Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Bayesian parameter estimation applied to (non-)Gaussian random fields

Thesis on behalf of the
Delft Institute of Applied Mathematics
as part to be obtained

the degree of

BACHELOR OF SCIENCE
in
APPLIED MATHEMATICS

by
LEVI KLOMP

under supervision of
dr.ir. J. Bierkens

Delft, The Netherlands
June 30th 2021

Abstract

The purpose of this study is to define and estimate multivariate statistic models, inspired by the data of the Cosmic Microwave Background Radiation, and apply those models to self-simulated data. Two distinct models are constructed using random fields, which include a set of parameters, structural covariance among random variables and a multivariate (non-)Gaussian distribution. To obtain the optimal parameter estimates for a model, methods like Maximum likelihood estimation and Bayesian parameter estimation, are utilised for these predefined models. Only the Bayesian parameter estimation is used, since it is illustrated to be superior, compared to Maximum likelihood estimation, for parameter estimation of statistical models consisting of random fields. The structure of the models makes it impossible to compute estimates analytically, hence the Metropolis-Hastings algorithm is employed to calculate the estimates. Finally a comparison about the performance of parameter estimation among several models with dissimilar mask sizes is established.

Contents

1 Introduction

The Cosmic Microwave Background Radiation, abbreviated (CMB), is one of the most important discoveries in the field of cosmology (Evans (2015)). It describes the composition of the early universe. This is because this specific radiation, observed by satellites, originates from a relative short epoch after the Big Bang. In every direction of the night sky the satellite points, it observes this CMB. The radiation is converted by the satellite into heat in order to create a heatmap of the universe which is shown in Figure 1. But because the universe consists of a lot of factors which block or distort the data (e.g. planets, black holes or other extraterrestrial phenomena), it is paramount to construct a model for the CMB map in order to eliminate those factors.

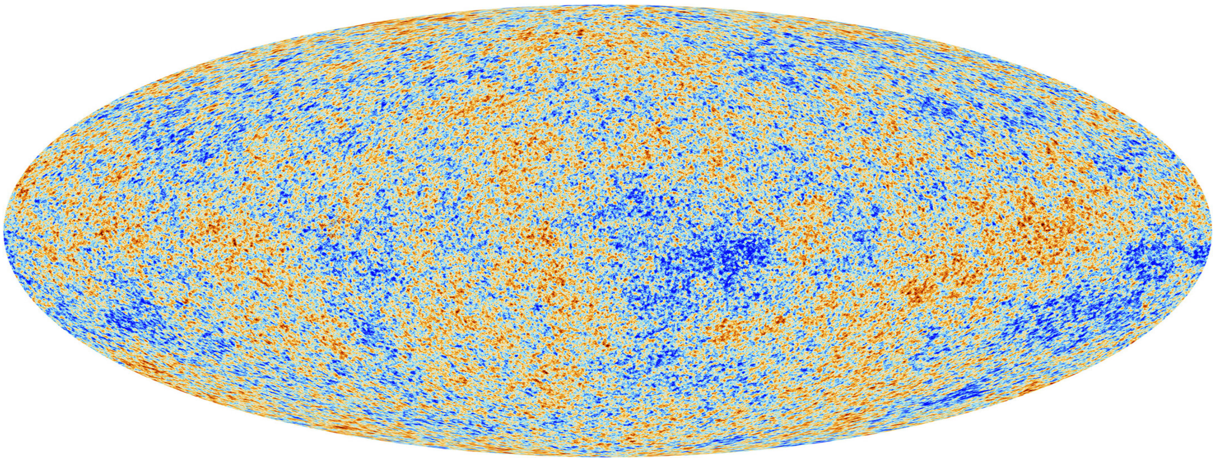


Figure 1: Heatmap of the Cosmic Microwave Background radiation (ESA and the Planck Collaboration (2013)).

1.1 Overview

The goal of this paper is to apply statistical models to simulated data, which has similarities with the observed CMB map in Figure 1. The building blocks of these models are called random fields, introduced by Vio et al. (2001). In **Section 2** the theory behind this important concept is treated and the Gaussian random field is defined. With Gaussian random fields we can create different types of non-Gaussian random fields, where all of them have diverse characteristics. This is discussed in **Section 3**, where the definitions and simulations of various random fields are given. These non-Gaussian random fields increase in complexity with respect to the Gaussian random fields. Some of these examples are inspired by Vio et al. (2001).

Using these random fields we can build models for the CMB data, where the first model is covered in **Section 4** and the second model in **Section 5**. These are two distinct models suggested respectively by Vio et al. (2001) and Taylor et al. (2008), which both are built on a number of parameters. The first model treats every coordinate in the random field as a random variable, whereas in the second model the random field is created by using an eigenfunction expansion. Here the eigenfunction coefficients will assume some distribution, which will together with the number of distinct predefined eigenfunctions govern the structure of the random field.

After constructing the two models , the next step will be to estimate the model for a given set of observations. In **Section 6** Maximum likelihood estimation (MLE) is applied to all the different examples we have encountered in the previous sections. However it is shown here that MLE is very inconvenient for these examples, because of the computational intractability it produces. Another parameter estimation method for the two models is Bayesian parameter estimation (BPE), which is covered in **Section 7**. In contrast to MLE, BPE can be used to estimate the parameters of the CMB models found in earlier sections. Still we cannot describe BPE analytically, because of the complexity of our models. The solution to this is in **Section 8**, where we use the theory of BPE to calculate the parameter estimates computationally with a tool called the Metropolis-Hastings algorithm (MH). We simulate all the previous examples in this section and after that we use the MH algorithm to estimate the parameters and compare these estimates with the real values. These results will be visualised in heat maps, tables and trace plots for every example.

An addition to the observation of data will be the possibility of using masks, which is introduced in **Section 9**. A mask will eliminate a specific selection of pixels of the data. The physical interpretation can be that these pixels are distorted or blocked by some spatial phenomena, like planets or stars. A consequence of this will be that it affects the performance of the quality of the parameter estimation. In **Section 10** one will compare the performance of parameters estimation, by using the MH algorithm, on different masks sizes. This can give us some useful information about the credibility about a given estimate of parameters. Finally a summary is given in **Section 11**, which also includes further research topics relating to the results in this paper. Furthermore a, more trivial, summary for laymen is given in **Section 12**.

2 Preliminaries on random fields and the Gaussian random field

The data of the CMB radiation, in Figure 1, seems difficult to model because all of its small 'lumps' of different temperatures. This is because there is some kind of structure between the values of the pixels, or equivalently it is not independent, identically distributed random noise that is observed by the satellite. For this research it is important to describe such observed data in a statistical model. Random fields, as suggested in Vio et al. (2001), are such statistical models which are frequently used for clustered data like the CMB data.

2.1 Defining a Gaussian random field

To build a random field one needs a set of coordinates. Each coordinate in the set gets assigned a random value. Here the set of coordinates are all the pixels and each pixel has a random value which represents the temperature. Formally written, if there are N coordinates, the d-dimensional field (set of coordinates) is defined as $V := \{\boldsymbol{t}_1, ..., \boldsymbol{t}_N\}$ where $\boldsymbol{t}_i \in \mathbb{R}^d, \ \forall i = 1, ..., N$. In this paper we will only make use of the case $d = 2$ (a 2-dimensional field).

The temperature value of a coordinate $\boldsymbol{t}$ is defined as $R(\boldsymbol{t})$, also the set of all temperatures is denoted by the vector $\boldsymbol{R}$ (temperature field). To describe the relation between a pair of random values, a covariance function is constructed, denoted by the letter $\xi_{\boldsymbol{\eta}}$ where $\boldsymbol{\eta}$ will be some vector with function parameters

$$\xi_{\boldsymbol{\eta},R}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \mathrm{Cov}(R(\boldsymbol{t}_1), R(\boldsymbol{t}_2)), \ \ \forall \boldsymbol{t}_1, \boldsymbol{t}_2 \in V \tag{1}$$

note that $R(\boldsymbol{t})$'s are assumed to be random.

For the first example we will use the notation for the random values $X(\boldsymbol{t})$ (instead of $R(\boldsymbol{t})$) to assume that the temperature for all points $\boldsymbol{t} \in V$ are standard Gaussian distributed, i.e. $X(\boldsymbol{t}) \sim \mathcal{N}(0, 1)$. Also the covariance structure will be created by the covariance function $\xi_{\boldsymbol{\eta},X}(\boldsymbol{t}_1, \boldsymbol{t}_2) := \mathrm{Cov}(X(\boldsymbol{t}_1), X(\boldsymbol{t}_2))$.

Thus a *standard Gaussian random field* denoted by the N-dimensional random vector $\boldsymbol{X} = \{X(\boldsymbol{t}_1), ..., X(\boldsymbol{t}_N)\}$ has $\forall \boldsymbol{t} \in V$ that $X(\boldsymbol{t}) \sim \mathcal{N}(0, 1)$ and for all pairs $\boldsymbol{t_1}, \boldsymbol{t_2} \in V$ that $\mathrm{Cov}(X(\boldsymbol{t}_1), X(\boldsymbol{t}_2)) = \xi_{\boldsymbol{\eta},X}(\boldsymbol{t}_1, \boldsymbol{t}_2)$ for some predefined covariance function $\xi_{\boldsymbol{\eta},X}$ with function parameters $\boldsymbol{\eta}$. Some alternative terminology is that we say that the field is Gaussian.

2.2 Theory behind simulating a Gaussian random field

Here we discuss how to sample from such a Gaussian random field. The difficulty here is that there is some underlying dependence among the random variables $X(\boldsymbol{t})$. In order to simulate from $\boldsymbol{X}$ it is mandatory to transform this univariate $\mathcal{N}(0, 1)$-distribution with covariance function $\xi_{\boldsymbol{\eta},X}$ into some multivariate distribution. Since $\boldsymbol{X}$ is a vector of standard Gaussian random variables it is possible to write $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$, where $\boldsymbol{\Sigma}_{\boldsymbol{\eta}} \in \mathbb{R}^{N \times N}$. Note that every element of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ is determined by the pairwise covariance among all random variables in the random field $\boldsymbol{X}$, which are defined by the function $\xi_{\boldsymbol{\eta},X}$. Hence the covariance matrix of the Gaussian random field $\boldsymbol{X}$ will look like

$$\boldsymbol{\Sigma}_{\boldsymbol{\eta}} = \begin{bmatrix} \xi_{\boldsymbol{\eta}}(\boldsymbol{t}_1, \boldsymbol{t}_1) & \xi_{\boldsymbol{\eta}}(\boldsymbol{t}_1, \boldsymbol{t}_2) & \dots & \xi_{\boldsymbol{\eta}}(\boldsymbol{t}_1, \boldsymbol{t}_N) \\ \xi_{\boldsymbol{\eta}}(\boldsymbol{t}_2, \boldsymbol{t}_1) & \xi_{\boldsymbol{\eta}}(\boldsymbol{t}_2, \boldsymbol{t}_2) & \dots & \xi_{\boldsymbol{\eta}}(\boldsymbol{t}_2, \boldsymbol{t}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{\boldsymbol{\eta}}(\boldsymbol{t}_N, \boldsymbol{t}_1) & \xi_{\boldsymbol{\eta}}(\boldsymbol{t}_N, \boldsymbol{t}_2) & \dots & \xi_{\boldsymbol{\eta}}(\boldsymbol{t}_N, \boldsymbol{t}_N) \end{bmatrix}. \tag{2}$$

Alternative terminology for the covariance matrix of a Gaussian random field, which will be used repeatedly in this paper, is the *Gaussian covariance matrix.* So in order to simulate a Gaussian random field we use the fact that the random vector $\boldsymbol{X}$ is representing a Gaussian random field with *Gaussian covariance function* $\xi_{\boldsymbol{\eta},X}$, with parameters $\boldsymbol{\eta}$. Then $\boldsymbol{X} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$ where the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ has the form of (2).

There are several techniques to sample from a multivariate Gaussian such as the Cholesky decomposition or the spectral decomposition, which will be applied to simulate the example in the next section.

2.2.1 Simulating an example for a Gaussian random field

The example for this section will be the 2-dimensional grid with coordinates $(-L : \Delta x : L)$ in the x-direction and $(-M : \Delta y : M)$ in the y-direction. This notation means that for x we have coordinates between and including $-L$ and L with step size Δx. For the example we take $L = M = 10$ and $\Delta x = \Delta y = 0.25$. This implies that there are $N = 6561$ grid points and thus a 6561-dimensional distribution. Now constructing the covariance function of the Gaussian random field with parameters $\boldsymbol{\eta} = \{A, B, C\}$ by

$$\xi_{\boldsymbol{\eta},X}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \xi_{\{A,B,C\},X}(\boldsymbol{t}_1, \boldsymbol{t}_2) = A\cos(B||\boldsymbol{t}_1 - \boldsymbol{t}_2||)\exp(-C||\boldsymbol{t}_1 - \boldsymbol{t}_2||)$$

where, in this example, we assume that $A = 1, B = 1, C = 0.1$. Then $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ will be evaluated by equation (2). A simulation of such $\boldsymbol{X} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$ will then be done by taking a sample from this multivariate distribution and plotted in a heatmap which is shown in Figure 2.

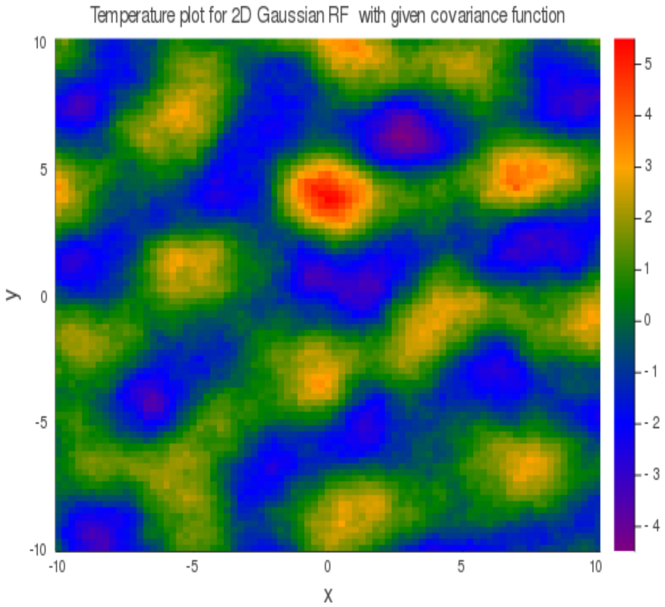


Figure 2: Simulated Gaussian random field for this section's example.

3 Theory behind non-Gaussian random fields

In the previous section the concepts of a random field were introduced. Specifically Section 2.2.1 describes how a Gaussian random field $\boldsymbol{X}$ for a given covariance function $\xi_{\boldsymbol{\eta},X}$ can be simulated. Figure 2 showed what such a sample looks like. But there are a lot of different random fields, with different probability distributions. These random fields will not be Gaussian and are thus called non-Gaussian random fields. We will denote such a random field with the symbol $\boldsymbol{R}$ and its definition follows the same concept as the definition of a Gaussian random field.

A *non-Gaussian random field* denoted by the N-dimensional random vector $\boldsymbol{R} = \{R(\boldsymbol{t}_1), ..., R(\boldsymbol{t}_N)\}$ has $\forall \boldsymbol{t} \in V$ that $R(\boldsymbol{t}) \sim \mathcal{M}_{\boldsymbol{\alpha}}$ for some parameters $\boldsymbol{\alpha}$, where the distribution $\mathcal{M}_{\boldsymbol{\alpha}}$ is a multivariate non-Gaussian distribution. For all pairs $\boldsymbol{t_1}, \boldsymbol{t_2} \in V$ that $\mathrm{Cov}(R(\boldsymbol{t}_1), R(\boldsymbol{t}_2)) = \xi_{\boldsymbol{\eta},R}(\boldsymbol{t}_1, \boldsymbol{t}_2)$ for some predefined covariance function $\xi_{\boldsymbol{\eta},R}$ with function parameters $\boldsymbol{\eta}$. Some alternative terminology is that we say that the field is Gaussian.

Since we are not restricted to Gaussian random fields, one can think of an endless amount of different examples of random fields. We are not restricted to a random field which is Gaussian, which provides a lot of freedom when trying to find the best model for the CMB data. In the following subsections some different examples for non-Gaussian random fields will follow.

3.1 Non-Gaussian random fields by using a transformation $g_{\boldsymbol{\theta}}[\cdot]$ on a Gaussian random field

A whole family of non-Gaussian distributions can be defined by mapping the Gaussian random field using a transformation $g_{\boldsymbol{\theta}} : \mathbb{R} \to \mathbb{R}$ (Vio et al. (2001)). Here $\boldsymbol{\theta}$ are the parameters in the transformation. A non-Gaussian random field $\boldsymbol{R}$ can be acquired by using a Gaussian random field and the following relation:

$$R(\boldsymbol{t}) = g_{\boldsymbol{\theta}}(X(\boldsymbol{t})), \quad \forall \boldsymbol{t} \in V \tag{3}$$

Or more shortly and conveniently written as $\boldsymbol{R} = g_{\boldsymbol{\theta}}(\boldsymbol{X})$. Important is that this function transforms the Gaussian random field pointwise to a non-Gaussian random field (it is $\mathbb{R} \to \mathbb{R}$).

First considering the case that $R(\boldsymbol{t}) = g_{a,b}(X(\boldsymbol{t})) = a + bX(\boldsymbol{t}), \ \forall \boldsymbol{t} \in V$ and $a, b \in \mathbb{R}$, i.e. it is a linear transformation. Since $X(\boldsymbol{t}) \sim \mathcal{N}(0, 1)$, equivalent multivariate notation $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$, by linearity of the normal distribution it holds that $R(\boldsymbol{t}) \sim \mathcal{N}(a, b^2), \ \forall \boldsymbol{t} \in V$. Which means that $\forall \boldsymbol{t} \in V, \ \mathbb{E}[R(\boldsymbol{t})] = a$ and

$$\mathrm{Cov}(R(\boldsymbol{t}_1), R(\boldsymbol{t}_2)) = \mathrm{Cov}(a + bX(\boldsymbol{t}_1), a + bX(\boldsymbol{t}_2)) = b^2\mathrm{Cov}(X(\boldsymbol{t}_1), X(\boldsymbol{t}_2)),$$

which implies that $\boldsymbol{R} \sim \mathcal{N}(a\boldsymbol{1}_N, b^2\boldsymbol{\Sigma}_{\boldsymbol{\eta}})$. Here $\boldsymbol{1}_N$ is a N-dimensional vector where all entries are equal to 1. This means that this linear mapping gives again a Gaussian random field.

The linear transformations are not interesting since it generates a Gaussian random field again. Hence we want to look into some non-linear transformations of $g_{\boldsymbol{\theta}}[\cdot]$. An example of such a non-linear transformation is a lognormal distribution, which is an exponential transformation of a Gaussian distribution.

For the simulation of such a mapped, non-Gaussian random field one only needs to generate a Gaussian random field. After that every pixel of the Gaussian random field is mapped using the transformation. This will be visualised in the next subsection.

3.1.1 Simulating an example of a transformation: Lognormal random field

For this example the mapping $R(\boldsymbol{t}) = g_{\mu,\sigma}(X(\boldsymbol{t})) = \exp(\mu + \sigma X(\boldsymbol{t}))$ is assumed, where again $X(\boldsymbol{t}) \sim \mathcal{N}(0, 1)$. By the definition this means that $R(\boldsymbol{t}) \sim \mathrm{Lognormal}(\mu, \sigma^2)$. Simulating a lognormal random field is not difficult. After generating a Gaussian random field one needs to map these values pixel per pixel. Such a simulation is illustrated in Figure 3. We assign the values $\mu = 2, \sigma = 0.1$ to the transformation parameters $\boldsymbol{\theta}$. Note that the same example is used as in Section 2.2.1

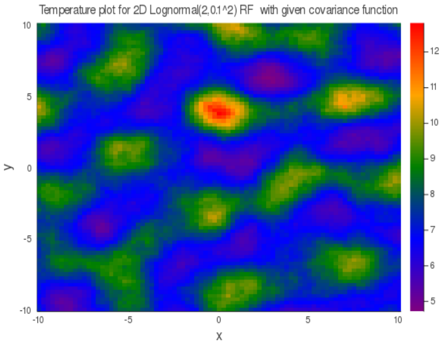


Figure 3: Simulated lognormal random field for the same Gaussian covariance function $\xi_{\{A,B,C\},X}$ and coordinates as the example in Section 2.2.1.

Comparing Figure 2 of the Gaussian example and Figure 3 of the lognormal example there is a clear difference for this example's values for $\boldsymbol{\theta} = \{\mu, \sigma\}$. The lognormal field has more extreme values then the Gaussian random field. The exponential distribution causes higher values to increase relatively harder compared to lower values. Even if we try to rescale this random field we will not end up with a Gaussian random field as in Figure 2. Also because of the exponential transform; all the temperature values are positive.

3.2 Examples of well-known non-Gaussian random fields

In Section 3 it was assumed that the non-Gaussian random field was a pointwise transformation of a Gaussian random field. However there are also non-Gaussian random fields which can not be described by such a transformation. But luckily these can be simulated using other expressions, which use several independently generated Gaussian random fields. Three examples of such are shown: Gamma(m), Beta(m, n) and Student-t_v random fields. The first two are from inspiration of Vio et al. (2001). Also a simulated heat map will be provided for each random field as well as the steps to simulate them.

3.2.1 Simulation of a Gamma(m) random field

The first group of random fields are called Gamma random fields. The definition used in Vio et al. (2001) at equation (16) actually implies that a Gamma(m) random field can be simulated using numerous independently simulated Gaussian random fields.

Given a set of coordinates $\boldsymbol{t} \in V$, a *Gamma(m) random field* $(m \in \mathbb{N})$ is defined coordinate wise by the formula $G_m(\boldsymbol{t}) = \frac{1}{2}\sum_{i=1}^{2m} X_i(\boldsymbol{t})$, where $X_i(\boldsymbol{t}) \sim \mathcal{N}(0,1), \quad \forall \boldsymbol{t} \in V$ are independent with some covariance function $\xi_{\boldsymbol{\eta},X_i}$ which is the same for all X_i. Or, equivalently, in the multivariate case a Gamma(m) random field $\boldsymbol{G}_m = \sum_{i=1}^{2m} \boldsymbol{X}_i$, where $\boldsymbol{X}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma_\eta})$ are independently distributed. Note that the covariance function acts on the structure of the Gaussian random field. These independently sampled Gaussian random fields will generate the Gamma random field.

Since it is known how to simulate Gaussian random fields, the definition of a Gamma random field can be used to simulate Gamma random fields using some independently sampled Gaussian random fields. Here in Figure 4 an example of a Gamma(3) random field has been simulated.

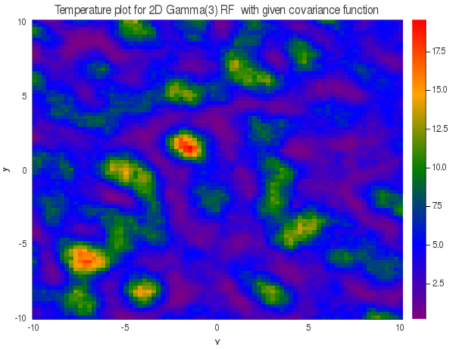


Figure 4: Simulated Gamma(3) random field for the covariance function $\xi_{\{A,B,C\},X}$, which acts on the Gaussian random field, and coordinates as the example in Section 2.2.1.

A noticable difference with the Gaussian and lognormal field in Figure 2 and 3 is that there seems to be a lot more cold spots in comparison to hot spots. This is an interesting characteristic for this random field since maybe this uneven proportion of hot an cold spots seem to fit the data of the CMB better.

3.2.2 Simulation of a Beta(m, n) random field

An even larger group of random fields called the Beta(m, n) random fields can be simulated using two independently generated Gamma(m) and Gamma(n) random fields, with the same covariance function. Again defined by the definition in Vio et al. (2001) using equation (21).

Given a set of coordinates $\boldsymbol{t} \in V$, a *Beta(m,n) random field* $(m, n \in \mathbb{N})$ is defined coordinate wise by the formula $B_{m,n}(\boldsymbol{t}) = \frac{G_m(\boldsymbol{t})}{G_m(\boldsymbol{t})+G_m(\boldsymbol{t})}$, where $G_m(\boldsymbol{t})$ and $G_n(\boldsymbol{t})$ are independent and generated by the same covariance function of the Gaussian random field $\xi_{\boldsymbol{\eta},X}$.
Or, equivalently, in the multivariate case a Beta(m, n) random field $\boldsymbol{B}_{m,n} = \frac{\boldsymbol{G}_m}{\boldsymbol{G}_m+\boldsymbol{G}_n}$ (pointwise division), where $\boldsymbol{G}_m$ and $\boldsymbol{G}_n$ are independently generated Gamma random fields.

Since in the previous section we have generated a Gamma random field, it is now trivial by the definition of a Beta(m, n) random field how to simulate such a field. In Figure 5 such a simulation ,where $m = 4$ and $n = 2$, has been created.

There seems to be a lot more hills and troughs in contrast to all the previously simulated random fields. There is a lot more variety in temperatures among all spots, which is an interesting characteristic.

3.2.3 Simulation of a Student-t_v random field

An other obvious interesting choice for a random field, because of its similar symmetric distribution as the Gaussian distribution, would be that temperatures $R(\boldsymbol{t}) \sim t_v,\ \ v \in \mathbb{R}_{>0}$. This is also because the Student-t_v distribution is, like the normal distribution, symmetrical as well; but it has thicker tails.

To take a sample T from a t_v-distribution one can generate it by calculating $T = Z \Big/ \sqrt{\frac{W_v}{v}}$, where $Z \sim \mathcal{N}(0, 1)$ and $W_v \sim \chi^2_v$. Using this fact a t_v random field can be defined it using a

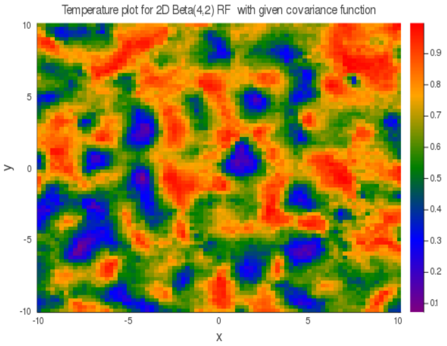


Figure 5: Simulated Beta(4,2) random field for the covariance function $\xi_{\{A,B,C\},X}$, which acts on the Gaussian random field, and coordinates as the example in Section 2.2.1.

single Gaussian random field.

Given a set of coordinates $\boldsymbol{t} \in V$ and a random variable $W \sim \chi_v^2$, a *Student-t_v random field* ($v \in \mathbb{R}_{>0}$) is defined coordinate wise by the formula $T_v(\boldsymbol{t}) = X(\boldsymbol{t}) \Big/ \sqrt{\frac{W}{v}}$, where $X(\boldsymbol{t})$ is generated by a Gaussian random field with some covariance function $\xi_{\boldsymbol{\eta},X}$.
Or, equivalently, in the multivariate case a t_v random field $\boldsymbol{T}_v = \boldsymbol{X} \Big/ \sqrt{\frac{W}{v}}$, where $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$.

From previous sections it is known how to simulate a Gaussian random field $\boldsymbol{X}$. The only extra thing for this random field is the generation of a single sample from χ_v^2 . Then a simulation of the Student t_v random field; for this example $v = 0.5$; is plotted in Figure 6.

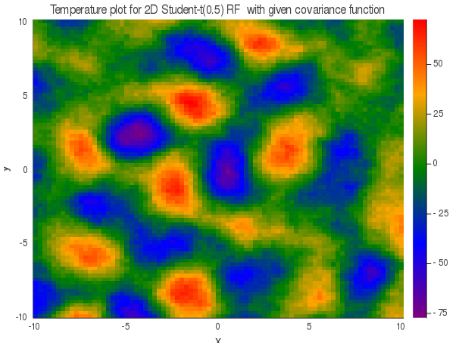


Figure 6: Simulated $t_{0.5}$ random field for the covariance function $\xi_{\{A,B,C\},X}$, which acts on the Gaussian random field, and coordinates as the example in Section 2.2.1.

As expected, in contrast with the Gaussian random field in Figure 2, the Student $t_{0.5}$ has a lot more extreme values in temperature.

In this section we have looked at several examples of random fields and the methods to simulate them. It seems plausible that some random field is the underlying structure of the CMB data, because of its similarities in structure. We see similar regions of cold and hot spots as in Figure 1. The next step is to create a model in order to express the observed data in random fields and an other additional factor. The examples of the random fields from previous sections could then be applied to the observed data.

4 Model of the first type: general random field with observation noise

In this section a first of two models for the observed data will be defined. In the previous sections random fields, which were introduced by Vio et al. (2001), have made an entry. These fields will function as an underlying structure of the first model that will be constructed.

4.1 Assumptions about the observed data of the first model

The observed data at a coordinate $\boldsymbol{t}$ will be denoted by the symbol $Y(\boldsymbol{t})$ and analogously the whole observed field by the vector $\boldsymbol{Y}$. Then, using the theory of random fields, the expression of the definition of the first model is not difficult.

We say the observed data $\boldsymbol{Y}$ is a *model of the first type*, if it can be written as

$$\boldsymbol{Y} = \boldsymbol{F} + \boldsymbol{\epsilon},$$

where $\boldsymbol{F}$ is any random field (Gaussian/non-Gaussian) and observation noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2 \boldsymbol{I})$. Or, equivalently, $\forall \boldsymbol{t} \in V$ it holds that $Y(\boldsymbol{t}) = F(\boldsymbol{t}) + \epsilon(\boldsymbol{t})$, where $F(\boldsymbol{t})$ is defined by some random field and independent $\epsilon(\boldsymbol{t}) \sim \mathcal{N}(0, \nu^2)$.

As suggested by Taylor et al. (2008) only one additional factor is added, namely the observation noise. Although in their research the random field has a different form, which will be discussed in an upcoming section. The additional noise factor is realistic since with every measure instrument, e.g. satellite, the observed value always differs from the real value. Note that the parameter ν^2 explains how significant this noise really is. The choice for letting the noise be independently Gaussian is because it is a simple and natural choice when working with physical phenomena like the CMB. The smaller ν^2 the more accurate the observation is. The underlying structure for the model is thus the random field $\boldsymbol{F}$.

This additional observation noise $\boldsymbol{\epsilon}$ does make the temperature distribution of the observed data $\boldsymbol{Y}$ more difficult. Since now two (possibly different) independent multivariate distributions are added.

4.2 First model with a Gaussian random field

For the trivial case if the observed data is described by a Gaussian random field, the expression of $\boldsymbol{Y}$ is again a Gaussian distribution. Because if $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$ is a Gaussian random field with covariance function $\xi_{\boldsymbol{\eta},X}$, then

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}} + \nu^2 \boldsymbol{I}). \tag{4}$$

This is because of the linearity of the Gaussian distribution. Now comparing the difference between the observed data and the random field, when still using the same example as in section 2.2.1, the plot in Figure 7 arises. Here the noise factor has the value $\nu^2 = 0.5^2$.

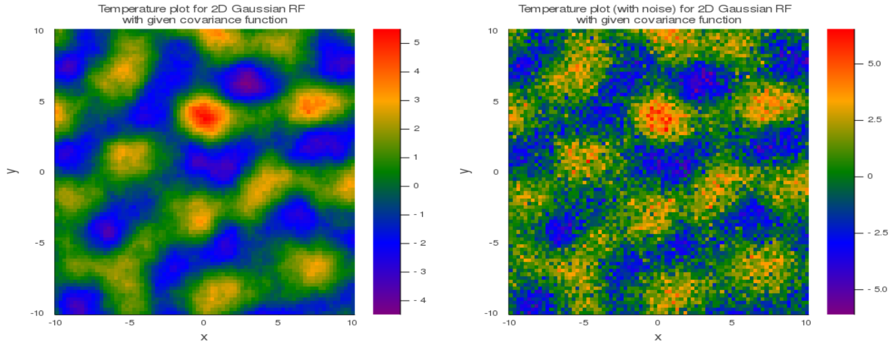


Figure 7: Simulated Gaussian random field for the example in section 2.2.1 with noise (right) and without noise (left).With $\nu^2 = 0.5^2$

4.3 First model with a mapped, non-Gaussian random field

If the observed data in the first model has a random field defined by $\boldsymbol{R} = g_{\boldsymbol{\theta}}(\boldsymbol{X})$, a transformation of a Gaussian random field $\boldsymbol{X}$ a, then the expression for the distribution of $\boldsymbol{Y}$ becomes more difficult. Luckily the two added distributions are independent which means that the probability density function of $\boldsymbol{Y}$ can be expressed as an convolution integral.

Also introducing notation, used in Bishop (2006), for the joint probability density function of the multivariate normal distribution:

$$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}}) = \frac{1}{(2\pi)^{N/2} \det(\boldsymbol{\Sigma}_{\boldsymbol{\eta}})^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_{\boldsymbol{\eta}}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right) \tag{5}$$

Theorem 4.3.1. *For the first model if the observed data has the form* $\boldsymbol{Y} = g_{\boldsymbol{\theta}}(\boldsymbol{X}) + \boldsymbol{\epsilon}$ *for some mapping* $g_{\boldsymbol{\theta}}[\cdot]$ *with parameters* $\boldsymbol{\theta}$*, then the probability density function of* $\boldsymbol{Y}$ *is equal to:*

$$p_{\boldsymbol{Y}|\boldsymbol{\eta},\boldsymbol{\theta},\nu^2}(\boldsymbol{y}) = \int_{\mathbb{R}^N} \mathcal{N}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r}) \mid \boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}}) \det[\mathcal{J}(\boldsymbol{r} \mid \boldsymbol{\theta})] \mathcal{N}(\boldsymbol{y} - \boldsymbol{r} \mid \boldsymbol{0}, \nu^2 \boldsymbol{I}) d\boldsymbol{x}, \tag{6}$$

where $\mathcal{J}(\boldsymbol{R} \mid \boldsymbol{\theta}) = \left[\frac{\partial g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r})}{\partial \boldsymbol{r}} \right]_{\boldsymbol{r}=\boldsymbol{R}|\boldsymbol{\theta}}$ *and* $d\boldsymbol{x} = dg_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r})$*.*

Proof. Recall that if $Z = X + Y$ and X, Y are independent we have that for the probability density function of Z:

$$p_Z(z) = \int_{-\infty}^{\infty} p_X(z - y) p_Y(y) dy,$$

which will also hold for the multivariate case as we have here. If $\boldsymbol{Y} \in \mathbb{R}^N$, we have $\boldsymbol{Y} = \boldsymbol{R} + \boldsymbol{\epsilon} = g_{\boldsymbol{\theta}}(\boldsymbol{X}) + \boldsymbol{\epsilon}$, $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$. So because of independence convolution be be applied for this data:

$$p_{\boldsymbol{Y}|\boldsymbol{\eta},\boldsymbol{\theta},\nu^2}(\boldsymbol{y}) = \int_{\mathbb{R}^N} p_{\boldsymbol{R}|\boldsymbol{\eta},\boldsymbol{\theta}}(\boldsymbol{r}) \mathcal{N}(\boldsymbol{y} - \boldsymbol{r} \mid \boldsymbol{0}, \nu^2 \boldsymbol{I}) d\boldsymbol{r} \tag{7}$$

Now considering the function $p_{\boldsymbol{R}|\boldsymbol{\theta}}(\boldsymbol{r})$. When looking at the cumulative density function $F_{\boldsymbol{R}|\boldsymbol{\theta}}$:

$$F_{\boldsymbol{R}|\boldsymbol{\eta},\boldsymbol{\theta}}(\boldsymbol{r}) = \mathbb{P}(\boldsymbol{R} \leq \boldsymbol{r} | \boldsymbol{\eta}, \boldsymbol{\theta}) = \mathbb{P}(g_{\boldsymbol{\theta}}(\boldsymbol{X}) \leq \boldsymbol{r} \mid \boldsymbol{\eta}, \boldsymbol{\theta}) = \mathbb{P}(\boldsymbol{X} \leq g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r}) \mid \boldsymbol{\eta}, \boldsymbol{\theta})$$

where $g_{\boldsymbol{\theta}}^{-1}[.]$ is the inverse of the mapping which we assume exists and that it is differentiable. The probability $\mathbb{P}(\boldsymbol{R} \leq \boldsymbol{r})$ means that $\forall i = 1, ...N$ $, R_i \leq r_i$ (element wise). Note that $\boldsymbol{X} \sim$

$\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{\eta}})$. Using the N-th derivative of the joint cumulative density function of $\boldsymbol{R}$, we can obtain $p_{\boldsymbol{R}|\boldsymbol{\theta}}(\boldsymbol{r})$ using the following expression:

$$p_{\boldsymbol{R}|\boldsymbol{\theta}}(\boldsymbol{r}) = \frac{\partial^N F_{\boldsymbol{R}|\boldsymbol{\eta},\boldsymbol{\theta}}}{\partial r_1 \cdots \partial r_N}(\boldsymbol{r}) = \frac{\partial^N \mathbb{P}(\boldsymbol{X} \leq g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r})|\boldsymbol{\eta},\boldsymbol{\theta})}{\partial r_1 \cdots \partial r_N} \tag{8}$$

Now because $g_{\boldsymbol{\theta}} : \mathbb{R} \to \mathbb{R}$ the expression becomes much simpler because of its easy Jacobian matrix. The Jacobian is a diagonal matrix and hence its inverse aswell. The expression becomes the following by using the chain rule;

$$\begin{aligned} p_{\boldsymbol{R}|\boldsymbol{\eta},\boldsymbol{\theta}}(\boldsymbol{r}) &= \frac{\partial^N \mathbb{P}(\boldsymbol{X} \leq g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r})|\boldsymbol{\eta},\boldsymbol{\theta})}{\partial g_{\boldsymbol{\theta}}^{-1}(r_1)\cdots\partial g_{\boldsymbol{\theta}}^{-1}(r_N)} \times \frac{\partial g_{\boldsymbol{\theta}}^{-1}(r_1)\cdots\partial g_{\boldsymbol{\theta}}^{-1}(r_N)}{\partial r_1 \cdots \partial r_N} \\ &= \frac{\partial^N F_{\boldsymbol{X}|\boldsymbol{\eta},\boldsymbol{\theta}}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r}))}{\partial g_{\boldsymbol{\theta}}^{-1}(r_1)\cdots\partial g_{\boldsymbol{\theta}}^{-1}(r_N)} \times \frac{\partial g_{\boldsymbol{\theta}}^{-1}(r_1)\cdots\partial g_{\boldsymbol{\theta}}^{-1}(r_N)}{\partial r_1 \cdots \partial r_N} \\ &= p_{\boldsymbol{X}|\boldsymbol{\eta},\boldsymbol{\theta}}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r}))\prod_{i=1}^{N}(g_{\boldsymbol{\theta}}^{-1})'(r_i) := p_{\boldsymbol{X}|\boldsymbol{\eta},\boldsymbol{\theta}}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r}))\det[\mathcal{J}(\boldsymbol{r} \mid \boldsymbol{\theta})], \end{aligned}$$

where $\mathcal{J}(\boldsymbol{R} \mid \boldsymbol{\theta}) = \left[\frac{\partial g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r})}{\partial \boldsymbol{r}}\right]_{\boldsymbol{r}=\boldsymbol{R}|\boldsymbol{\theta}}$ denotes the Jacobian matrix of the inverse mapping $g_{\boldsymbol{\theta}}^{-1}$ at a point $\boldsymbol{R}$ given mapping parameters $\boldsymbol{\theta}$.

Then because the joint probability density function of $\boldsymbol{X}$ given $\boldsymbol{\eta}$ is known the final expression will be

$$p_{\boldsymbol{R}|\boldsymbol{\eta},\boldsymbol{\theta}}(\boldsymbol{r}) = \mathcal{N}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r}) \mid \mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{\eta}})\det[\mathcal{J}(\boldsymbol{r} \mid \boldsymbol{\theta})] \tag{9}$$

then combining (7) and (9) the joint probability function of the data $\boldsymbol{Y}$, given $\boldsymbol{\eta}, \boldsymbol{\theta}$ and ν^2, for a transformation $g_{\boldsymbol{\theta}}[.]$ is

$$p_{\boldsymbol{Y}|\boldsymbol{\eta},\boldsymbol{\theta},\nu^2}(\boldsymbol{y}) = \int_{\mathbb{R}^N} \mathcal{N}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r}) \mid \mathbf{0}, \mathbf{\Sigma}_{\boldsymbol{\eta}})\det[\mathcal{J}(\boldsymbol{r} \mid \boldsymbol{\theta})]\mathcal{N}(\boldsymbol{y} - \boldsymbol{r} \mid \mathbf{0}, \nu^2\boldsymbol{I})d\boldsymbol{x} \tag{10}$$

where $d\boldsymbol{x} = dg_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r})$.

□

For a general mapping of a Gaussian random field the joint probability density function is really complex. It consists of a N-dimensional integral which needs to be solved numerically. This will take a lot of time to compute as N increases. Still it is possible to simulate such a non-Gaussian random field which is mapped. In Section 3.2 a lognormal random field was simulated, which is a transformed Gaussian random field as well. Now for comparison in Figure 8 , for $\nu^2 = 0.5^2$, both the observed data and the random field is simulated for a lognormal random field.

4.4 First model with other non-Gaussian random fields

The last case of the first model is that the random field is non-Gaussian, but can not be expressed in a closed form of a transformation. These examples have been shown in Section 3.2 and include the Gamma(m), Beta(m, n) and Student-t_v random field. For the Student-t_v (Bishop (2006)) the joint probability density function is known:

$$\mathcal{ST}(\boldsymbol{x}|\boldsymbol{\mu}, \mathbf{\Sigma}_{\boldsymbol{\eta}}, v) = \frac{\Gamma(N/2 + v/2)}{\Gamma(v/2)(\pi v)^{N/2}\det(\mathbf{\Sigma}_{\boldsymbol{\eta}})^{1/2}}\left[1 + \frac{1}{v}(\boldsymbol{x} - \boldsymbol{\mu})^T\mathbf{\Sigma}_{\boldsymbol{\eta}}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right]^{-N/2-v/2}, \tag{11}$$

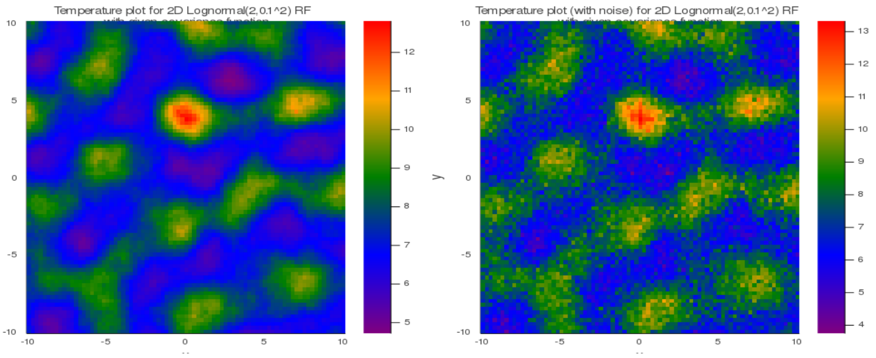


Figure 8: Simulated lognormal random field for the example in section 2.2.1 with noise (right) and without noise (left). With $\nu^2 = 0.5^2$

where $\Gamma(s) = \int_0^\infty x^s e^{-s} dx$ is the Gamma-function.

Now considering a non-Gaussian random field where $\boldsymbol{R} \sim t_d(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$. Again assuming that the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2 \boldsymbol{I}_N)$. This results in the model

$$\boldsymbol{Y} = \boldsymbol{R} + \boldsymbol{\epsilon} = t_v(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}}) + \mathcal{N}(\boldsymbol{0}, \nu^2 \boldsymbol{I}_N)$$

which is the sum of two independent N-dimensional multivariate independent random variables. Using convolution this results in the joint probability density function for the observed data $\boldsymbol{Y}$ with known Gaussian covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ constructed by the Gaussian covariance function $\xi_{\boldsymbol{\eta},X}$:

$$p_{\boldsymbol{Y}|\boldsymbol{\eta},v,\nu^2}(\boldsymbol{y}) = \int_{\mathbb{R}^N} \mathcal{ST}(\boldsymbol{r}|\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}}, v) \mathcal{N}(\boldsymbol{y} - \boldsymbol{r}|\boldsymbol{0}, \nu^2 \boldsymbol{I}) d\boldsymbol{r} \tag{12}$$

Note that $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ is the covariance of the Gaussian random field $\boldsymbol{X}$. It is not the covariance matrix of the $t_v(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$ random field, because this matrix is defined as $\text{Cov}(\boldsymbol{R}) = \frac{v}{v-2}\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ (Bishop (2006)). Here in Figure 9 both the Student-t_v random field, from Figure 6, and the random field with added noise are plotted. In the corresponding figure the noise factor $\nu^2 = 5^2$.

For the other two cases, a joint probability density function has not been found, or simply does not exist in an explicit form. Hence such a density function can not be written explicitly. Vio et al. (2001) does calculate the moments and covariance/correlation function of both the Gamma(m) and Beta(m, n) random fields in relation to the covariance of the Gaussian random field, but this will not be discussed in this paper. This does not mean that it is impossible to simulate the data for these random fields. These simulations for the observed data for the Gamma(3) and Beta$(4, 2)$ random field are respectively shown in Figure 10 and 11. Here for Gamma(3) we have $\nu^2 = 2.5^2$ and for Beta(4,2) we take $\nu^2 = 0.5^2$. We use different noise factors, because we want to see a noticeable difference between the random field and its corresponding observed data, which includes the observation noise.

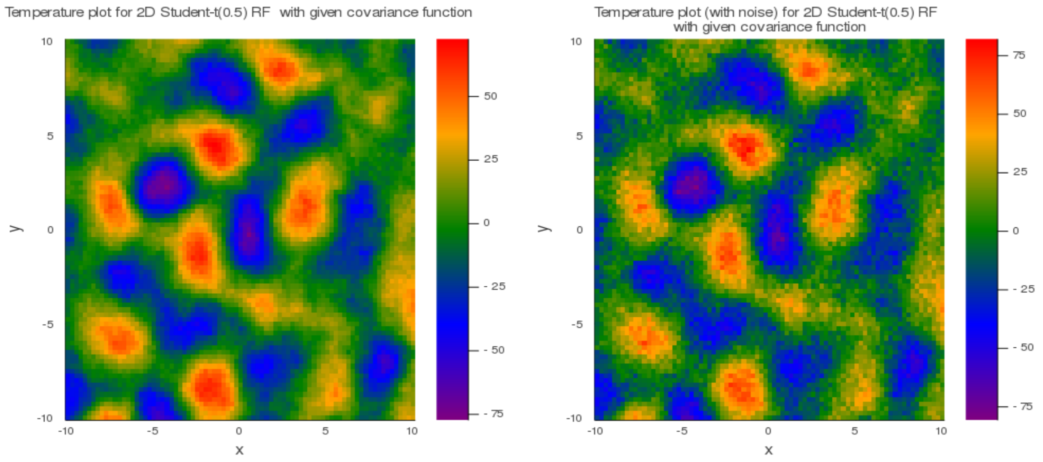


Figure 9: Simulated Student-$t_{0.5}$ random field for the example in Section 2.2.1 with noise (right) and without noise (left). With $\nu^2 = 5^2$

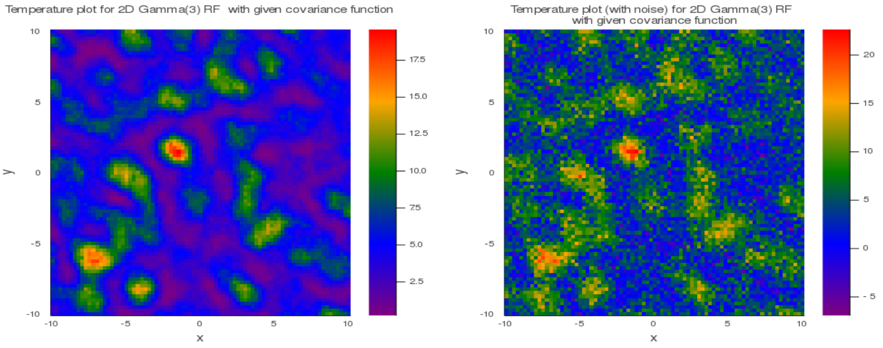


Figure 10: Simulated Gamma(3) random field for the example in Section 2.2.1 with noise (right) and without noise (left). With $\nu^2 = 2.5^2$

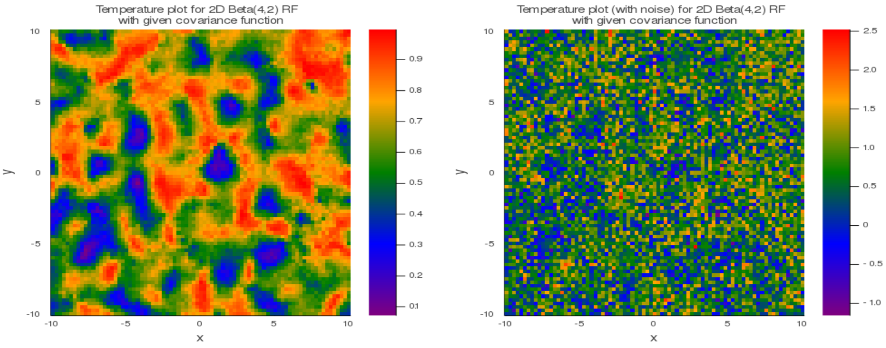


Figure 11: Simulated Beta(4, 2) random field for the example in Section 2.2.1 with noise (right) and without noise (left).With $\nu^2 = 0.5^2$

.

5 Model of the second type: random field using an eigenfunction representation with random coefficients and observation noise

The model introduced in this section has similarities to the first model. Similarly, the only additional factor to the underlying structure will be the observed noise. The only difference will be that there is an assumption made about the form of the random field. It will be decomposed in a sum of eigenfunctions, whose coefficients will be randomly distributed, in addition with some covariance structure among the coefficients. This model is first encountered in Taylor et al. (2008) and will be more generalised in this paper, since in Taylor et al. (2008) only a Gaussian case is assumed of the model we will define.

5.1 Assumptions about the observed data of the second model

The model will still assume that the data has the form $\boldsymbol{Y} = \boldsymbol{F} + \boldsymbol{\epsilon}$ for a general random field $\boldsymbol{F}$. Here it is assumed that $\boldsymbol{F}$ can be written as a sum of eigenfunctions with random coefficients with, i.e.

$$F(\boldsymbol{t}) = \sum_{i=1}^{n} c_i e_i(\boldsymbol{t}), \ \ \forall \boldsymbol{t} \in V = \{\boldsymbol{t}_1, ..., \boldsymbol{t}_N\} \tag{13}$$

where $e_i(\boldsymbol{t})$ is an eigenfunction evaluated at the coordinate $\boldsymbol{t} \in V$. For $i = 1, ..., n$ the coefficients c_i are random variables with some covariance structure $\mathrm{Cov}(c_i, c_j) = \zeta_{c,\boldsymbol{\gamma}}(i, j)$, where $\boldsymbol{\gamma}$ are function parameters. The multivariate notation is

$$\boldsymbol{F} = \boldsymbol{E}\boldsymbol{c},$$

where $\boldsymbol{E}$, the eigenfunction matrix, is a matrix with a row for all the coordinates and a column for all the eigenfunctions:

$$\boldsymbol{E} = \begin{bmatrix} e_1(t_1) & e_2(t_1) & \cdots & e_n(t_1)) \\ e_1(t_2) & e_2(t_2) & \cdots & e_n(t_2)) \\ \vdots & \vdots & & \vdots \\ e_1(t_N) & e_2(t_N) & \cdots & e_n(t_N)) \end{bmatrix} \in \mathbb{R}^{N \times n}, \quad \boldsymbol{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \in \mathbb{R}^n. \tag{14}$$

Here $\boldsymbol{c}$ has a multivariate distribution which depends on the covariance function $\zeta_{c,\boldsymbol{\gamma}}(\cdot,\cdot)$.

We say that the observed data $\boldsymbol{Y}$ is a *model of the second type*, if it can be written as

$$\boldsymbol{Y} = \boldsymbol{F} + \boldsymbol{\epsilon} = \boldsymbol{E}\boldsymbol{c} + \boldsymbol{\epsilon}$$

where $\boldsymbol{F}$ is any random field (Gaussian/non-Gaussian) and observation noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2\boldsymbol{I})$. The eigenfunction matrix $\boldsymbol{E}$ and vector of random coefficients $\boldsymbol{c}$ are defined in (14).

In the upcoming subsections two cases will be discussed for different kind of distributions for the random coefficients $\boldsymbol{c}$. Note that this model is also used in Taylor et al. (2008) whom use in their specific case spherical harmonics as eigenfunctions and a normal distribution for the random coefficients.

5.2 Second model with Gaussian eigenfunction coefficients

Still using the letter $\boldsymbol{X}$ to denote a Gaussian random field, the definition for the second model will look like $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{\epsilon}$. The only difference is the assumed form of $\boldsymbol{X}$. It will be expressed as a sum of eigenfunctions multiplied with random variables as coefficients, i.e. :

$$X(\boldsymbol{t}) = \sum_{i=1}^{n} a_i e_i(\boldsymbol{t}), \;\; \forall \boldsymbol{t} \in V = \{\boldsymbol{t}_1, ..., \boldsymbol{t}_N\}. \tag{15}$$

Now writing $\boldsymbol{X} = \boldsymbol{Ea}$ where the random vector $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_\gamma})$. Note that the use of $\boldsymbol{a}$ is done to prevent later confusion in Gaussian and non-Gaussian examples. Since $\boldsymbol{a}$ is normally distributed, also $\boldsymbol{X} = \boldsymbol{Ea}$ has a normal distribution, namely $\boldsymbol{X} = \boldsymbol{Ea} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{EC_\gamma E}^T)$. Implying , since $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2 \boldsymbol{I})$, that the observed data $\boldsymbol{Y}$ has the distribution

$$\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{EC_\gamma E}^T + \nu^2 \boldsymbol{I}).$$

Hence the data $\boldsymbol{Y}$ also has an explicit expression for the probability density function, which will be helpful for parameter estimation later on.

To show how such a simulation will look like a couple of assumptions have to be made about the existing model. First the field consists of the grid $(-L : \Delta x : L)$ in the x-direction and $(-M : \Delta y : M)$ in the y-direction. For the example we take $L = M = 10$ and $\Delta x = \Delta y = 0.25$. We will assume that:

$$X(\boldsymbol{t}_j) = \sum_{l=1}^{5} \sum_{m=1}^{5} a_{lm} e^{-i\pi(lx_j/L + my_j/M)}, \;\; \forall j = 1, ..., N \;\; \boldsymbol{t_j} = (x_j, y_j) \tag{16}$$

Here the eigenfunction $e_{lm}(\boldsymbol{t}_j) = e^{-i\pi(lx_j/L + my_j/M)}, \;\; l, m = 1, ..., 5$. This means that the number of eigenfunctions is $n = 25$. The coefficients assume the normal distribution $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_\gamma})$ where the matrix elements have the value

$$C_{lml'm',\boldsymbol{\gamma}} = \zeta_{a,\boldsymbol{\gamma}}((l,m),(l',m')) = \gamma_{lm} \delta_{ll'} \delta_{mm'}$$

where the vector $\boldsymbol{\gamma}$ consists of coefficients $\gamma_{lm} = \frac{1}{lm}$ and $\delta_{ij} = 1_{\{i=j\}}$ is the Kronecker-delta function. Note that this will imply that $\boldsymbol{C_\gamma}$ is a diagonal matrix.

Simulating coefficients $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_\gamma})$, noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2 \boldsymbol{I})$, where $\nu^2 = 1^2$, and constructing $\boldsymbol{E}$ using (14) is self-explanatory. Then using these two generated vectors and the matrix to simulate the observed data $\boldsymbol{Y} = \boldsymbol{Ea} + \boldsymbol{\epsilon}$, gives the following sample in Figure 12. As usual comparing the observed data with its underlying structure (random field).

5.3 Second model with mapped, non-Gaussian eigenfunction coefficients

The second model of random fields for Gaussian coefficients can be extended to non-Gaussian examples. In a similar fashion as in Section 3.1 the Gaussian coefficients $\boldsymbol{a}$ can be transformed to non-Gaussian coefficients using a mapping $g_{\boldsymbol{\theta}}[\cdot]$, with parameters $\boldsymbol{\theta}$. Non-Gaussian distributed eigenfunction coefficients; denoted by the bold letter $\boldsymbol{b}$; will in this section be defined by a transformation. This means that the expression $\boldsymbol{b} = g_{\boldsymbol{\theta}}[\boldsymbol{a}]$, where $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_\gamma})$ and the Gaussian covariance matrix $\boldsymbol{C_\gamma}$ is constructed using some Gaussian covariance function $\zeta_{a,\boldsymbol{\gamma}}(\cdot,\cdot)$ with parameters $\boldsymbol{\gamma}$.

The example for this section is again a lognormal example. The mapping $g_{\boldsymbol{\theta}}[a] = e^{\mu + \sigma a}$ creates lognormal coefficients when $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_\gamma})$. Hence we get that $b_i = e^{\mu + \sigma a_i}, \;\; \forall i = 1, ..., n$; or shortly written as $\boldsymbol{b} = e^{\mu + \sigma \boldsymbol{a}}$ (pointwise addition and exponentiation). We use the same example as in Section 5.2. Additional for the mapping in this example the parameters $\mu = -1$ and $\sigma = 0.1$. Then simulating the random field presents the heat maps in Figure 13. Here the noise parameter $\nu^2 = 1^2$.

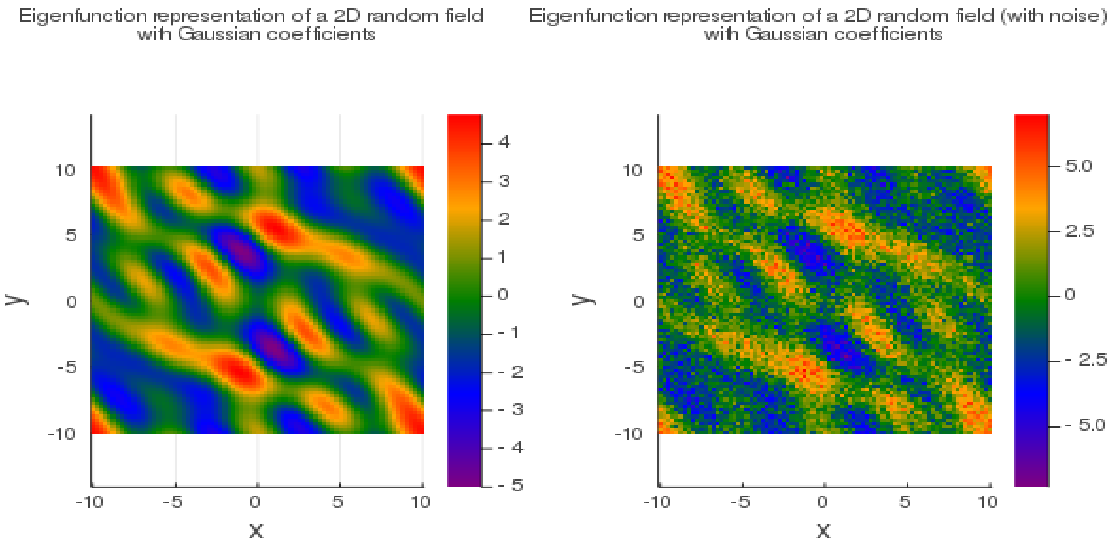


Figure 12: Simulated random field with Gaussian eigenfunction coefficients. With noise (right) and without noise (left). With $\nu^2 = 1^2$

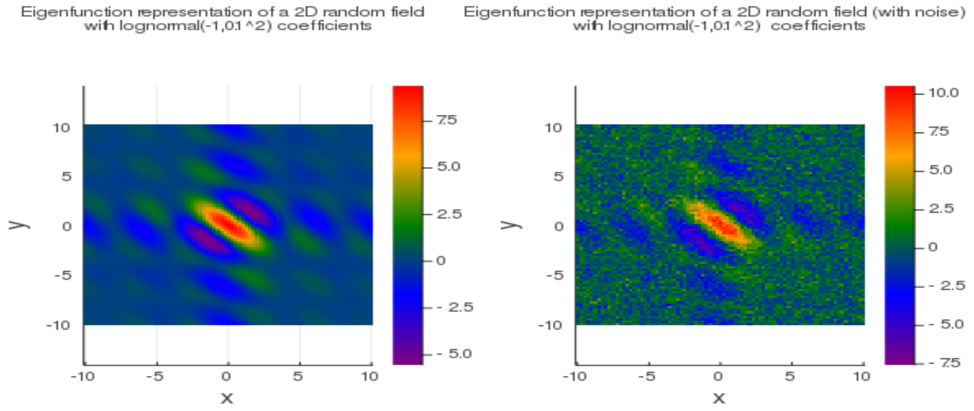


Figure 13: Simulated random field with lognormal distributed eigenfunction coefficients. With noise (right) and without noise (left). With $\nu^2 = 1^2$

5.4 Second model with other non-Gaussian eigenfunction coefficients

Similar to the coefficients in previous subsection, these coefficients will also have a non-Gaussian distribution. In this case the mapping will be unknown, or maybe not possible to write explicitly. Again denoted by the bold letter $\boldsymbol{b}$ it will have a non-Gaussian distribution $\boldsymbol{b} \sim \mathcal{M}_{\boldsymbol{\alpha},\boldsymbol{\gamma}}$ with parameters $\boldsymbol{\alpha}$ and covariance parameters $\boldsymbol{\gamma}$. The covariance matrix is then described by the function $\zeta_{b,\boldsymbol{\gamma}}(\cdot,\cdot)$.

For this section we take the example of Student-t_v distributed random coefficients. This means that $\boldsymbol{b} \sim t_v(\boldsymbol{0}, \boldsymbol{C}_\gamma)$. We will use that $v = 4$ and use the same matrix form $\boldsymbol{C}_\gamma$ as in all previous subsections of the second model. The noise factor $\nu^2 = 2.5^2$. Then a simulation of this model will be painted in Figure 14.

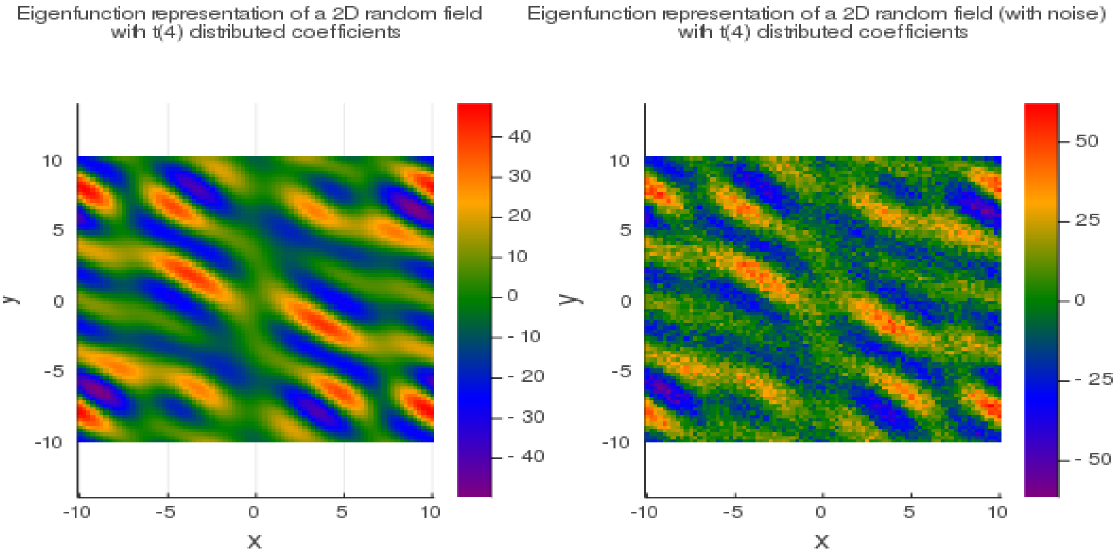


Figure 14: Simulated random field with Student-t_4 distributed eigenfunction coefficients. With noise (right) and without noise (left). With $\nu^2 = 2.5^2$

The way the t_v random coefficients $\boldsymbol{b}$ are generated is by using the Gaussian coefficients $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_\gamma)$ and use that $\boldsymbol{b} = \boldsymbol{a} \Big/ \sqrt{\frac{W}{v}} \sim t_v(\boldsymbol{0}, \boldsymbol{C}_\gamma)$, where $W \sim \chi^2_v$. Note that the covariance matrix $\boldsymbol{C}_\gamma$ is the covariance matrix of the Gaussian distributed eigenfunction vector $\boldsymbol{a}$, which is established using the Gaussian covariance function $\zeta_{a,\gamma}(\cdot, \cdot)$. The covariance matrix of the vector $\boldsymbol{b}$ is then $\text{Cov}(\boldsymbol{b}) = \frac{v}{v-2}\boldsymbol{C}_\gamma$, using Bishop (2006).

6 Parameter estimation of the CMB models using maximum likelihood estimation (MLE)

Now that the assumptions about the observed data of the Cosmic Microwave Background have been made, the parameters of the model have to be tuned such that it creates the best fit for the observed data. This can be applied to both models that were introduced as the first and the second type.

The data we want to fit a model on is the observed CMB data which is represented with the letter $\boldsymbol{Y}$. We will assume that there are K observations of the multivariate data $\boldsymbol{Y}$ (do not all have to be the same size). This can be interpreted as a set of snapshots of the CMB map. The set of observations is denoted by the symbol $\mathcal{Y} = \{\boldsymbol{Y_1}, \boldsymbol{Y_2}, ..., \boldsymbol{Y}_K\}$ with their corresponding coordinate set $\mathcal{V} = \{V_1, V_2, ..., V_K\}$, with the number of coordinates of observation i is $|V_i| = N_i$. The reason that we do not assume to only have one big observation is that this is also included in the general form of K observations. Simply take $K = 1$ and take $\boldsymbol{Y}$, for example, as the whole CMB map in Figure 1 .

Maximum likelihood estimation (MLE) is a method to optimise the parameters of a model given a set of observations. For reference check Bijma et al. (2017) (Chapter 3.3). It works by calculating the maximum of a objective function called the *log-likelihood function.* This log-likelihood function is defined by:

$$\boldsymbol{\rho} \mapsto \mathcal{L}(\boldsymbol{\rho}; \mathcal{Y}) = \prod_{i=1}^{K} p_{\boldsymbol{Y}|\boldsymbol{\rho}}(\boldsymbol{Y}_i) \tag{17}$$

where $\boldsymbol{\rho}$ are all the parameters in the model (e.g. $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\eta}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \nu^2$) and $p_{\boldsymbol{Y}|\boldsymbol{\rho}}$ is the joint probability density function of $\boldsymbol{Y}$ given all the parameters of the model $\boldsymbol{\rho}$. The parameters $\hat{\boldsymbol{\rho}}$ which maximise function (17) are then called the maximum likelihood estimates. Of course since there are two distinct models, two different log-likelihood functions arise, because of the difference in parameters and joint probability density functions.

In this section and the next one is also shown that the use of MLE is inferior to an other parameter estimation method. That is why, in this section, only is shown how MLE should be applied and why it is not convenient for these examples.

6.1 Parameter estimation of the first model using MLE

For the first model of the observed data it is shown, for the three cases, that MLE is not really convenient for parameter estimation. Also for all three the cases it is assumed the reader has made an assumption about the form of the covariance function $\xi_{\boldsymbol{\eta},X}$, which will impact the structure of all $\boldsymbol{\Sigma}_{\boldsymbol{\eta},i},\ \ i = 1, ..., K$. Here $\boldsymbol{\Sigma}_{\boldsymbol{\eta},i}$ is the Gaussian covariance matrix constructed using a Gaussian covariance function $\xi_{\boldsymbol{\eta},X}$ with parameters $\boldsymbol{\eta}$ given the random field $\boldsymbol{X}_i$. The size of the matrix depends on the number of coordinates in the random field.

6.1.1 MLE on Gaussian random field

The log-likelihood function for this example, using its joint pdf in (4), is defined as:

$$(\boldsymbol{\eta}, \nu^2) \mapsto \mathcal{L}(\boldsymbol{\eta}, \nu^2; \mathcal{Y}) = \prod_{i=1}^{K} \mathcal{N}_{N_i}(\boldsymbol{Y}_i \mid \boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta},i} + \nu^2 \boldsymbol{I}) \tag{18}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\eta},i}$ is the covariance matrix of $\boldsymbol{Y}_i$, $\forall i = 1, ..., K$. Note that these matrices can have different sizes because of the different sizes of $\boldsymbol{Y}_i$. Analytically it is impossible to solve a maximization problem like this one, since there are parameters inside the matrix. Also the joint pdf of the multivariate normal (5) includes its determinant (depending on variable $\boldsymbol{\eta}$), which creates another level of complexity to this problem. Computationally this problem becomes more exhausting as the total size of $\mathcal{Y}$ increases. There does exist an algorithm which can calculate the maximum likelihood estimates for this specific example. This algorithm is called the EM algorithm (Expectation-Maximization algorithm). This is an iterative process which can be used to calculate the estimates for MLE (Bijma et al. (2017), Chapter 3.3.2). We will not use this algorithm for an example since our focus will be on the Bayesian parameter estimation, which will come along in Section 7.

6.1.2 MLE on mapped, non-Gaussian random field

The probability density function of the observed data given a mapped non-Gaussian random field is even worse since it contains an N-dimensional integral (7). Also there are additional parameters $\boldsymbol{\theta}$ of the mapping. This means the log-likelihood function will look like

$$(\boldsymbol{\eta}, \boldsymbol{\theta}, \nu^2) \mapsto \mathcal{L}(\boldsymbol{\eta}, \boldsymbol{\theta}, \nu^2; \mathcal{Y}) = \prod_{i=1}^{K} \int_{\mathbb{R}^{N_i}} \mathcal{N}_{N_i}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r}) \mid \mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta},i}) \det[\mathcal{J}_{N_i}(\boldsymbol{r} \mid \boldsymbol{\theta})] \mathcal{N}_{N_i}(\boldsymbol{y} - \boldsymbol{r} \mid \mathbf{0}, \nu^2 \boldsymbol{I})\, d\boldsymbol{x} \tag{19}$$

$$= \prod_{i=1}^{K} \int_{\mathbb{R}^{N_i}} \mathcal{N}_{N_i}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r}) \mid \mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta},i}) \left[\prod_{i=1}^{N_i} (g_{\boldsymbol{\theta}}^{-1})'(r_i) \right] \mathcal{N}_{N_i}(\boldsymbol{y} - \boldsymbol{r} \mid \mathbf{0}, \nu^2 \boldsymbol{I})\, d\boldsymbol{x}. \tag{20}$$

Now the log-likelihood function consists of multidimensional integrals, including the calculation of determinants which will drastically increase the computational time of the calculation of the maximum likelihood estimates.

6.1.3 MLE on an other example of a non-Gaussian random field

An example of a different non-Gaussian random field, not represented by a mapping $g_{\boldsymbol{\theta}}[\cdot]$, is the Student-$t_v$ random field. Applying MLE to calculate the parameters (now including the degrees of freedom v) using (12), we acquire the following log-likelhood function:

$$(\boldsymbol{\eta}, v, \nu^2) \mapsto \mathcal{L}(\boldsymbol{\eta}, v, \nu^2; \mathcal{Y}) = \prod_{i=1}^{K} \int_{\mathbb{R}^{N_i}} \mathcal{ST}_{N_i}(\boldsymbol{r} | \mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta},i}, v) \mathcal{N}_{N_i}(\boldsymbol{y} - \boldsymbol{r} | \mathbf{0}, \nu^2 \boldsymbol{I})\, d\boldsymbol{r} \tag{21}$$

Again like previous the case, the function consists of multiple higher dimensional integral and determinants which will hinder the efficiency of the maximization problem.

6.2 Parameter estimation of the second model using MLE

Similar to the MLE in the previous subsection about the first model, one needs the density functions for all the cases of the second model in order to perform Maximum likelihood estimation.

6.2.1 MLE on model with Gaussian eigenfunction coefficients

The first and most trivial example is again the model with Gaussian distributed eigenfunction coefficients. The log-likelihood function will look like:

$$(\gamma, \nu^2) \mapsto \mathcal{L}(\gamma, \nu^2; \mathcal{Y}) = \prod_{i=1}^{K} \mathcal{N}_{N_i}(\boldsymbol{Y}_i \mid \boldsymbol{0}, \boldsymbol{E}\boldsymbol{C}_{\boldsymbol{\gamma}}\boldsymbol{E}^T + \nu^2\boldsymbol{I}). \tag{22}$$

The log-likelihood function in equation (22) is valid, but it does not contain the parameters $\boldsymbol{a}$. This is because they are integrated out of the density. This simplifies the equation, but it eliminates the parameters that we wanted to estimate using MLE. Hence if there is no interest in the parameter $\boldsymbol{a}$, the log-likelihood function (22) can be used. The same difficulties arise as in Section 6.1.1. Again the unknown parameters are inside the matrix, which increase the difficulty in maximising the function. Also making it impossible to solve analytically. However it is possible to use an EM algorithm to obtain the maximum likelihood estimates.

To calculate the log-likelihood function for the parameters $(\boldsymbol{a}, \boldsymbol{\gamma}, \nu^2)$ we need to find a joint probability density function where the parameters $\boldsymbol{a}$ are present. This is a difficult task. We will not be able to find the joint density for this example, but we will intuitively explain what it will look like. The difficulty comes with the matrix $\boldsymbol{E} \in \mathbb{R}^{N \times n}$, which will not be a square matrix. Note that the rank of this matrix is at most n. Because the columns consists of eigenfunctions which are pairwise orthogonal we can say all of these column vectors are independent. This means that $\text{Rank}(\boldsymbol{E}) = n$. The column space of the matrix will thus be a subset of the whole $\mathbb{R}^N$, i.e. $\text{Col}(\boldsymbol{E}) \subset \mathbb{R}^N$. So the product $\boldsymbol{Ea}$ can not get to every vector in $\mathbb{R}^N$, since it is always in the column space. Since the observed data is defined by $\boldsymbol{Y} = \boldsymbol{Ea} + \boldsymbol{\epsilon}$, the product $\boldsymbol{Ea}$ will function as some sort of underlying structure of the observed data. The observational noise vector $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2\boldsymbol{I})$ will be the difference vector between the observed data $\boldsymbol{Y}$ and a vector $\boldsymbol{Ea} \in \text{Col}(\boldsymbol{E}) \subset \mathbb{R}^N$. Also this coefficients vector $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\gamma}})$, where the covariance matrix $\boldsymbol{C}_{\boldsymbol{\gamma}}$ is defined by some covariance function $\zeta_{a,\gamma}$ with unknown parameters $\boldsymbol{\gamma}$. Hence we will not be able to apply MLE for this example if we wanted to estimate the parameters $(\boldsymbol{a}, \boldsymbol{\gamma}, \nu^2)$. If we were only interested in the parameters $(\boldsymbol{\gamma}, \nu^2)$ we could use the EM algorithm and use the log-likelihood function in equation (22). Luckily Bayesian parameter estimation in Section 7 will work on all models in this section.

6.2.2 MLE on model with non-Gaussian eigenfunction coefficients

We will generalise the two different cases of non-Gaussianity (mapped and non-mapped) into one section, because the same problem arises for the two models. In the Gaussian case, i.e. the coefficients $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\gamma}})$, the expression $\boldsymbol{Ea}$ also had a well-known multivariate Gaussian distribution. Here the data will look like $\boldsymbol{Y} = \boldsymbol{Eb} + \boldsymbol{\epsilon}$ where $\boldsymbol{b}$ is some non-Gaussian vector with some covariance structure. In this case the distribution of the expression $\boldsymbol{Eb}$ will not be known. The intuition remains the same for the distribution of the observed data. The underlying structure of the data will be the vector $\boldsymbol{Eb} \in \text{Col}(\boldsymbol{E}) \subset \mathbb{R}^N$, where the vector $\boldsymbol{b} \sim \mathcal{M}_{\boldsymbol{\alpha},\boldsymbol{\gamma}}$ has some multivariate non-Gaussian distribution with parameters $\boldsymbol{\alpha}$ and covariance parameters $\boldsymbol{\gamma}$. Then again the observational noise vector $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2\boldsymbol{I})$ will be the difference vector between the vector $\boldsymbol{Eb} \in \text{Col}(\boldsymbol{E})$ and the observed data $\boldsymbol{Y}$.

We have shown in this section that MLE is very inconvenient for these types of models. The only model which would qualified to solve with MLE is the first model Gaussian case, which could be solved using the EM algorithm The log-likelihood function of the non-Gaussian examples of the first model result in products of higher dimensional integrals, which are inefficient to solve compared to other Bayesian methods which we will discuss in upcoming sections. For the second model the probability density function of the Gaussian case results in a nice expression, but this expression does not consist of the eigenfunction coefficients $\boldsymbol{a}$. We were not able to write the joint probability density function of this observed data for the second model in terms of Gaussian and non-Gaussian coefficients. Hence the log-likelihood function could not be defined in terms of these eigenfunction coefficients, which implied that MLE is not the suitable parameter estimation method for this type of model. In the next section we will use Bayesian methods which are more convenient for these types of models, because we can describe the likelihood of a set of parameters more neatly in something called a posterior distribution.

7 Parameter estimation of the CMB models using Bayesian parameter estimation (BPE)

In the previous section maximum likelihood estimation, a frequentist approach, has been applied to the models to estimate the parameters. There was not a lot of success in estimating the parameters using MLE. The log-likelihood functions were to computationally heavy, because of their higher dimensional integrals. For this section, without further referencing, we will use Bijma et al. (2017) (Chapter 3.5).

In contrast to the frequentist approach; the Bayesian approach has something called the posterior distribution which gives an inside about the likelihood of a selection of parameters when the observed data is applied to it. In order to apply this estimation method, again some assumptions about the data has to be known (the two models) . Furthermore some prior knowledge about the parameters is mandatory in order to perform this estimation method.

In this section it is clarified how to apply Bayesian parameter estimation (BPE) for to the specific cases in the two models of the observed data, in order to obtain estimates for the parameters of the model. It will be shown that it is still impossible to calculate analytically, but also in the section hereafter, some convenient computational methods will be used for parameter estimation, which build on the theory treated in this section.

7.1 Posterior density and the Bayes estimate

The Bayesian formula describes that for some parameters $\boldsymbol{\rho}$ and observed data $\boldsymbol{y}$ the *posterior density*

$$p(\boldsymbol{\rho} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{\rho})\pi(\boldsymbol{\rho})}{p(\boldsymbol{y})} \propto p(\boldsymbol{y} \mid \boldsymbol{\rho})\pi(\boldsymbol{\rho}). \tag{23}$$

Here $p(\boldsymbol{y} \mid \boldsymbol{\rho})$ is called the likelihood function and $\pi(\boldsymbol{\rho})$ the prior of the parameters. Note that $\boldsymbol{y}$ in the example for the CMB data is the set of observations $\mathcal{Y} = \{\boldsymbol{Y_1}, \boldsymbol{Y_2}, \boldsymbol{...}, \boldsymbol{Y_K}\}$, which was also used in the section about MLE. Here $\boldsymbol{\rho}$ are all the unknown parameters and latent variables of the model.

The *Bayes estimate* for a parameter $h(\boldsymbol{\rho})$, denoted $T_{h(\boldsymbol{\rho})}(\boldsymbol{y})$, can then be calculated by calculating

$$T_{h(\boldsymbol{\rho})}(\boldsymbol{y}) = \frac{\int_{\mathrm{P}} h(\boldsymbol{\rho})p(\boldsymbol{y} \mid \boldsymbol{\rho})\pi(\boldsymbol{\rho})\,d\boldsymbol{\rho}}{p(\boldsymbol{y})} = \frac{\int_{\mathrm{P}} h(\boldsymbol{\rho})p(\boldsymbol{y} \mid \boldsymbol{\rho})\pi(\boldsymbol{\rho})\,d\boldsymbol{\rho}}{\int_{\mathrm{P}} p(\boldsymbol{y} \mid \boldsymbol{\rho'})\pi(\boldsymbol{\rho'})\,d\boldsymbol{\rho'}}, \tag{24}$$

where we integrate over the whole parameter space P. Since there are K observations denoted by the set $\mathcal{Y}$, the likelihood function becomes a product resulting in the formula

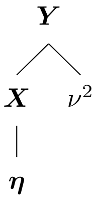
$$T_{h(\boldsymbol{\rho})}(\mathcal{Y}) = \frac{\int_{\mathrm{P}} h(\boldsymbol{\rho}) \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{\rho})\right] \pi(\boldsymbol{\rho})\,d\boldsymbol{\rho}}{\int_{\mathrm{P}} \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{\rho'})\right] \pi(\boldsymbol{\rho'})\,d\boldsymbol{\rho'}}. \tag{25}$$

One could use for example the mean of the posterior as the Bayes estimate, which would imply that $h(\boldsymbol{\rho}) = \boldsymbol{\rho}$. In the next subsections the posterior will be calculated for the two models that have been created.

7.2 Bayesian parameter estimation of the first model

7.2.1 BPE on Gaussian random field

The model will look like $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{\epsilon}$, $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma_\eta})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2\boldsymbol{I})$, from which the dependent structure of the model can be visualised in a tree



Using a Gaussian random field in the first model results in the likelihood function

$$p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{\eta}, \nu^2) = p(\boldsymbol{Y} \mid \boldsymbol{X}, \nu^2) p(\boldsymbol{X} \mid \boldsymbol{\eta}). \tag{26}$$

Note that the latent variables $\boldsymbol{X}$ are introduced (value of the Gaussian random field). This will result in the posterior density

$$p(\boldsymbol{X}, \boldsymbol{\eta}, \nu^2 \mid \boldsymbol{Y}) = \frac{p(\boldsymbol{Y} \mid \boldsymbol{X}, \nu^2) p(\boldsymbol{X} \mid \boldsymbol{\eta}) \pi(\boldsymbol{\eta}) \pi(\nu^2)}{p(\boldsymbol{Y})}. \tag{27}$$

Again because of multiple observations we introduce the set of underlying Gaussian random fields as $\mathcal{X}$ to there corresponding observation in $\mathcal{Y}$. Hence acquiring the following posterior

$$\begin{aligned} p(\mathcal{X}, \boldsymbol{\eta}, \nu^2 \mid \mathcal{Y}) &= \frac{p(\mathcal{Y} \mid \mathcal{X}, \nu^2) p(\mathcal{X} \mid \boldsymbol{\eta}) \pi(\boldsymbol{\eta}) \pi(\nu^2)}{p(\mathcal{Y})} && (28) \\ &= \frac{\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{X}_i, \nu^2) p(\boldsymbol{X}_i \mid \boldsymbol{\eta})\right] \pi(\boldsymbol{\eta}) \pi(\nu^2)}{\prod_{i=1}^{K} p(\boldsymbol{Y}_i)} && (29) \\ &\propto \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{X}_i, \nu^2) p(\boldsymbol{X}_i \mid \boldsymbol{\eta})\right] \pi(\boldsymbol{\eta}) \pi(\nu^2). && (30) \end{aligned}$$

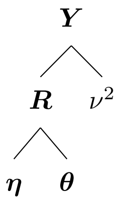
When calculating an Bayes estimate for the parameter $h(\boldsymbol{\rho})$, where $\boldsymbol{\rho} = \{\mathcal{X}, \boldsymbol{\eta}, \nu^2\}$, one needs to fill in the found posterior in equation (25) to get the estimate

$$p(\mathcal{X}, \boldsymbol{\eta}, \nu^2 \mid \mathcal{Y}) = \frac{\int_{\mathrm{P}} h(\boldsymbol{\rho}) \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{X}_i, \nu^2) p(\boldsymbol{X}_i \mid \boldsymbol{\eta})\right] \pi(\boldsymbol{\eta}) \pi(\nu^2) \, d\boldsymbol{\rho}}{\int_{\mathrm{P}} \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{X}_i', \nu'^2) p(\boldsymbol{X}_i' \mid \boldsymbol{\eta}')\right] \pi(\boldsymbol{\eta}') \pi(\nu'^2) \, d\boldsymbol{\rho}'}. \tag{31}$$

Note that the denominator is some normalisation constant dependent on the parameters $\boldsymbol{\rho}$. The expression in the denominator is too difficult to solve analytically, because of its high dimension and there will still be parameters in the covariance matrix of all $\boldsymbol{X} \in \mathcal{X}$. Luckily there is an algorithm, called the Metropolis-Hastings algorithm, that will bypass these problems. This will calculate the Bayes estimate numerically and will be discussed in the next main section.

7.2.2 BPE on mapped, non-Gaussian random field

The only addition to this example of the first model in contrast with the previous one is that there are additional mapping parameters $\boldsymbol{\theta}$, since the model looks like $\boldsymbol{Y} = \boldsymbol{R} + \boldsymbol{\epsilon} = g_{\boldsymbol{\theta}}(\boldsymbol{X}) + \boldsymbol{\epsilon}$, $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2 \boldsymbol{I})$. The tree will thus have one extra branch



The same procedure, as with the previous example, has been done to calculate the likelihood function:

$$p(\boldsymbol{Y} \mid \boldsymbol{R}, \boldsymbol{\eta}, \boldsymbol{\theta}, \nu^2) = p(\boldsymbol{Y} \mid \boldsymbol{R}, \nu^2) p(\boldsymbol{R} \mid \boldsymbol{\eta}, \boldsymbol{\theta}). \tag{32}$$

This will result in the following posterior, given observations $\mathcal{Y}$ and set of corresponding random fields $\mathcal{R}$

$$\begin{aligned} p(\mathcal{R}, \boldsymbol{\eta}, \boldsymbol{\theta}, \nu^2 \mid \mathcal{Y}) &= \frac{p(\boldsymbol{Y} \mid \mathcal{R}, \nu^2) p(\mathcal{R} \mid \boldsymbol{\eta}, \boldsymbol{\theta}) \pi(\boldsymbol{\eta}) \pi(\boldsymbol{\theta}) \pi(\nu^2)}{p(\mathcal{Y})} & (33) \\ &= \frac{\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{R}_i, \nu^2) p(\boldsymbol{R}_i \mid \boldsymbol{\eta}, \boldsymbol{\theta})\right] \pi(\boldsymbol{\eta}) \pi(\boldsymbol{\theta}) \pi(\nu^2)}{\prod_{i=1}^{K} p(\boldsymbol{Y}_i)}, & (34) \end{aligned}$$

now using (9) to obtain

$$p(\mathcal{R}, \boldsymbol{\eta}, \boldsymbol{\theta}, \nu^2 \mid \mathcal{Y}) = \frac{\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{R}_i, \nu^2) \mathcal{N}_{N_i}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{R}_i) \mid \boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta},i}) \det[\mathcal{J}_{N_i}(\boldsymbol{R}_i \mid \boldsymbol{\theta})]\right] \pi(\boldsymbol{\eta}) \pi(\boldsymbol{\theta}) \pi(\nu^2)}{\prod_{i=1}^{K} p(\boldsymbol{Y}_i)} \tag{35}$$

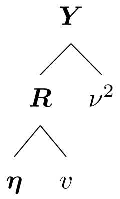
Analogously denoting the set of non-Gaussian mapped random fields with $\mathcal{R}$ by their corresponding observed data $\mathcal{Y}$. Then the Bayes estimate for $\boldsymbol{\rho} = \{\mathcal{R}, \boldsymbol{\eta}, \boldsymbol{\theta}, \nu^2\}$, for observations $\mathcal{Y}$, will look like

$$p(\mathcal{R}, \boldsymbol{\eta}, \boldsymbol{\theta}, \nu^2 \mid \mathcal{Y}) = \frac{\int_{\mathrm{P}} h(\boldsymbol{\rho}) \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{R}_i, \nu^2) \mathcal{N}_{N_i} \det[\mathcal{J}_{N_i}(\boldsymbol{R}_i \mid \boldsymbol{\theta})]\right] \pi(\boldsymbol{\eta}) \pi(\boldsymbol{\theta}) \pi(\nu^2) \, d\boldsymbol{\rho}}{\int_{\mathrm{P}} \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{R}'_i, \nu'^2) \mathcal{N}_{N_i}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{R}'_i) \mid \boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}',i}) \det[\mathcal{J}_{N_i}(\boldsymbol{R}'_i \mid \boldsymbol{\theta}')]\right] \pi(\boldsymbol{\eta}') \pi(\boldsymbol{\theta}') \pi(\nu'^2) \, d\boldsymbol{\rho}'}. \tag{36}$$

This integral cannot be calculated analytically.

7.2.3 BPE on an other example of a non-Gaussian random field

This example of a Student-t_v random field is very similar as the previous example. Except now there is no mapping and the likelihood function is defined with a density function of the t_v-distribution (11). Now the data is $\boldsymbol{Y} = \boldsymbol{R} + \boldsymbol{\epsilon}$, $\boldsymbol{R} \sim t_v(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2 \boldsymbol{I})$. The structure of parameters will thus look like



This will result in the likelihood function

$$p(\boldsymbol{Y} \mid \boldsymbol{R,\eta}, v, \nu^2) = p(\boldsymbol{Y} \mid \boldsymbol{R}, \nu^2) p(\boldsymbol{R} \mid \boldsymbol{\eta}, v). \tag{37}$$

Given the observations $\mathcal{Y}$, the posterior will thus look like

$$p(\mathcal{R}, \boldsymbol{\eta}, v, \nu^2 \mid \mathcal{Y}) = \frac{p(\boldsymbol{Y} \mid \mathcal{R}, \nu^2) p(\mathcal{R} \mid \boldsymbol{\eta}, v) \pi(\boldsymbol{\eta}) \pi(v) \pi(\nu^2)}{p(\mathcal{Y})} \tag{38}$$

$$= \frac{\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{R}_i, \nu^2) p(\boldsymbol{R}_i \mid \boldsymbol{\eta}, v)\right] \pi(\boldsymbol{\eta}) \pi(v) \pi(\nu^2)}{\prod_{i=1}^{K} p(\boldsymbol{Y}_i)} \tag{39}$$

and the Bayes estimate for a $h(\boldsymbol{\rho})$ where $\boldsymbol{\rho} = \{\mathcal{R}, \boldsymbol{\eta}, v, \nu^2\}$ will be defined as

$$p(\mathcal{R}, \boldsymbol{\eta}, \boldsymbol{\theta}, \nu^2 \mid \mathcal{Y}) = \frac{\int_{\mathrm{P}} h(\boldsymbol{\rho}) \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{R}_i, \nu^2) p(\boldsymbol{R}_i \mid \boldsymbol{\eta}, v)\right] \pi(\boldsymbol{\eta}) \pi(v) \pi(\nu^2) \, d\boldsymbol{\rho}}{\int_{\mathrm{P}} \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{R}'_i, \nu'^2) p(\boldsymbol{R}'_i \mid \boldsymbol{\eta'}, , v')\right] \pi(\boldsymbol{\eta'}) \pi(v') \pi(\nu'^2) \, d\boldsymbol{\rho'}}, \tag{40}$$

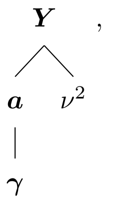
which again is impossible to solve analytically.

For all three cases of the first model it is demonstrated, in this subsection, that the Bayes estimates for the parameters result again in higher dimensional integrals with a difficult expression in it. The covariance matrix will namely be variable and thus unknown as well as its determinant.

7.3 Bayesian parameter estimation of the second model

7.3.1 BPE on model with Gaussian eigenfunction coefficients

For the second model using Gaussian eigenfunction coefficients; the observed data will look like $\boldsymbol{Y} = \boldsymbol{Ea} + \boldsymbol{\epsilon}$, $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_\gamma})$and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2 \boldsymbol{I})$. Here it is assumed that there are n different eigenfunction, i.e. $\boldsymbol{a} \in \mathbb{R}^n$. The following structure of the model arises



which results in the likelihood function

$$p(\boldsymbol{Y} \mid \boldsymbol{a,\gamma}, \nu^2) = p(\boldsymbol{Y} \mid \boldsymbol{a}, \nu^2) p(\boldsymbol{a} \mid \gamma). \tag{41}$$

Given observations $\mathcal{Y}$, the posterior density will be defined as

$$p(\boldsymbol{a}, \gamma, \nu^2 \mid \mathcal{Y}) = \frac{p(\mathcal{Y} \mid \boldsymbol{a}, \nu^2) p(\boldsymbol{a} \mid \gamma) \pi(\gamma) \pi(\nu^2)}{p(\mathcal{Y})} \tag{42}$$

$$= \frac{\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{a}, \nu^2) p(\boldsymbol{a} \mid \gamma)\right] \pi(\gamma) \pi(\nu^2)}{\prod_{i=1}^{K} p(\boldsymbol{Y}_i)} \tag{43}$$

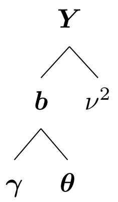
and the Bayes estimate for a $h(\boldsymbol{\rho})$ where $\boldsymbol{\rho} = \{\boldsymbol{a}, \boldsymbol{\gamma}, \nu^2\}$ will be defined as

$$p(\boldsymbol{a},\boldsymbol{\theta},\boldsymbol{\gamma},v,\nu^2 \mid \mathcal{Y}) = \frac{\int_{\mathrm{P}} h(\boldsymbol{\rho})\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{a},\nu^2)p(\boldsymbol{a}\mid\boldsymbol{\gamma})\right]\pi(\boldsymbol{\gamma})\pi(\nu^2)\,d\boldsymbol{\rho}}{\int_{\mathrm{P}}\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i\mid \boldsymbol{a}',\nu'^2)p(\boldsymbol{a}'\mid\boldsymbol{\gamma}')\right]\pi(\boldsymbol{\gamma}')\pi(\nu'^2)\,d\boldsymbol{\rho}'}, \tag{44}$$

As seen in the previous subsection about the first model, we again get a very difficult expression. Note that we afresh have parameters inside the covariance matrix $\boldsymbol{C}_{\boldsymbol{\gamma}}$, and thus also in its determinant. It makes it impossible to calculate analytically. This is also, because the dimension of the integral increases as the number of eigenfunctions and data points increases.

7.3.2 BPE on model with mapped non-Gaussian eigenfunction coefficients

The next example is for mapped Gaussian eigenfunction coefficients to a non-Gaussian distribution. This means that the observed data $\boldsymbol{Y} = \boldsymbol{E}\boldsymbol{b} + \boldsymbol{\epsilon}$, where $\boldsymbol{b} = g_{\boldsymbol{\theta}}(\boldsymbol{a})$ with $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\gamma}})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2\boldsymbol{I})$. Then the structure is a little more extended compared to the Gaussian case



which results in the likelihood function

$$p(\boldsymbol{Y} \mid \boldsymbol{b}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \nu^2) = p(\boldsymbol{Y} \mid \boldsymbol{b}, \nu^2)p(\boldsymbol{b} \mid \boldsymbol{\gamma}, \boldsymbol{\theta}). \tag{45}$$

Given observations $\mathcal{Y}$, the posterior density will be

$$\begin{aligned} p(\boldsymbol{b},\boldsymbol{\gamma},\boldsymbol{\theta},\nu^2 \mid \mathcal{Y}) &= \frac{p(\mathcal{Y} \mid \boldsymbol{b},\nu^2)p(\boldsymbol{b}\mid\boldsymbol{\gamma},\boldsymbol{\theta})\pi(\boldsymbol{\gamma})\pi(\boldsymbol{\theta})\pi(\nu^2)}{p(\mathcal{Y})} && (46)\\ &= \frac{\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{b},\nu^2)p(\boldsymbol{b}\mid\boldsymbol{\gamma},\boldsymbol{\theta})\right]\pi(\boldsymbol{\gamma})\pi(\boldsymbol{\theta})\pi(\nu^2)}{\prod_{i=1}^{K} p(\boldsymbol{Y}_i)} && (47)\\ &= \frac{\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{b},\nu^2)p(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{b})\mid\boldsymbol{\gamma},v)\det[\mathcal{J}(\boldsymbol{b}\mid\boldsymbol{\theta})]\right]\pi(\boldsymbol{\gamma})\pi(\boldsymbol{\theta})\pi(\nu^2)}{\prod_{i=1}^{K} p(\boldsymbol{Y}_i)}, && (48)\end{aligned}$$

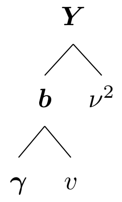
where the Jacobian $\mathcal{J}(\boldsymbol{b} \mid \boldsymbol{\theta}) = \left[\frac{\partial g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r})}{\partial \boldsymbol{r}}\right]_{\boldsymbol{r}=\boldsymbol{b}}$. Then the Bayes estimate for a $h(\boldsymbol{\rho})$ where $\boldsymbol{\rho} = \{\boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \nu^2\}$ will be defined as

$$p(\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{\gamma},\nu^2 \mid \mathcal{Y}) = \frac{\int_{\mathrm{P}} h(\boldsymbol{\rho})\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{b},\nu^2)p(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{b})\mid\boldsymbol{\gamma},\boldsymbol{\theta})\det[\mathcal{J}(\boldsymbol{b}\mid\boldsymbol{\theta})]\right]\pi(\boldsymbol{\gamma})\pi(\boldsymbol{\theta})\pi(\nu^2)\,d\boldsymbol{\rho}}{\int_{\mathrm{P}}\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{b}',\nu'^2)p(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{b}')\mid\boldsymbol{\gamma}',\boldsymbol{\theta}')\det[\mathcal{J}(\boldsymbol{b}'\mid\boldsymbol{\theta}')]\right]\pi(\boldsymbol{\gamma}')\pi(\boldsymbol{\theta}')\pi(\nu'^2)\,d\boldsymbol{\rho}'}. \tag{49}$$

Again this will be impossible to solve analytically.

7.3.3 BPE on model with other non-Gaussian eigenfunction coefficients

Finally we check what happens if we have some other eigenfunction coefficients which are non-Gaussian, but do not have a known mapping from the Gaussian eigenfunction coefficients. Again we take the example of the Student-t_v distribution. We let $\boldsymbol{Y} = \boldsymbol{R} + \boldsymbol{\epsilon}$, $\boldsymbol{R} \sim t_v(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\gamma}})$ and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu^2 \boldsymbol{I})$. This results in the following tree:



which results in the likelihood function

$$p(\boldsymbol{Y} \mid \boldsymbol{b}, \boldsymbol{\gamma}, v, \nu^2) = p(\boldsymbol{Y} \mid \boldsymbol{b}, \nu^2) p(\boldsymbol{b} \mid \boldsymbol{\gamma}, v). \tag{50}$$

Given observations $\mathcal{Y}$, the posterior density will be defined as

$$\begin{align} p(\boldsymbol{b}, \boldsymbol{\gamma}, v, \nu^2 \mid \mathcal{Y}) &= \frac{p(\mathcal{Y} \mid \boldsymbol{b}, \nu^2) p(\boldsymbol{b} \mid \boldsymbol{\gamma}, v) \pi(\boldsymbol{\gamma}) \pi(v) \pi(\nu^2)}{p(\mathcal{Y})} \tag{51} \\ &= \frac{\left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{b}, \nu^2) p(\boldsymbol{b} \mid \boldsymbol{\gamma}, v)\right] \pi(\boldsymbol{\gamma}) \pi(v) \pi(\nu^2)}{\prod_{i=1}^{K} \boldsymbol{Y}_i} \tag{52} \\ & \tag{53} \end{align}$$

and the Bayes estimate for a $h(\boldsymbol{\rho})$ where $\boldsymbol{\rho} = \{\boldsymbol{b}, \boldsymbol{\gamma}, v, \nu^2\}$ will be defined as:

$$p(\boldsymbol{b}, \boldsymbol{\gamma}, v, \nu^2 \mid \mathcal{Y}) = \frac{\int_{\mathrm{P}} h(\boldsymbol{\rho}) \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{b}, \nu^2) p(\boldsymbol{b} \mid \boldsymbol{\gamma}, v)\right] \pi(\boldsymbol{\gamma}) \pi(v) \pi(\nu^2) \, d\boldsymbol{\rho}}{\int_{\mathrm{P}} \left[\prod_{i=1}^{K} p(\boldsymbol{Y}_i \mid \boldsymbol{b'}, \nu'^2) p(\boldsymbol{b'} \mid \boldsymbol{\gamma'}, v')\right] \pi(\boldsymbol{\gamma'}) \pi(v') \pi(\nu'^2) \, d\boldsymbol{\rho'}}, \tag{54}$$

As all previous example, this estimate is impossible to solve analytically. However in the upcoming section we will use the Metropolis-Hastings algorithm to surpass this problem and estimate the parameter estimates numerically.

8 Parameter estimation of the CMB models using the Metropolis-Hastings algorithm (MH)

In the previous section the theory about BPE has been applied to the example models we encountered earlier in this paper. By calculating the parameter estimates for a given parameter vector $h(\boldsymbol{\rho})$, one could estimate it by calculating its Bayes estimate. These could not be calculated analytically. Also numerically calculating the integrals would be highly inefficient.

Intuitively if one was able to sample from this posterior distribution it would be possible, after many samples, to estimate for example the sample mean of the parameters. However the posterior distribution, in the latter cases of the first and second model, is not a 'well-known' density.

Something called the Metropolis-Hastings algorithm surpasses this problem. It is something called a MCMC method (Markov Chain Monte Carlo) which will 'walk' through this posterior density. It is called a Markov Chain because its sample only depends on the previous sample that it was at (in the parameter space P). The algorithm will a large amount of successive samples from the posterior density. This means that approximate posterior mean can be calculated numerically using this MH algorithm. The algorithm will, in this section, be applied to the two models with different examples. Without further referencing, contents of this section is based on Bijma et al. (2017) (Chapter 3.5.1).

8.1 Introduction to the Metropolis-Hastings algorithm

The posterior distribution will be denoted by the letter $p(\boldsymbol{\rho})$. To start the walk through the parameter space P one needs an initial starting point $\boldsymbol{\rho}_0 \in \mathrm{P}$. From this the next point $\boldsymbol{\rho}_1$ needs to be generated. This will be done by using a *transition kernel* Q. This will propose a new value in P which could be accepted, in which it will move to the next value, or rejected, in which it will stay at the current point. The *transition density* q is a function which calculates the likelihood of the proposed value $\boldsymbol{\rho}_{i+1}$ given the current value $\boldsymbol{\rho}_i$, denoted by $\boldsymbol{\rho}_{i+1} \mapsto q(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1})$. This density is associated with the transition kernel, which is a sampler for the transition density. Then given the current state $\boldsymbol{\rho}_i$, the MH algorithm will accept the proposed state $\boldsymbol{\rho}_{i+1}$ with probability

$$A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1}) = \min\left\{\frac{p(\boldsymbol{\rho}_{i+1})q(\boldsymbol{\rho}_{i+1}, \boldsymbol{\rho}_i)}{p(\boldsymbol{\rho}_i)q(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1})}, 1\right\}.$$

Hence if one samples a uniform random variable $U_{i+1} \sim \mathcal{U}(0,1)$ and $U_{i+1} < A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1})$ the MH algorithm will accept the proposed state, else it will just stay at the current state (hence it will sample the same value again). This process will be iterated for a given number of times. After the MH has finished for all iterations, something called the *burn-in* will remove some number of samples from the beginning of the algorithm. This is because the initial value can be in a really low density state which will impact the calculation of the estimated posterior means. Hence the parameter estimation will most likely be more precise if those samples are removed.

In all of the examples the transition density will be symmetric, this means that $q(\boldsymbol{a}, \boldsymbol{b}) = q(\boldsymbol{b}, \boldsymbol{a}), \ \forall \boldsymbol{a}, \boldsymbol{b} \in \mathrm{P}$. Then the acceptance probability can be simplified:

$$A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1}) = \min\left\{\frac{p(\boldsymbol{\rho}_{i+1})}{p(\boldsymbol{\rho}_i)}, 1\right\},$$

since $\frac{q(\boldsymbol{\rho}_{i+1}, \boldsymbol{\rho}_i)}{q(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1})} = 1$. Also for our examples we need to use logarithms, because the densities get really small for higher dimensions, which can cause numerical underflow (giving zero values while it is not equal to zero). Hence we accept if

$$\log U_{i+1} < \log A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1}) = \min\{\log p(\boldsymbol{\rho}_{i+1}) - \log p(\boldsymbol{\rho}_i), 0\},$$

which is equivalent to accept when $U_{i+1} < A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1})$ because of the strict monotonicity of the logarithm.

8.2 Parameter estimation of the first model using the Metropolis-Hastings algorithm

First the example, that will be used for the different cases of the first model, needs to be defined. The grid will be $(-L : \Delta x : L)$ in the x-direction and $(-M : \Delta y : M)$ in the y-direction. There will thus be one observation, i.e. $\mathcal{Y} = \boldsymbol{Y}$. Here $L = M = 1$ and $\Delta x = \Delta y = 0.25$. The covariance function of the Gaussian random field $\boldsymbol{X}$ will be the function with parameters $\boldsymbol{\eta} = \{A, B, C\}$:

$$\xi_{\boldsymbol{\eta},X}(\boldsymbol{t}_1, \boldsymbol{t}_2) = \xi_{\{A,B,C\},X}(\boldsymbol{t}_1, \boldsymbol{t}_2) = A\cos(B||\boldsymbol{t}_1 - \boldsymbol{t}_2||)\exp(-C||\boldsymbol{t}_1 - \boldsymbol{t}_2||)$$

where, in this example, we assume that $A = 1, B = 1, C = 0.5$. Then $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ will be constructed by equation (2). The noise factor for generating the observed data will be $\nu^2 = 1^2$

Using section 7.3 we can get the posterior density for each different case, which we will need for the Metropolis-Hastings algorithm. For the MH algorithm we take a symmetric transition kernel with distribution $\boldsymbol{\rho}_{i+1} \sim \mathcal{N}(\boldsymbol{\rho}_i, \beta^2\boldsymbol{I})$ for every iteration i, where $\beta^2 =$. This means the transition density q is $\boldsymbol{\rho}_{i+1} \mapsto q(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1}) = \mathcal{N}(\boldsymbol{\rho}_{i+1} \mid \boldsymbol{\rho}_i, \beta^2\boldsymbol{I})$. This value β^2 has to be chosen wisely, because it acts kind of like a step size and will effect the performance of the MH algorithm. Too big and the acceptance rate will be very low, i.e. a very inefficient algorithm. Too small and the sample will not efficiently move through the parameter space. It will take a long time to through the whole space P. Finally the number of iterations $T = 25000$ with a burn-in of $B = 500$.

A key note is that we already assume that we know the distribution of the random field and of the covariance structure. However the parameters are unknown, because those are going to be estimated by the MH algorithm. This is of course a very precise assumption, because we have simulated the models ourselves with the same covariance structure and distribution.

8.2.1 MH algorithm on Gaussian random field

For the first example using the posterior in (27) the acceptance probability of the Metropolis-Hastings algorithm for this case, given parameters $\boldsymbol{\rho}_i = \{\boldsymbol{X}_i, \boldsymbol{\eta}_i, \nu_i^2\}$, is defined by:

$$\begin{aligned} A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1}) &= \min\left\{\frac{\frac{p(\boldsymbol{Y}|\boldsymbol{X}_{i+1},\nu_{i+1}^2)p(\boldsymbol{X}_{i+1}|\boldsymbol{\eta}_{i+1})\pi(\boldsymbol{\eta}_{i+1})\pi(\nu_{i+1}^2)}{p(\boldsymbol{Y})}}{\frac{p(\boldsymbol{Y}|\boldsymbol{X}_i,\nu_i^2)p(\boldsymbol{X}_i|\boldsymbol{\eta}_i)\pi(\boldsymbol{\eta}_i)\pi(\nu_i^2)}{p(\boldsymbol{Y})}}, 1\right\} \\ &= \min\left\{\frac{p(\boldsymbol{Y} \mid \boldsymbol{X}_{i+1}, \nu_{i+1}^2)p(\boldsymbol{X}_{i+1} \mid \boldsymbol{\eta}_{i+1})\pi(\boldsymbol{\eta}_{i+1})\pi(\nu_{i+1}^2)}{p(\boldsymbol{Y} \mid \boldsymbol{X}_i, \nu_i^2)p(\boldsymbol{X}_i \mid \boldsymbol{\eta}_i)\pi(\boldsymbol{\eta}_i)\pi(\nu_i^2)}, 1\right\}. \end{aligned}$$

One thing which is useful, is that the normalisation factors $p(\boldsymbol{Y})$ cancel eachother out. The algorithm thus prevents us from calculating a high dimensional integral. Also in this case because of our symmetric transition kernel we do not have to take that calculation of the transition densities into account, since those two will cancel eachother out. In the long run this will same us some computational time.

Now some assumptions about the priors have to be made. The parameters $\boldsymbol{\eta}$ will assumed to have a prior density of $\pi(\boldsymbol{\eta}) = \mathcal{N}(\boldsymbol{\eta} \mid \tilde{\boldsymbol{\eta}}, \boldsymbol{v}_{\boldsymbol{\eta}}\boldsymbol{I})$ where $\tilde{\boldsymbol{\eta}}$ is some expectation vector and $\boldsymbol{v}_{\boldsymbol{\eta}}$ is some variance vector for the parameters. The parameter ν^2 can only be positive, so the prior density will be $\pi(\nu^2) = \mathcal{N}(\nu^2 \mid \mu_{\nu^2}, \sigma^2_{\nu^2})$, from the normal distribution with parameters μ_{ν^2} and $\sigma^2_{\nu^2}$. For this section $\tilde{\boldsymbol{\eta}} = [1, 1, 0.1]^T$, $\boldsymbol{v}_{\boldsymbol{\eta}} = [0.25, 0.25, 0.25]^T$, $\mu_{\nu^2} = 1$ and $\sigma^2_{\nu^2} = 0.1^2$. Henceforth the present functions inside the acceptance probability will be:

$$
\begin{aligned}
p(\boldsymbol{Y} \mid \boldsymbol{X}, \nu^2) &= \mathcal{N}(\boldsymbol{Y} - \boldsymbol{X} \mid \boldsymbol{0}, \nu^2\boldsymbol{I}) \\
p(\boldsymbol{X} \mid \boldsymbol{\eta}) &= \mathcal{N}(\boldsymbol{X} \mid \boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}}) \\
\pi(\boldsymbol{\eta}) &= \mathcal{N}(\boldsymbol{\eta} \mid \tilde{\boldsymbol{\eta}}, \boldsymbol{v}_{\boldsymbol{\eta}}\boldsymbol{I}) \\
\pi(\nu^2) &= \mathcal{N}(\nu^2 \mid \mu_{\nu^2}, \sigma^2_{\nu^2})
\end{aligned}
$$

Note that $\xi_{\boldsymbol{\eta},X}$ will be known, which implies that the structure of $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ will be known.

The transition kernel will be $\boldsymbol{\rho}_{i+1} \sim \mathcal{N}(\boldsymbol{\rho}_i, \beta^2\boldsymbol{I})$ where $\beta^2 = 0.02^2$. Now using $T = 25000$ iterations and $B = 500$ burn-in we can start sampling from the posterior distribution using the MH algorithm above, starting at the initial value

$$\boldsymbol{X}_0 = \boldsymbol{Y}, \quad \boldsymbol{\eta}_0 = [1; 1; 1], \quad \nu_0^2 = 0.75.$$

This walk through the parameter space P result in something called a *trace plot*, which indicates the value per parameter at a given iteration. The trace plot for all parameters can be seen in Figure 15.

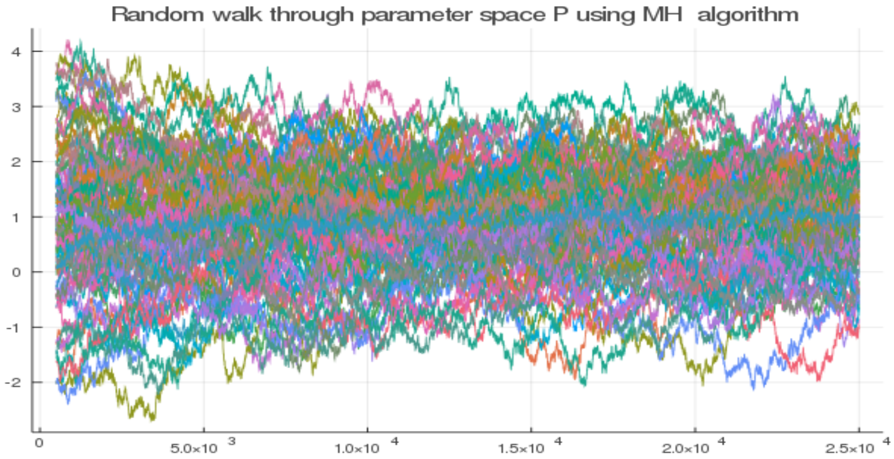


Figure 15: Metropolis-Hastings algorithm for given example for the observed data with with a Gaussian random field of the first model with 25000 iterations and 500 burn-in. One line represents one parameter value.

If we use the posterior mean as Bayes estimate $\hat{\boldsymbol{X}}$ for the Gaussian random field parameters, we would obtain the random field, which is plotted in Figure 17. For comparison it is plotted next to the real values of the random field.

From the trace plot of all parameters we can not read clearly what happens for specific parameters, that is why we do not show this plot for the upcoming examples of applying the MH algorithm to various random fields. The more interesting parameters $\boldsymbol{\eta}$ and ν^2 can be isolated in their own trace plot as shown in Figure 16. Here the horizontal lines represent the real values

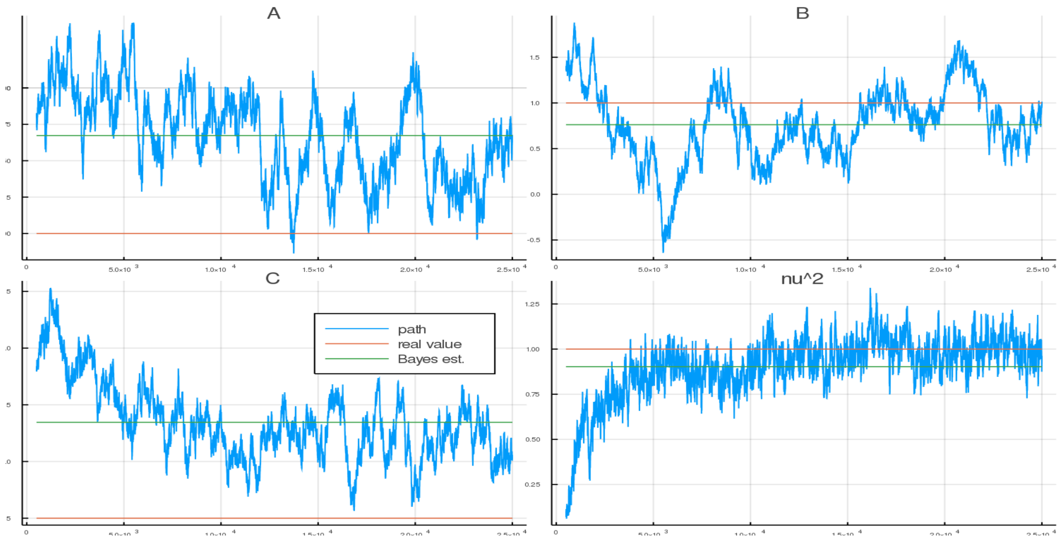


Figure 16: The same trace plot but now only for the parameters $\boldsymbol{\eta}$ and ν^2. The blue line represents the path, the red line represents the real value used in the simulation and the green line is the posterior mean for the path (sample mean)

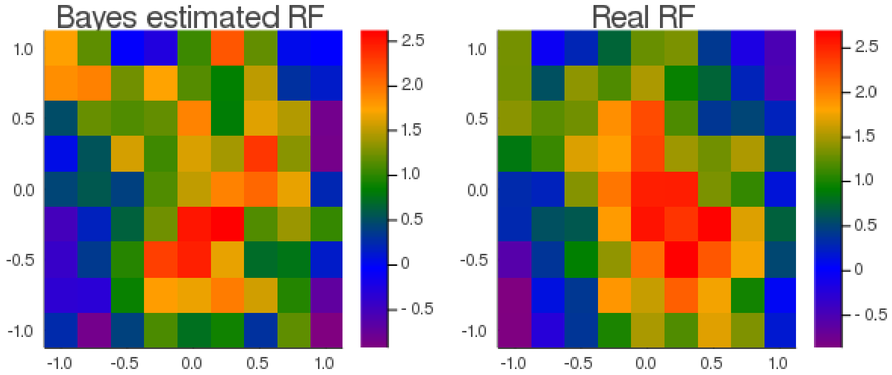


Figure 17: The posterior mean of the random field, acquired from the trace plot, plotted against the real random field for the given example.

of the parameters. The estimates of the parameters for this example and their real values are respectively the green and the red horizontal lines in Figure 16. Looking at the trace plots all the paths get close to the real value. The posterior mean of the paths (green lines) are also relatively close to the real value (red lines), especially for the parameter ν^2.

In this section all different examples of (non-)Gaussian random fields for the two different models will be used to apply the MH algorithm for their corresponding self-simulated observed data. Here the real values of the parameters are already known, which is convenient for making a comparison among all its designated parameters and their estimate.

8.2.2 MH algorithm on mapped, non-Gaussian random field

As an example of mapped non-Gaussian random field we take a lognormal random field. This means $\boldsymbol{\theta} = \{\mu, \sigma\}$ and

$$R(\boldsymbol{t}) = g_{\boldsymbol{\theta}}(X(\boldsymbol{t})) = g_{\mu,\sigma}(X(\boldsymbol{t})) = e^{\mu+\sigma X(\boldsymbol{t})},$$

with for this example $\mu = -1$ and $\sigma = 1$. These are two extra parameters that have to be estimated by the MH algorithm. For this example using the posterior (33) the acceptance probability, given parameters $\boldsymbol{\rho}_i = \{\boldsymbol{R}_i, \boldsymbol{\eta}_i, \boldsymbol{\theta}_i, \nu_i^2\}$, will be defined as:

$$A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1}) = \min\left\{\frac{p(\boldsymbol{Y} \mid \boldsymbol{R}_{i+1}, \nu_{i+1}^2)p(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{R}_{i+1}) \mid \boldsymbol{\eta}_{i+1}, \boldsymbol{\theta}_{i+1})\det[\mathcal{J}(\boldsymbol{R}_{i+1}, \boldsymbol{\theta}_{i+1})]\pi(\boldsymbol{\eta}_{i+1})\pi(\nu_{i+1}^2)}{p(\boldsymbol{Y} \mid \boldsymbol{R}_i, \nu_i^2)p(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{R}_i) \mid \boldsymbol{\eta}_i, \boldsymbol{\theta}_i)\det[\mathcal{J}(\boldsymbol{R}_i, \boldsymbol{\theta}_i)]\pi(\boldsymbol{\eta}_i)\pi(\nu_i^2)}, 1\right\},$$

where $g_{\boldsymbol{\theta}}^{-1}(R(\boldsymbol{t})) = \frac{\log R(\boldsymbol{t}) - \mu}{\sigma}$. For this example we also need the prior for the mapping parameters $\boldsymbol{\theta}$. Analogous to the prior of $\boldsymbol{\eta}$ we take that $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \tilde{\boldsymbol{\theta}}, \boldsymbol{v_\theta I})$. Here we take that $\tilde{\boldsymbol{\theta}} = [-1, 1]^T$ and $\boldsymbol{v_\theta} = [0.5, 0.5]^T$. Thus the function inside the acceptance probability are defined by:

$$\begin{aligned}
p(\boldsymbol{Y} \mid \boldsymbol{R}, \nu^2) &= \mathcal{N}(\boldsymbol{Y} - \boldsymbol{R} \mid \boldsymbol{0}, \nu^2\boldsymbol{I}) \\
p(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{R}) \mid \boldsymbol{\eta}, \boldsymbol{\theta}) &= \mathcal{N}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{R}) \mid \boldsymbol{0}, \boldsymbol{\Sigma_\eta}) \\
\pi(\boldsymbol{\eta}) &= \mathcal{N}(\boldsymbol{\eta} \mid \tilde{\boldsymbol{\eta}}, \boldsymbol{v_\eta I}) \\
\pi(\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{\theta} \mid \tilde{\boldsymbol{\theta}}, \boldsymbol{v_\theta I}) \\
\pi(\nu^2) &= \mathcal{N}(\nu^2 \mid \mu_{\nu^2}, \sigma_{\nu^2}^2)
\end{aligned}$$

$$\mathcal{J}(\boldsymbol{R}, \boldsymbol{\theta}) = \left[\frac{\partial g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r})}{\partial \boldsymbol{r}}\right]_{\boldsymbol{r}=\boldsymbol{R}|\boldsymbol{\theta}}$$

Note that $\boldsymbol{\Sigma_\eta}$ is the Gaussian covariance matrix, defined by the function $\xi_{\boldsymbol{\eta},X}$. The transition kernel will be $\boldsymbol{\rho}_{i+1} \sim \mathcal{N}(\boldsymbol{\rho}_i, \beta^2\boldsymbol{I})$ where $\beta^2 = 0.005^2$. Now given $T = 50000$ and $B = 500$ we can start sampling from the posterior distribution given the MH algorithm above, starting at the initial values

$$R(\boldsymbol{t}_i)_0 = \max(Y(\boldsymbol{t}_i), 0.01), \quad \boldsymbol{\eta}_0 = [0.5; 0.5; 0.5], \quad \boldsymbol{\theta}_0 = [-0.5; 0.5], \quad \nu_0^2 = 0.5, \quad \forall i = 1, ..., N.$$

A trace plot is shown with the parameters $\boldsymbol{\eta}, \boldsymbol{\theta}$ and ν^2 in Figure 18. The real values are again shown as the red horizontal lines and their corresponding posterior mean as the green horizontal lines. Taking the posterior mean as our Bayes estimate for the parameters we obtain the following estimated underlying random field, which is plotted next to the real random field in Figure 19.

The values of the estimates are also put in the table in Table 1. Also the real values and the variances of the values are put in the table.

The posterior means of σ and C are still some standard deviations away from the real values. Especially the estimator $\hat{\sigma}$ which is around 5 standard deviations away. If we run the algorithm for some more steps, these estimated posterior means will get closer to the real values of the parameters.

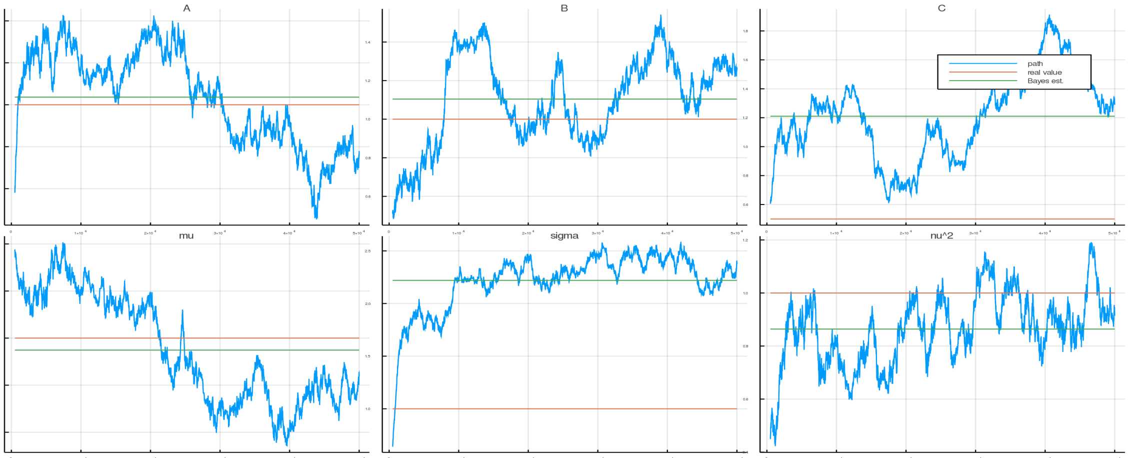


Figure 18: The same trace plot but now only for the parameters $\boldsymbol{\eta}, \boldsymbol{\theta}$ and ν^2. The blue line represents the path, the red line represents the real value used in the simulation and the green line is the posterior mean for the path (sample mean).

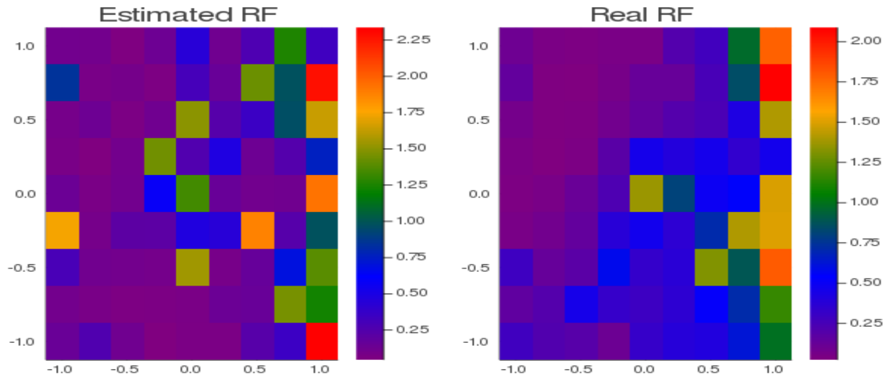


Figure 19: The posterior mean of the random field, acquired from the trace plot, plotted against the real random field for the given example of the lognormal random field.

parameter	real value (red line)	posterior mean (green line)	variance of post. mean
A	1.0000	1.0363	0.0504
B	1.0000	1.1042	0.0518
C	0.5000	1.2103	0.0853
μ	-1.0000	-1.0638	0.0810
σ	1.0000	2.2200	0.0691
ν^2	1.0000	0.8640	0.0176

Table 1: Table corresponding to the parameter trace plot in Figure 18.

18

A possibility for the inaccurate estimation is because there are multiple true value possible for the parameters. This means that the model is *unidentifiable*. Here the coefficients A and σ interfere with each other. So we could simplify the model by taking away one of the parameters.

Possibly the parameter C also is related to the parameter σ, but for this further investigation is needed.

8.2.3 MH algorithm on an other example of a non-Gaussian random field

The last case of the first model would be a non-mapped non-Gaussian random field. Again the Student-t_v example is demonstrated. For our simulation we take the real value of $v = 6$. This parameters will also be estimated by the MH algorithm. This leads to a acceptance probability, given parameters $\boldsymbol{\rho}_i = \{\boldsymbol{R}_i, \boldsymbol{\eta}_i, v_i, \nu_i^2\}$, of:

$$A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1}) = \min\left\{\frac{p(\boldsymbol{Y} \mid \boldsymbol{R}_{i+1}, \nu_{i+1}^2)p(\boldsymbol{R}_{i+1} \mid \boldsymbol{\eta}_{i+1}, v_{i+1})\pi(\boldsymbol{\eta}_{i+1})\pi(v_{i+1})\pi(\nu_{i+1}^2)}{p(\boldsymbol{Y} \mid \boldsymbol{R}_i, \nu_i^2)p(\boldsymbol{R}_i \mid \boldsymbol{\eta}_i, v_i)\pi(\boldsymbol{\eta}_i)\pi(v_i)\pi(\nu_i^2)}, 1\right\}.$$

Now the prior of v has to be defined. Let $\pi(v) = \mathcal{N}(v \mid \mu_v, \sigma_v^2)$, where $\mu_v = 6$ and $\sigma_v = 0.25$. This means the present functions inside the acceptance probability are:

$$\begin{aligned}
p(\boldsymbol{Y} \mid \boldsymbol{R}, \nu^2) &= \mathcal{N}(\boldsymbol{Y} - \boldsymbol{R} \mid \boldsymbol{0}, \nu^2 \boldsymbol{I}) \\
p(\boldsymbol{R} \mid \boldsymbol{\eta}, v) &= \mathcal{ST}(\boldsymbol{R} \mid \boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}}, v) \\
\pi(\boldsymbol{\eta}) &= \mathcal{N}(\boldsymbol{\eta} \mid \tilde{\boldsymbol{\eta}}, \boldsymbol{v}_{\boldsymbol{\eta}} \boldsymbol{I}) \\
\pi(v) &= \mathcal{N}(v \mid \mu_v, \sigma_v^2) \\
\pi(\nu^2) &= \mathcal{N}(\nu^2 \mid \mu_{\nu^2}, \sigma_{\nu^2}^2)
\end{aligned}$$

Note that $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ is the Gaussian covariance matrix, defined by the function $\xi_{\boldsymbol{\eta},X}$. The transition kernel will be $\boldsymbol{\rho}_{i+1} \sim \mathcal{N}(\boldsymbol{\rho}_i, \beta^2 \boldsymbol{I})$ where $\beta^2 = 0.01^2$. Now given $T = 50000$ and $B = 500$ we can start sampling from the posterior distribution given the MH algorithm above, starting at the initial values

$$\boldsymbol{R}_0 = \boldsymbol{Y}, \quad \boldsymbol{\eta}_0 = [0.75; 0.75; 0.75], \quad v_0 = 6, \qquad \nu_0^2 = 0.75.$$

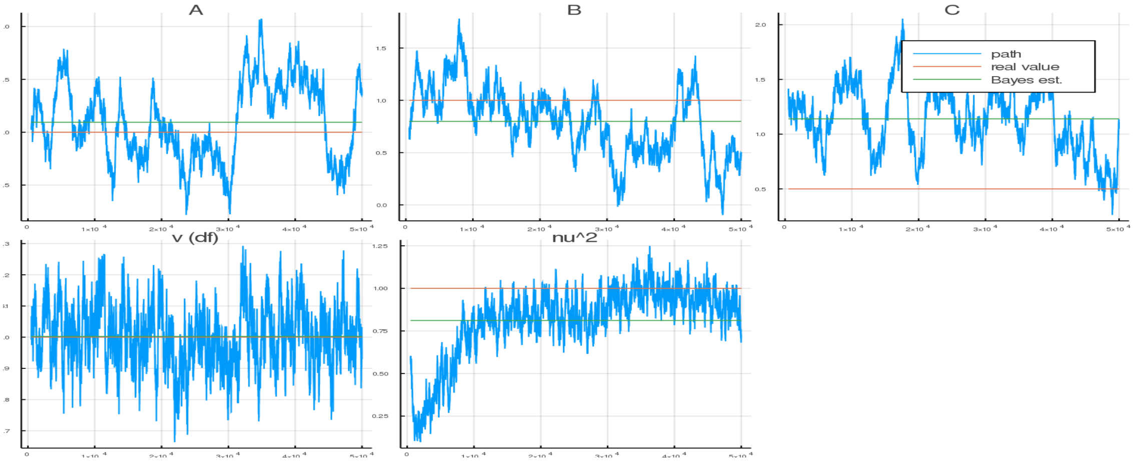


Figure 20: The same trace plot but now only for the parameters $\boldsymbol{\eta}, v$ and ν^2. The blue line represents the path, the red line represents the real value used in the simulation and the green line is the posterior mean for the path (sample mean)

A trace plot is shown with the parameters $\boldsymbol{\eta}, v$ and ν^2 in Figure 20. The real values are again shown as the horizontal lines. If we take the posterior mean as our Bayes estimate for the parameters we obtain the following estimated underlying random field, which is plotted next to the real random field in Figure 21.

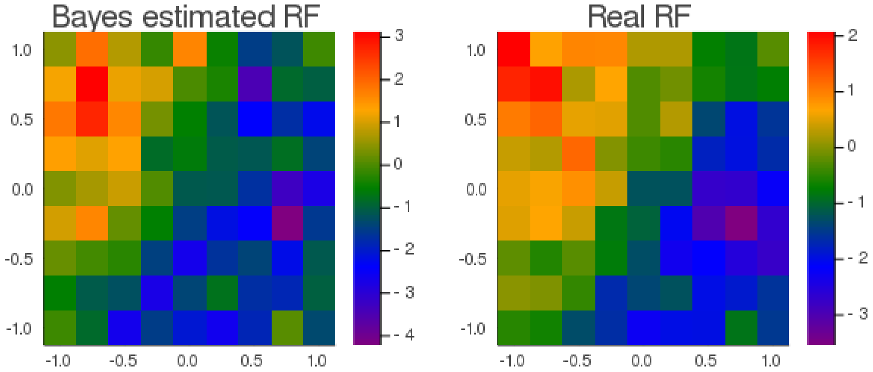


Figure 21: The posterior mean of the random field, acquired from the trace plot, plotted against the real random field for the given example.

The other estimated parameters have been put in Table 2 next to their real value. These values correspond to the parameter trace plot. In the figure with the two random fields (Figure 21) is shown that the estimated field has higher and lower temperatures. A cause of this is probably that the estimate of the noise factor ν^2 is lower than the real value, which implies that more extreme temperatures are more likely since there is less noise.

parameter	real value (red line)	posterior mean (green line)	variance of post. mean
A	1.0000	1.0943	0.1339
B	1.0000	0.7987	0.1078
C	0.5000	1.1401	0.0919
v	6.0000	6.0023	0.0096
ν^2	1.0000	0.8114	0.0446

Table 2: Table corresponding to the parameter trace plot in Figure 20

8.3 Parameter estimation of the second model using the Metropolis-Hastings algorithm

The example that we work with for the second model has the grid same grid as in the example for the first model, i.e $(-L : \Delta x : L)$ in the x-direction and $(-M : \Delta y : M)$ in the y-direction, with $L = M = 2.5$ and $\Delta x = \Delta y = 0.25$. There will thus be one observation, i.e. $\mathcal{Y} = \boldsymbol{Y}$. The covariance function of the Gaussian eigenfunction parameters $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\gamma}})$ where the matrix elements have the value

$$C_{lml'm',\gamma} = \zeta_{a,\boldsymbol{\gamma}}((l,m),(l',m')) = \gamma_{lm}\delta_{ll'}\delta_{mm'}$$

where the vector $\boldsymbol{\gamma}$ consists of coefficients $\gamma_{lm} = \frac{1}{lm}$ and $\delta_{ij} = 1_{\{i=j\}}$ is the Kronecker-delta function. Then the Gaussian random field can be generated using

$$X(\boldsymbol{t}_j) = \sum_{l=1}^{5}\sum_{m=1}^{5} a_{lm}e^{-i\pi(lx_j/L+my_j/M)}, \ \ \forall j = 1, ..., N \ \ \boldsymbol{t_j} = (x_j, y_j), \tag{55}$$

or equivalently $\boldsymbol{X} = \boldsymbol{E}\boldsymbol{a}$. This means we have $n = 25$ distinct eigenfunctions. To simulate the observed data the noise has to be added, since $\boldsymbol{Y} = \boldsymbol{E}\boldsymbol{a} + \boldsymbol{\epsilon}$. The noise factor will be different for all examples; its value will be assigned before every example.

For the Metropolis-Hastings algorithm in this section, the same assumptions have been made about the transition kernel/density as in the previous section on applying MH on the first model. Note that this (second) model will be way more efficient when applying the MH algorithm, since its speed is affected less by the number of pixels in comparison to the first model. This is because the dimension of the covariance matrix $\boldsymbol{C}_{\boldsymbol{\gamma}}$ of the second model only grows when the number of eigenfunctions increases. Not when the number of pixels increases as in the first model. This can be convenient when dealing with larger observations like the CMB data.

8.3.1 MH algorithm on model with Gaussian eigenfunction coefficients

Using the posterior (42) the acceptance probability, given parameters $\boldsymbol{\rho}_i = \{\boldsymbol{a}_i, \boldsymbol{\gamma}_i, \nu_i^2\}$, will look like:

$$A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1}) = \min\left\{\frac{p(\boldsymbol{Y} \mid \boldsymbol{a}_{i+1}, \nu_{i+1}^2)p(\boldsymbol{a}_{i+1} \mid \gamma_{i+1})\pi(\boldsymbol{\gamma}_{i+1})\pi(\nu_{i+1}^2)}{p(\boldsymbol{Y} \mid \boldsymbol{a}_i, \nu_i^2)p(\boldsymbol{a}_i \mid \gamma_i)\pi(\boldsymbol{\gamma}_i)\pi(\nu_i^2)}, 1\right\}$$

The prior of $\boldsymbol{\gamma}$ will be defined in a similar fashion as the prior for $\boldsymbol{\eta}$ or $\boldsymbol{\theta}$ as in the previous subsections. We define $\pi(\gamma_{lm}) = \text{Exp}(\gamma_{lm} \mid \lambda_{lm})$, where λ_{lm} is the exponential rate for the coefficients γ_{lm}, where in this example $\lambda_{lm} = 1/lm$. The prior of ν^2 will remain the same,but with different $\mu_{\nu^2} = 5$ and $\sigma_{\nu^2} = 1$. This means for the acceptance probability we have the functions:

$$\begin{aligned} p(\boldsymbol{Y} \mid \boldsymbol{a}, \nu^2) &= \mathcal{N}(\boldsymbol{Y} - \boldsymbol{E}\boldsymbol{a} \mid \boldsymbol{0}, \nu^2\boldsymbol{I}) \\ p(\boldsymbol{a} \mid \gamma) &= \mathcal{N}(\boldsymbol{a} \mid \boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\gamma}}) \\ \pi(\gamma_{lm}) &= \text{Exp}(\gamma_{lm} \mid \lambda_{lm}) \\ \pi(\nu^2) &= \mathcal{N}(\nu^2 \mid \mu_{\nu^2}, \sigma_{\nu^2}^2). \end{aligned}$$

$\boldsymbol{C}_{\boldsymbol{\gamma}}$ is the Gaussian covariance matrix, defined by the function $\zeta_{a,\boldsymbol{\gamma}}$. The real value of the noise factor will be $\nu^2 = 5$.

Then the transition kernel will be $\boldsymbol{\rho}_{i+1} \sim \mathcal{N}(\boldsymbol{\rho}_i, \beta^2 \boldsymbol{I})$ where $\beta^2 = 0.02^2$. Now given $T = 25000$ and $B = 500$ we can start sampling from the posterior distribution given the MH algorithm above, starting at the initial value

$$\boldsymbol{a}_0 = \boldsymbol{0}, \quad \gamma_{lm,0} = 1, \quad \nu_0^2 = 1, \quad \forall l, m = 1, ..., n = 5.$$

A trace plot is shown, in Figure 22, where only the parameter ν^2 is shown next to its real value in red and posterior mean in green. The purple line represents the posterior mean upto the current point.

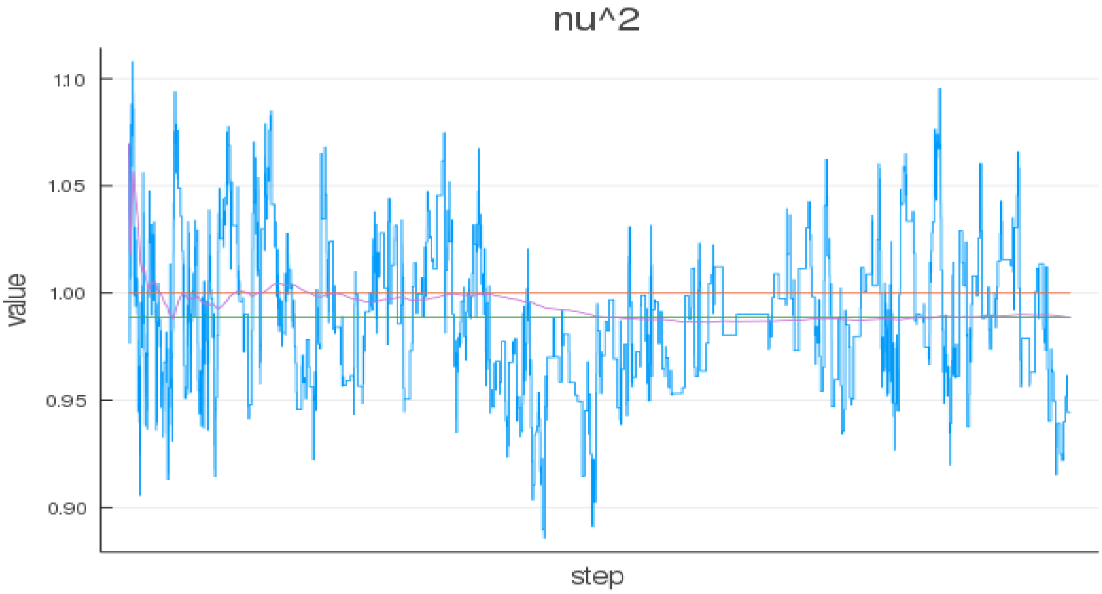


Figure 22: Trace plots of the parameter ν^2.

The estimated eigenfunctions coefficients $\hat{\boldsymbol{a}}$ generate the estimated Gaussian random field $\hat{\boldsymbol{X}} = \boldsymbol{E}\hat{\boldsymbol{a}}$. The real random field $\boldsymbol{X}$ and estimated random field $\hat{\boldsymbol{X}}$ are compared in Figure 23.

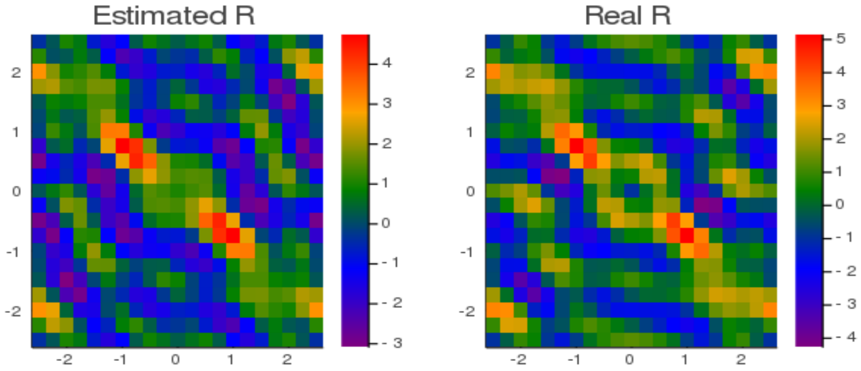


Figure 23: Random field given the posterior mean of the random coefficients, acquired from the trace plot; plotted against the random field given the real coefficients for the given example.

Since we have 51 parameters which is an extensive amount of work to put in a table, we describe the precision of the estimates by the Mean Squared Error (MSE), which is defined in Bijma et al. (2017) (Section 3.5) and slightly altered for our example as the following formula:

$$\mathrm{MSE} = \mathbb{E}_{\boldsymbol{\rho}}((\hat{\boldsymbol{\rho}} - \boldsymbol{\rho})^2) = \frac{1}{d}\sum_{i=1}^{d}(\hat{\boldsymbol{\rho}}_i - \boldsymbol{\rho}_i)^2. \tag{56}$$

Here $\hat{\boldsymbol{\rho}}$ is an estimate vector of the real values of the parameter vector $\boldsymbol{\rho} \in \mathbb{R}^d$. A convention is that the estimation gets more accurate as the MSE decreases.

For this simulation of the MH algorithm we have acquired some estimates and MSE's for parameter vectors $\boldsymbol{\rho}, \boldsymbol{a}$ and $\boldsymbol{\gamma}$, which are put in Table 3.

parameter	real value (red line)	posterior mean (green line)	variance of post. mean
ν^2	5.0000	4.6801	0.0467

parameter vector	MSE(.)
$\boldsymbol{\rho}$	0.1697
$\boldsymbol{a}$	0.0013
$\boldsymbol{\gamma}$	0.3448

Table 3: Table corresponding to the parameter trace plot in Figure 22 and the MSE's for several parameter vectors.

8.3.2 MH algorithm on model with mapped, non-Gaussian eigenfunction coefficients

Using the same example again of mapped, non-Gaussian eigenfunction coefficients. We generate lognormal eigenfunction coefficients. This means $\boldsymbol{\theta} = \{\mu, \sigma\}$ and

$$b = g_{\boldsymbol{\theta}}(X(\boldsymbol{t})) = g_{\mu,\sigma}(a) = e^{\mu+\sigma a},$$

with for this example $\mu = -1$ and $\sigma = 1$. The two mapping parameters have to be estimated as well in the MH algorithm. For this example using the posterior (46) the acceptance probability, given model parameters $\boldsymbol{\rho}_i = \{\boldsymbol{b}_i, \boldsymbol{\gamma}_i, \boldsymbol{\theta}_i, \nu_i^2\}$, will be defined as:

$$A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1}) = \min\left\{\frac{p(\boldsymbol{Y} \mid \boldsymbol{b}_{i+1}, \nu_{i+1}^2)p(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{b}_{i+1}) \mid \boldsymbol{\gamma}_{i+1}, \boldsymbol{\theta}_{i+1})\det[\mathcal{J}(\boldsymbol{b}_{i+1} \mid \boldsymbol{\theta})]\pi(\boldsymbol{\gamma}_{i+1})\pi(\boldsymbol{\gamma}_{i+1})\pi(\nu_{i+1}^2)}{p(\boldsymbol{Y} \mid \boldsymbol{b}_i, \nu_i^2)p(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{b}_i) \mid \boldsymbol{\gamma}_i\boldsymbol{\theta}_i)\det[\mathcal{J}(\boldsymbol{b}_i \mid \boldsymbol{\theta})]\pi(\boldsymbol{\gamma}_i)\pi(\boldsymbol{\theta}_i)\pi(\nu_i^2)}, 1\right\}$$

The prior of $\boldsymbol{\theta}$ is the same as in the previous example for the first model, with prior mean $\tilde{\boldsymbol{\theta}} = [-1; 1]$. For the other priors we assume $\lambda_{lm} = 1/lm$, $\mu_{\nu^2} = 5$ and $\sigma_{\nu^2} = 1$. Hence we have the densities:

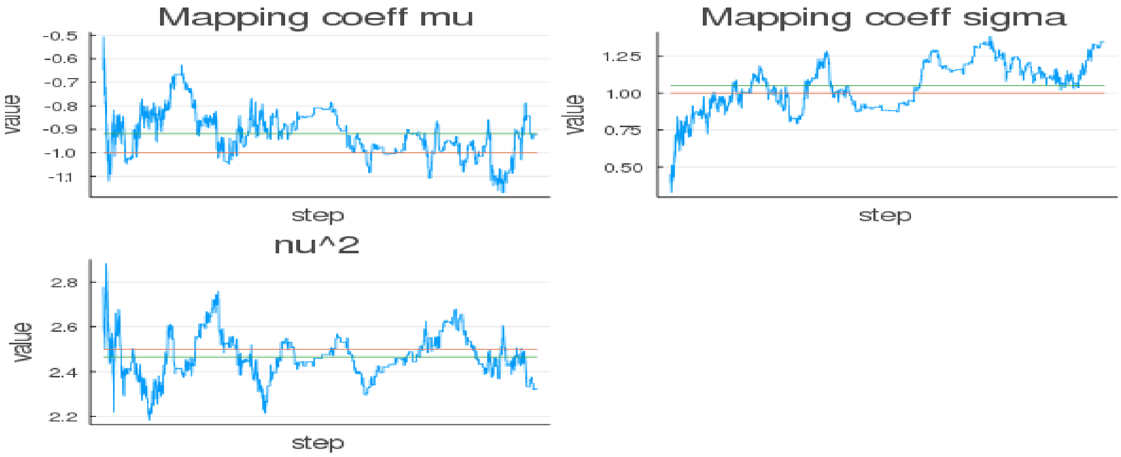


Figure 24: Trace plots which seperates the parameters $\boldsymbol{\theta} = \{\mu, \sigma\}$ and ν^2. The blue line represents the path, the red line represents the real value used in the simulation and the green line is the posterior mean for the path.

$$
\begin{aligned}
p(\boldsymbol{Y} \mid \boldsymbol{b}, \nu^2) &= \mathcal{N}(\boldsymbol{Y} - \boldsymbol{E}\boldsymbol{b} \mid \boldsymbol{0}, \nu^2 \boldsymbol{I}) \\
p(\boldsymbol{b} \mid \boldsymbol{\gamma}, \boldsymbol{\theta}) &= \mathcal{N}(g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{b}) \mid \boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\gamma}}) \\
\pi(\gamma_{lm}) &= \mathrm{Exp}(\gamma_{lm} \mid \lambda_{lm}) \\
\pi(\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{\theta} \mid \tilde{\boldsymbol{\theta}}, \boldsymbol{v_\theta} \boldsymbol{I}) \\
\pi(\nu^2) &= \mathcal{N}(\nu^2 \mid \mu_{\nu^2}, \sigma^2_{\nu^2}) \\
\mathcal{J}(\boldsymbol{b}, \boldsymbol{\theta}) &= \left[\frac{\partial g_{\boldsymbol{\theta}}^{-1}(\boldsymbol{r})}{\partial \boldsymbol{r}} \right]_{\boldsymbol{r} = \boldsymbol{b} | \boldsymbol{\theta}}
\end{aligned}
$$

For this example, the real value of the noise factor $\boldsymbol{C}_{\boldsymbol{\gamma}}$ is again the Gaussian covariance matrix, defined by the function $\zeta_{a,\boldsymbol{\gamma}}$. The real value of the noise factor, in this example, will be $\nu^2 = 2.5$.

Now for the MH algorithm; transition kernel will be $\boldsymbol{\rho}_{i+1} \sim \mathcal{N}(\boldsymbol{\rho}_i, \beta^2 \boldsymbol{I})$ where $\beta^2 = 0.02^2$. Now given $T = 25000$ and $B = 500$ we can begin sampling given the initial values

$$b_{lm,0} = 1, \quad \gamma_{lm,0} = 1, \quad \boldsymbol{\theta}_0 = [-0.5; 1], \quad \nu_0^2 = 1, \quad \forall l, m = 1, ..., n = 5.$$

A trace plot is shown in Figure 24, which groups the parameters $\boldsymbol{b}, \boldsymbol{\gamma}, \boldsymbol{\theta}$ and ν^2. For $\boldsymbol{\theta} = \{\mu, \sigma\}$ and ν^2 the real and estimated values are also plotted in Figure 24.

The estimated eigenfunction coefficients $\hat{\boldsymbol{b}}$ generate the estimated random field $\hat{\boldsymbol{R}} = \boldsymbol{E}\hat{\boldsymbol{b}}$. The real random field $\boldsymbol{R}$ and estimated random field $\hat{\boldsymbol{R}}$ are compared in Figure 25. Furthermore Table 4 shows the estimates in comparison to their real values, as well as some Mean Squared errors of the parameter vectors $\boldsymbol{\rho}, \boldsymbol{b}$ and $\boldsymbol{\gamma}$.

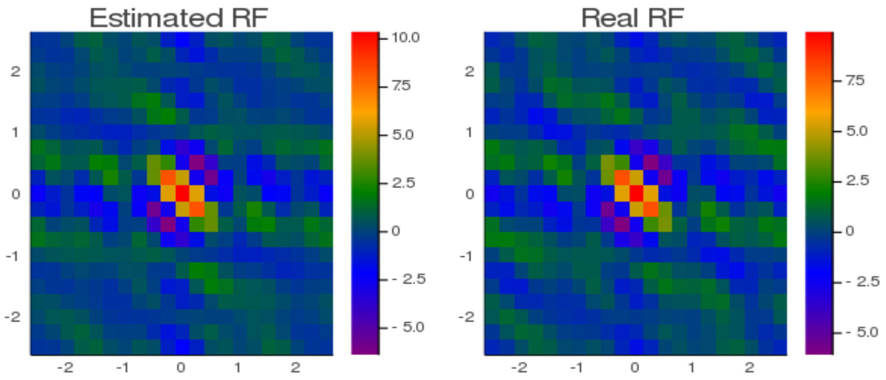


Figure 25: Random field given the posterior mean, acquired from the trace plot; plotted against the random field given the real coefficients for the given example.

parameter	real value (red line)	posterior mean (green line)	variance of post. mean
μ	-1.0000	-0.9187	0.0093
σ	1.0000	1.0508	0.0279
ν^2	2.5000	2.4651	0.0096

parameter vector	MSE(.)
$\boldsymbol{\rho}$	0.1562
$\boldsymbol{b}$	0.2108
$\boldsymbol{\gamma}$	0.1120

Table 4: Table corresponding to the parameter trace plot in Figure 24 and the MSE's for several parameter vectors.

8.3.3 MH algorithm on model with other non-Gaussian eigenfunction coefficients

The final example of applying the MH algorithm will be on non-mapped, non-Gaussian eigenfunction coefficients. The same example is used, namely a Student-t_v distribution. For the example we take that $v = 10$ to simulate the t_v random field. The noise factor for this model will be $\nu^2 = 2.5$ The acceptance probability, with model parameters $\boldsymbol{\rho}_i = \{\boldsymbol{b}_i, \boldsymbol{\gamma}_i, v_i, \nu_i^2\}$, of the algorithm will look like:

$$A(\boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1}) = \min\left\{\frac{p(\boldsymbol{Y} \mid \boldsymbol{b}_{i+1}, \nu_{i+1}^2)p(\boldsymbol{b}_{i+1} \mid \boldsymbol{\gamma}_{i+1}, v_{i+1})\pi(\boldsymbol{\gamma}_{i+1})\pi(v_{i+1})\pi(\nu_{i+1}^2)}{p(\boldsymbol{Y} \mid \boldsymbol{b}_i, \nu_i^2)p(\boldsymbol{b}_i \mid \boldsymbol{\gamma}_i, v_i)\pi(\boldsymbol{\gamma}_i)\pi(v_i)\pi(\nu_i^2)}, 1\right\}$$

The prior for v is the same as used in Section 8.2.3,but here we use $\mu_v = 10$ and $\sigma_v^2 = 0.5$. The prior for the noise factor will be the same, but with different values $\mu_{\nu^2} = 2.5$ and $\sigma_{\nu^2}^2 = 0.5$. The prior of $\boldsymbol{\gamma}$ will remain the same. The list of densities inside the probability are then

$$
\begin{aligned}
p(\boldsymbol{Y} \mid \boldsymbol{b}, \nu^2) &= \mathcal{N}(\boldsymbol{Y} - \boldsymbol{E}\boldsymbol{b} \mid \boldsymbol{0}, \nu^2 \boldsymbol{I}) \\
p(\boldsymbol{b} \mid \gamma) &= \mathcal{ST}(\boldsymbol{b} \mid \boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\gamma}}, v) \\
\pi(\gamma_{lm}) &= \text{Exp}(\gamma_{lm} \mid \lambda_{lm}) \\
\pi(v) &= \mathcal{N}(v \mid \mu_v, \sigma_v^2) \\
\pi(\nu^2) &= \mathcal{N}(\nu^2 \mid \mu_{\nu^2}, \sigma_{\nu^2}^2),
\end{aligned}
$$

where the covariance matrix $\boldsymbol{C}_{\boldsymbol{\gamma}}$ is defined by the function $\zeta_{a,\boldsymbol{\gamma}}$. The transition kernel will be $\boldsymbol{\rho}_{i+1} \sim \mathcal{N}(\boldsymbol{\rho}_i, \beta^2 \boldsymbol{I})$ where $\beta^2 = 0.02^2$. Now given $T = 25000$ and $B = 500$ we can start sampling starting at the initial values

$$\boldsymbol{b}_0 = \boldsymbol{0}, \gamma_{lm,0} = 1, v_0 = 9, \nu_0^2 = 1.$$

The real random field $\boldsymbol{R} = \boldsymbol{E}\boldsymbol{b}$ and estimated random field $\hat{\boldsymbol{R}} = \boldsymbol{E}\hat{\boldsymbol{b}}$ are compared in Figure 27.

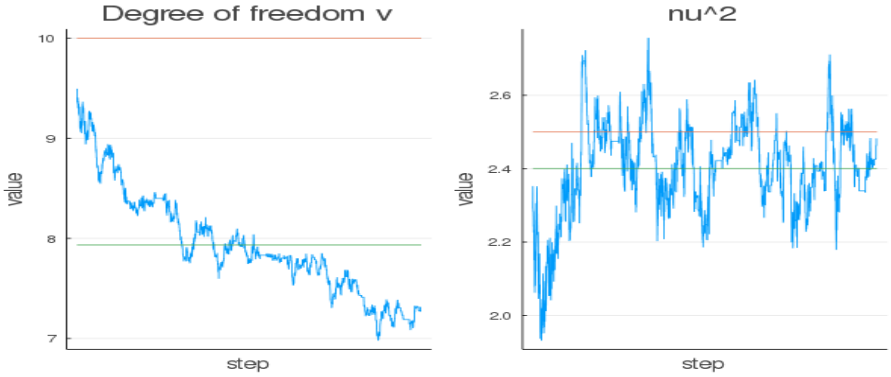


Figure 26: Trace plots which seperates the parameters v and ν^2. Tthe blue line represents the path, the red line represents the real value used in the simulation and the green line is the posterior mean for the path.

The estimators of the parameters and their real values are represented in Table 5. Furthermore the accuracy of the parameter vectors $\boldsymbol{\rho}, \boldsymbol{b}$ and $\boldsymbol{\gamma}$, given by the MSE, are in the same table. The estimator of the parameter v is inaccurate. Possibly we need more iterations of the MH algorithm in order to get the posterior mean of the parameter v closer to the real value. What also could be the case is that the model is unidentifiable. Further investigation is mandatory to figure this out.

The estimation of the random field $\boldsymbol{E}\hat{\boldsymbol{b}}$ is really accurate. The estimated and the real random field in Figure 27 look almost identical. This is because the MSE of the eigenfunction coefficients $\boldsymbol{b}$ is really close to zero.

In this section we have used the Metropolis-Hastings algorithm for estimating parameters, using BPE, for a given CMB model. The only aspect that changes when using the MH algorithm among different models is the calculation of the acceptance probability. This is probability is defined using the posterior distribution and will determine whether a new proposal is accepted or rejected. One convenient trademark from this acceptance probability is that it cancels out some probabilities which are hard to calculate ($p(\boldsymbol{Y})$). Using this algorithm one could calculate Bayes

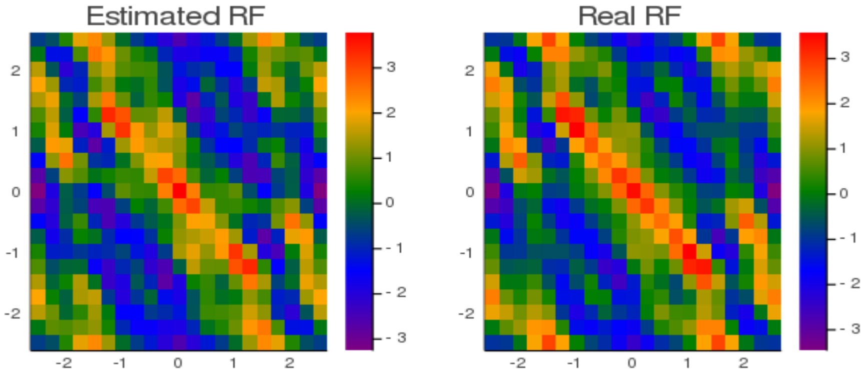


Figure 27: Random field given the posterior mean of the random coefficients, acquired from the trace plot, plotted against the random field given the real coefficients for the given example.

parameter	real value (red line)	posterior mean (green line)	variance of post. mean
v	10.000	7.9346	0.2746
ν^2	2.5000	2.4004	0.0160

parameter vector	MSE(.)
$\boldsymbol{\rho}$	0.2559
$\boldsymbol{b}$	0.0205
$\boldsymbol{\gamma}$	0.3407

Table 5: Table corresponding to the parameter trace plot in Figure 26 and the MSE's for several parameter vectors.

estimates (e.g. posterior means) for the parameters of the model. For example by calculating the sample mean of the 'walk' through the parameter space P. Those walks are visualised in trace plots. The models that were estimates here were pre-simulated by ourselves. This algorithm can be applied to the real CMB data using our two models.

For the next sections this algorithm will be applied to data which consists of pixels that will be thrown away from the data set. Without these pixels we can still use this algorithm to estimate the value of these eliminated pixels.

9 Utilisation of masks on the CMB models

The data of the Cosmic Microwave Background (Figure 1) can be modeled using the two models created in the previous sections of this paper (Section 4 and 5). We create the observed data ourselves by simulating the models. We did not use the real observed data of the CMB. After the simulation the parameters of the assumed model were estimated using the Metropolis-Hastings algorithm (Section 8).

One thing to keep in mind is that outer space consists of a myriad of extraterrestrial factors which will distort or disrupt the observation of the CMB. Radiation emitted by stars can add some heat to some pixels or black holes can map the radiation to some other pixels. Even some pixels will be useless because a physical object, for example a planet or asteroid, is blocking that region in the night sky. To eliminate those pixels something called a mask is created. In Taylor et al. (2008) the model created also consisted of a map, but it was not clear here if it was solely for elimination purposes.

9.1 Definition of a mask

A *mask* $\boldsymbol{M}$ is essentially a linear mapping (matrix) of the random field which eliminates a certain set of pixels, so that only the pixels with relevant information remain. Here $\boldsymbol{M} \in \mathbb{R}^{m\times N}$ is a matrix which keeps m out of N observations. This means that the observed data $\boldsymbol{Y}$ will now look like:

$$\boldsymbol{Y} = \boldsymbol{M}\boldsymbol{R} + \boldsymbol{\epsilon},$$

where the random field $\boldsymbol{R} \in \mathbb{R}^N$, observed data $\boldsymbol{Y} \in \mathbb{R}^m$ and $\boldsymbol{\epsilon} \sim \mathcal{N}_m(\boldsymbol{0}, \nu^2\boldsymbol{I})$.

The construction of $\boldsymbol{M}$ is as follows; first starting with a $N \times N$ identity matrix $\boldsymbol{I}$. If a pixel $\boldsymbol{t}_i$ for $i = 1, ..., N$ needs to be removed from the random field, then row i of the identity matrix has to be removed. This results in a $(N-1) \times N$ matrix. Iterate this process until all the irrelevant, unwanted pixels are removed and only m pixels remain. For example if we had a random field with ten pixels and we want to remove pixels 2,3,6,9,10, the mask would look like:

$$\boldsymbol{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Then $\boldsymbol{M}\boldsymbol{R}$ only keeps the wanted pixels of the random field (here pixels 1,4,5,7,8):

$$\boldsymbol{M}\boldsymbol{R} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R(\boldsymbol{t}_1) \\ R(\boldsymbol{t}_2) \\ \vdots \\ R(\boldsymbol{t}_9) \\ R(\boldsymbol{t}_{10}) \end{bmatrix} = \begin{bmatrix} R(\boldsymbol{t}_1) \\ R(\boldsymbol{t}_4) \\ R(\boldsymbol{t}_5) \\ R(\boldsymbol{t}_7) \\ R(\boldsymbol{t}_8). \end{bmatrix}$$

Essentially the mask we use only reduces the set of data points used for our model. Still BPE can be applied to this model, using the Metropolis-Hastings algorithm. The only difference will be that we do not have a rectangular grid, because of some removed pixels. This will create holes in the data. The covariance structure will remain the same when simulating a model using a mask. For later applications in this paper using masks, we will assume that the pixels that are removed form a rectangle. In the physical world this can represent obstructing by a planet or star.

9.2 Example of using a mask on simulated data

The example for this section consists of the grid $(-L : \Delta x : L)$ in the x-direction and $(-M : \Delta y : M)$ in the y-direction. For the example we take $L = M = 5$ and $\Delta x = \Delta y = 0.25$. Here the same covariance function $\zeta_{a,\gamma}$ is assumed for the Gaussian coefficients $\boldsymbol{a} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_\gamma)$ as in Section 5.2. The mask $\boldsymbol{M}$ will be rectangular and consists of all pixels that lie in $D = [-2, 3] \times [-2, 3]$. This means that the bottom left corner is $(-2, -2)$ and the top right corner is $(3, 3)$. Since $\Delta x = \Delta y = 0.25$, we have that 441 pixels out of 1681 have been removed from the observed data. Hence we have the model

$$\boldsymbol{Y} = \boldsymbol{M}\boldsymbol{E}\boldsymbol{a} + \boldsymbol{\epsilon},$$

where we used that $\nu^2 = 1$ for the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \nu^2\boldsymbol{I})$. The number of distinct eigenfunction are $n = 25$.

This is visualised in Figure 28 where the original observations and the observations with mask are illustrated.

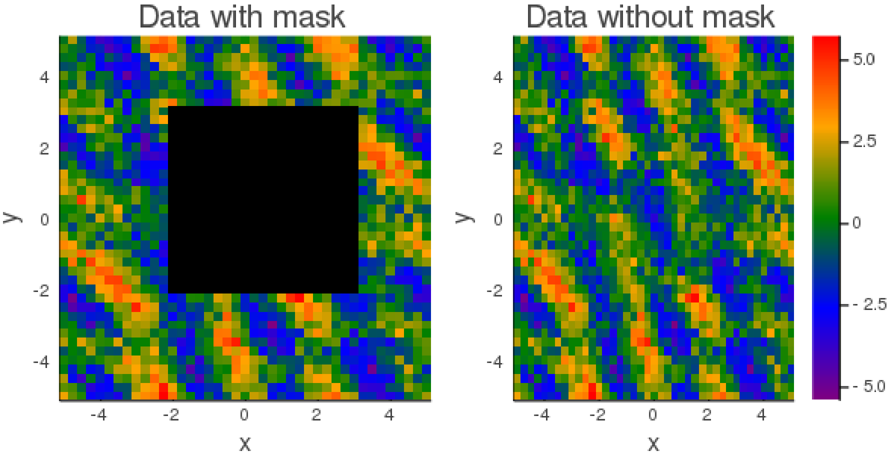
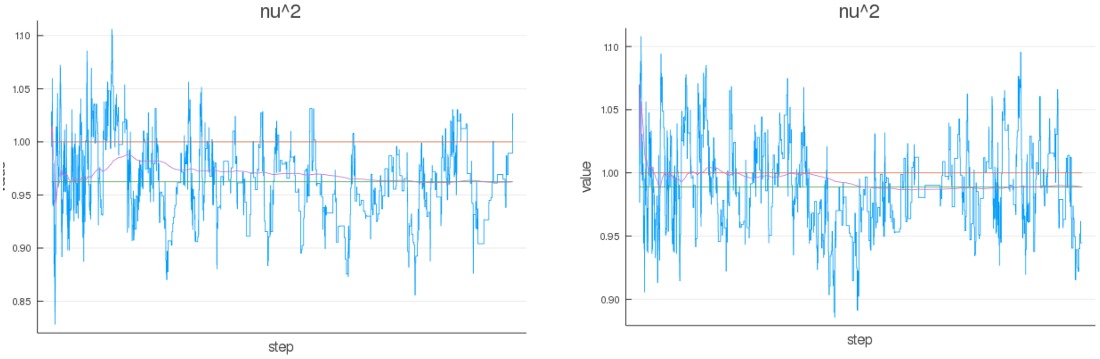


Figure 28: Generated observed data with and without rectangular mask. The mask is displayed with the black pixels.

9.3 MH algorithm on the example of a Gaussian random field with rectangular mask

Using the Metropolis-Hastings algorithm on this example is actually no different then in Section 8.3.1. Again we assume that we know the complete structure of the data, except for the real values of the parameters. Because of the presence of a mask, the only difference is that our data set has shrunk, because of the elimination of a specific selection of pixels. Here we will compare the performance of the MH algorithm on the data with and without mask to check if there are any noticeable difference in posterior density. Also the same MH scheme is used to calculate the acceptance probability, that is the posterior, priors, transition kernel and initial value all remain the same. The only difference is that we run it two times with two different data sets. One with mask and one without mask.

In Figure 29, the trace plot of parameter ν^2 is shown separately for both the example with and without mask. Also the real value and posterior mean estimate is shown in the trace plot.



(a) With mask

(b) Without mask

Figure 29: Trace plots for ν^2 the example of all parameters with mask and without mask using the Metropolis-Hastings algorithm. Red line represents the real value, green line represents the posterior mean, the purple line represents the mean of all values of the walk upto the current state.

As well as an extra line (purple) which calculates the mean of all previous values upto its current value.

In order to visualise the estimates for $\boldsymbol{a}$ we can generate the random field $\hat{\boldsymbol{R}}_{\boldsymbol{M}} = \boldsymbol{E}\hat{\boldsymbol{a}}_{\boldsymbol{M}}$ with mask and $\hat{\boldsymbol{R}} = \boldsymbol{E}\hat{\boldsymbol{a}}$ without mask and compare it with the real random field $\boldsymbol{R} = \boldsymbol{E}\boldsymbol{a}$. This is shown in Figure 30. Hence three different random fields (but similar looking) are shown. Two of those are build on the estimates calculated by the two simulations of the MH algorithm. The other one is the real random field generated by the real values of $\boldsymbol{a}$.

Both the MH algorithms estimated the random field well as seen in Figure 30. The random field without mask seems to be estimated better when comparing it to the real random field. To see the differences in estimations we put all the data in Table 6 .

parameter	real value (red line)	posterior mean (green line)	variance of post. mean
ν^2 with mask	1.0000	1.0351	0.0017
ν^2 without mask	1.0000	1.0188	0.0012

parameter vector	MSE(.) on mask	MSE(.) on no mask
$\boldsymbol{\rho}$	0.0570	0.1697
$\boldsymbol{a}$	0.0059	0.0013
$\boldsymbol{\gamma}$	0.1103	0.3448

Table 6: Table corresponding to the parameter trace plot in Figure 26 and the MSE's for several parameter vectors.

The table shows us that the estimation using the posterior mean of the parameter ν^2 is indeed better for the model without mask. Furthermore the estimation of the parameter vector $\boldsymbol{a}$ is more accurate for the model without mask, which is also substantiated visually in Figure 30 .

An interesting result arises when we compare the MSE's of the parameter vector $\boldsymbol{\gamma}$ for the two different cases. Using the MSE, Table 6 states that these parameters of the model with mask are better estimated than for the model without mask. This sounds debatable since there are more observations in the model without mask than with mask. This phenomena will probably disappear when we take more iterations in the MH algorithm. Because of this result also the

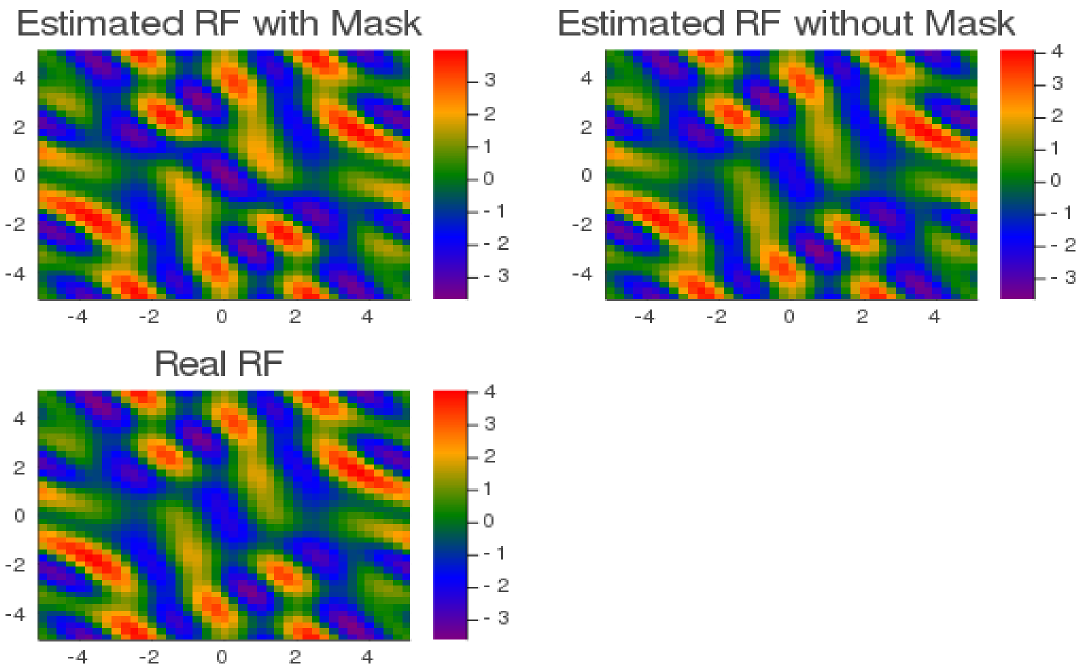


Figure 30: Real random field against the estimated random field for the MH algorithm on the data with and without mask.

MSE for the parameter vector $\boldsymbol{\rho}$, which indicates all parameters, of the model with mask is more accurate than for the model without mask. This is a counter-intuitive result.

In this section the concept of a mask was introduced. The way that a mask affects the data is by eliminating uninformative pixels. Here we checked the difference in estimation using the MH algorithm for the model with and without mask. For this example the acquired parameter estimates of the observed data with and without mask were almost similar. The two estimated random fields almost look similar, which indicates that the MH algorithm performs well for this example. This section only compared the performance of the parameter estimation using the MH algorithm between two models. In the next section this will be taken one step further. We will calculate parameter estimates for the same observed data multiple times for a number of different masks, which will be applied independently to the same data set.

10 Performance analysis on parameter estimation among various mask sizes

An interesting topic on calculating the parameters of the model will be how the estimation of these parameters is affected by the presence of a mask with a given size. In this section we will just assume that the mask on the data is a single rectangle with four given vertices. Each vertex is assigned to a pixel in the field. For a rectangle only two non-adjacent vertices have to be assigned, for example bottom left corner and top right. This is enough to create the rectangle.

In Section 9 we have caught sight of a comparison of the performance of the MH algorithm used for parameters estimation on the same model with and without mask. In this section the rectangular mask size will be altered multiple times, increased in small steps. Each different mask size will then be compared with some measure that conveys the parameter estimation performance using the MH algorithm.

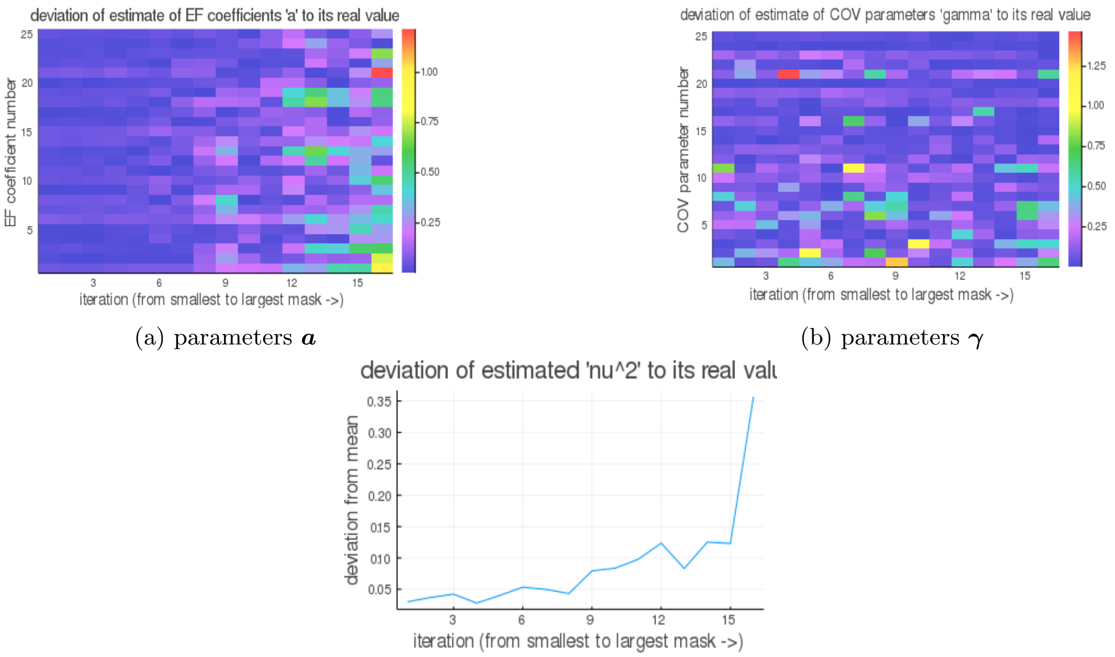
10.1 Rectangular mask size against posterior variance for second model example using MH algorithm

The performance of the posterior density given the observed data can be represented using the posterior variance. To illustrate this we will just take one example of a random field which is generated by Gaussian eigenfunction coefficients. This is the same example as in section 5.2. The number of available pixels inside the data shrinks as the mask size increase. Still we want to use MH algorithm to estimate the parameters of the model, even if there is a mask. Here the posterior variances of the parameters will be plotted against it iteration (this means the given mask size/available data points). In this context the beginning of a new iteration means that we are running the MH algorithm again for a new mask size.

The grid for this section consists of coordinates $(-L : \Delta x : L)$ in the x-direction and $(-M : \Delta y : M)$ in the y-direction. For the example we take $L = M = 4$ and $\Delta x = \Delta y = 0.25$. Also the same random field with Gaussian eigenfunction coefficients is assumed as in Section 9. In total we will have 16 different mask sizes, including the case where we apply no mask. Per iteration we let the MH algorithm run for the given observations, which depend on the mask. This means that for every iteration we obtain the posterior mean and variances/standard deviations of all parameters. We will compare the deviation of the estimated parameters to its real values per parameter per iteration in Figure 31. An important note is that for every iteration we will have the same initial value for the MH algorithm, as well as the same transition kernel/density.

Also the standard deviation of all parameters at any iteration is given in Figure 33. We have seperated the parameters in their corresponding group. The groups are $\boldsymbol{a}, \boldsymbol{\gamma}$ and ν^2. The heatmap represents the value of the standard deviation or the deviation of the estimate from the real value. On the x-axis we have the indices of the parameters and on the y-axis we have the iteration. Here the iterations start from no mask, to the largest mask. The results in the heatmaps and line plot are as expected. For Figure 31 we see that the smaller the mask size (low iteration) the closer the estimate gets to the real value. For the $\boldsymbol{\gamma}$ parameters this is less convincingly, because its heatmap is more ‘cloudy’. For the parameters $\boldsymbol{a}$ and ν^2 we see a clear increase of the deviation of the posterior mean from the real value as the iterations begin to go up. This deviation is defined as the absolute value of the difference between the real and estimated value.

Another measure for the performance of the parameter estimation is the Mean Squared Error (56). For the example in Figure 32 we used that the estimator is the posterior mean calculated by the MH algorithm. We can see in these three subfigures that there is a clear trend between the MSE of the total parameter vector and the iteration. The larger the mask size the higher



(a) parameters $\boldsymbol{a}$

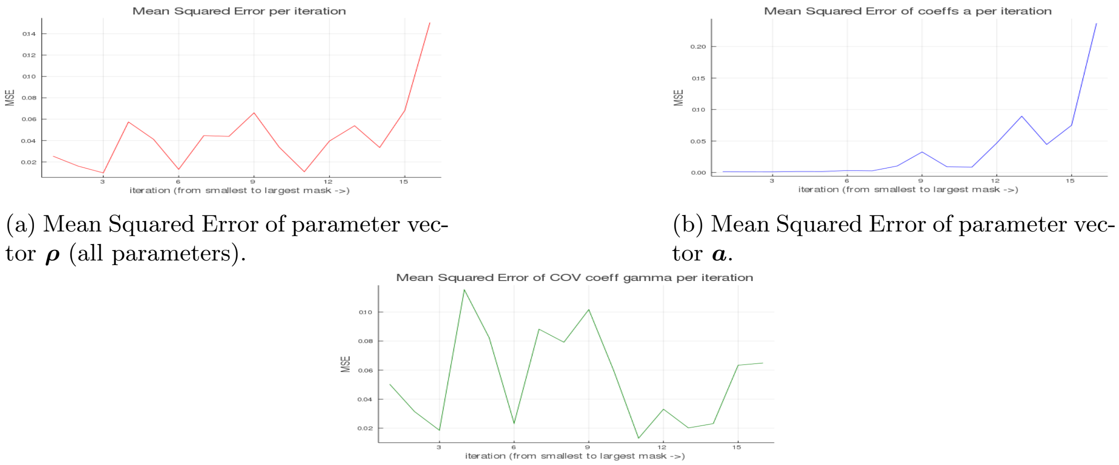
(b) parameters $\boldsymbol{\gamma}$

(c) parameters $\boldsymbol{\gamma}$

Figure 31: Heatmaps and line plot of parameters $\boldsymbol{a}, \boldsymbol{\gamma}, \nu^2$. For the heatmaps the color represents the deviation of a parameter estimate at a given iteration from its real value. For the line plot, the height represents the deviation varied over different iterations for the parameter ν^2.

the MSE for the total parameter vector. This is also the case for the MSE of the estimate of the eigenfunction coefficient vector $\hat{\boldsymbol{a}}$. The estimate of the vector covariance parameter $\hat{\boldsymbol{\gamma}}$ does not show a clear trend when the mask size increases. This can imply that this model is unidentifiable.

In combination with Figure 31 the standard deviation visualised in Figure 33 also seems to be lower as the mask size is smaller. Again for the parameters of $\boldsymbol{\gamma}$ the results are not convincing, because we do not see a clear relation between the colors and the iterations. For the other parameters the standard deviation drops as the number of the iteration decreases. Those two Figures combined tells us that the MH algorithm makes less accurate predictions, given its estimate is the posterior mean, when the mask size increases. For this example we particularly see this in the estimation of the parameters $\boldsymbol{a}$ and ν^2. When the mask size gets smaller the estimates seem to get closer to the real value, as well as its standard deviations decrease.

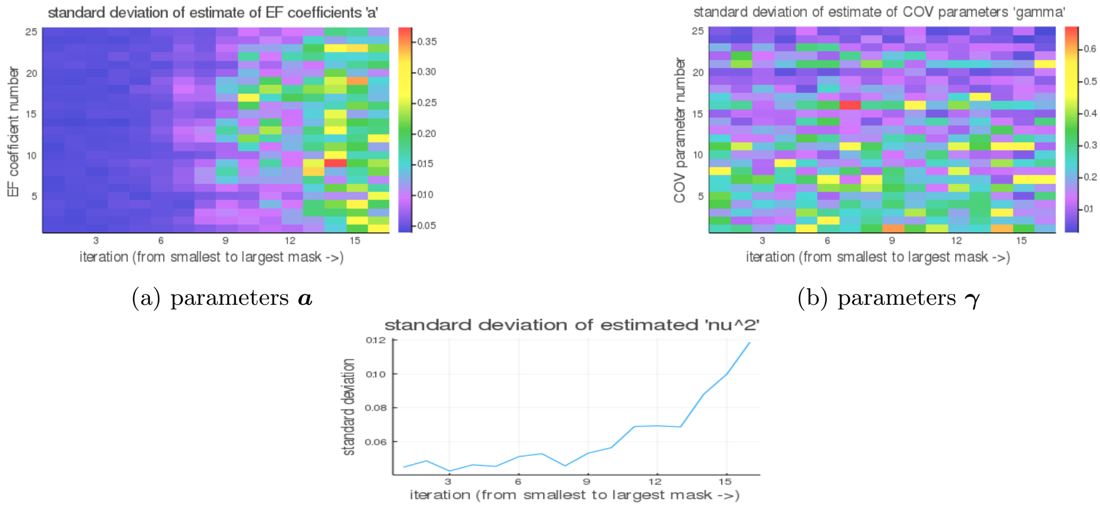


(a) Mean Squared Error of parameter vector $\boldsymbol{\rho}$ (all parameters).

(b) Mean Squared Error of parameter vector $\boldsymbol{a}$.

(c) Mean Squared Error of parameter vector $\boldsymbol{\gamma}$.

Figure 32: Comparison of the Mean Squared Error and its iteration, which is a single simulation of the MH algorithm. Every iteration assigns a different mask size.



(a) parameters $\boldsymbol{a}$

(b) parameters $\boldsymbol{\gamma}$

(c) parameter ν^2

Figure 33: Heatmaps and line plots of parameters $\boldsymbol{a}, \boldsymbol{\gamma}, \nu^2$ and the MSE. For the heatmaps the color represents the standard deviation of a parameter estimate at a given iteration. For the line plot, the height represents the standard deviation for the parameter ν^2 or MSE over different iterations.

As an extra aid we will plot the empirical probability density function of the parameter ν^2 for different simulations of the Metropolis-Hastings algorithm. Again a noticeable difference will arise, when we plot four different EPDF's given their iteration, which is shown in Figure 34.

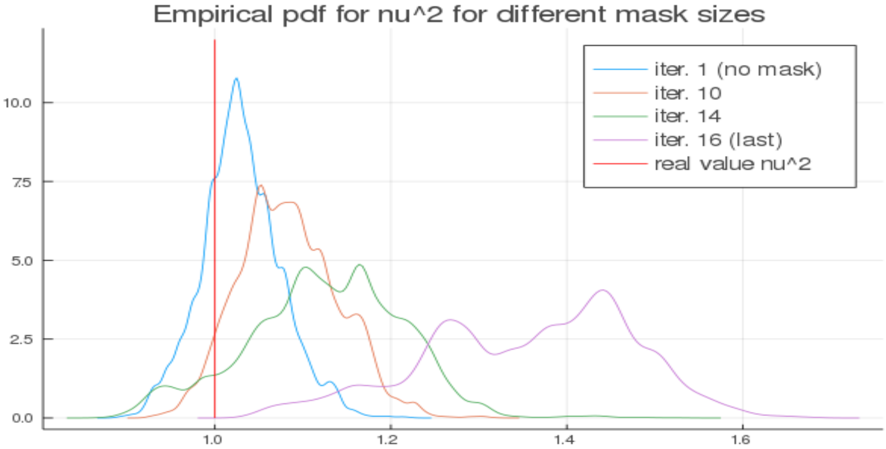


Figure 34: Empirical probability density functions for the parameter ν^2 given its iteration of the MH algorithm. The higher the iteration the bigger the mask. Also the real value of $\nu^2 = 1$ is shown as a vertical red line.

We see that as the iteration increases (mask size gets larger) the mode of the EPDF gets further away from the real value. Also the variance/standard deviation of the density increases. This can also be seen in Figure 31c and 33c, where these two values also increase as the iteration gets higher.

In Table 6 in Section 9 we saw that the accuracy of the estimation of ν^2 increased as the mask size decreased (no mask), but furthermore we saw that the estimates for $\boldsymbol{a}$ and $\boldsymbol{\gamma}$ of the model with no mask were less precise than the model with mask. In this section we have shown that the estimation of $\boldsymbol{\gamma}$ is less affected by the mask size, because there is no recognisable pattern illustrated in both of its heatmaps in Figure 31b and 33b. But for coefficients $\boldsymbol{a}$ there is a clear trend, shown in Figure 31a and 33a, when we increase the mask size and compare it to the two precision factors. This probably means that, in the previous section, we just had an unfortunate event for the estimation of the parameters $\boldsymbol{a}$. This occurance is always probable because of the random nature of the MH algorithm.

11 Summary

We started in this paper by defining the random fields, which are multivariate distributions with some covariance structure, constructed using a function, among all random variables. These random fields are the underlying structures of the two models we defined. These models, suggested by Vio et al. (2001) and Taylor et al. (2008), where used to simulate observed data which we would later use to compare estimates of the parameters of these models with their real values.

First we looked at Maximum likelihood estimation for parameter estimation. Analytically it was impossible to calculate the estimates. We could use the EM algorithm for the most trivial Gaussian models, but for the more difficult non-Gaussian models the log-likelihood function would consist of higher dimensional integrals. It is possible to calculate these, but for these examples more convenient Bayesian methods are available. Using Bayesian parameter estimation it is still impossible to calculate Bayes estimates analytically, but with a computational tool called the Metropolis-Hastings algorithm one could make the Bayesian parameter estimation work. The algorithm samples from the posterior density using some transition kernel and acceptance probability defined by the assumed prior distributions, likelihood functions and transition densities. This MH algorithm was used on all kinds of examples on the two distinct models. When the algorithm is finished, it creates trace from which we can calculate the estimates of the parameters. The results of these estimations were mostly accurate, except for one case which could imply that designated model is unidentifiable (Section 8.2.2).

Furthermore masks were introduced which eliminate irrelevant pixels from the observed data. Something we researched is the comparison of estimated posterior densities, using the MH algorithm, among the same observed data given different mask sizes. Intuitively the results were right. The more pixels were eliminated by the mask, the more uncertain and inaccurate the Bayes estimate got. This meant that the error with the real value compared to the posterior mean increased, as well as the posterior standard deviation got bigger as the size of the mask increased. This was visualised in a heatmap.

11.1 Further research

There are a lot of different routes one can take from here. In this paper we have only applied the MH algorithm to self-simulated data, but it is obvious that we want to apply this to the real observed data of the CMB. The models we have defined can be altered a little bit. For example the observation noise, which is now assumed independent noise, can be dependent. The physical interpretation behind this is that some spots in the night sky come along with more noise compared to other spots. A possibility is to assign some covariance matrix constructed by a function to the observation noise (similar to the random field covariance function). Another small modification can be to make the mask not only eliminate pixels, but also map the values of one pixel to the other. The physical motive is that radiation can be bend or morphed by heavy objects in outer space. This implies that some fraction of radiation from one pixel can be mapped to a set of other pixels.

Another assumption about the second model will be about its eigenfunction. In this paper we assumed that we had a flat surface for our random field. In reality the observations lie on a sphere. This means that spherical harmonics will be more suitable than regular exponentials. This is also suggested in Taylor et al. (2008).

Finally some other MCMC methods can be applied for the constructed models. Something like a Gibbs sampler takes some more work to program, but will have faster convergence to the real posterior density. Also the Hamiltonian Monte Carlo (HMC), discussed in Taylor et al. (2008), can be more opportune than the Metropolis-Hastings algorithm for these examples.

12 Layman's summary

The Cosmic Microwave Background raditiation (CMB) is observed by satellites. These observe radiation and convert it into heat, which is displayed in the heatmap of Figure 1. Because of its structure we wanted to create models, which had similarities with this CMB data. We started to design models for this observed data using some statistical models, which were built on a concept called random fields. Random fields are a set of random variables, which influence each other (covariance) in a structural fashion. This structure also depended on some covariance parameters. The random fields formed the underlying structure of the observed data models. We looked at the most trivial, Gaussian models and some more complex, non-Gaussian models. When the models increased in complexity the number of different parameters also increased.

For a given set of observed data like the CMB data we want to estimate these parameters of these different examples among the two models to obtain the best possible model. Two parameter estimation methods were introduced, were we only used one to estimate parameters called the Bayesian parameter estimation (BPE). Analytically these parameter estimates could not be acquired, so we had to use some algorithm to calculate the estimates numerically. This algorithm is called the Metropolis-Hastings algorithm. Using this algorithm we were able to estimate the parameters of all our self-simulated observed data, which we were then able to compare with the real values of the parameters.

Lastly masks were introduced. These are objects that eliminate a selected family of pixels, because we assume they are irrelevant. The increase of mask size on the data decreased our precision on parameter estimation of the underlying model for the observed data.

References

Bijma, F., Jonker, M., and van der Vaart, A. (2017). *An Introduction to Mathematical Statistics.* Amsterdam University Press, Amsterdam.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer, New York.

ESA and the Planck Collaboration (2013). Planck CMB. `https://www.esa.int/ESA_Multimedia/Images/2013/03/Planck_CMB#.YNnirGtbR80.link`.

Evans, R. (2015). *The Cosmic Microwave Background, How It Changed Our Understanding of the Universe.* Springer, New York.

Taylor, J., Ashdown, M., and Hobson, M. (2008). Fast optimal CMB power spectrum estimation with Hamiltonian sampling. *Mon. Not. R. Astron. Soc.*, 389:1284–1292.

Vio, R., Andreani, P., and Wamsteker, W. (2001). Numerical Simulation of Non-Gaussian Random Fields with Prescribed Correlation Structure. *Publications of the Astronomical Society of the Pacific*, 113:1009–1020.

GitHub repository

The code of this project is written in Julia and can be found in the following repository:
`https://github.com/leviklomp/BEP_project`