letting “random walkers” transverse a parameter space, guided by the likelihood function, the probability that the data fits the model for the given parameter configuration. If a proposal step yields a likelihood greater than the current likelihood, then random walker accepts the step immediately. If the likelihood is less, then the walker will with a certain probability step “down” the likelihood surface. Eventually, the histogram of all the random walkers will converge to the posterior, the full underlying distribution.

Assume you have a model with \( n \) parameters, \( \theta = \{ \theta_k \} \) and you wish to map out a joint distribution from \( P(\theta) \).

Usually, one calculates the ratio \( R \) between the posteriors at the two steps \( P(\theta_1) \) and \( P(\theta_2) \), such that
\[ R = \frac{P(\theta_1)}{P(\theta_2)} \frac{T(\theta_2|\theta_1)}{T(\theta_1|\theta_2)} \]
where \( T(\theta_2|\theta_1) \) is the proposal distribution for going left or right. If the proposal distribution is symmetric (i.e. the probability of going left-right is equal for all \( \theta_k \)), then \( T(\theta_2|\theta_1) = T(\theta_1|\theta_2) \) such that:
\[ R = \frac{P(\theta_1)}{P(\theta_2)} \]

The MH acceptance rule now states: if \( R \) is larger than 1, accepted the step unconditionally. If \( R > 1 \), then accept the step if a random uniform variable \( x = U(0,1) < R \).

2.2. Review of the Gibbs algorithm

Assume you have a model with two parameters, \( \theta_1 \) and \( \theta_2 \), and you wish to map out a joint distribution from \( P(\theta_1, \theta_2) \). Now, also assume that you have prior knowledge of the conditional distributions, \( P(\theta_1|\theta_2) \) and \( P(\theta_2|\theta_1) \). A general proposal density is not necessary symmetric, and one must therefore consider the asymmetric proposal term as described in equation 4. However, we now define the proposal density \( T(\theta_2|\theta_1) \) to be the conditional distributions:
\[ T(\theta_2|\theta_1) = \delta(\theta_2 - \theta_1)P(\theta_2|\theta_1) \]

In words, the proposal is only considered when \( \theta_2 = \theta_1 \), which means that \( \theta_1 \) is fixed while \( \theta_2 \) can vary. If so, the acceptance is then given as the conditional distribution \( P(\theta_2|\theta_1) \), which we must have prior knowledge of. The reason for choosing such a proposal density becomes clear when investigating the Metropolis Hastings acceptance rate:
\[ R = \frac{P(\theta_2, \theta_1)\delta(\theta_2 - \theta_1)P(\theta_2|\theta_1)}{P(\theta_2, \theta_1)\delta(\theta_2 - \theta_1)P(\theta_2|\theta_1)} \]

Using the conditional sampling proposal 4 one obtains
\[ R = \frac{P(\theta_2|\theta_1)P(\theta_1|\theta_2)}{P(\theta_2|\theta_1)P(\theta_1|\theta_2)} \]

We now enforce the delta-function such that \( \theta_2 = \theta_1 \). This sampling from the conditional distributions is the crucial step in the Gibbs sampler, such that all terms cancel out:
\[ R = 1 \]

This implies that all steps are valid, and none are ever rejected. Hence one alternates between sampling from the known conditional distributions, where each step is independently accepted and can be performed as many times as needed.

3. THE CONDITIONAL DISTRIBUTIONS

In section 2.2, it was explained how the Gibbs sampler requires previous knowledge about the underlying conditional distributions. The CMB Gibbs sampler will alternate between sampling power spectra \( C_l \) and CMB signal \( s \), where each proposed step will always be valid. In order to enable sampling from the joint distribution,
we therefore need to derive the analytical properties of the conditional distributions:

\[ P(C|s, d) \quad \text{and} \quad P(s|C, d). \tag{10} \]

The derivations described here were first presented in \textit{Jewell et al.} (2004), \textit{Wandelt et al.} (2004) and \textit{Eriksen et al.} (2004b). The full, joint distribution is expressed as

\[ P(C, s|d) \propto P(d|C, s)P(C, s) \tag{11} \]

\[ = P(d|C, s)P(s|C)P(C) \tag{12} \]

where \( P(C) \) is a prior on \( C \), typically chosen to be flat. The first term, \(-2\ln P(d|C, s)\), is nothing but the \( \chi^2 \). The \( \chi^2 \) measures the goodness-of-fit between model and data, leaving only fluctuations in noise. As \( n = d - s \) is distributed accordingly to a Gaussian with mean 0 and variance \( N \), we find that

\[ P(d|C, s, d) \propto e^{-\frac{1}{2}(d-s)^T N^{-1}(d-s)}. \tag{13} \]

As we now assume that the signal \( s \) is known and fixed, the data \( d \) becomes redundant and \( P(C|s, d) = P(C|s) \propto P(s|C) \). We therefore first need to obtain an expression for \( P(C|s, d) \).

### 3.1. Deriving \( P(C|s, d) \)

Assuming that the CMB map consists of Gaussian fluctuations, we can express the conditional probability density for a power spectrum \( C \) given a sky signal \( s \) as follows:

\[ P(C_s|s, d) = \frac{e^{-\frac{1}{2}s^TC^{-1}s}}{\sqrt{C}}. \tag{14} \]

where \( C = C(C_s) \) is the covariance matrix. We now perform a transformation to spherical harmonics space, where \( s = \sum_{l,m} a_{\ell m} Y_{\ell m} \) and \( C_{l,l'} = \sum_{l,l'} \sum_{m,m'} Y_{\ell m}^* Y_{\ell' m'} C_{\ell' m'} Y_{\ell' m'}^* Y_{\ell m} \). Then equation (14) transforms to

\[ s^TC^{-1}s = \sum_{l,m} a_{\ell m}^* C_{\ell}^{-1} a_{\ell m} \tag{15} \]

As the spherical harmonics are orthogonal, they all cancel out and leave delta functions for \( \delta_{l' l} \delta_{mm'} \) such that

\[ s^TC^{-1}s = \sum_{\ell m} a_{\ell m}^* C_{\ell}^{-1} a_{\ell m} = \sum_{\ell m} a_{\ell m}^* \frac{1}{C_{\ell}} a_{\ell m}. \tag{16} \]

We now define a power spectrum \( \sigma_\ell = \frac{1}{2\ell+1} \sum_m |a_{\ell m}|^2 \) such that

\[ s^TC^{-1}s = \sum_{\ell} (2\ell + 1) \sigma_\ell. \tag{17} \]

Similarly, the determinant is given as the product of the diagonal matrix \( C \), which for each \( l \) has \( 2\ell + 1 \) values of \( C_{ll} \). The determinant is thus \( |C| = \prod_l C_{ll}^{2\ell+1} \). Expression (14) can now be written as

\[ P(C|s) = \prod_{\ell} e^{-\frac{2\ell+1}{2} \frac{\sigma_{\ell}}{C_{\ell}}} \tag{18} \]

which by definition means that the \( C_\ell \)'s are distributed as an inverse Gamma function. In the computational section, we will discuss how to draw random variables from this distribution.

### 3.2. Deriving \( P(s|C_\ell, d) \)

Again, we begin with the full, joint distribution:

\[ P(C_s, s|d) \propto P(d|C_s, s)P(C_s|s). \tag{19} \]

We now know from equation (18) and (13) that the joint distribution can be expressed as

\[ P(C_s, s|d) \propto e^{-\frac{1}{2}(d-s)^T N^{-1}(d-s)} \prod_{\ell} e^{-\frac{2\ell+1}{2} \frac{\sigma_{\ell}}{C_{\ell}}} \tag{20} \]

omitting the prior \( P(C_s) \). Again, note that it would be nearly impossible to sample directly from the full distribution. We now investigate what happens with equation (20) when \( C_s \) becomes a fixed quantity. As the \( C_{\ell} \)'s in the denominator vanishes, we use equation (14) to obtain

\[ P(s|C_s, d) \propto e^{-\frac{1}{2}(s-s')^T N^{-1}(s-s')} e^{-\frac{1}{2}s^TC^{-1}s}. \tag{21} \]

We now introduce a residual variable \( r = d - s \), such that \( r \) roughly consist of noise. As noise was uncorrelated, we can expect that \( r \) follows a Gaussian distribution with zero mean and \( N \) variance. Also, if \( s \) is known, then \( C_s \) is redundant. We complete the square, and introduce \( \hat{s} = (S^{-1} + N^{-1})^{-1} N^{-1} d \). Equation (21) can now be rewritten as

\[ P(s|C_s, d) \propto e^{-\frac{1}{2}(s-\hat{s})^T (C_s^{-1} + N^{-1}) (s-\hat{s})}. \tag{22} \]

Hence \( P(s|C_s, d) \) is a Gaussian distribution with mean \( \hat{s} \) and covariance \((C_s^{-1} + N^{-1})^{-1}\). In the computational section, we will discuss how to draw random variables from this distribution.

### 4. NUMERICAL IMPLEMENTATION

In its utter simplicity, the mechanics of the Gibbs sampler can be summarized as follows:

- load data
- initialize \( s \) and \( cl \)
- loop number of chains
  - \( s = \text{generate from } p(s | cl, d) \)
  - \( cl = \text{generate from } p(cl | s, d) \)
  - save \( s \) and \( cl \)
- end loop

We now present the computational methods for drawing from \( P(s|C_s, d) \) and \( P(C_s|s, d) \).
4.1. $P(C_\ell | s, d)$

We show that equation 18 is an inverse Gamma distribution. A general gamma-distribution is proportional to

$$P_T(x; k, \theta) \propto x^{k-1} e^{-\frac{x}{\theta}}.$$  \hspace{1cm} (23)

Equation 18 can be expressed as

$$P(C_\ell | s) = C_\ell^{2s \ell - 1} e^{-\beta/C_\ell}$$  \hspace{1cm} (24)

where $\beta = \frac{2s \ell + 1}{2}$. If we now perform a substitution $y = 1/C_\ell$, we see that

$$P(y|s) = y^{2s \ell - 1} e^{-\beta y} \cdot y^{-2}$$  \hspace{1cm} (25)

which is a gamma-distribution for $k = \frac{2s \ell - 1}{2}$. We now show that this particular distribution also happens to be a special case of the $\chi^2$ distribution:

$$\chi^2(x; k) = \frac{x^2}{2} e^{-\beta x}.$$  \hspace{1cm} (27)

Letting $z = 2\beta y$ and ignoring the constants, we find that

$$P(z|s) = z^{k-1} e^{-z^2/2}$$  \hspace{1cm} (28)

such that if $k' = 2k = 2\ell - 1$, $z$ is distributed according to a $\chi^2$ distribution with $2\ell - 1$ degrees of freedom. A random variable following such a distribution can be drawn as follows:

$$z_\chi = \sum_{i=0}^{2\ell-1} |N_i(0, 1)|^2$$  \hspace{1cm} (29)

where $N_i(0, 1)$ are random Gaussian variables with mean 0 and variance 1. Since $z = 2\beta y = 2\beta/C_\ell$, we find that

$$C_\ell = (2\ell + 1)\sigma_i/z_\chi.$$  \hspace{1cm} (30)

Numerically, one can implement this as

for each 1 
for i = 0 to 2\ell-1 
    z = z+ rand_gauss()^2 
end 
C(l) = (2l+1)*sigma(l)/z
end

An example of this method can be found in the SLAVE libraries, within class “powerspectrum” method “draw_gamma”.

4.2. $P(s, C_\ell, d)$

From equation 22, it is easy to see that $P(s|C_\ell, d)$ is a Gaussian distribution with mean $\hat{s}$ and variance $(C^{-1} + N^{-1})^{-1}$. Instead of deriving a method for drawing a random variable from this distribution, we present the solution and show that this solution indeed has the necessary properties [Jewell et al. 2004]. Let

$$s = (C^{-1} + N^{-1})^{-1}(N^{-1}d + N^{-\frac{1}{2}}\omega_1 + C^{-\frac{1}{2}}\omega_2)$$  \hspace{1cm} (31)

where $\omega_1$ and $\omega_2$ are independent, random $N(0, 1)$ variables. We now show that the random variable $s$ indeed has mean $\hat{s}$ and variance $(C^{-1} + N^{-1})^{-1}$. First,

$$\langle s \rangle = (C^{-1} + N^{-1})^{-1}(N^{-1}\langle d \rangle + N^{-\frac{1}{2}}\langle \omega_1 \rangle + C^{-\frac{1}{2}}\langle \omega_2 \rangle).$$  \hspace{1cm} (32)

As $\langle \omega_1 \rangle = \langle \omega_2 \rangle = 0$, $\langle s \rangle = (C^{-1} + N^{-1})^{-1}N^{-1}(d) = \hat{s}$ by definition.

The covariance is then

$$\langle (s-\hat{s})(s-\hat{s})^T \rangle.$$  \hspace{1cm} (33)

Note that in the term $s - \hat{s}$, we have $(C^{-1} + N^{-1})^{-1}(N^{-1}d - N^{-1}d) = 0$, so we are only left with the terms with the random variables $\omega$:

$$\langle (s-\hat{s})(s-\hat{s})^T \rangle = (C^{-1} + N^{-1})^{-1}.$$  \hspace{1cm} (34)

But, as $\omega_1$ and $\omega_2$ are independently drawn from a $N(0, 1)$ distribution, then $\langle \omega_1 \omega_2 \rangle = \delta_{ij} I$, and we end up with

$$\langle (s-\hat{s})(s-\hat{s})^T \rangle = (C^{-1} + N^{-1})^{-1}$$  \hspace{1cm} (35)

which shows that a random variable drawn using equation 31 has the desired properties of being drawn from $P(s|C_\ell, d)$.

Having implemented a “real alm” class in SLAVE with operator overloading, it is possible to directly translate equation 31 into code:

omega1.gaussian_draw(0, 1, rng); 
omega2.gaussian_draw(0, 1, rng); 
calculate_CNi(); 
S = CNI* (NI*D + NI.square_root()*omega1 - CI.square_root()*omega2);

where the code has been slightly optimized: both $C^{-1}$, $N^{-1}$ and $(C^{-1} + N^{-1})^{-1}$ has been pre-calculated for efficiency. Note that this is only possible to do when assuming full-sky coverage with constant RMS noise. If the noise isn’t constant on the sky, then $N$ is a dense off-diagonal matrix, nearly impossible to calculate directly for large $\ell$. However, it is still possible to perform the calculation in pixel space, but this requires that we assume $N$ to be an operator instead of a matrix. We will address this issue in section 4.6.

We have now presented the main simplified Gibbs steps for calculating $P(s|C_\ell, d)$ and $P(C_\ell | s, d)$, without convolution, uniform noise and no sky cut. Sampling from these two distributions is then done alternating between the two Gibbs steps, and the chain output - $s$ and $C_\ell$ - are saved to disk during each step.

We now investigate the behavior of these fields, as each have special properties.

4.3. Field properties

Equation 31 can be broken into two separate parts: the Wiener filter $(C^{-1} + N^{-1})^{-1}(N^{-1}d)$ and the fluctuation map $(C^{-1} + N^{-1})^{-1}(N^{-\frac{1}{2}}\omega_1 + C^{-\frac{1}{2}}\omega_2)$. In figure 3 each of these maps are depicted. The Wiener filter map determines the fluctuations outside the sky cut - where they are heavily constrained by the known data, given cosmic variance and noise. However, within the sky cut, large-scale fluctuations are possible to pin down statistically while small-scales are repressed. The fluctuation map determines the small-scale fluctuations within the unknown sky cut, and are constrained by cosmic variance and noise effects. Outside the sky cut, the fluctuation map is constrained by the data, yielding very low small-scale fluctuations. The sum of these two parts make up the full CMB signal sample.
A thing we did not address in the previous section was the inclusion of the instrumental beam convolution $A$. Including this in equation (31) we obtain

$$(C^{-1} + A^T N^{-1} A)s = AN^{-1}d + AN^{-\frac{1}{2}}\omega_1 + C^{-\frac{1}{2}}\omega_2.$$  

(38)

In SLAVE, the beam is loaded directly from a fits file, or generated as a Gaussian beam given a full width half-maximum (FWHM) range. The beam is then multiplied with the corresponding pixel window, and stored in the $a_{inx}$-object $A$ throughout the code.

4.6. The sky cut

Until now, we have only assumed full-sky data sets contaminated by constant noise. However, in order to be able to investigate real data, we need to take into account both the foreground galaxy and anisotropic noise. The galaxy contributes to almost 20% of the WMAP data, and needs to be removed with a mask. This means that the usable pars of the maps becomes anisotropic, giving rise to correlations in the spherical harmonics $a_{lm}$s. In other words, all the previously diagonal and well-behaved matrices now have off-diagonal elements, which for large $m_{\max}$ is an impossible feat to perform for dense matrices.

One way to get around these problems is to perform the calculations containing the sky cut mask in pixel space. This means that every time one needs to take into account the sky cut, one transforms from harmonic to pixel space, performs the operation including the sky cut before transforming back to harmonic space. While this operation in itself is trivial, equation (31) provides a few other problems:

$$(C^{-1} + A^T N^{-1} A)s = AN^{-1}d + AN^{-\frac{1}{2}}\omega_1 + C^{-\frac{1}{2}}\omega_2.$$  

(39)

The right-hand side can easily be calculated, letting $N^{-1}$ be an operator acting on $d$ and $\omega$, switching from spherical harmonics to pixel space and back. However, the left-hand side is troublesome - one cannot solve this equation explicitly. First, we need to rewrite (39) a bit:

$$(1 + C^{-\frac{1}{2}} A^T N^{-1} A C^{-\frac{1}{2}})(C^{-\frac{1}{2}} s) =$$  

(40)

$$C^{-\frac{1}{2}} AN^{-1}d + C^{-\frac{1}{2}} AN^{-\frac{1}{2}}\omega_1 + \omega_2 = b.$$  

(41)

The first thing one should note about equation (41) is that the left-hand term is proportional to $(1 + S/N)$, where the diagonal parts are just the signal-to-noise ratios of the corresponding mode. Another nice feature about this form is that the variance of the signal is kept constant, that is, $\text{Var}(s) \sim \ell^{-2}$, but $\text{Var}(C^{-\frac{1}{2}} s) \sim I$. Hence we obtain better numerical stability. In order to solve the equation $(1 + S/N)x = b$, we implement a direct-from-textbook Conjugate Gradient (CG) algorithm presented on page 40 in [Shewchuk (1994)]. The code looks like this:

```c
b = L*(A*Ni(D) + A*Ni(map_work2,true)) + omega2; MI = setup_preconditioner(); x = mult_by_A(x); r = b - x; d = MI*r; r0 = r.norm_L1(r); do {
  Ad = mult_by_A(d);
  alpha = r.dot(MI*r) / (d.dot(Ad));
  x = x + d*alpha;
  rn = r - Ad*alpha;
  ...}
```

4.5. Convolution

The trick lies with the noise. As

$$\chi^2 = \sum \frac{(d - As)^2}{\sigma_{\text{RMS}}^2},$$  

(37)

and the $\chi^2$ should be close to the number of pixels in the map plus minus $\sqrt{2n}$. Usually, when an incorrect parameter is used, the $\chi^2$ comes out far away from the expected value.

Calculating the $\chi^2$ is not particularly time-consuming, but it has other uses as well: the $\chi^2$ is used in the estimation of noise, as presented in section 3.
Obviously, \( \sigma \) space. The variance is then given as

\[
\text{For a single binned set with } n \text{ multipoles ranging from } \ell_{\text{low}} \text{ to } \ell_{\text{high}}, \text{ the average value of the power spectrum is given as}
\]

\[
D_\ell = \frac{1}{n} \sum_{\ell_{\text{low}}}^{\ell_{\text{high}}} C_\ell.
\]

Similarly for the noise power spectrum,

\[
N_b = \frac{1}{n} \sum_{\ell_{\text{low}}}^{\ell_{\text{high}}} N_\ell.
\]

Thus, the variance of the noise is given as

\[
\sigma_N^2 = \text{Var}(N_b) = \frac{1}{n^2} \sum_{\ell_{\text{low}}}^{\ell_{\text{high}}} \text{Var}(N_\ell).
\]

Obviously, \( \sigma_N \) is reduced as the number of multipoles in the bin \( n \) is increased. We now select bins such that the noise variance in a single bin is always less than three times the value of the angular power spectrum, or \( \sigma_n < 3D_\ell \).

The only affected part of the code is where one determines \( P(C_\ell|s,d) \). Instead of generating a power spectrum \( C_\ell \) given a set of \( \sigma_\ell \), the calculation is now performed via a binning class that calculates the binned power spectrum \( C_b \). That is,

\[
P(C_b|\sigma) = \prod_{\ell_{\text{low}}}^{\ell_{\text{high}}} \left( \frac{e^{-\frac{2(\ell+1)\sigma^2}{C_b^2}}}{\sigma^2} \right).
\]

Absorbing the product into the exponential, this becomes

\[
P(C_b|\sigma) = e^{-\frac{1}{\sigma^2} \sum_{\ell} (2\ell+1)\sigma^2} C_b^2 \sum_{\ell} (2\ell+1). 
\]

We now sample the signal with flat bins in \( \ell(\ell+1)/(2\pi) \), not in \( \ell \).

5. GENERALIZING THE MODEL: NOISE ESTIMATION

In this section, we give a direct example of how one could extend the data model to the SLAVE Gibbs sampler. We derive the necessary conditional distribution, explain how this was integrated, and present some results from [Groeneboom et al. (2009a)], where a full analysis of the noise levels in the WMAP data was performed using the SLAVE framework.

Traditionally, the noise properties used in the Gibbs sampler (e.g., [Eriksen et al. 2004]) have been assumed known to infinite precision. In this section, however, we relax this assumption, and introduce a new free parameter, \( \alpha \), that scales the fiducial noise covariance matrix, \( N^{\text{fid}} \), such that \( N = \alpha N^{\text{fid}} \). Thus, if there is no deviation between the assumed and real noise levels, then \( \alpha \) should equal 1. The full analysis of the 5-yr WMAP data was presented in [Groeneboom et al. (2009a)], with interesting results. For the foreground-reduced 5-year WMAP sky maps, we find that the posterior means typically range between \( \alpha = 1.005 \pm 0.001 \) and \( \alpha = 1.010 \pm 0.001 \) depending on differencing assembly, indicating that the noise level of these maps are underestimated by 0.5-1.0%. The same problem is not observed for the uncorrected WMAP sky maps.
and variance $\sigma$, where the first term is the likelihood, $P(d | s, \alpha) \cdot P(s, C_t | \alpha) \cdot P(\alpha)$

where the first term is the likelihood,

$$P(d | s, \alpha) = \frac{e^{-\frac{1}{2}(d-s)(\alpha N)^{-1}(d-s)}}{\sqrt{|\alpha N|}},$$

the second term is a CMB prior, and the third term is a prior on $\alpha$. Note that the latter two are independent, given that these describe two a-priori independent objects. In this paper, we adopt a Gaussian prior centered on unity mean and standard deviation $\sigma$. Typically, we choose a very loose prior, such that the posterior is completely data-driven.

The conditional distribution for $\alpha$ can now be expressed as

$$P(\alpha | s, C_t, d) \propto \frac{e^{-\frac{1}{2}d(s)\gamma^{-1}(d-s)}}{\alpha^{n/2}}$$

where $n = N_{\text{pix}}$ and $\beta = (d - s)N^{-1}(d - s)$ is the $\chi^2$. (Note that the $\chi^2$ is already calculated within the Gibbs sampler, as it is used to validate that the input noise maps and beams are within a correct range for each Gibbs iteration. Sampling from this distribution within the Gibbs sampler represents therefore a completely negligible extra computational cost.) For the Gaussian prior with unity mean and standard deviation $\sigma_n$, we find that

$$P(\alpha | s, C_t, d) \propto \frac{e^{-\frac{1}{2}(\frac{d}{\alpha_n^2} + \frac{s}{\alpha^2})}}{\alpha^{n/2}}$$

For large degrees of freedom, $n$, the inverse gamma function converges to a Gaussian distribution with mean $\mu = b/(k+1)$, where we have defined $k = n_{\text{pix}}/2 - 1$, and variance $\sigma^2 = b^2/((k-1)(k-1)(k-2))$. A good approximation is therefore letting $\alpha_{n+1}$ be drawn from a product of two Gaussian distributions, which itself is a Gaussian, with mean and standard deviation

$$\mu = \frac{\mu_1 \sigma_1^2 + \mu_2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

$$\sigma = \frac{\sigma_1 \sigma_2}{\sigma_1^2 + \sigma_2^2}.$$
6.2. \(C_\ell\) likelihoods

The first important step is to verify that the output \(C_\ell\)s follow the desired inverse-Gamma distribution for low \(\ell\), but converges to Gaussians for larger \(\ell\). The SLAVE processing utility \texttt{SLAVE\_PROCESS} can generate a set of \(C_\ell\)s from the \(\sigma_\ell\)s and output the corresponding values for a single \(\ell\). It is then straightforward to use a graphical utility such as \texttt{XMGRACE} to obtain the histogram. Such histograms are plotted together with the analytical likelihoods in figure 7. Note the good match between the histogram of the \(C_\ell\)s and the likelihoods obtained from the Blackwell-Rao estimator. The analysis for producing these plots was performed on simulated high-detail data, in order to verify the validity of the BR-estimator.

To save the cls for a specific \(\ell\), type

```
./process 4 [sigma_l file] [1] [generate no cls] [output textfile]
```

6.3. The Blackwell-Rao estimator

Our primary objective is obtaining the best-fit power spectrum from the estimated signal power spectra. If the \(C_\ell\)s are completely distributed according to a Gaussian, one would only need to select the maximum of the distribution for each \(C_\ell\). However, as we saw in equation 18, this is not the case, and we need a better way to obtain the likelihood \(L(C_\ell)|\ell\) for each \(\ell\).

Luckily, we can obtain an analytical expression of the likelihood for the \(C_\ell\)s via the Blackwell-Rao (BR) estimator, as presented in [Chu et al. 2003]. By using prior knowledge of the distributions of the \(C_\ell\)s, we can build an analytical expression for the distribution for each \(C_\ell\) given the signal power spectrum \(\sigma_\ell\), or \(P(C_\ell|\sigma_\ell)\).

Note that since the power spectrum only depends on the data through the signal and thus \(\sigma_\ell\), then

\[
P(C_\ell | s, d) = P(C_\ell | s) = P(C_\ell | \sigma_\ell).
\]

(55)

It is therefore possible to approximate the distribution...
Fig. 7.— The histograms of the $C_{l,s}$ (red) and the BR-estimated likelihoods (black) for various $l$. Note how the distribution converges to a Gaussian for larger multipoles $l$. The analysis has been performed on simulated WMAP-like data.

$$P(C_l | d)$$ as such:

$$P(C_l | d) = \int P(C_l, s | d) \, ds \quad \text{(56)}$$

$$= \int P(C_l | s, d) P(s | d) \, ds \quad \text{(57)}$$

$$= \int P(C_l | \sigma_l) P(\sigma_l | d) \, D\sigma_l \quad \text{(58)}$$

$$\approx \frac{1}{N_G} \sum_{i=1}^{N_G} P(C_l | \sigma_l^i) \quad \text{(59)}$$

where $N_G$ is the number of Gibbs samples in the chain. This method of estimating the $P(C_l | d)$ is called the Blackwell-Rao estimator. Now, for a Gaussian field,

$$P(C_l | \sigma_l) \propto \prod_{\ell=0}^{\infty} \frac{1}{\sigma_l \sqrt{C_l}} e^{2\ell+1 \frac{\sigma_l}{C_l}}. \quad \text{(60)}$$

Taking the logarithm, we obtain a nice expression

$$\ln P(C_l | \sigma_l) = \sum \left( \frac{2\ell+1}{2} \left[ -\frac{\sigma_l}{C_l} + \ln \left( \frac{\sigma_l}{C_l} \right) \right] - \ln \sigma_l \right) \quad \text{(61)}$$

which is straight-forward to implement numerically. To output the BR-estimated likelihood for one $l$, type

./process 3 [sigma_l file] [output likelihood]

6.4. Power spectrum estimation

The best-fit BR-estimated power spectrum is obtained by choosing the maximum likelihood value of $C_l$ for each $l$. To do so, type

./process 2 [sigma_l file] [output power spectrum file]

An example of a BR-estimated power spectrum can be seen in figure 8. In addition, both the input-and noise power spectra are shown. Note how the BR-estimated power spectrum is exact on small scales (low $l$), while the convolution and noise dominated on higher scales.

7. CONCLUSIONS

We have presented a self-contained guide to a CMB Gibbs sampler, having focused on both deriving the conditional probability distributions and code design. We described in detail how one can draw samples from the conditional distributions, and saw how the Gibbs sampler is numerically superior to conventional MCMC methods, scaling as $O(n^{1.5})$. We have also introduced a new object-oriented CMB Gibbs framework, which employs the existing HEALPix (Górski et al. 2005) C++
package. We presented a small guide to the usage of SLAVE, including post-processing tools and the Blackwell-Rao estimator for obtaining the likelihoods and the best-fit power spectrum. We also reviewed a new way of estimating noise levels in CMB maps, as presented in Groeneboom et al. [2009]. The software package SLAVE will hopefully be released when it is completed during 2009, and will run on all operating systems supporting the GNU C++ compiler. Please see http://www.irio.co.uk for release details and information.

Nicolaas E. Groeneboom acknowledges financial support from the Research Council of Norway. Nicolaas especially wishes to thank Hans Kristian Eriksen, but also Jeffrey Jewell, Kris Gorski, Benjamin Wandelt and the whole “Gibbs team” at Jet Propulsion Laboratories (JPL) for useful discussions, comments and input. The computations presented in this paper were carried out on Titan, a cluster owned and maintained by the University of Oslo and NOTUR. We acknowledge use of the HEALPix software [Górski et al. 2005] and analysis package for deriving the results in this paper. We acknowledge the use of the Legacy Archive for Microwave Background Data Analysis (LAMBDA). Support for LAMBDA is provided by the NASA Office of Space Science.

REFERENCES

Bennett, C. L., et al. 2003, ApJS, 148, 1
Chu, M., Eriksen, H. K., Knox, L., Górski, K. M., Jewell, J. B., Larson, D. L., O’Dwyer, I. J., & Wandelt, B. D. 2005, Phys. Rev. D, 71, 103002
Chu, I.-W. M. 2005, Ph.D. Thesis, Cumberbatch, D. T., Zuntz, J., Kamfjord Eriksen, H. K., & Silk, J. 2009, arXiv:0902.0039
Dickinson, C., Eriksen, H. K., Jewell, J., Banday, A. J., Gorski, K. M., & Lawrence, C. R. 2007, Bulletin of the American Astronomical Society, 38, 883
Dickinson, C., et al. 2009, arXiv:0903.4311
Dunkley, J., et al. 2008, arXiv:0811.4289
Eriksen, H. K., et al. 2004b, ApJS, 155, 227
Eriksen, H. K., & Lawrance, C. R. 2008a, ApJ, 676, 10
Eriksen, H. K., Dickinson, C., Jewell, J. B., Banday, A. J., Górski, K. M., & Lawrence, C. R. 2008b, ApJ, 672, L87
Górski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelmann, M. 2005, ApJ, 622, 759
Groeneboom, N. E and Eriksen, H. K, 2008, ApJ, 690.1807G
Groeneboom, N. E., et al. 2009 [arXiv:0904.2554]
Hinshaw, G., et al. 2007, ApJS, 170, 288
Hivon, E. Górski, K. M., Netterfield, C. B., Crill, B. P., Prunet, S., & Hansen, F. 2002, ApJ, 567, 2
Jewell, J. B., Eriksen, H. K., Wandelt, B. D., O’Dwyer, I. J., Huey, G., & Górski, K. M. 2009, ApJ, 697, 258
Jewell, J., Levin, S., & Anderson, C. H., 2004, ApJ, 609
Rudjord, Ø., Groeneboom, N. E., Eriksen, H. K., Huey, G., Górski, K. M., & Jewell, J. B. 2009, ApJ, 692, 1669
Larson, D. L., Eriksen, H. K., Wandelt, B. D., Górski, K. M., Huey, G., Jewell, J. B., & O’Dwyer, I. J. 2007, ApJ, 656, 653
Shewchuk, Jonathan R., 1994, Carnegie Mellon University
Shewchuk, Jonathan R., 1994, Carnegie Mellon University
Stash, K. M., Zahn, O., & Doré, O. 2007, Phys. Rev. D, 76, 043510
Smoot et al., 1992, ApJ396, L1
Wandelt, Benjamin D. and Larson, David L. and Lakshminarayanan, Arun Phys. Rev. D70,8
