NTNU – Trondheim
Norwegian University of
Science and Technology

Quantification of an Approximate forward-backward Algorithm applied to a Convolutional Model

Trine-Lise P Lorentsen

Master of Science in Physics and Mathematics
Submission date: June 2014
Supervisor: Håkon Tjelmeland, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Preface

This master thesis is the conclusive work of the advanced engineering/master's degree program; "Applied Physics and Mathematics" (MTFYMA) at the Norwegian university of science and technology (NTNU). The thesis is written under the main profile course "Industrial Mathematics" and is a statistics master. It is part of the tenth and final semester of the master program, and counts for 30p of 300p. Industrial mathematics is given under the Department of Mathematical Sciences at NTNU.

First of all I would like to say thank you to my supervisor for all the help he has given me on this thesis. It has been a privilege. I would also like to thank my family for their love and support, you are truly amazing. Driving 500 km just to babysit for a couple of days, before making the return trip says it all. Most of all I would like to say thank you to Frank and Nikolai. If not for you, I would not be handing in this thesis. I love you to the moon and back.

Trine-Lise Lorentsen
June 24, 2014

Sammendrag

I denne masteroppgaven er en tilnærmet forover-bakover algoritme for binære Markov felt kvantifisert for en konvolusjonerende Bayesiansk modell. Den Bayesianske modellen transformeres til sin unikt korresponderende energi funksjon bestående av binære variabler, hvor interaksjons parametrer definerer funksjonen.

Vi kvantifisere tilnærmingens kvalitet ved hjelp av en Metropolis-Hastings algoritme, hvor vi anvender den tilnærmede forover-bakover algoritmen og Metropolis-Hastings på en rekke syntetiske tilfeller. Resultatene viser at akseptratene øker når antallet maksimale naboer øker, noe som var å forvente. Høyeste akseptprosent ble funnet for tilfellene hvor støyen er økt i sannsynlighetstettheten, med en resulterende akseptprosent på $94,95\%$ for 10 naboer. De laveste akseptratene forekom for tilfellene med lite støy, og for førfordelingen modellert som en binære Markov-kjede resulterte dette i en akseptprosent på $8,03\%$. For dette sistnevnte tilfellet ble det også simulert tilnærminger uten bruk av Hastings-Metropolis algoritmen, og sammenlignet med den Bayesianske sluttfordelingen har disse to tilfellene omtrent samme marginale sannsynlighetsfordeling. Dette var også tilfelle når førfordelingen ble modellert som en Markov kjede med fire mulige tilstander. Dermed konkluderer vi med at den tilnærmede forover-bakover algoritmen gir gode resultater selv når Metropolis-Hastings genererer lave akseptrater.

Abstract

In this master thesis an approximated forward-backward algorithm for binary Markov random fields is applied to and evaluated for a convolutional Bayesian model. The Bayesian model is transformed into its unique corresponding energy function of binary variables, where interaction parameters defines the function.

We quantify the quality of the approximation by using an independent proposal Metropolis-Hastings algorithm, where we apply the approximation to a variety of synthetic test cases. The acceptance rates increases as the maximum number of neighbors increase, which was to be expected. Highest percentage was generated for a case with increased noise in the likelihood, with a resulting acceptance rate of 94.95% for 10 neighbors. The lowest acceptance rates were gained from low noise cases, and for the binary Markov chain prior an acceptance rate of 8.03% was registered. For this last mentioned case the approximation was also simulated without the use of the Metropolis-Hastings algorithm, and compared with the aposteriori, where these two cases have approximately the same marginal probabilities. The same was seen for the four state Markov chain prior. Thus we conclude that the approximated forward-backward algorithm is viable even when the Metropolis-Hastings algorithm generate low acceptance rates.

Contents

1 Introduction

In Rimstad and Omre (2013) a convolutional two-level hidden Markov chain, formed as a Bayesian model, is considered. The model has application to seismic inversion, where solving the inversion generates one dimensional lithology-fluid (LF) profiles. The convolution generates a dependency throughout the model that makes drawing from the distribution impossible, and thus approximations are made. Rimstad and Omre (2013) suggests three approximations for the likelihood of the model, and uses the forward-backward algorithm to solve the inversion. Testing involves synthetic test cases and a real data study. A similar convolutional Bayesian model is given in Ulvmoen and Hammer (2010), where the inversion is solved in a similar fashion. The likelihood is given an approximation and the inversion is also here solved by the forward-backward algorithm.

In the doctoral dissertation of Austad (2011) and corresponding article by Tjelmeland and Austad (2012) an approximated forward-backward algorithm for an energy functions of binary variables is developed. This approximation considers discrete variables in the form of distributions for binary Markov random fields, where the approximation is based upon minimizing the error sum of square for the interaction parameters of the energy function. Through their work, a transformations from Bayesian models to binary Markov random fields is discussed, but it is not more than mentioned as a possibility. Further, a transformation from a convolutional Bayesian model has never been done before.

We aim to quantify the quality of the above mentioned approximation for a convolutional Bayesian model. This thesis adopts the model presented in Rimstad and Omre (2013), and we transform it into the formulation of a binary Markov random field. This is done by expressing the posterior with interaction parameters for binary variables of the energy function. In our studies we consider both a two state Markov chain prior and a four state Markov chain prior. We adopt covariance matrices from synthetic test cases in Rimstad and Omre (2013), and adapt parameters to use in our evaluations. All cases are generated realizations for using the approximated forward-backward algorithm provided by Tjelmeland and Austad (2012), and we evaluate the approximation based on acceptance rates from an independent proposal Metropolis-Hastings algorithm.

The succeeding sections in this thesis are organized in the following way. Section 2 contains some of the statistical tools used throughout this thesis. The theory is in short form, but when needed references to complete work have been given. The problem description is found in Section 3, where the Bayesian model is first presented and is then followed by the transformation into the energy function. Section 4 contains the test cases used and the results for two Markov chain priors, one with two states and the other with four. Remarks, conclusions and further work areas is elaborated on in Section 5. The appendix contains complete calculations and equations for certain parts of the transformations, it also contains result plots of realizations for completeness of this thesis.

2 Statistics Theory

The content of this section is a brief description of some of the statistical tools used in this master thesis. For a thorough and complete elaboration the reader is referred to cited articles and books.

2.1 Bayesian Hidden Markov Model

In this section the concept of Bayesian modeling is introduced. This is presented for the reader to have an understanding of which parts a Bayesian model is built upon. A more thorough interpretation of the subject is given by Gamerman and Lopes (2006) and for a complete elaboration see Lee (2012). We conclude this section by introducing hidden Markov models (HMM). For an elaborated discussion on this subject see Blake et al. (2011).

The main interest of the Bayesian model is to establish a distribution for n unknown real-valued quantities, which is here denoted by the vector

$$\boldsymbol{\theta} = [\theta_1, \ldots, \theta_n]^T, \qquad n \geq 1. \tag{1}$$

The approach is to find an expression for the distribution of $\boldsymbol{\theta}$ given some observations. Before any observations are made, we assume $\boldsymbol{\theta}$ to have a probability distribution denoted by $\pi(\boldsymbol{\theta})$. This is known as a prior distribution, as it is an assumption from before observations are made. Note that the generic term $\pi(\cdot)$, is from here on out used as denotation for probability distributions.

Let $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ for $n \geq 1$ denote observations from an arbitrary stochastic process. The observations are assumed to be conditionally distributed given $\boldsymbol{\theta}$, i.e. $\mathbf{x} \sim \pi(\mathbf{x}|\boldsymbol{\theta})$. This is known as a likelihood. After observations are made, the posterior distribution is established. This is the distribution of $\boldsymbol{\theta}$ conditioned on the observations, $\mathbf{x}$. To obtain the posterior distribution, Bayes' theorem is used yielding

$$\pi(\boldsymbol{\theta}|\mathbf{x}) = \frac{\pi(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\mathbf{x})}, \tag{2}$$

where

$$\pi(\mathbf{x}) = \int \pi(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}. \tag{3}$$

Equation (3) is a hard to assess normalizing constant, which involves exhaustive calculation of all possible values of $\boldsymbol{\theta}$. The posterior model is however fully defined by the likelihood and the prior distribution, and thus the normalizing constant is often omitted when considering other aspects of the distribution. The posterior distribution is therefore often written as

$$\pi(\boldsymbol{\theta}|\mathbf{x}) \propto \pi(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}). \tag{4}$$

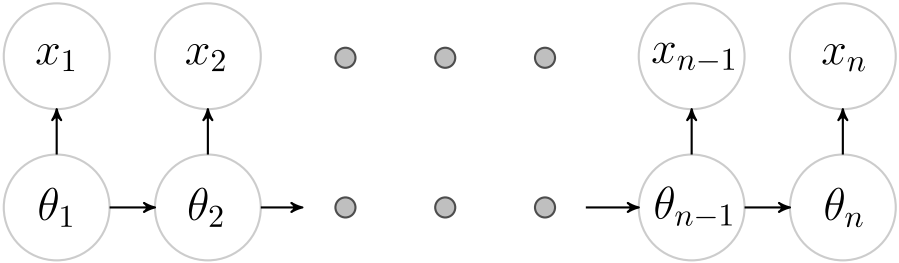


Figure 1: *First order hidden Markov model, illustrated as a directed graph to show dependencies. Here $\boldsymbol{\theta}$ is a Markov chain of assessment interest and $x_1, x_2, \ldots, x_n$ are observed quantities.*

Now, let us assume that $\boldsymbol{\theta}$ is a Markov chain, where a Markov chain is a stochastic process of transitions between states in a given state space. A Markov chain is entirely defined by a transition probability matrix. Let the chain posses a memoryless first-order Markov property, which means that the transition probability from time $i-1$ to i is independent of all previous states of the Markov chain. The transition probability is thus given by

$$\pi(\theta_i|\theta_{i-1}, \ldots, \theta_2, \theta_1) = \pi(\theta_i|\theta_{i-1}). \tag{5}$$

Let us further assume that the relationship between the observations in $\mathbf{x}$ and the Markov chain $\boldsymbol{\theta}$ is as illustrated in Figure 1. This is known as a first-order one level hidden Markov model, where the Markov chain is seen as something masked or as an underlying property of the observations, $\mathbf{x}$. The observations are in this HMM independent of one another when conditioned on the variables of the hidden Markov chain, i.e. if you consider elements i and j in $\mathbf{x}$ and $\boldsymbol{\theta}$, then x_i and x_j are independent of one another given θ_i and θ_j, $\forall i \neq j$. This condition is seen in the directed graph structure of the illustrated HMM.

Hidden Markov models are often of a more complex form and can consist of many layers and orders, e.g. the convolutional Bayesian model considered in Rimstad and Omre (2013) is for a two-level hidden Markov model.

2.2 Markov Random Fields

This section addresses the topic of Markov random fields (MRF). A great book on this subject is Blake et al. (2011), which also discusses Markov models on graphs in direct correspondence to HMMs, a topic we introduced in the previous section. We here follow the work of Austad (2011), which is based on the Hammersley-Clifford theorem and follows the work of Besag (1974). In the following section we adapt the notation of Austad (2011), and use it to define an energy function of binary variables with interaction parameters.

A Markov random field or MRF is an undirected graph representing stochastic variables with a Markov property. The main difference between a Bayesian model and an MRF is that the graph is undirected and may be cyclic in the MRF case, which the Bayesian model never is. To discuss graphical representation, we first need to establish some notation. An undirected graph is usually denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a nonempty set of vertices or nodes, and $\mathcal{E}$ is a set containing the edges between the vertices. The edge set thus comprises of 2-elemented subsets of the vertex set $\mathcal{V}$, which each represents the connection of two vertices in $\mathcal{V}$. The order of a graph is given by $|\mathcal{G}| = |\mathcal{V}|$ and is thus the number of vertices, and the size of the graph refers to the number of edges in the graph.

To give a proper definition of MRFs, we first need to establish a general notation for a neighborhood system. We start with letting $\mathcal{V} = \{1, 2, 3, \dots, n\}$ be an arbitrary vertex set for an undirected graph that has n nodes. A neighborhood system for $\mathcal{V}$ is defined as a set $\mathcal{N} = \{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_n\}$, where $\mathcal{N}_i \subseteq \mathcal{V} \backslash \{i\}, \forall i \in \mathcal{V}$. There is also a one-to-one correspondence between these neighbor sets and the edge set of a graph, and for element $i, j \in \mathcal{V}$ the neighbor set is thus defined by $\mathcal{N}_i = \{j | \{i, j\} \in \mathcal{E}\}$. Now, if the elements $i, j \in \mathcal{V}$ is such that $i \in \mathcal{N}_j$, then i is said to be a neighbor of j. A natural consequence of $i \in \mathcal{N}_j$ is that $j \in \mathcal{N}_i$, i.e. i and j are neighbors of each other.

Let us now consider stochastic variables $\boldsymbol{\theta} \in \mathbb{R}^{n \times 1}$, where the undirected graph of these variables have $\mathcal{V}$ as a corresponding vertex set. Note that the vertex set $\mathcal{V}$ is in direct correspondence to the indices of $\boldsymbol{\theta}$. Let $\pi(\boldsymbol{\theta})$ denote the probability distribution of the variables, and let $\mathcal{N}$ denote the neighborhood system of the graph. For $i \in \mathcal{V}$ let $\boldsymbol{\theta}_{-i}$ denote all of $\boldsymbol{\theta}$ with the exception of element i, and let $\boldsymbol{\theta}_{\mathcal{N}_i}$ denote the corresponding variables of the neighbor set $\mathcal{N}_i$ for element i in $\boldsymbol{\theta}$. If we assume that $\pi(\boldsymbol{\theta}) > 0$ and that the variables of $\boldsymbol{\theta}$ possesses a Markov property with respect to the neighborhood system $\mathcal{N}$, i.e.

$$\pi(\theta_i | \boldsymbol{\theta}_{-i}) = \pi(\theta_i | \boldsymbol{\theta}_{\mathcal{N}_i}), \qquad \forall i \in \mathcal{V}, \tag{6}$$

then $\boldsymbol{\theta}$ is said to be an MRF.

Another set we need to establish notation for, is the clique set. To define cliques we look to the power set of $\mathcal{V}$, denoted by $\mathcal{P}(\mathcal{V})$. This is the set containing all subsets of $\mathcal{V}$, such that for an arbitrary subset $\Lambda \subseteq \mathcal{V}$ we always have $\Lambda \in \mathcal{P}(\mathcal{V})$. Now consider any $\Lambda \in \mathcal{P}(\mathcal{V})$, if every pair $i, j \in \Lambda$ is such that $i \in N_j$, then Λ is said to be a clique. The set containing all possible cliques we denote by $\mathcal{C}$.

We round up the elaboration on MRFs, their neighborhood systems and clique sets by presenting a small example by the use of a graph. Consider the posterior distribution in (4), and assume that it is the distribution of the HMM illustrated in Figure 1. This distribution can be represented as a Markov

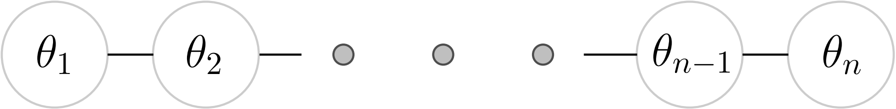


Figure 2: *The posterior distribution, $\pi(\boldsymbol{\theta}|\mathbf{x})$, illustrated as a Markov random field.*

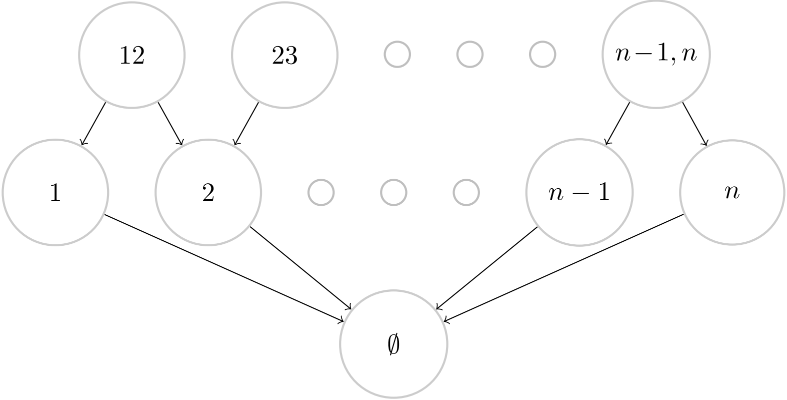


Figure 3: *Directed acyclic graph of the clique set $\mathcal{C}$ in (8).*

random field consisting of the variables of interest, i.e. $\boldsymbol{\theta}$. The MRF of $\pi(\boldsymbol{\theta}|\mathbf{x})$ is therefore as illustrated by the undirected graph in Figure 2. We denote this graph by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex set is given by $\mathcal{V} = \{1, 2, \ldots, n\}$ and the edge set $\mathcal{E} = \{\{i, i+1\}\,|i = 1, \ldots, n-1\}$. The order of this graph is $|\mathcal{V}| = n$ and it has size $n - 1$. This graph is a first-order nearest neighbor scheme. The neighborhood system of the graph in Figure 2 is given by

$$\begin{aligned} \mathcal{N} &= \{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_{n-1}, \mathcal{N}_n\} \\ &= \{\{2\}, \{1,3\}, \ldots, \{n-2, n\}, \{n-1\}\}\,, \end{aligned} \tag{7}$$

and the clique set for this MRF is given by

$$\mathcal{C} = \{\emptyset, \{1\}, \{2\}, \ldots, \{n\}, \{1,2\}, \ldots, \{n-1, n\}\}\,. \tag{8}$$

This clique set can be represented as a directed acyclic graph (DAG), and is illustrated in Figure 3. Each element of $\mathcal{C}$ is here represented as a node, where the directed edges from each node points at the subsets of that vertex. Thus the graph is built in a decreasing order of subsets in a downwards direction. The purpose of the graph becomes clear in the following section.

2.2.1 Energy Function for Binary MRFs

In this section we address binary Markov random fields and their distributions. The form of these distribution are defined by an energy function of binary variables and their interaction parameters.

Consider a vector $\boldsymbol{\theta}$ containing n binary variables, i.e. $\theta_i \in \{0,1\}$ for $i = 1, \dots, n$. Assume that $\boldsymbol{\theta}$ is an MRF possessing a Markov property with respect to a neighborhood $\mathcal{N}$, this is then a binary MRF. We denote $\Omega = \{0,1\}^n$ as our sample space, and thus $\boldsymbol{\theta} \in \Omega$. The fact that we are dealing with an MRF ensures that its distribution is greater than zero, i.e. if we let $\pi(\boldsymbol{\theta})$ denote this distribution then $\pi(\boldsymbol{\theta}) > 0$. Based on the strictly positive criteria and the Markov property, the distribution of this binary MRF can be expressed as

$$\pi(\boldsymbol{\theta}) \propto \exp\{U(\boldsymbol{\theta})\}, \tag{9}$$

where $U(\boldsymbol{\theta})$ is known as the energy function. The energy function only depends on binary variables and their interaction parameters, and the goal of this section is to establish an expression for this function.

Recall that each binary variable θ_i in $\boldsymbol{\theta}$ has a corresponding vertex $i \in \mathcal{V}$, where $\mathcal{V}$ is the vertex set of the MRF. Now, the power set of $\mathcal{V}$ has a one-to-one correspondence with the sample space Ω. To see this we present a small example where $n = 3$ such that $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]$. Assume that $\boldsymbol{\theta}$ is still a binary MRF as described above, and corresponding to this MRF is the vertex set $\mathcal{V} = \{1,2,3\}$. The power-set of $\mathcal{V}$ then has $2^3 = 8$ subsets;

$$\mathcal{P}(\mathcal{V}) = \{\{\emptyset\}, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}. \tag{10}$$

The sample space $\{0,1\}^3$, which corresponds to the binary MRF of $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]$, has the form

$$\begin{aligned} \{0,1\}^3 = \{ & \{0,0,0\}, \{1,0,0\}, \{0,1,0\}, \{0,0,1\}, \\ & \{1,1,0\}, \{1,0,1\}, \{0,1,1\}, \{1,1,1\} \}. \end{aligned} \tag{11}$$

The one-to-one correspondence between $\mathcal{P}(\mathcal{V})$ and $\{0,1\}^3$ is seen for any $\boldsymbol{\theta} \in \{0,1\}^3$. If for example $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3] = [0,1,1]$, then the corresponding subset $\Lambda \in \mathcal{P}(\mathcal{V})$ is the subset containing the indices of $\boldsymbol{\theta}$ which has an associating variable equal to 1, i.e. $\Lambda = \{i \in \mathcal{V} | \theta_i = 1\} = \{2,3\}$. Due to this one-to-one relationship we know that the power set of $\mathcal{V}$, where $|\mathcal{V}| = n$, contains 2^n subsets. Also this one-to-one correspondence yields that for an arbitrary $\boldsymbol{\theta} \in \Omega$, $\exists \Lambda \in \mathcal{P}(\mathcal{V})$ where $\Lambda = \{i \in \mathcal{V} | \theta_i = 1\}$.

Using the above notation, $U(\boldsymbol{\theta})$ can be written in terms of interaction parameters denoted by $\{\beta(\Lambda), \Lambda \in \mathcal{P}(\mathcal{V})\}$. The energy function is by these parameters uniquely defined by

$$U(\boldsymbol{\theta}) = \sum_{\Lambda \in \mathcal{P}(\mathcal{V})} \beta(\Lambda) \prod_{i \in \Lambda} \theta_i, \tag{12}$$

which is a sum with 2^n terms. However, in many cases some of the interaction parameters are zero, and can therefore be omitted from the sum. It can be

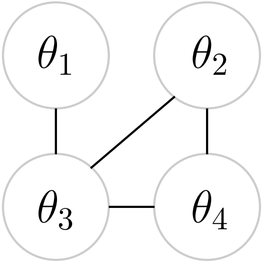


Figure 4: *MRF for the small example* $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]$.

shown that a binary function, such as $U(\boldsymbol{\theta})$ in (12), which possesses a Markov property with respect to a neighborhood system $\mathcal{N}$ and a clique set $\mathcal{C}$, have interaction parameters that are zero for all $\Lambda \notin \mathcal{C}$, i.e. $\beta(\Lambda) = 0$ when $\Lambda \in \mathcal{P}(\mathcal{V}) \backslash \mathcal{C}$. A proof of this is provided by Austad (2011).

Some of the cliques in the clique set can also produce interaction parameters that are zero. We remove these, and a final set of cliques is established. Note however that for a clique $\Lambda \in \mathcal{C}$ which has $\beta(\Lambda) \neq 0$, we also need to include $\mathcal{P}(\Lambda)$ to the final set. This is due to the structure of the DAG for the final set, which is very much used in the computational algorithm provided by Tjelmeland and Austad (2012). The set is denoted by

$$\mathcal{B} = \bigcup_{\Lambda \in \mathcal{C} : \beta(\Lambda) \neq 0} \mathcal{P}(\Lambda) \tag{13}$$

where $\mathcal{B} \subseteq \mathcal{C}$. Using this set, the energy function is restated as

$$U(\boldsymbol{\theta}) = \sum_{\Lambda \in \mathcal{B}} \beta(\Lambda) \prod_{i \in \Lambda} \theta_i. \tag{14}$$

Tjelmeland and Austad (2012) builds a DAG of the set $\mathcal{B}$ and constructs an interaction parameter weighted graph for the forward-backward algorithm to use. If we assume that the MRF of $\boldsymbol{\theta}$ in Figure 2 takes binary variables, and assume that every clique for this MRF has an interaction parameter that is not zero, then the graph of $\mathcal{B}$ for this particular case is as illustrated in Figure 3.

As a final example we look at $\boldsymbol{\theta}$ with $n = 4$, i.e. $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]$. Let the MRF of $\boldsymbol{\theta}$ be as illustrated in Figure 4, and assume that $\boldsymbol{\theta} \in \{0,1\}^4$. The neighborhood system for $\mathcal{V} = \{1,2,3,4\}$ corresponding to $\boldsymbol{\theta}$ is given by

$$\mathcal{N} = \{\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3, \mathcal{N}_4\} = \{\{3\}, \{3,4\}, \{1,2,4\}, \{2,3\}\}, \tag{15}$$

which gives the clique set the following form

$$\begin{aligned} \mathcal{C} = &\{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1,3\}, \\ &\{2,3\}, \{2,4\}, \{3,4\}, \{2,3,4\}\}. \end{aligned} \tag{16}$$

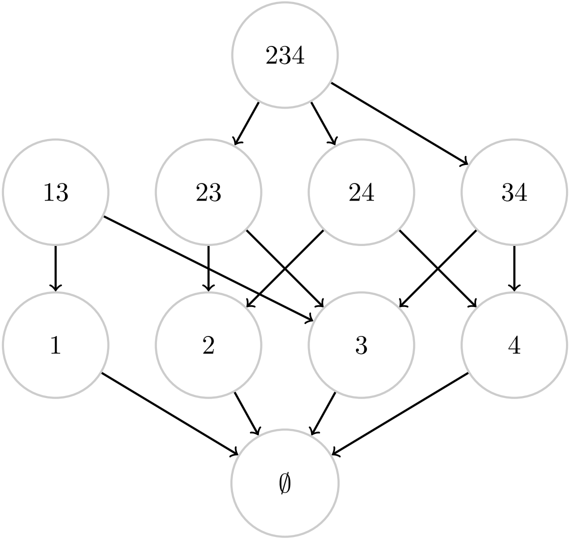


Figure 5: *DAG of the clique set for* $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]$.

A DAG of this clique set is illustrated in Figure 5. If, say the node $\Lambda = \{2, 3\}$ here produces a zero interaction parameter and $\beta(\{2, 3, 4\}) \neq 0$, then this subset still needs to be contained in the graph for programming purposes. However, if $\beta(\{1, 3\}) = 0$ this subset may be omitted from the set $\mathcal{B}$ and its DAG.

2.3 Algorithm and Approximation

In this section we introduce the forward-backward algorithm and an approximation for binary MRFs as in Austad (2011). The approximation is not exhaustively elaborated on, so for a complete description see Austad (2011) or Tjelmeland and Austad (2012).

Consider the n stochastic variables in the vector $\boldsymbol{\theta}$ with distribution $\pi(\boldsymbol{\theta}) = \pi(\theta_1, \ldots, \theta_n)$. This distribution can be written in the form of independent conditional distributions by letting

$$\pi(\theta_1, \ldots, \theta_n) = \pi(\theta_1|\theta_2, \ldots, \theta_n)\pi(\theta_2|\theta_3, \ldots, \theta_n) \cdots \pi(\theta_{n-1}|\theta_n)\pi(\theta_n). \tag{17}$$

To obtain realizations from $\pi(\theta_1, \ldots, \theta_n)$ the forward-backward algorithm can be used. The aim of this algorithm is to gain the conditional distributions in (17), and then draw samples from each of the them. Among papers that gains their results from using the forward-backward algorithm are Austad (2011), Rimstad and Omre (2013) and Ulvmoen and Hammer (2010).

We now present the structure of how the forward-backward algorithm operates. We start by considering the forward part of the algorithm where we wish

$$\begin{tikzcd}
\boxed{\pi(\theta_1, \theta_2, \theta_3, \theta_4)} \arrow[r] \arrow[d] & \boxed{\pi(\theta_1|\theta_2, \theta_3, \theta_4)} \\
\boxed{\pi(\theta_2, \theta_3, \theta_4)} \arrow[r] \arrow[d] & \boxed{\pi(\theta_2|\theta_3, \theta_4)} \\
\boxed{\pi(\theta_3, \theta_4)} \arrow[r] \arrow[d] & \boxed{\pi(\theta_3|\theta_4)} \\
\boxed{\pi(\theta_4)} &
\end{tikzcd}$$

Figure 6: *The arrows in the figure indicate the forward part of the forward-backward algorithm, which sums out one variable at the time. The process is repeated until reaching the end, then conditional distributions have been obtained to draw from one after the other in a backward fashion. The figure shows a small example of this structure for* $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]$.

to obtain the conditional distributions in (17). First we want to find the conditional distribution $\pi(\theta_1|\theta_2 \ldots, \theta_n)$, i.e. the distribution θ_1 is to be sampled from at a later point in the algorithm. However, this distribution is proportional to the marginal distribution, so we are done. In mathematic terms this is denoted by

$$\pi(\theta_1|\theta_2 \ldots, \theta_n) = \frac{1}{c} \cdot \pi(\theta_1, \ldots, \theta_n) \propto \pi(\theta_1, \ldots, \theta_n), \tag{18}$$

where the proportional constant c is a summation over all possible values of θ_1,

$$c = \pi(\theta_2, \ldots, \theta_n) = \sum_{\theta_1} \pi(\theta_1, \ldots, \theta_n). \tag{19}$$

The next step is to establish the distribution generated from the proportional constant, i.e. we find $\pi(\theta_2, \ldots, \theta_n)$ in (19). This is done to obtain the conditional distribution of $\pi(\theta_2|\theta_3, \ldots, \theta_n)$, which we find in the same way as we did for θ_1. This process is then repeated until the very last marginal distribution in (17) is gained, i.e. the distribution $\pi(\theta_n)$. A small example of the forward structure when considering the distribution of $\boldsymbol{\theta}$ for $n = 4$, is illustrated in Figure 6. Here the arrows indicate how the algorithm moves forward, hence the name. After this part of the algorithm is completed, we can start drawing samples in a backwards direction, again hence the name. For the backward part of the algorithm, we start by drawing a sample $\theta_n^* \sim \pi(\theta_n)$, where $*$ is used to indicate that it is a drawn sample. Given the newly drawn sample the algorithm moves on to draw θ_{n-1}^* from the next distribution $\pi(\theta_{n-1}|\theta_n^*)$. Then both sampled values, θ_{n-1}^* and θ_n^*, are used to draw $\theta_{n-2}^* \sim \pi(\theta_{n-2}|\theta_{n-1}^*, \theta_n^*)$, and so on. The

$$\begin{tikzcd}
\boxed{\pi(\theta_1, \theta_2, \theta_3, \theta_4)} \arrow[r] \arrow[d] & \boxed{\pi(\theta_1|\theta_2, \theta_3, \theta_4)} \\
\boxed{\tilde{\pi}(\theta_2, \theta_3, \theta_4)} \arrow[r] \arrow[d] & \boxed{\tilde{\pi}(\theta_2|\theta_3, \theta_4)} \\
\boxed{\tilde{\tilde{\pi}}(\theta_3, \theta_4)} \arrow[r] \arrow[d] & \boxed{\tilde{\tilde{\pi}}(\theta_3|\theta_4)} \\
\boxed{\tilde{\tilde{\tilde{\pi}}}(\theta_4)} &
\end{tikzcd}$$

Figure 7: *Asmall example for the structure of where approximations take place, this for the distribution of* $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]$.

algorithm continuous with the same procedure until all conditional distributions have been drawn from. So, for the small example which is illustrated in Figure 6, the backward part of the algorithm starts by drawing $\theta_4^* \sim \pi(\theta_4)$ and then uses this to draw $\theta_3^* \sim \pi(\theta_3|\theta_4^*)$. Further both samples, i.e. θ_3^* and θ_4^*, are used to draw $\theta_2^* \sim \pi(\theta_2|\theta_3^*, \theta_4^*)$, and then finally given all three newly gained sampled variables, we draw the last one $\theta_1^* \sim \pi(\theta_1|\theta_2^*, \theta_3^*, \theta_4^*)$. Hence a sample $\boldsymbol{\theta}^* = [\theta_1^*, \theta_2^*, \theta_3^*, \theta_4^*]$ is at hand, which is a sample drawn from the distribution

$$\pi(\theta_1, \theta_2, \theta_3, \theta_4) = \pi(\theta_1|\theta_2, \theta_3, \theta_4)\pi(\theta_2|\theta_3, \theta_4)\pi(\theta_3|\theta_4)\pi(\theta_4). \tag{20}$$

When n is large, the implementation of the forward-backward algorithm is in general hard to attain, and in many cases completely impossible to achieve. For this reason approximations are made.

Austad (2011) present an approximation for binary Markov random fields, which is based on minimizing the error sum of squares (SSE). Consider the variables $\boldsymbol{\theta} \in \Omega = \{0, 1\}^n$, and let $\tilde{\pi}(\cdot)$ denote an approximation of a probability distribution $\pi(\boldsymbol{\theta})$. The SSE is then given by

$$\text{SSE}(\pi, \tilde{\pi}) = \sum_{\boldsymbol{\theta} \in \Omega} \left(\pi(\boldsymbol{\theta}) - \tilde{\pi}(\boldsymbol{\theta})\right)^2. \tag{21}$$

Each time the forward part of the algorithm is to sum out a variable, an approximation is made such that this becomes possible. Figure 7 illustrates where the approximations is applied in the forward part of the algorithm for $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4]$, where the symbol $\sim$ is added to the distribution each time an approximation is made, and a variable has been summed out. Based on a maximum number of neighbors a DAG is generated for the interaction parameters that minimizes the SSE, which then becomes the approximated distribution. Basically interaction parameters that are close to zero are put to zero, and this is done with respect to the maximum number of neighbors. We do not go into further details concerning the approximation, so for a complete explanation and understanding of the subject we encourage the reader to read Austad (2011) or Tjelmeland and Austad (2012).

2.4 Metropolis-Hastings

This section contains a presentation of the Metropolis-Hastings (MH) algorithm. We here explain how we use it as a tool to evaluate the approximation discussed in the previous section. A thorough discussion of the MH algorithm can be found in the article written by Chib and Greenberg (1995) or books such as Gamerman and Lopes (2006), Hamada et al. (2008) or Lee (2012). The MH algorithm is in this section used in the same way as in the paper by Rimstad and Omre (2013).

The MH algorithm is a Markov chain Monte Carlo (MCMC) method used in stochastic simulation. It is most commonly used to obtain samples from probability distributions that are difficult to draw from directly, but we use it to quantify the quality of the approximated forward-backward algorithm of Austad (2011). In the following we present a general form of the MH algorithm, and thereafter we discuss how we use the algorithm in our evaluation. The MH algorithm generates a Markov chain through its simulations, where the chain has a target distribution as its limiting distribution. Let $\boldsymbol{\theta} \in \mathbb{R}^{n \times 1}$ denote the variables of interest, $\mathbf{x} \in \mathbb{R}^{n \times 1}$ denote observations and let $\pi(\boldsymbol{\theta}|\mathbf{x})$ be the target distribution. Further, let $\boldsymbol{\theta}^{(i-1)}$ denote the state of the Markov chain after $i-1$ steps. Iteration i then consists of first drawing a new sample $\boldsymbol{\theta}^*$ from a proposal distribution we denote as $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(i-1)})$, and then accepting the transition from state $\boldsymbol{\theta}^{(i-1)}$ to $\boldsymbol{\theta}^*$ with probability

$$\alpha_i(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(i-1)}) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*|\mathbf{x})}{\pi(\boldsymbol{\theta}^{(i-1)}|\mathbf{x})} \cdot \frac{q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{i-1})}\right\}. \tag{22}$$

If the suggested sample $\boldsymbol{\theta}^*$ is not accepted, the Markov chain does not move, and the ith state is given by $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$. The Markov chain is initialized with an arbitrary sample $\boldsymbol{\theta}^0$, and the sampling procedure is repeated until convergence is reached.

Assume that we have run M iterations, and that the algorithm has converged. It is usual to cast aside the first L samples of these iteration samples, i.e. $\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(L)}$. This is known as a *burn-in period*, where the Markov chain is assumed to have reached convergency after the burn in. The rest of the $M-L$ simulations are taken as the resulting realizations.

In theory, any density function can be used as a proposal distribution, and for the purpose of our area of study we use the approximated distribution generated by the forward-backward algorithm provided by Tjelmeland and Austad (2012). We denote this approximation by $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{x})$, and thus let the proposal distribution take the form $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(i-1)}) = \tilde{\pi}(\boldsymbol{\theta}^*|\mathbf{x})$ in (22). Note that the approximated distribution does not depend on previous states of the Markov chain, and the proposed samples are therefore independent of one another. To evaluate the approximation we look to the acceptance rate of the MH algorithm, where a higher acceptance rate is desirable. The acceptance, or test ratio, is a measure of two ratios. The first fraction is a ratio which evaluates the posterior density. It considers the drawn sample in comparison to the last accepted sample, where a higher density is favored. The second ratio is used as a correction term, if in case the approximated distribution favors some sampled quantities, this is supposed to dampen the effect. We use the acceptance rate to evaluate the performance of the proposal distribution, and it basically tells us if the distribution is a sound choice or not. By this the approximated forward-backward algorithm is evaluated for the transformation of a convolutional Bayesian model into a binary MRF.

3 Problem Description

The purpose of this master thesis is to evaluate the approximation of Austad (2011) for a convolutional Bayesian model. We adapt the Bayesian model for a convolutional two-level hidden Markov chain from Rimstad and Omre (2013), and present it in the following section. The Bayesian model is then reformulated into a binary Markov random field, and interaction parameters for an energy function of binary variables is established for a two state Markov chain prior and a four state Markov chain prior.

3.1 Bayesian Model

Rimstad and Omre (2013) consider a convolutional two-level hidden Markov model, which is constructed in a Bayesian setting. The model has application to seismic inversion, and the paper revolves around assessment of one dimensional lithology-fluid profiles. In Ulvmoen and Hammer (2010) a similar model is considered, with many of the same aspects as Rimstad and Omre (2013). In this section a Bayesian model, similar to the models of the mentioned papers, is presented for our purpose and usage. Note however that the purpose of this master thesis is not the application to seismic inversion, but it is included for a better understanding of the model.

The variable of interest is a categorical-valued vector $\boldsymbol{\theta}$ of size $n \times 1, n \geq 1$. In the interest of application to a seismic inversion, this vector takes categorical values from rock-types or lithology, which can be both permeable and imperme-

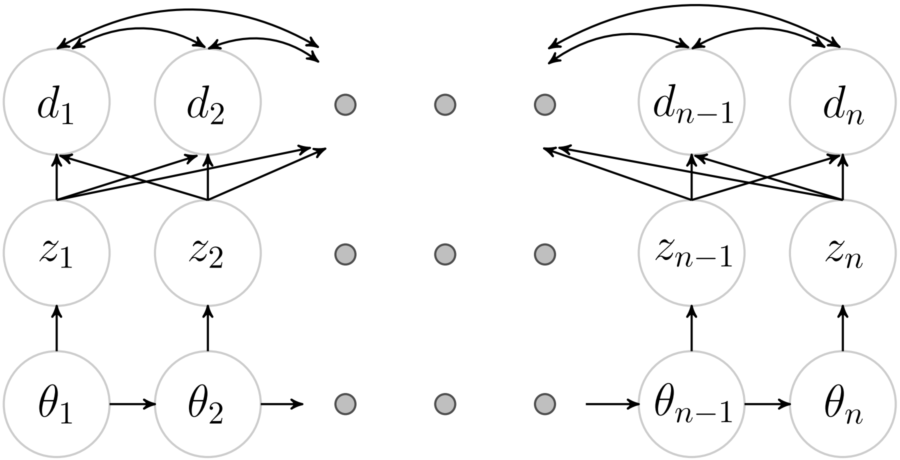


Figure 8: *Convolutional two-level hidden Markov model, illustrated as a directed graph to show dependencies. The vector of interest, $\boldsymbol{\theta}$, is a hidden Markov chain, $\mathbf{z} = [z_1, \ldots, z_n]^T$ is a hidden layer and the seismic data, $\mathbf{d} = [d_1, d_2, \ldots, d_n]^T$ are known quantities.*

able. Examples of lithology-fluids are sandstone, shale, gas, sandstone saturated with oil and sandstone saturated with water. The prior of $\boldsymbol{\theta}$ is assumed to be a first order Markov chain, such that the distribution is given by

$$\pi(\boldsymbol{\theta}) = \pi(\theta_1) \prod_{i=2}^{n} \pi(\theta_i|\theta_{i-1}), \tag{23}$$

where $\pi(\theta_i|\theta_{i-1})$ denotes the transition probability from θ_{i-1} to θ_i, and $\pi(\theta_1)$ denotes the limiting probability distribution of the Markov chain.

The Bayesian model we are considering is a two-level hidden Markov model. The vector of interest, $\boldsymbol{\theta}$, is in this case located beneath two continuous valued processes denoted as $\mathbf{z}$ and $\mathbf{d}$, both of which are discretized into real-valued vectors of size $n \times 1$. Their dependency and connection to $\boldsymbol{\theta}$ is as shown in Figure 8, where the bottom part is the same as the HMM we saw in Figure 1. The HMM is now extended by an extra level, with a more complex dependency throughout the model. This is due to the vector $\mathbf{d}$ depending on multiple variables of the level containing $\mathbf{z}$. In correspondence to the seismic inversion application, $\mathbf{d}$ is a vector of discretized seismic data and $\mathbf{z}$ consists of lithology-fluid properties.

The likelihood of the Bayesian model is split into two parts, one considering the dependencies between $\mathbf{z}$ and $\boldsymbol{\theta}$ and the other considering $\mathbf{z}$ and $\mathbf{d}$. We include the $\mathbf{z}$ level of the two-level likelihood in the following manner

$$\pi(\mathbf{d}|\boldsymbol{\theta}) = \int \pi(\mathbf{d}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} = \int \pi(\mathbf{d}|\mathbf{z}, \boldsymbol{\theta}) \pi(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} = \int \pi(\mathbf{d}|\mathbf{z}) \pi(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}, \tag{24}$$

and we now address the two distributions, $\pi(\mathbf{z}|\boldsymbol{\theta})$ and $\pi(\mathbf{d}|\mathbf{z})$, in the respective order. The elements contained in the vector $\mathbf{z}$ are as mentioned lithology-fluid properties, which are in direct correspondence with the same located element of $\boldsymbol{\theta}$. The properties are assumed to be conditionally independent of one another, i.e. we may write $\pi(\mathbf{z}|\boldsymbol{\theta}) = \prod_{i=1}^{n} \pi(z_i|\theta_i)$, where the conditional distributions are assumed to be Gaussian. Now, let the ith element of the level containing $\mathbf{z}$ have a conditional expectation denoted by $\mathrm{E}[z_i|\theta_i] = \mu_{z_i|\theta_i}$ and variance by $\mathrm{Var}(z_i|\theta_i) = \sigma^2_{z_i|\theta_i}$. We denote the expectation vector by $\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}} = [\mu_{z_1|\theta_1}, \ldots, \mu_{z_n|\theta_n}]^T$ and the covariance matrix by $\boldsymbol{\Sigma}_{\mathbf{z}|\boldsymbol{\theta}} = diag[\sigma^2_{z_1|\theta_1}, \ldots, \sigma^2_{z_n|\theta_n}]$. All other entries of the covariance matrix are zero due to the conditional independence between the elements. Using the given notation the distribution of $\mathbf{z}|\boldsymbol{\theta}$ is then given by

$$\begin{aligned}\pi(\mathbf{z}|\boldsymbol{\theta}) &= \prod_{i=1}^{n} \left(2\pi\sigma^2_{z_i|\theta_i}\right)^{-1/2} \exp\left\{-\frac{1}{2\sigma^2_{z_i|\theta_i}}(z_i - \mu_{z_i|\theta_i})^2\right\} \\ &= \frac{1}{(2\pi)^{n/2}} \left|\boldsymbol{\Sigma}_{\mathbf{z}|\boldsymbol{\theta}}\right|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}})^T \boldsymbol{\Sigma}^{-1}_{\mathbf{z}|\boldsymbol{\theta}}(\mathbf{z} - \boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}})\right\},\end{aligned} \tag{25}$$

which is a multivariate Gaussian distribution. As a side note we mention that the elements of the expectation vector and the covariance matrix does not necessarily have to be scalars, they may as well be of a higher dimension. For example in Ulvmoen and Hammer (2010) and Rimstad and Omre (2013), three elastic material properties are considered for a real-data analysis, which makes $\mu_{z_i|\theta_i}$ a vector containing three expected values and $\sigma^2_{z_i|\theta_i}$ a 3×3 matrix, for $i = 1, \ldots, n$. This would necessarily mean that the system becomes three times as big, e.g. $\mathbf{z} \in \mathbb{R}^{3n\times 1}$. We are however considering the variables to be scalar in the test cases of this thesis.

In the two top levels of the HMM (see Figure 8), $\mathbf{z}$ and $\mathbf{d}$ are located. Here a convolution in the model creates a more complex dependency throughout the model, i.e. each d_i depends on multiple variables of $\mathbf{z}$. If n is large, a worst case scenario is that each d_i depends on all of $\mathbf{z}$. This dependency is what makes $\boldsymbol{\theta}$ hard to assess in many cases. The assumed dependencies are modeled such that the seismic data, $\mathbf{d}$, is seen as a convolution of the material properties, $\mathbf{z}$, with a wavelet. It also includes a Gaussian error. The wavelet is a very much used tool in seismic inversion. We are considering two types of wavelet, a Gaussian and a Ricker wavelet, both of which are illustrated in Figure 9. These wavelets are defined by the following functions, for the Gaussian case we have

$$f(t) = \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left\{-\frac{t^2}{2\sigma_w^2}\right\}, \tag{26}$$

and for the Ricker case the function is given by

$$g(t) = \frac{2}{\sqrt{3\sigma_w}\pi^{1/4}}\Big(1 - \frac{t^2}{\sigma_w^2}\Big) \exp\Big\{-\frac{t^2}{2\sigma_w^2}\Big\}, \tag{27}$$

where σ_w^2 denotes the variance. Note that the Gaussian function is strictly speaking not a wavelet as it does not oscillate, but it is very much used either

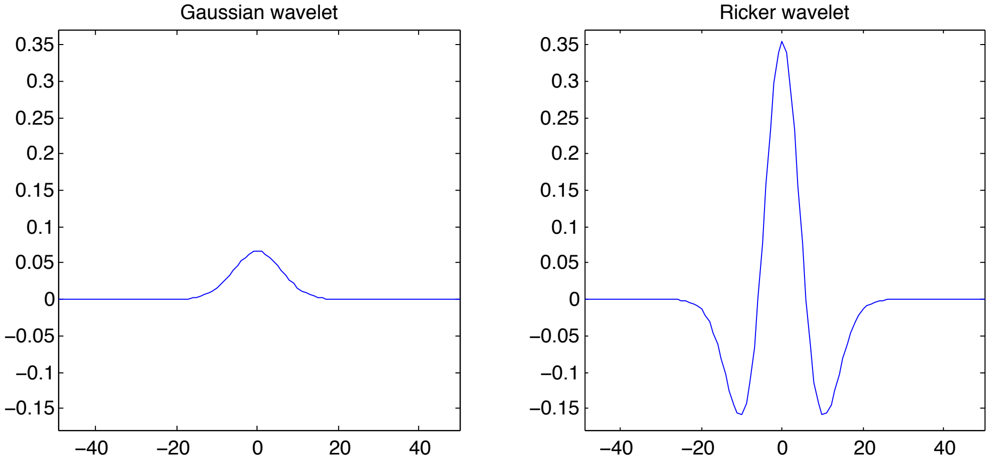


Figure 9: *Left: Gaussian wavelet and Right: Ricker wavelet. Both with variance* $\sigma_w^2 = 6^2$.

way. The wavelet which is under consideration is discretized into a symmetric matrix $\mathbf{W}$, where each row of the matrix has the center of the wavelet located at the main diagonal. We use the Gaussian curve to illustrate how the discretization works, but the same procedure applies for the Ricker wavelet. The function $f(t)$ is discretized by letting $w_0 = f(0)$, $w_1 = w_{-1} = f(1)$, $w_2 = w_{-2} = f(2)$ and so on. This is put into the matrix, $\mathbf{W}$, with w_0 on the main diagonal and the increasing indices following on the outer diagonals. This results in the matrix having the symmetric form,

$$\mathbf{W} = \begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_{n-1} & w_n \\ w_1 & w_0 & w_1 & & & w_{n-1} \\ w_2 & w_1 & w_0 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & w_1 & w_2 \\ w_{n-1} & & & w_1 & w_0 & w_1 \\ w_n & w_{n-1} & \cdots & w_2 & w_1 & w_0 \end{bmatrix}. \tag{28}$$

In Figure 10, the discretization of the Gaussian curve is illustrated and the resulting wavelet is shown as an image. Using the wavelet matrix, the seismic data is given by $\mathbf{d} = \mathbf{Wz} + \mathbf{e}$, where $\mathbf{e} \sim N(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{d}|\mathbf{z}})$, $\mathbf{\Sigma}_{\mathbf{d}|\mathbf{z}} = \sigma^2_{d|z} \cdot \mathbf{I}$ and $\mathbf{I}$ is the identity matrix of size $n \times n$. Since the error terms of the seismic data are all Gaussian, the distribution of $\mathbf{d}|\mathbf{z}$ is multivariate Gaussian. Recall that the distribution of $\mathbf{z}|\boldsymbol{\theta}$ is also multivariate Gaussian. The conditional expectation of the seismic data, i.e. $\mathrm{E}[\mathbf{d}|\mathbf{z}]$, is found from

$$\boldsymbol{\mu}_{\mathbf{d}|\mathbf{z}} = \mathrm{E}[\mathbf{d}|\mathbf{z}] = \mathrm{E}[\mathbf{Wz} + \mathbf{e}|\mathbf{z}] = \mathrm{E}[\mathbf{Wz}|\mathbf{z}] + \mathrm{E}[\mathbf{e}|\mathbf{z}] = \mathbf{Wz}, \tag{29}$$

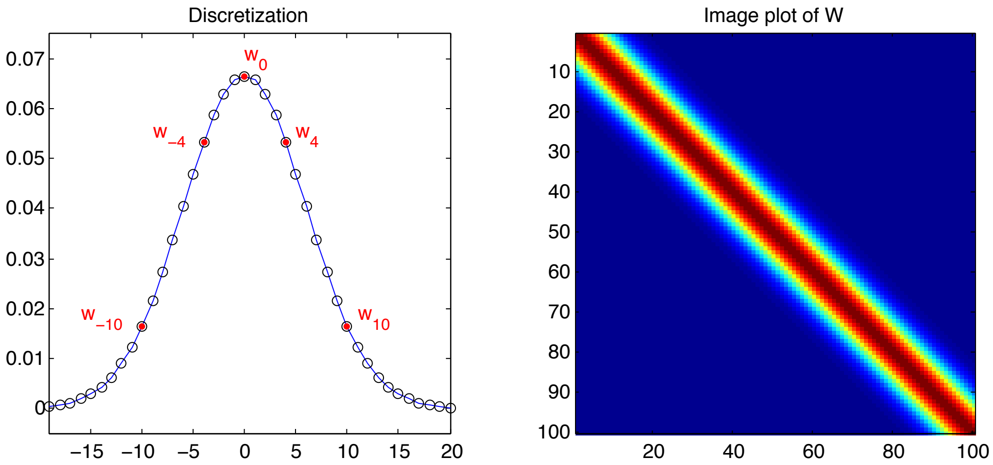


Figure 10: *Gaussian wavelet, $\sigma_w^2 = 6^2$. The wavelet curve is shown with discretization points, with some of the points highlighted. The right image shows the resulting wavelet* $\mathbf{W}$ *plotted as an image.*

and the covariance matrix is given by

$$\begin{aligned}\mathrm{Cov}(\mathbf{d}|\mathbf{z}) &= \mathrm{Cov}(\mathbf{W}\mathbf{z}+\mathbf{e}|\mathbf{z}, \mathbf{W}\mathbf{z}+\mathbf{e}|\mathbf{z}) \\ &= \mathrm{Cov}(\mathbf{e}^T\mathbf{e}|\mathbf{z}) \\ &= \boldsymbol{\Sigma}_{\mathbf{d}|\mathbf{z}},\end{aligned} \tag{30}$$

and thus

$$\mathbf{d}|\mathbf{z} \sim N(\mathbf{W}\mathbf{z}, \boldsymbol{\Sigma}_{\mathbf{d}|\mathbf{z}}). \tag{31}$$

The two parts presented above together defines the likelihood, $\pi(\mathbf{d}|\boldsymbol{\theta})$, see (24). We are dealing with multivariate Gaussian distributions for both $\mathbf{z}|\boldsymbol{\theta}$ and $\mathbf{d}|\mathbf{z}$, and since the data in $\mathbf{d}$ are linearly depending on $\mathbf{z}$ we know that the likelihood will also be multivariate Gaussian. We denote the expectation vector of $\mathbf{d}|\boldsymbol{\theta}$ in the same manner as before, and elaborate

$$\boldsymbol{\mu}_{\mathbf{d}|\boldsymbol{\theta}} = \mathrm{E}[\mathbf{d}|\boldsymbol{\theta}] = \mathrm{E}[\mathbf{W}\mathbf{z}+\mathbf{e}|\boldsymbol{\theta}] = \mathbf{W}\mathrm{E}[\mathbf{z}|\boldsymbol{\theta}] = \mathbf{W}\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}}, \tag{32}$$

where $\boldsymbol{\mu}_{\mathbf{d}|\boldsymbol{\theta}} = [\mu_{d_1|\theta_1}, \ldots, \mu_{d_n|\theta_n}]^T \in \mathbb{R}^{n\times 1}$. The covariance matrix of the distribution can be found in the same way as was done for $\mathbf{d}|\mathbf{z}$. We have that

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{d}|\boldsymbol{\theta}} &= \mathrm{Cov}(\mathbf{d}|\boldsymbol{\theta}, \mathbf{d}|\boldsymbol{\theta}) = \mathrm{Cov}(\mathbf{W}\mathbf{z}+\mathbf{e}|\boldsymbol{\theta}, \mathbf{W}\mathbf{z}+\mathbf{e}|\boldsymbol{\theta}) \\ &= \mathbf{W}\mathrm{Cov}(\mathbf{z}^T\mathbf{z}|\boldsymbol{\theta})\mathbf{W}^T + \mathrm{Cov}(\mathbf{e}^T\mathbf{e}|\boldsymbol{\theta}) \\ &= \mathbf{W}\boldsymbol{\Sigma}_{\mathbf{z}|\boldsymbol{\theta}}\mathbf{W}^T + \boldsymbol{\Sigma}_{\mathbf{d}|\mathbf{z}},\end{aligned} \tag{33}$$

such that

$$\mathbf{d}|\boldsymbol{\theta} \sim N(\mathbf{W}\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}}, \mathbf{W}\boldsymbol{\Sigma}_{\mathbf{z}|\boldsymbol{\theta}}\mathbf{W}^T + \boldsymbol{\Sigma}_{\mathbf{d}|\mathbf{z}}). \tag{34}$$

The posterior distribution is entirely defined by the prior in (23) and the likelihood in (34). We let $\pi(\theta_1|\theta_0) = \pi(\theta_1)$, and thus the Bayesian models resulting posterior distribution yields

$$\pi(\boldsymbol{\theta}|\mathbf{d}) \propto \exp\big\{ -\frac{1}{2}(\mathbf{d} - \mathbf{W}\boldsymbol{\mu}_{\mathbf{d}|\boldsymbol{\theta}})^T \boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}(\mathbf{d} - \mathbf{W}\boldsymbol{\mu}_{\mathbf{d}|\boldsymbol{\theta}})\big\} \cdot \prod_{i=1}^{n} \pi(\theta_i|\theta_{i-1}). \tag{35}$$

3.2 Interaction Parameters in Posterior Model

This section contains the reformulation of the Bayesian model in (35) into a form depending on the energy function of binary variables defined in (14). The goal is to formulate the posterior in terms of interaction parameters for the approximated algorithm of Tjelmeland and Austad (2012) to handle. Interaction parameters are here established for two cases, a binary Markov chain prior case and a four state Markov chain prior case. Some of the calculations generate large equations, which have been exhaustively presented in an appendix.

We start by simplifying the posterior distribution given in (35). Explanatory calculations for the following expansion and defined products are included in Appendix A. We redefine some of the vector-matrix multiplications in the posterior by creating two new vectors and a matrix, $\boldsymbol{\mu}(\boldsymbol{\theta}) = \boldsymbol{\mu}_{\mathbf{d}|\boldsymbol{\theta}} \in \mathbb{R}^{n\times 1}$, $\mathbf{y}^T = \mathbf{d}^T\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}\mathbf{W} \in \mathbb{R}^{n\times 1}$ and $\mathbf{Q} = -\frac{1}{2}\mathbf{W}^T\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}\mathbf{W} \in \mathbb{R}^{n\times n}$. In general the vector $\mathbf{y}$ and matrix $\mathbf{Q}$ may also be functions of $\boldsymbol{\theta}$ because of their dependency of $\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}$, e.g. the real data analysis in Ulvmoen and Hammer (2010) and Rimstad and Omre (2013) consider parameters for this covariance matrix which depends on the categorical values of $\boldsymbol{\theta}$. However, in our test cases the covariance matrix is taken to be constant. Using the new denotational expressions, the posterior can be written as

$$\pi(\boldsymbol{\theta}|\mathbf{d}) \propto \exp\big\{\mathbf{y}^T\boldsymbol{\mu}(\boldsymbol{\theta}) + \boldsymbol{\mu}(\boldsymbol{\theta})^T\mathbf{Q}\boldsymbol{\mu}(\boldsymbol{\theta})\big\} \cdot \prod_{i=1}^{n} \pi(\theta_i|\theta_{i-1}), \tag{36}$$

where the data input, $\mathbf{d}$, is now contained in the vector $\mathbf{y}$. We continue by pulling the transition probabilities from the prior distribution into the exponential function and expand the expressions using summations. Let $\mu_i(\theta_i)$ and y_i denote the ith element of their respective vectors, and Q_{ij} represent the element in the ith row and jth column of the matrix $\mathbf{Q}$. This results in the posterior being written as

$$\pi(\boldsymbol{\theta}|\mathbf{d}) \propto \exp\left\{ \sum_{i=1}^{n} y_i\mu_i(\theta_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}\mu_i(\theta_i)\mu_j(\theta_j) + \sum_{i=1}^{n} \ln\left(\pi(\theta_i|\theta_{i-1})\right) \right\}, \tag{37}$$

where the energy function is given by

$$U(\boldsymbol{\theta}) = \sum_{i=1}^{n} y_i\mu_i(\theta_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}\mu_i(\theta_i)\mu_j(\theta_j) + \sum_{i=1}^{n} \ln(\pi(\theta_i|\theta_{i-1})). \tag{38}$$

The expression for the posterior in (37) will be the same regardless of how many states the Markov chain prior has. However, the energy function in (38) still needs to be expanded some more such that it depends on the values of $\boldsymbol{\theta}$ directly. The goal is to express (38) as a binary function with interaction parameters as in (14). To do this, an expression is needed for $\mu_i(\theta_i)$ and for $\ln(\pi(\theta_i|\theta_{i-1}))$, and these expressions depends on how many states the Markov chain has. In the following section we discuss the case where the Markov chain in $\boldsymbol{\theta}$ has two states, and we assemble the interaction parameters for the corresponding energy function. Afterwards we expand and let the Markov chain in $\boldsymbol{\theta}$ have four possible states, and then determine the interaction parameters for this particular case.

3.2.1 Binary Markov chain

In this section we consider a binary Markov chain $\boldsymbol{\theta} \in \Omega = \{0,1\}^n$, and find interaction parameters for the energy function in this case.

To begin with, we define two constants $a = \mu(1) - \mu(0)$ and $b = \mu(0)$, where $\mu(0)$ is the expectation of state 0 and $\mu(1)$ of state 1. Using these constants the expectation variable, $\mu_i(\theta_i)$, in (38) may be written as

$$\begin{aligned} \mu_i(\theta_i) &= \mu(1)\theta_i + \mu(0)(1-\theta_i) \\ &= a\theta_i + b. \end{aligned} \tag{39}$$

Further, for the Markov chain states $\theta_{i-1}, \theta_i \in \{0,1\}$, let the constant denoted $t_{\theta_{i-1},\theta_i} = \ln(\pi(\theta_i|\theta_{i-1}))$ be defined as the logarithm of the transition probability from state θ_{i-1} to state θ_i. Using this we may write

$$\begin{aligned} \ln(\pi(\theta_i|\theta_{i-1})) =& \ln(\pi(0|0))(1-\theta_i)(1-\theta_{i-1}) + \ln(\pi(0|1))(1-\theta_i)\theta_{i-1} \\ &+ \ln(\pi(1|0))(1-\theta_{i-1})\theta_i + \ln(\pi(1|1))\theta_{i-1}\theta_i \\ =& t_{00}(1-\theta_i)(1-\theta_{i-1}) + t_{10}(1-\theta_i)\theta_{i-1} \\ &+ t_{01}(1-\theta_{i-1})\theta_i + t_{11}\theta_{i-1}\theta_i \\ =& t_{00} + (t_{10} - t_{00})\theta_{i-1} + (t_{01} - t_{00})\theta_i \\ &+ (t_{00} + t_{11} - t_{10} - t_{01})\theta_{i-1}\theta_i. \end{aligned} \tag{40}$$

We once more introduce new constants to simplify this expression. Let $c_0 = t_{00}$, $c_1 = (t_{10} - t_{00})$, $c_2 = (t_{01} - t_{00})$ and $c_3 = (t_{00} + t_{11} - t_{10} - t_{01})$ such that

$$\ln(\pi(\theta_i|\theta_{i-1})) = c_0 + c_1\theta_{i-1} + c_2\theta_i + c_3\theta_{i-1}\theta_i, \tag{41}$$

which holds for $i = 2, \ldots, n$. For $i = 1$ we have a special case, and we introduce another constant to express the equation. Let $t_i = \ln(\pi(\theta_i))$ denote the logarithm of the limiting probability distribution of the Markov chain. Using this we find that

$$\begin{aligned} \ln(\pi(\theta_1)) =& \ln(\pi(1))\theta_1 + \ln(\pi(0))(1-\theta_1) \\ =& (t_1 - t_0)\theta_1 + t_0. \end{aligned} \tag{42}$$

Equations (39), (41) and (42) are then put into the expression we found for the energy function in (38). This leads to the following energy function for the binary Markov chain prior case

$$\begin{aligned}
U(\boldsymbol{\theta}) =&\sum_{i=1}^{n} y_i(a\theta_i + b) + \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}(a\theta_i + b)(a\theta_j + b)\\
&+ (p_1 - p_0)\theta_1 + p_0 + \sum_{i=2}^{n}(c_0 + c_1\theta_{i-1} + c_2\theta_i + c_3\theta_{i-1}\theta_i)\\
=&bn\bar{\mathbf{y}} + (n-1)c_0 + p_0 + b^2\sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij} \qquad (43)\\
&+ (p_1 - p_0)\theta_1 + a\sum_{i=1}^{n} y_i\theta_i + 2ab\sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}\theta_i\\
&+ c_1\sum_{i=1}^{n-1}\theta_i + c_2\sum_{i=2}^{n}\theta_i + c_3\sum_{i=2}^{n}\theta_{i-1}\theta_i + a^2\sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}\theta_i\theta_j,
\end{aligned}$$

where $\bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

The equation in (43) is now in such a form that it can be connected to the interaction parameters of the energy function. The neighborhood system of the two level-hidden Markov model in Figure 8 is quite complex due to the convolution of the model, but based on (43) we see that the interaction parameters in (12) are all zero when the clique is larger than two, i.e. $|\Lambda| > 2 \Rightarrow \beta(\Lambda) = 0$. If shown as a graph, the clique set for this model would be an extension of Figure 3, where all possible two-pairings of the vertex set $\mathcal{V} = \{1, 2, \dots, n\}$ would be in the top level of the graph.

We now present the interaction parameters for the energy function in (43). For the empty set, $\Lambda = \{\emptyset\}$, the interaction parameter becomes

$$\beta\left(\{\emptyset\}\right) = bn\bar{\mathbf{y}} + (n-1)c_0 + p_0 + b^2\sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}. \qquad (44)$$

When we consider the singular sets interaction parameters, i.e. $\beta(\Lambda)$ for $|\Lambda| = 1$, we have two boundary cases that needs to be taken into account. Let $I(\cdot)$ denote the indicator function, giving one if the argument in the function is true and zero otherwise. Then the two boundary-cases we need to consider, $\Lambda = \{1\}$ and $\Lambda = \{n\}$, is handled using the indicator function. Thus the interaction parameters for $i = 1, 2, \dots, n$ are given by

$$\begin{aligned}
\beta\left(\{i\}\right) =& ((p_1 - p_0) + c_1)\cdot I(i = 1) + c_2 \cdot I(i = n)\\
&+ (c_1 + c_2)\cdot I(i \notin \{1, n\}) + ay_1 + a^2 Q_{ii} + 2ab\sum_{j=1}^{n} Q_{ij}. \qquad (45)
\end{aligned}$$

Last we consider the interaction parameters for cliques of order $|\Lambda| = 2$, which gains terms from the quadratic terms in the energy-function in (43). Here we have special cases for $\Lambda = \{i, j\}, i, j \in \mathcal{V}$, if $|i-j| = 1$. In other words, there are special cases for the elements in $\boldsymbol{\theta}$ that are right next to one another. We use the indicator function again and the interaction parameters are then given by

$$\beta\left(\{i, j\}\right) = c_3 \cdot I(|i-j| = 1) + 2a^2 Q_{ij}. \tag{46}$$

This concludes the interaction parameters for the binary Markov chain prior case. As mentioned above, the interaction parameters for cliques of higher order than 2 are zero.

3.2.2 Four State Markov chain

The Markov chain of interest often has more than two classes, and in this section we let $\boldsymbol{\theta} \in \{0, 1, 2, 3\}^n$ such that the Markov chain has four possible states. The energy function in (14) is based on binary variables, and therefore the variables in $\boldsymbol{\theta}$ needs to be expressed in a binary fashion. In this section we present one strategy of doing this, where the Markov states are assigned a pair of binary variables, where the aim is to find interaction parameters for the energy function.

For each $\theta_i \in \boldsymbol{\theta}$ we assign a pair of corresponding binary variables $[\phi_i, \phi_{n+i}] \in \{[0, 0], [1, 0], [0, 1], [1, 1]\}$, where

$$\theta_i = \begin{cases} 0, & \text{if } \phi_i = 0 \text{ and } \phi_{n+i} = 0, \\ 1, & \text{if } \phi_i = 1 \text{ and } \phi_{n+i} = 0, \\ 2, & \text{if } \phi_i = 0 \text{ and } \phi_{n+i} = 1, \\ 3, & \text{if } \phi_i = 1 \text{ and } \phi_{n+i} = 1. \end{cases} \tag{47}$$

We now have a new vector of interest $\boldsymbol{\phi} \in \{0, 1\}^{2n \times 1}$, where these variables are used to expand the expression in (38) into the form of the energy function in (14). We start in the same manner as we did for the binary Markov chain case, by looking at the expectation of θ_i expressed using $[\phi_i, \phi_{n+i}]$. The following calculation, expansions and constants are fully described in Appendix B. Let $K_1 = \mu(0)$, $K_2 = (\mu(1) - \mu(0))$, $K_3 = (\mu(2) - \mu(0))$ and $K_4 = (\mu(0) - \mu(1) - \mu(2) + \mu(3))$ be constants defined by the expectations of the four states in the Markov chain, i.e. $\mu(0)$, $\mu(1)$, $\mu(2)$ and $\mu(3)$. By using these constants, the expectation of θ_i can be expressed as

$$\mu_i(\theta_i) = K_1 + K_2\phi_i + K_3\phi_{n+i} + K_4\phi_i\phi_{n+i}. \tag{48}$$

The quadratic term of the expectation $\mu_i(\theta_i)\mu_j(\theta_j)$, which is a part of the energy function in (38), yields the following expression when using the constants defined above

$$\begin{aligned}\mu_i(\theta_i)\mu_j(\theta_j) =& K_1^2 + K_1K_2\left(\phi_i + \phi_j\right) + K_1K_3\left(\phi_{n+i} + \phi_{n+j}\right) + K_1K_4\left(\phi_i\phi_{n+i} + \phi_j\phi_{n+j}\right)\\ &+ K_2^2\phi_i\phi_j + K_2K_3\left(\phi_i\phi_{n+j} + \phi_{n+i}\phi_j\right) + K_2K_4\left(\phi_i\phi_j\phi_{n+j} + \phi_i\phi_{n+i}\phi_j\right)\\ &+ K_3^2\phi_{n+i}\phi_{n+j} + K_3K_4\left(\phi_{n+i}\phi_j\phi_{n+j} + \phi_i\phi_{n+i}\phi_{n+j}\right)\\ &+ K_4^2\phi_i\phi_{n+i}\phi_j\phi_{n+j}.\end{aligned} \tag{49}$$

We now find an expression for the logarithmic transition probabilities between the state variables. Let us once again denote the logarithm of the transition probability from state θ_{i-1} to state θ_i by $t_{\theta_{i-1},\theta_i} = \ln\left(\pi(\theta_i|\theta_{i-1})\right)$, and the logarithm of the limiting probability distribution of the Markov chain by $t_{\theta_i} = \ln(\pi(\theta_i))$, for $\theta_i \in \{0, 1, 2, 3\}$. We first address the boundary case of $i = 1$, which for θ_1 expressed by the binary form $[\phi_1, \phi_{n+1}]$ yields the expression

$$\begin{aligned}\ln(\pi(\theta_1)) =& t_0 + \left(t_1 - t_0\right)\phi_1 + \left(t_2 - t_0\right)\phi_{n+1}\\ &+ \left(t_0 - t_1 - t_2 + t_3\right)\phi_1\phi_{n+1}.\end{aligned} \tag{50}$$

This has the same build up as the expectation, however this is not the case when considering the general expression for the logarithm of the transition probability. This general expression is of a more complex form and consist of many constant built ups defined by combinations of the logarithmic transitions between the different states. These constants are defined as

$$\begin{aligned}G_1 =& \left(t_{00} - t_{01} - t_{10} + t_{11}\right),\\ G_2 =& \left(t_{00} - t_{10} - t_{20} + t_{30}\right),\\ G_3 =& \left(t_{00} - t_{02} - t_{10} + t_{12}\right),\\ G_4 =& \left(t_{00} - t_{01} - t_{20} + t_{21}\right),\\ G_5 =& \left(t_{00} - t_{01} - t_{02} + t_{03}\right),\\ G_6 =& \left(t_{00} - t_{02} - t_{20} + t_{22}\right),\\ H_1 =& \left(t_{01} + t_{10} - t_{11} + t_{20} - t_{21} - t_{30} + t_{31} - t_{00}\right),\\ H_2 =& \left(t_{01} + t_{02} - t_{03} + t_{10} - t_{11} - t_{12} + t_{13} - t_{00}\right),\\ H_3 =& \left(t_{02} + t_{10} - t_{12} + t_{20} - t_{22} - t_{30} + t_{32} - t_{00}\right),\\ H_4 =& \left(t_{01} + t_{02} - t_{03} + t_{20} - t_{21} - t_{22} + t_{23} - t_{00}\right),\\ J_1 =& (t_{00} - t_{01} - t_{02} + t_{03} - t_{10} + t_{11} + t_{12} - t_{13}\\ &- t_{20} + t_{21} + t_{22} - t_{23} + t_{30} - t_{31} - t_{32} + t_{33}),\end{aligned} \tag{51}$$

where all constants are explained in Appendix B. These constants are collected for the combinations of the quadratic, cubic and quartic terms of ϕ_{i-1}, ϕ_i, ϕ_{n+i-1} and ϕ_{n+i}, and it has been done to simplify the resulting energy function. The logarithm of the transition probability from state θ_{i-1} to state θ_i can, when using the constants in (51) and by using the respective binary representation of the states $[\phi_{i-1}, \phi_{n+i-1}]$ and $[\phi_i, \phi_{n+i}]$, be written as

$$\begin{aligned}
\ln(\pi(\theta_i|\theta_{i-1})) =& p_{00} + (p_{10} - p_{00})\,\phi_{i-1} + (p_{20} - p_{00})\,\phi_{n+i-1} \\
&+ (p_{01} - p_{00})\,\phi_i + (p_{02} - p_{00})\,\phi_{n+i} \\
&+ G_1\phi_{i-1}\phi_i + G_2\phi_{i-1}\phi_{n+i-1} + G_3\phi_{i-1}\phi_{n+i} \\
&+ G_4\phi_i\phi_{n+i-1} + G_5\phi_i\phi_{n+i} + G_6\phi_{n+i-1}\phi_{n+i} \\
&+ H_1\phi_{i-1}\phi_i\phi_{n+i-1} + H_2\phi_{i-1}\phi_i\phi_{n+i} \\
&+ H_3\phi_{i-1}\phi_{n+i-1}\phi_{n+i} + H_4\phi_i\phi_{n+i-1}\phi_{n+i} \\
&+ J_1\phi_i\phi_{i-1}\phi_{n+i-1}\phi_{n+i},
\end{aligned} \tag{52}$$

which applies for $i = 2, 3, \ldots, n$. The full calculation of this expression can be found in Appendix B.

Finally, we take the expressions given in (48), (49), (50) and (52) and put them into the energy function in (38). The final form of the energy function for the Markov chain when $\boldsymbol{\theta} \in \{0, 1, 2, 3\}^n \Rightarrow \boldsymbol{\phi} \in \{0, 1\}^{2n\times 1}$ is thus given by

$$\begin{aligned}
U(\boldsymbol{\phi}) =& \sum_{i=1}^{n} y_i \Big[K_1 + K_2\phi_i + K_3\phi_{n+i} + K_4\phi_i\phi_{n+i}\Big] \\
&+ \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}\Big[K_1^2 + K_1K_2\,(\phi_i + \phi_j) + K_1K_3\,(\phi_{n+i} + \phi_{n+j}) + K_1K_4\,(\phi_i\phi_{n+i} + \phi_j\phi_{n+j}) \\
&+ K_2^2\phi_i\phi_j + K_2K_3\,(\phi_i\phi_{n+j} + \phi_{n+i}\phi_j) + K_2K_4\,(\phi_i\phi_j\phi_{n+j} + \phi_i\phi_{n+i}\phi_j) \\
&+ K_3^2\phi_{n+i}\phi_{n+j} + K_3K_4\,(\phi_{n+i}\phi_j\phi_{n+j} + \phi_i\phi_{n+i}\phi_{n+j}) \\
&+ K_4^2\phi_i\phi_{n+i}\phi_j\phi_{n+j}\Big] \\
&+ p_0 + (p_1 - p_0)\,\phi_1 + (p_2 - p_0)\,\phi_{n+1} + (p_0 - p_1 - p_2 + p_3)\,\phi_1\phi_{n+1} \\
&+ \sum_{i=2}^{n}\Big[p_{00} + (p_{10} - p_{00})\,\phi_{i-1} + (p_{20} - p_{00})\,\phi_{n+i-1} \\
&+ (p_{01} - p_{00})\,\phi_i + (p_{02} - p_{00})\,\phi_{n+i} \\
&+ G_1\phi_{i-1}\phi_i + G_2\phi_{i-1}\phi_{n+i-1} + G_3\phi_{i-1}\phi_{n+i} + G_4\phi_i\phi_{n+i-1} \\
&+ G_5\phi_i\phi_{n+i} + G_6\phi_{n+i-1}\phi_{n+i} + H_1\phi_{i-1}\phi_i\phi_{n+i-1} + H_2\phi_{i-1}\phi_i\phi_{n+i} \\
&+ H_3\phi_{i-1}\phi_{n+i-1}\phi_{n+i} + H_4\phi_i\phi_{n+i-1}\phi_{n+i} + J_1\phi_i\phi_{i-1}\phi_{n+i-1}\phi_{n+i}\Big].
\end{aligned} \tag{53}$$

Corresponding to the MRF graph of the n elements in $\boldsymbol{\theta}$, we have the vertex set $\mathcal{V} = \{1, \ldots, n\}$. However, we are using the binary representation $\boldsymbol{\phi}$ in the approximated forward-backward algorithm by Austad (2011), which has twice the elements of $\boldsymbol{\theta}$. Let $\boldsymbol{\phi}$ have corresponding vertex set given by $\mathcal{V}_{\boldsymbol{\phi}} = \{1, \ldots, n, n+1, \ldots, 2n\}$. The energy function in (53) shows that for all cliques $\Lambda \subseteq \mathcal{V}_{\boldsymbol{\phi}}$ where $|\Lambda| > 4$, we have that the interaction parameters are zero, i.e. $|\Lambda| > 4 \Rightarrow \beta(\Lambda) = 0$. Let $\mathcal{B} \subseteq \mathcal{C} \subseteq \mathcal{P}(\mathcal{V}_{\boldsymbol{\phi}})$ be as defined in (13), where it is the set of cliques producing nonzero interaction parameters. The DAG of $\mathcal{B}$ is in this case an extension of the graph for the binary Markov chain prior. Here the level containing the singular elements such as $\{i\}$ and $\{j\}$, where $i, j \in \mathcal{V}_{\boldsymbol{\phi}}$, has twice the number of nodes. We still have all possible two pairings of $\{i\}$ and $\{j\}$ in the level above, in the same way as for the binary Markov chain prior. Further we have two more extended levels in the graph for three element subsets and four element subsets. The cliques of order three and four does however need to be of a certain form, and these will be given in the following sections. We now present the interaction parameters for the cliques of order $|\Lambda| \leq 4$, which generates nonzero cases for $\Lambda \in \mathcal{B}$.

Empty Set

We start by finding the interaction parameter for the empty set,

$$\beta(\{\emptyset\}) = K_1 \sum_{i=1}^{n} y_i + K_1^2 \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} + t_0 + (n-1)t_{00}. \tag{54}$$

This however is a constant that can be embedded in the proportional constant of the posterior distribution, and we may therefore put it to zero in the algorithm.

Linear Terms

In this section we address the interaction parameters for cliques of order $|\Lambda| = 1$, where $\Lambda \in \mathcal{V}_{\boldsymbol{\phi}}$. The linear terms of the energy function gives special cases for the boundary points, i.e. for $\Lambda = \{1\}$, $\Lambda = \{n\}$, $\Lambda = \{n+1\}$ and $\Lambda = \{2n\}$. We use the indicator function to include these cases, and we define the two following equations for the linear terms. First let $i \in \mathcal{V}$, then for $\Lambda = \{i\}$ the interaction parameter is given by

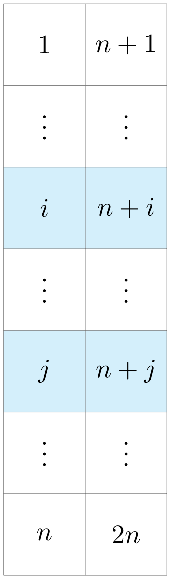
$$\begin{aligned}\beta(\{i\}) =& y_i K_2 + 2K_1K_2 \sum_{j=1}^{n} Q_{ij} + K_2^2 Q_{ii} + (t_1 - t_0) \cdot I(i=1) \\ &+ (t_{10} - t_{00}) \cdot I(i \neq n) + (t_{01} - t_{00}) \cdot I(i \neq 1),\end{aligned} \tag{55}$$

and for $\Lambda = \{n+i\}$ we have

$$\begin{aligned}\beta(\{n+i\}) =& y_i K_3 + 2K_1K_3 \sum_{j=1}^{n} Q_{ij} + K_3^2 Q_{ii} + (t_2 - t_0) I(i=1) \\ &+ (t_{20} - t_{00}) \cdot I(i \neq n) + (t_{02} - t_{00}) I(i \neq 1).\end{aligned} \tag{56}$$

Quadratic Terms

In this section the interaction parameters for the cliques of order $|\Lambda| = 2$ for $\Lambda \in \mathcal{B}$ is addressed. These interaction parameters mostly consists of coefficients from the quadratic terms of the energy function in (53), but they also get some contribution from the cubic and quartic terms.



1	$n+1$
$\vdots$	$\vdots$
i	$n+i$
$\vdots$	$\vdots$
j	$n+j$
$\vdots$	$\vdots$
n	$2n$

Figure 11: *Light blue shaded lattice points for $\Lambda = \{i, j, n+i, n+j\}$, where $\boldsymbol{\phi}$ has been cut in half and the resulting vectors are placed next to each other.*

We start the study with cliques of the form $\Lambda = \{i, n+i\}$. For this type of quadratic term there are special cases for $\Lambda = \{1, n+1\}$ and $\Lambda = \{n, 2n\}$. We again use the indicator function and look at the general case of interaction parameters for cliques of the form $\Lambda = \{i, n+i\}$, where $i \in \mathcal{V}$. These are given by

$$\begin{aligned} \beta(\{i, n+i\}) =& y_i K_4 + 2K_1K_4 \sum_{j=1}^{n} Q_{ij} + K_4^2 Q_{ii} \\ & + 2Q_{ii}(K_2K_3 + K_2K_4 + K_3K_4) \\ & + (t_0 - t_1 - t_2 + t_3) \cdot I(i=1) \\ & + G_2 \cdot I(i \neq n) + G_5 \cdot I(i \neq 1). \end{aligned} \tag{57}$$

For other pairs $i, j \in \mathcal{V}$, we still need expressions for $\Lambda = \{i, j\}$, $\Lambda = \{i, n+j\}$ and $\Lambda = \{n+i, n+j\}$. These are the possible two pairings of the shaded grid locations in Figure 11. There are also special cases for neighboring elements in $\boldsymbol{\theta}$ for these cliques, i.e. when $j - i = 1$ and $j - i = -1$, which are all handled by the indicator function. The interaction parameters for these types of cliques are thus given by

$$\beta(\{i, j\}) = 2Q_{ij}K_2^2 + G_1 \cdot I(j - i = -1), \tag{58}$$

$$\begin{aligned}\beta\left(\{i,n+j\}\right) =&2Q_{ij}K_2K_3\\&+G_3\cdot I\left(j-i=1\right)\\&+G_4\cdot I\left(j-i=-1\right),\end{aligned}\tag{59}$$

and

$$\beta\left(\{n+i,n+j\}\right) =2Q_{ij}K_3^2+G_6\cdot I(j-i=-1).\tag{60}$$

Cubic Terms

In this section we study the interaction parameters for cliques of order $|\Lambda|=3$, i.e. for the cubic terms of the energy function in (14). We study triples for $i,j\in\mathcal{V}$ of the form $\Lambda=\{i,j,n+i\}$ and $\Lambda=\{i,n+i,n+j\}$, where both the case $j>i$ and $j<i$ has to be taken into account. There will be special cases when $|i-j|=1$, i.e. triples in $\boldsymbol{\phi}$ corresponding to neighboring elements in $\boldsymbol{\theta}$. For a triple of the form $\Lambda=\{i,j,n+i\}$, the interaction parameter becomes

$$\begin{aligned}\beta\left(\{i,j,n+i\}\right) =&2Q_{ij}K_2K_4\\&+H_1\cdot I(j-i=1)\\&+H_2\cdot I(j-i=-1),\end{aligned}\tag{61}$$

and for $\Lambda=\{i,n+i,n+j\}$ we have

$$\begin{aligned}\beta\left(\{i,n+i,n+j\}\right) =&2Q_{ij}K_3K_4\\&+H_3\cdot I(j-i=1)\\&+H_4\cdot I(j-i=-1).\end{aligned}\tag{62}$$

Triples which are not in the specified forms of this section, have interaction parameter that are zero.

Quartic Terms

We now address cliques of order $|\Lambda|=4$, where $\Lambda=\{i,j,n+i,n+j\}$, see Figure 11. There are special cases for $i,j\in\mathcal{V}$ when $|i-j|=1$, which again is handled by the indicator function. For the quartic terms of the energy function, we have interaction parameters for Λ of the specified form given by

$$\beta\left(\{i,j,n+i,n+j\}\right)=2Q_{ij}K_4^2+J_1\cdot I(|i-j|=1).\tag{63}$$

A clique of order 4 that does not have this specific form, has an interaction parameter equal to zero. Also, all cliques with higher order than 4, have interaction parameters that are zero.

4 Simulation and Results

We now present test cases for the covariance matrices in the Bayesian model followed by approximation results for a two state- and a four state Markov chain

1. Base Case $\mathbf{\Sigma}_{\mathbf{z}\vert\boldsymbol{\theta}} = 0.5 \times \mathbf{I}$ $\mathbf{\Sigma}_{\mathbf{d}\vert\mathbf{z}} = 0.5 \times 10^{-4} \times \mathbf{I}$	2. Low Noise $\mathbf{\Sigma}_{\mathbf{z}\vert\boldsymbol{\theta}} = 0.25 \times 0.5 \times \mathbf{I}$ $\mathbf{\Sigma}_{\mathbf{d}\vert\mathbf{z}} = 0.1 \times 0.5 \times 10^{-4} \times \mathbf{I}$
3. High Noise - for $\mathbf{z}\vert\boldsymbol{\theta}$ $\mathbf{\Sigma}_{\mathbf{z}\vert\boldsymbol{\theta}} = 4 \times 0.5 \times \mathbf{I}$ $\mathbf{\Sigma}_{\mathbf{d}\vert\mathbf{z}} = 0.5 \times 10^{-4} \times \mathbf{I}$	4. High Noise - for $\mathbf{d}\vert\mathbf{z}$ $\mathbf{\Sigma}_{\mathbf{z}\vert\boldsymbol{\theta}} = 0.5 \times \mathbf{I}$ $\mathbf{\Sigma}_{\mathbf{d}\vert\mathbf{z}} = 10 \times 0.5 \times 10^{-4} \times \mathbf{I}$

Table 1: *Noise cases 1-4.*

prior.

We borrow synthetic test cases for the covariance matrices $\mathbf{\Sigma}_{\mathbf{z}|\boldsymbol{\theta}}$ and $\mathbf{\Sigma}_{\mathbf{d}|\mathbf{z}}$ from Rimstad and Omre (2013), which both are a part of the likelihood in the Bayesian model. As was mentioned in Section 3.1, the covariances matrices are chosen to be constant and does not depend on the variables of $\boldsymbol{\theta}$. Thus, in each case we consider the constants $Var(z_i|\theta_i) = \sigma^2_{z_i|\theta_i} = \sigma^2_{z|\theta}$ and $Var(d_i|z_i) = \sigma^2_{d_i|z_i} = \sigma^2_{d|z}$ for all $i \in \mathcal{V}$, such that we may write $\mathbf{\Sigma}_{\mathbf{z}|\boldsymbol{\theta}} = \sigma^2_{z|\theta} \cdot \mathbf{I}$ and $\mathbf{\Sigma}_{\mathbf{d}|\mathbf{z}} = \sigma^2_{d|z} \cdot \mathbf{I}$, where $\mathbf{I}$ is the $n \times n$ identity matrix. We vary the two noise parts of the likelihood and define four cases; a base case, a low noise case and two high noise cases. All these have been presented in Tabel 1.

As for the expected values, these will differ for the two state- and the four state prior case, and are therefore presented later in the text.

For the wavelets defined by the functions in (26) and (27), and illustrated in Figure 9, we use two values for the variance σ^2_w, one basic variance denoted by A, and a wide kernel case, B, where

$$\sigma^2_w = \begin{cases} 6^2 & \text{in case } A, \\ 12^2 & \text{in case } B. \end{cases} \tag{64}$$

4.1 Binary Markov Chain

We now present the two state Markov chain for the prior. The transformation of the Bayesian model into the form of the binary energy function can be found in Section 3.2.1. To evaluate the quality of the approximation we use the MH algorithm, where each case is simulated for 20000 iterations. We remove a burn-in period from the sampled Markov chain, and base the statistics of this section on the last 10000 realizations.

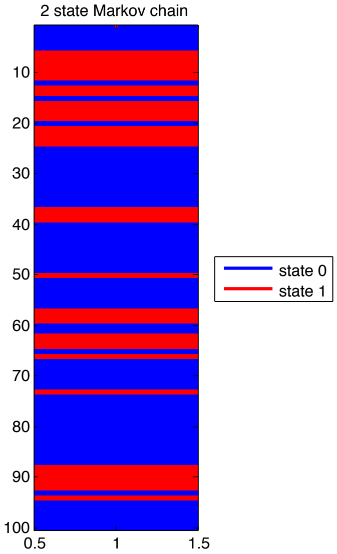


Figure 12: *Simulated Markov chain of size $n = 100$ with two states. This is referred to as the reference profile, $\boldsymbol{\theta}_R$.*

The vector of interest is here found in the state space $\boldsymbol{\theta} \in \{0, 1\}^n = \Omega$. The Markov chain is define by a 2×2 transition matrix given by

$$\mathbf{P}_2 = \begin{bmatrix} 0.83 & 0.17 \\ 0.46 & 0.54 \end{bmatrix}, \tag{65}$$

which has limiting probabilities given by $p_{l2} = [0.7302, 0.2698]$. Inspired by a three state synthetic test case in Rimstad and Omre (2013), we define expected values for $z_i|\theta_i$ given by

$$\mu_{z_i|\theta_i} = \begin{cases} -1 & \text{if } \theta_i = 0, \\ 1 & \text{if } \theta_i = 1. \end{cases} \tag{66}$$

To evaluate the approximation of Austad (2011), the Markov chain illustrated in Figure 12 is used throughout the study. This chain was obtained using the transition matrix given in (65) and is referred to as the reference profile $\boldsymbol{\theta}_R$. This is the profile the approximation aims to reproduce given simulated data. We establish data for all the cases $1-4$ presented in Table 1. We code a case by first referring to the wavelet used, see Figure 9, followed by a variance from (64) and a case number $1 - 4$. As an example, for the base case using the Gaussian wavelet with variance A we denote the case by "Gaussian $A1$". The data is initiated by simulating $\mathbf{z}|\boldsymbol{\theta}$, and then calculating $\mathbf{d} = \mathbf{W}\mathbf{z} + \mathbf{e}$, as we established in Section 3.1, and all cases considered are presented in Figure 13.

For the Gaussian case $A1$, multiple neighbor cases are studied, and for Gaussian cases $A\ 2-4$ the approximation is evaluated for 5 and 10 neighbors. This is

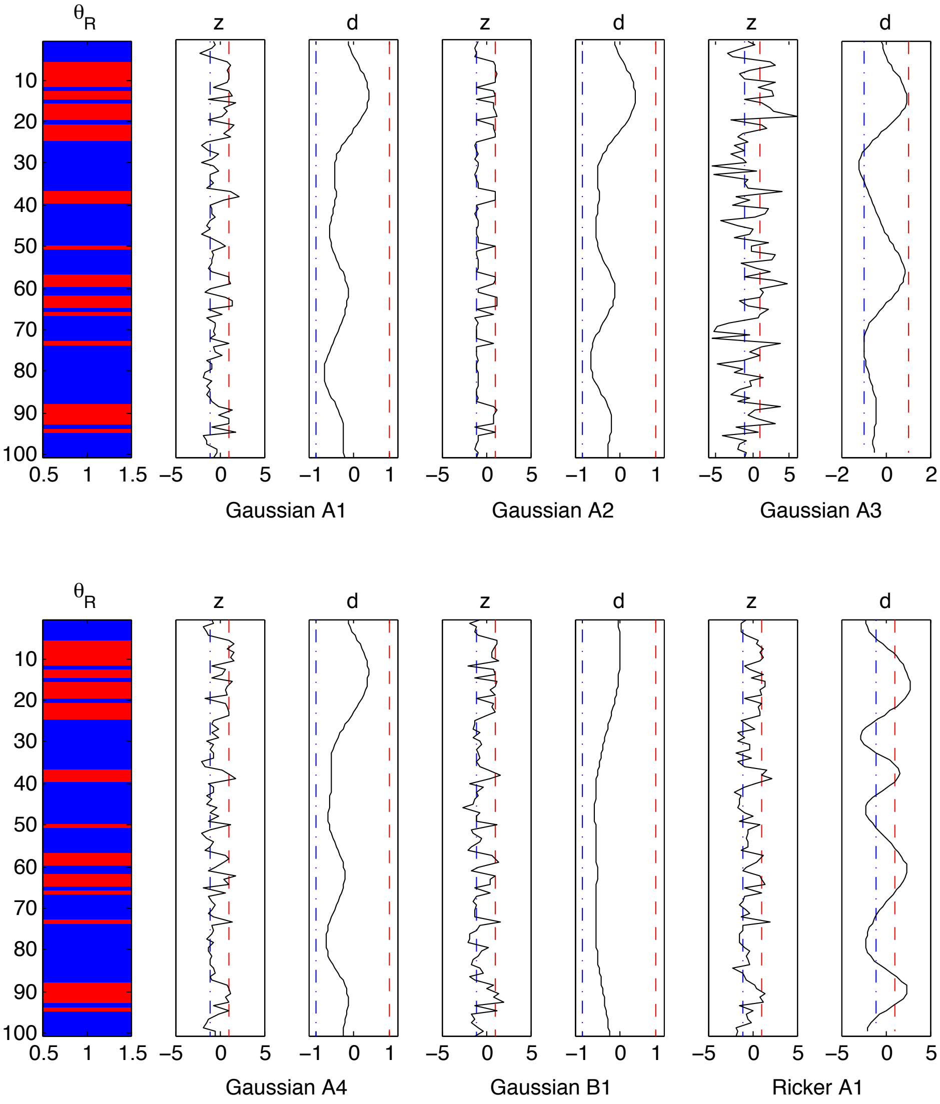


Figure 13: *To the left in this figure, the binary Markov chain reference profile,* $\boldsymbol{\theta}_R$*, is found. It has been included both in the top row as well as the lower row for comparison to the data* $\mathbf{d}$*, which is here plotted with the corresponding hidden layer* $\mathbf{z}$*. All test cases are here represented, i.e Gaussian cases A* $1-4$*,* $B1$ *and Ricker case* $A1$*. Expected values are plotted as striped lines in the plots of* $\mathbf{z}$ *and* $\mathbf{d}$*, where each states expectance line has its respective color.*

	Gaussian $A1$
0 neighbors	13.46%
1 neighbor	37.11%
2 neighbors	55.76%
3 neighbors	52.69%
4 neighbors	58.22%
5 neighbors	54.10%
10 neighbors	78.19%
15 neighbors	87.81%

Table 2: *Acceptance rates for Gaussian case* $A1$.

also done for a wide kernel case, Gaussian $B1$, and for the Ricker base case, $A1$. We now present each case one after the other, and in the last section they are reviewed together. There we present marginal probabilities, the marginal maximum aposteriori and simulations for some chosen cases where the MH algorithm is not applied.

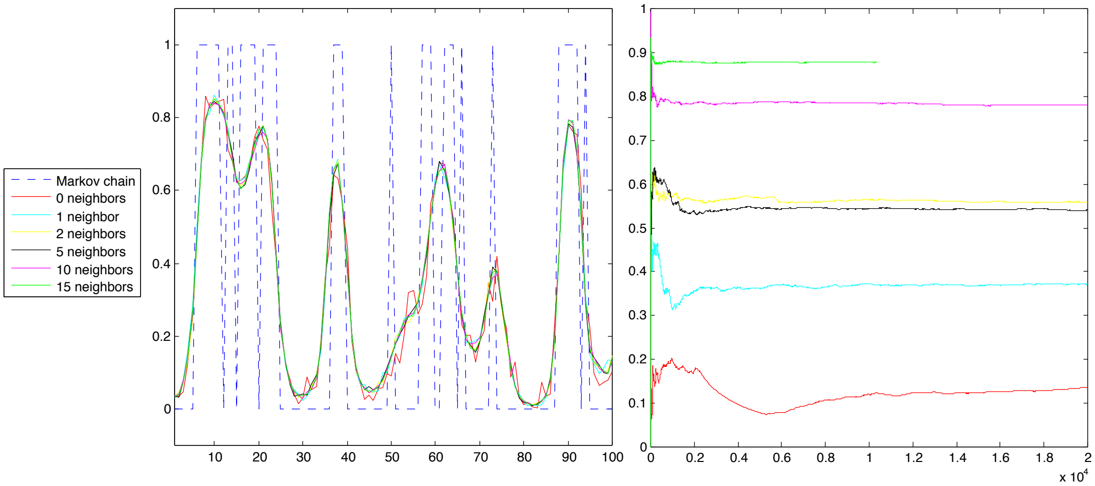
4.1.1 Gaussian A1

We now present the results for the approximation of the Gaussian case A1. This is the base case in Table 1, using the Gaussian wavelet in Figure 9 with variance A in (64). For this case we simulate realizations for $0, 1, 2, 3, 4, 5, 10$ and 15 neighbors.

We base the approximation evaluation on the acceptance rates of the MH algorithm for all the neighbor cases. The acceptance rates are presented in Table 2, and we now address each individual case. If and when the MH algorithm reaches convergency, we gain the limiting marginal distribution from the realizations. We have plotted the distributions and the acceptance rates for some of the neighbor cases in Figure 14.

For 0 neighbors we have an acceptance rate of 13.46%, see Table 2, which is the lowest acceptance rate of all the neighbor cases. In Figure 14a the zero neighbor case is plotted in red, and shows a slightly noisier marginal distribution. This is because the MH algorithm has not reached a satisfactory convergence, and when the acceptance rate is this low the MH algorithm usually uses a long time to reach convergency. This is seen in the acceptance plot for the zero case in Figure 14b, which shows a curve that has not quite settled yet. The MH algorithm here favors some simulated profiles causing this low acceptance, and thus the Markov chain get stuck on certain profiles for some time. This can be seen for the zero neighbors case in Figure 15, where the 20000 iteration profiles have been plotted.

For one neighbor the average acceptance rate has increased significantly and



(a) *Realizations marginal distribution, based on the last* 10000 *samples.*

(b) *Acceptance rates for all the neighbor cases.*

Figure 14: *Gaussian case A1 for* $0, 1, 2, 5, 10$ *and* 15 *neighbors.*

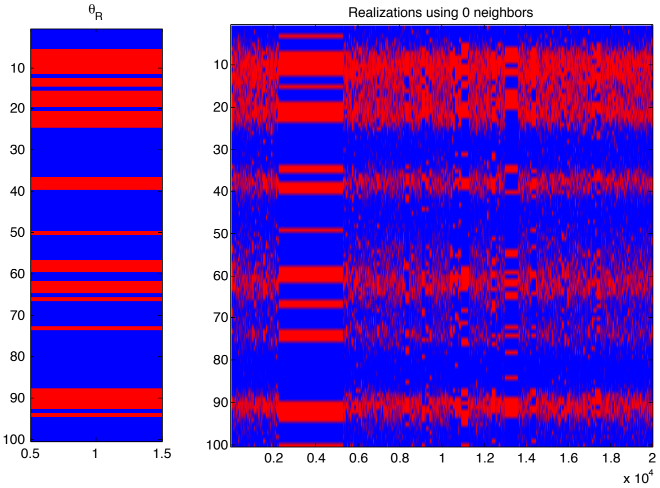


Figure 15: *Simulations for Gaussian case A1 using* 0 *neighbors plotted next to the reference profile. Because of the low acceptance rate, the algorithm get stuck on favoring profiles.*

is here 37.11%. Compared to the zero case, the acceptance rate is now nearly three times higher, and the MH algorithm has for one neighbor converged. There are now no profiles getting stuck in the simulations, and the marginal distribution is smoother. However, for the approximation, this is still not the best result since a high acceptance rate is desirable. For two to five neighbors the average acceptance rate bounces between 52.69% and 58.22%, see Table 2. Here it might seem like the approximation is better for neighbors of an even nature, but this is only speculation. For ten and fifteen neighbors, the average acceptance rate is 78.19% and 87.81%, respectively. As expected the approximation gets better for an increasing number of neighbors.

For further study of the other cases, we have chosen to concentrate on using 5 and 10 neighbors. The 15 neighbors case has a very high cost in CPU time, so we do not consider it. As for the 5 neighbor case, we chosen this since it had a good percentage and we wanted an odd number neighbor case as well. Plots of the simulations for 5 and 10 neighbors of the Gaussian case $A1$, similar to Figure 15, is supplied in Appendix C.1, see Figure 30. This is for visual comparison to the other noise cases, which we present next.

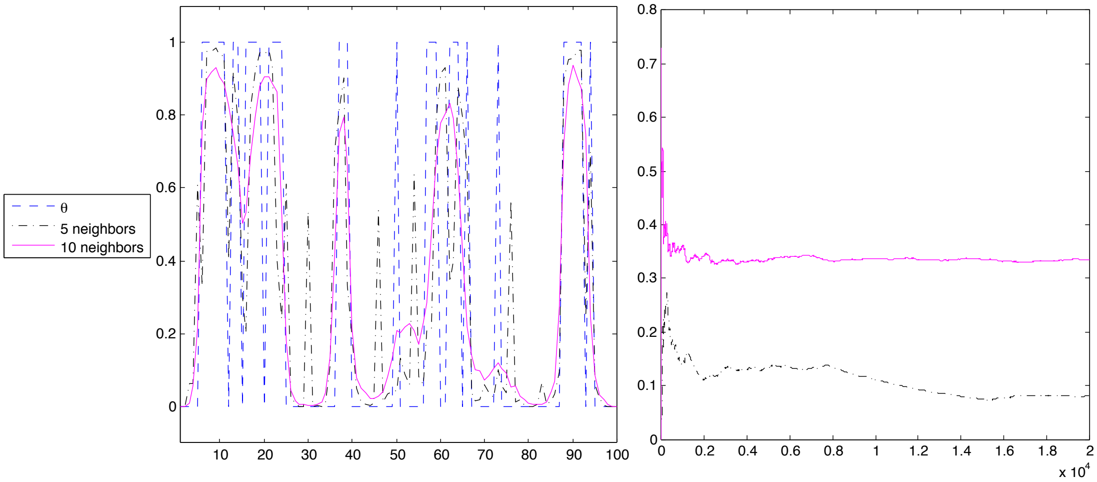
4.1.2 Gaussian A2

The Gaussian case $A2$ is a low noise case for the model. This is case 2 in Table 1, where we again use the Gaussian wavelet in Figure 9 with variance A in (64). The model is here more dependent of the likelihood, since much of the random noise has been removed. For this case we simulate realizations using 5 and 10 neighbors.

The resulting acceptance rate for 5 neighbors is only 8.03%, which is not a very good result for the approximation. The rate is lower than for the zero neighbors Gaussian case $A1$, and we here see a more rapid fluctuation in the marginal distribution. Also, the acceptance rate is presumed to be even lower, since the MH algorithm has not reached a satisfying convergency yet. Compared to the acceptance rate for the 5 neighbors Gaussian case $A1$, this rate is almost seven times lower.

The acceptance rate for 10 neighbors is 33.49%. This is lower than the 1 neighbor case for the Gaussian case $A1$, which was 37.11%, and compared to the 10 neighbors case it is over half the percentage lower. The reason for these lower acceptance rates is that the likelihood demands that the approximation needs to be more accurate when the noise decreases. The white noise to the data does not have much to say, but the noise in the hidden layer properties is the main reason of this occurrence. The marginal distribution and the acceptance rates are shown in Figure 16, and the simulation plots for both the 5 neighbors case and the 10 neighbors case are included in Appendix C.1, Figure 31.

As a side note we consider the inversion problem. The simulation plot in



(a) *Realizations marginal distribution.*

(b) *Acceptance rates.*

Figure 16: *Gaussian case A2 for* 5 *and* 10 *neighbors.*

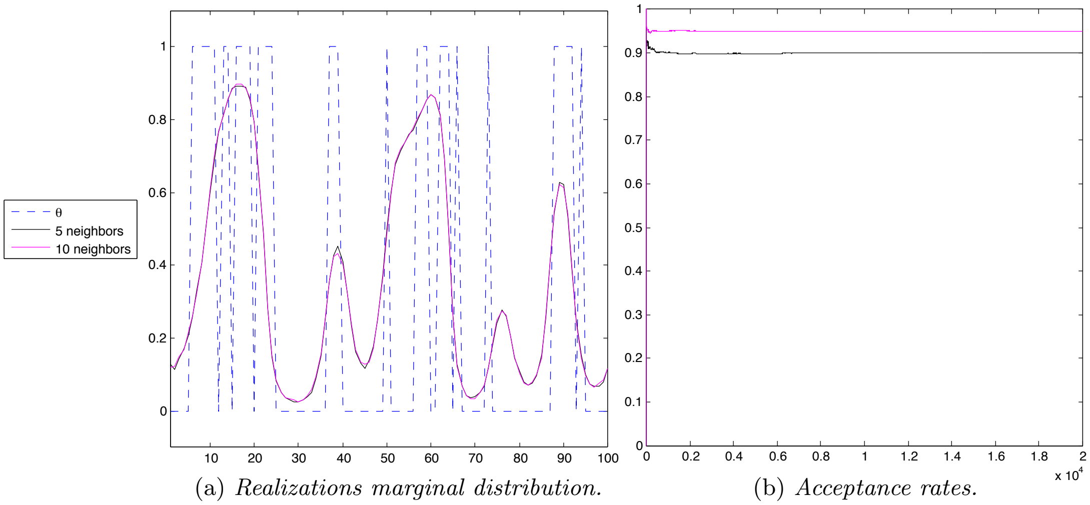
the appendix shows a visually cleaner appearance compared to the results for the base case, $A1$. The approximation does not recognize thin layers of states very well, but one can see tendencies of other states in the larger areas. The transition from larger areas of a state to the other state, the approximation manages to find with a pleasing result.

4.1.3 Gaussian A3

We here present the results for case 3 in Table 1, where we have used the Gaussian wavelet in Figure 9 and variance A in (64). For the Gaussian case $A3$, we increase the noise in the lower part of the two-level hidden Markov chain, i.e. for the properties of the hidden layer, $\mathbf{z}|\boldsymbol{\theta}$. We simulate approximations for 5 and 10 neighbors for this case also.

The acceptance rates here is for 5 neighbors 89.93% and for 10 neighbors, 94.95%, which are very good results regarding the approximation. The Gaussian case $A3$ did in fact give the best acceptance rates of the entire study. The marginal distribution and acceptance rates are illustrated in Figure 17, and the simulations for these two neighbor cases are embedded in Appendix C.1, see Figure 32.

We again make a short side note regarding the seismic inversion. The marginal distribution in Figure 17a and the simulation results found in the appendix, shows that the area from lattice point $10-20$ does not seem to notice state 0 in between the broad band of dominant state 1. For the references to states, see Figure 12. In Gaussian case $A1$ and Gaussian case $A2$ one can see



(a) *Realizations marginal distribution.*

(b) *Acceptance rates.*

Figure 17: *Gaussian case A3 for* 5 *and* 10 *neighbors.*

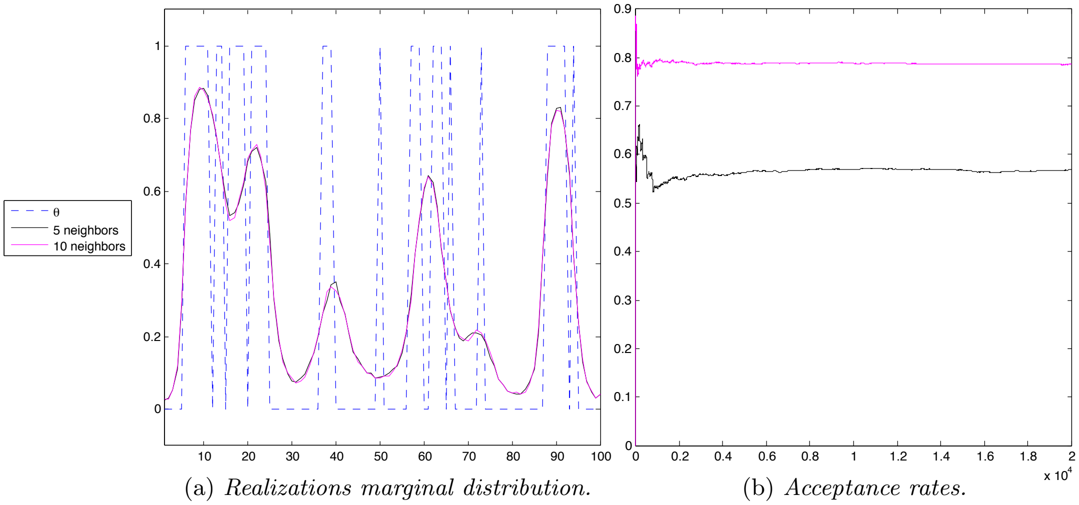
tendencies of state 0, but here there is a clear dominance of state 1, which the simulations in the appendix also illustrate.

4.1.4 Gaussian A4

In this section we consider case 4 in Table 1, with the Gaussian wavelet using variance A, i.e. Gaussian case $A4$. For the Gaussian case $A4$, the noise is increased in the upper part of the two-level hidden Markov chain, $\mathbf{d}|\mathbf{z}$. In other words we are increasing the white noise to the data. Again we generate realizations using 5 and 10 neighbors.

When using a maximum of 5 neighbors the result became a 56.89% acceptance rate, and for 10 neighbors this gave 78.61%. This is more in correspondence with the acceptance rates that we saw for the base case, see Table 2, only slightly higher. Adding more noise to the data does not seem to make that much of a difference to the approximation. Further, the marginal distribution of the realizations and the acceptance rates for the neighbor cases are shown in Figure18.

We now make some remarks regarding the application to the seismic inversion. The simulations, see Appendix C.1 Figure 33, also looks more similar to the Gaussian base case $A1$. As an example of this, we see that around lattice points $5-25$ state 0 can be seen as random noise in between the broad band of dominant state 1. If we compare it to the Gaussian case $A3$, the tendency of state 0 in between the broad band of state 1 for this particular area, is for the Gaussian case $A4$ significantly higher.



(a) *Realizations marginal distribution.* (b) *Acceptance rates.*

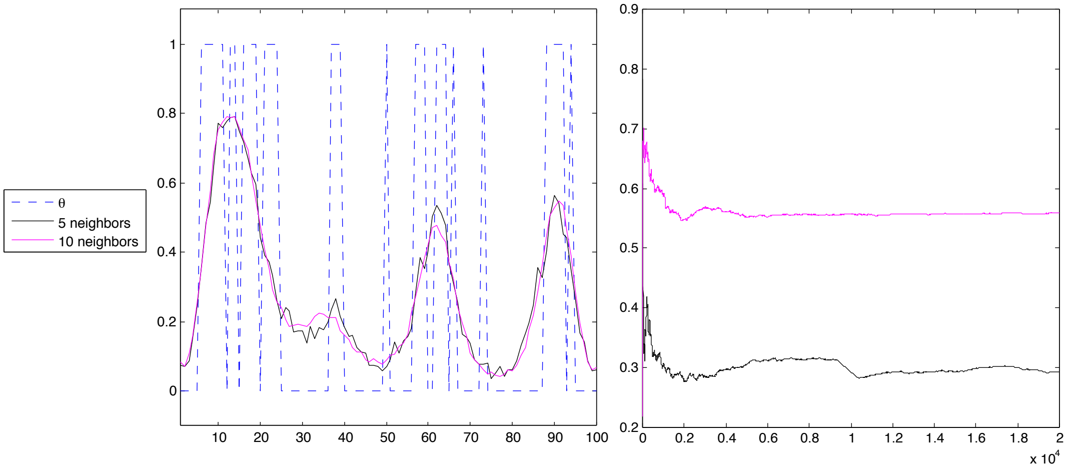
Figure 18: *Gaussian case A4 for* 5 *and* 10 *neighbors.*

4.1.5 Gaussian B1

The Gaussian case $B1$ is a wide kernel case, which is a higher variance in the wavelet used. We double the standard deviation, i.e. we are considering variance B in (64), and simulate for case 1 in Table 1. In Figure 13 the data, $\mathbf{d}$, for the Gaussian case $B1$ shows a less fluctuant curve when compared to all other cases.

The acceptance rates from using the MH algorithm and marginal distribution of the realizations are presented in Figure 19. These are supplemented with the image plot of the simulations in Appendix C.1, Figure 34. Empirical results for the acceptance rates when using 5 and 10 neighbors are 29.20% and 55.91%, respectively. The results for using 5 neighbors has not converged to a satisfying degree, which can be seen both in the acceptance plot and in the marginal distribution. However, it is not as bad as it was for the low noise Gaussian case $A2$ when using 5 neighbors. The acceptance rates for the Gaussian case $B1$ are the second worst we have come across, where only the Gaussian case $A2$ gave lower results. Compared to the Gaussian base case $A1$ the acceptance is over 20% lower in both neighbor cases.

Regarding the seismic inversion, the approximation now has great trouble finding thinner areas. The samples accepted by the MH algorithm shows that the thinner slices are only registered slightly in the approximations. This is seen in the simulation plots in Figure 34 in Appendix C.1, where the result is noisier than for any of the other cases. Figure 19a also shows a more flattening tendency in the marginal distribution curve.



(a) *Realizations marginal distribution.*

(b) *Acceptance rates.*

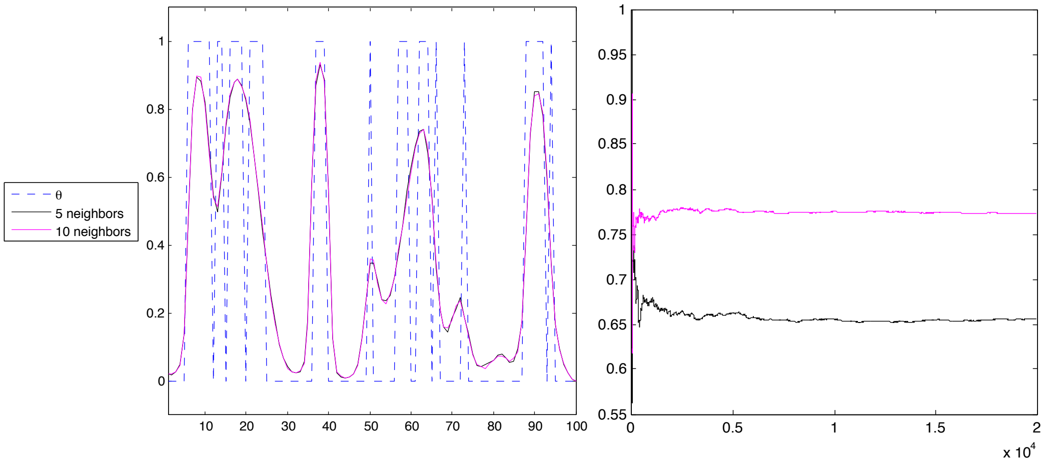
Figure 19: *Gaussian case B1 for* 5 *and* 10 *neighbors.*

4.1.6 Ricker A1

We are in this section considering case 1 in Table 1, i.e. the base case. We have change the type of wavelet and now consider the Ricker wavelet, see Figure 9, with variance A in (64). We here discuss the approximation results for the Ricker case $A1$ using 5 and 10 neighbors.

The acceptance rates from using the MH algorithm are shown in Figure 20, together with the marginal distributions. The acceptance rates for 5 and 10 neighbors are 65.58% and 77.35%, respectively. Compared to the Gaussian base case, this is a higher rate for the 5 neighbors case, but a slightly lower rate for the 10 neighbors case.

We also for this case make a note regarding the application to the inversion problem. The simulations have been supplemented to the Appendix C.1, see Figure 35. The Ricker case shows a cleaner result in the simulations, much like what we saw for the low noise case, Gaussian $A2$. However for the Ricker case we have a higher acceptance rate than for the low noise case. The thin area layers, consisting usually of singular points, is shown tendencies for and are clearer here than for most of the other cases. In the larger band of state 1 between lattice point $5-25$, there is a strong tendency of state 0. This is also noticeable in the band of state 1 around lattice point $85-95$. When compared to the Gaussian case $A1$, the approximation algorithm combined with the MH algorithm generates a better result for the Ricker case $A1$. Both when considering the acceptance rates and the application to seismic inversion.



(a) *Realizations marginal distribution.* (b) *Acceptance rates.*

Figure 20: *Ricker case A1 for* 5 *and* 10 *neighbors.*

4.1.7 Review of the Cases

In this section we present the cases together using statistical tools such as the marginal maximum aposteriori (MMAP) and marginal probabilities. We have also taken some of the cases and simulated approximations without the use of the MH algorithm, which are presented in this section.

The MMAP is a useful tool to gain a view of the dominant states, and the marginal probabilities are great for viewing the distribution of each state. Using the last 10000 realizations of all the cases presented for 10 neighbors, these two statistics have been plotted and are presented in Figure 21. Here the seismic inversion is at center, and shows the quality and the accuracy of the approximation in each case. Even though the MMAP does not register thinner layers, tendencies can be seen in the marginal probabilities that something might be present in certain areas. However, for the wide kernel case, Gaussian case $B1$, many of these areas are not registered. The marginal probabilities that seems to register the thinner areas best to some degree are the Gaussian case $A2$ and the Ricker case $A1$.

The acceptance rates for all presented cases using 5 and 10 neighbors are given in Table 3. We choose some of these cases to apply the approximated forward-backward algorithm to without using the MH algorithm, and we simulate 10000 iterations for each of these cases. This was done for the Gaussian base case $A1$ for 5 neighbors, to see how a relatively good approximation looks like without filtering. We simulated the low noise Gaussian case $A2$ using 5 neighbors, since only 8.03% of the suggested profiles were accepted for this

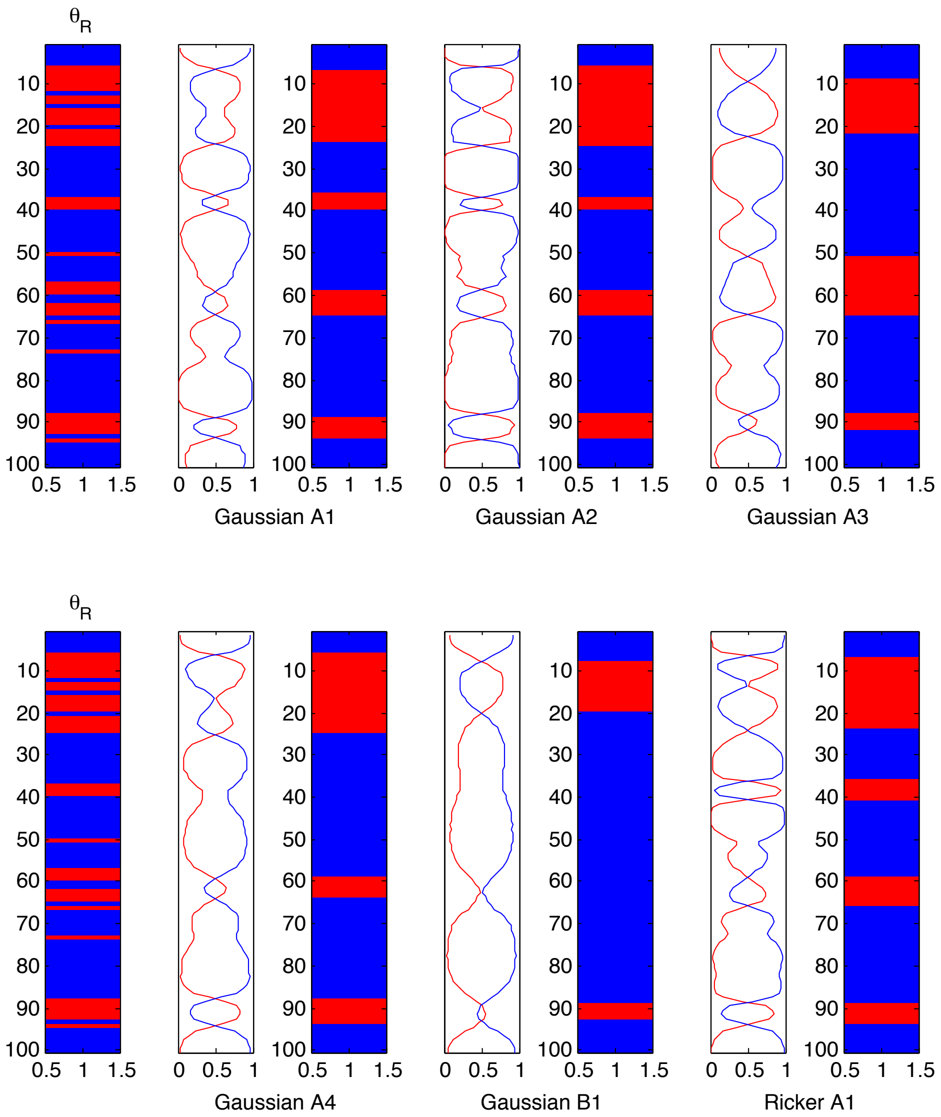


Figure 21: *Marginal probabilities for all cases plotted left of the respective marginal maximum aposteriori, MMAP. Reference profile, $\boldsymbol{\theta}_R$, are presented in each row for comparison.*

	5 neighbors	10 neighbors
Gaussian $A1$	54.10%	78.19%
Gaussian $A2$	8.03%	33.49%
Gaussian $A3$	89.93%	94.95%
Gaussian $A4$	56.89%	78.61%
Gaussian $B1$	29.20%	55.91%
Ricker $A1$	65.58%	77.35%

Table 3: *Acceptance rates when using* 5 *and* 10 *neighbors as maximum number of neighbors. The acceptance rates are for the Gaussian A cases* $1-4$, *Gaussian case $B1$ and Ricker case $A1$.*

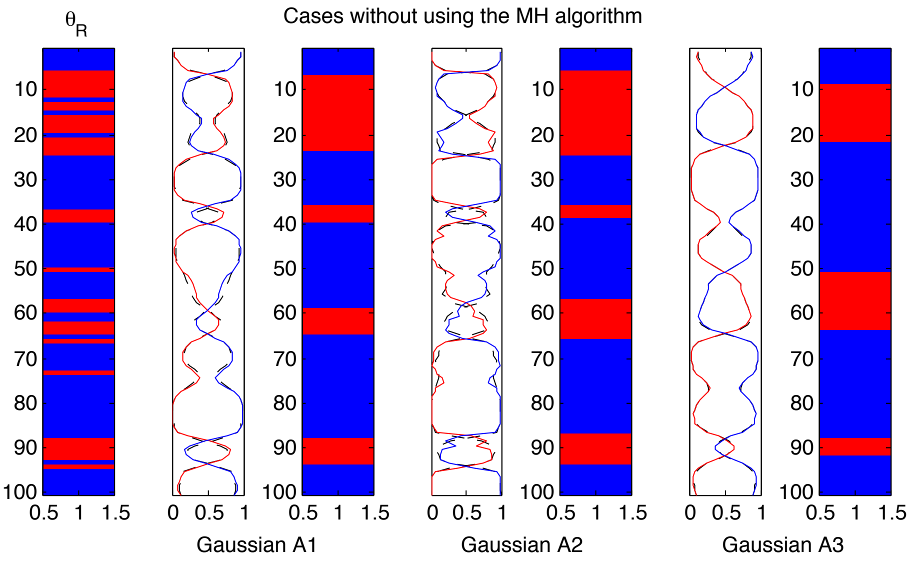


Figure 22: *Marginal probabilities of the approximation plotted next to the maximum marginal approximation for cases without using the MH algorithm. Dashed black lines shows converged marginal probabilities for each corresponding case using the MH algorithm. Reference profile $\boldsymbol{\theta}_R$ is found to the left.*

case, and we also chose to simulate the 10 neighbors Gaussian case $A3$, which has the highest acceptance rate of all considered cases. For these three cases the simulations are presented in Appendix C.2. The maximum marginal approximation and marginal probabilities are found in Figure 22, where they are plotted with the corresponding converged marginal probabilities from using the MH algorithm, which are shown as black dashed lines. The figure shows that the approximation results, when not applying the MH algorithm, gives very similar results compared to applying the MH algorithm. The Gaussian case $A3$, which had an acceptance of 94.95%, has overlapping curves that are hard to tell apart. Even the Gaussian case $A2$ shows approximately the same curve for the marginal probabilities. We therefore conclude that the approximated forward-backward algorithm by Austad (2011) is viable even when the acceptance rate is as low as 8.03%.

4.2 Multiple State Markov Chain

We here present approximation results for the convolutional Bayesian model with a four state Markov chain prior. The transformation of the Bayesian model into the form of a binary MRF can for this case be found in Section 3.2.2. For our four state Markov chain prior we have that $\boldsymbol{\theta} \in \{0, 1, 2, 3\}^n$, and the chain is defined by a transition matrix of size 4×4. We adapt a transition matrix from a study of a real-data case presented in Rimstad and Omre (2013), and use this to define a transition matrix for our four state case. We define the transition matrix as

$$\mathbf{P}_4 = \begin{bmatrix} 0.60 & 0.01 & 0.09 & 0.30 \\ 0.01 & 0.63 & 0.069 & 0.30 \\ 0.01 & 0.01 & 0.699 & 0.299 \\ 0.15 & 0.03 & 0.12 & 0.70 \end{bmatrix}, \tag{67}$$

which has limiting probabilities $p_{l4} = [0.145, 0.049, 0.262, 0.544]$. As for the expectations of $z_i|\theta_i$, we find inspiration in the three state synthetic test case in Rimstad and Omre (2013), and take the expected values to be

$$\mu_{z_i|\theta_i} = \begin{cases} -1 & \text{if } \theta_i = 0, \\ 0 & \text{if } \theta_i = 1, \\ 1 & \text{if } \theta_i = 2, \\ 2 & \text{if } \theta_i = 3. \end{cases} \tag{68}$$

We also consider a case with the expected values

$$\mu'_{z_i|\theta_i} = \begin{cases} -1 & \text{if } \theta_i = 0, \\ 0 & \text{if } \theta_i = 1, \\ 0.7 & \text{if } \theta_i = 2, \\ 1 & \text{if } \theta_i = 3, \end{cases} \tag{69}$$

which is done for the Gaussian case $A1'$. Here some of the expected values are moved closer together such that they have the same range as the expectance values for the binary Markov chain prior.

For the evaluation of the approximation we simulate a single four state Markov chain to use in our studies. This chain is based on the transition matrix in (67) and is presented in Figure 23, together with the corresponding binary representation. Recall that we for the four state case have corresponding binary variables $\boldsymbol{\phi} \in \{0,1\}^{2n\times 1}$, which is the form the approximated forward-backward algorithm is handling. We denote this reference chain by $\boldsymbol{\theta}_R$ and the binary correspondence by $\boldsymbol{\phi}_R$, which contains twice the elements of $\boldsymbol{\theta}_R$.

For the four state case we generate data using the covariance cases $1-4$ in Table 1. We code the cases is the same way as we did for the two state Markov chain prior, and all cases up for evaluation are found in Figure 24. We now present approximation results for the Gaussian cases $A\,1-4$ and the Ricker base case $A1$, where we are using variance A in (64) and the expected values in (68). For the second set of expected values in (69), we simulate realizations for the Gaussian case $A1'$. Each case is simulated for 10 neighbors, which is comparable to the two state Markov chain prior when using 5 neighbors. The approximation is now handling twice the elements, than for the binary Markov chain prior, and this costs in CPU time. There is also a cost for considering many neighbors. We

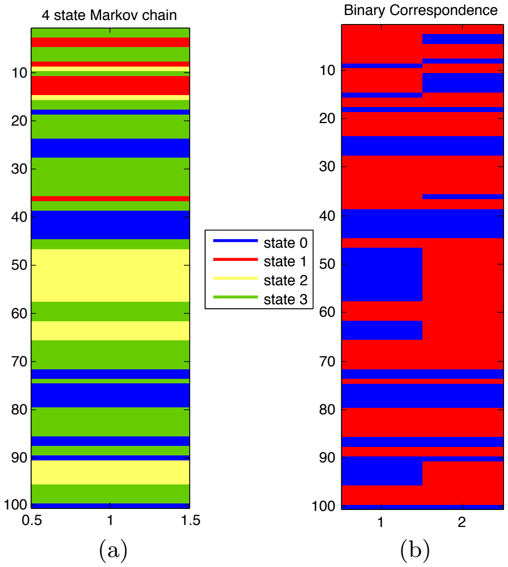


Figure 23: *Figure (a) shows a four state Markov chain, $\boldsymbol{\theta}_R$, generated from (67) with size $n = 100$, and (b) shows the corresponding binary variables, $\boldsymbol{\phi}_R$, for the profile in (a). The color chart in the middle applies for both profiles.*

generate 10000 profiles trying to match $\boldsymbol{\phi}_R$ from the approximation algorithm, and we use the MH algorithm to obtain the realizations. The last 6000 simulated profiles are taken as the final realizations, and these are used to calculate the statistics in the following sections.

4.2.1 Gaussian $A1$

In this section we consider case 1 in Table 1 using the Gaussian wavelet with variance A. We are here presenting results for the expected values in (68).

The acceptance rate for this case is 29.06%, which is a lower rate compared to the 5 neighbors case of the two state Markov chain which gave 54.10%. The acceptance rate of the MH algorithm is plotted in Figure 25a, where we see that the algorithm has converged. The simulated profiles have been supplemented to Appendix C.3, see Figure 39. For the four state Markov chain cases the number of binary variables in the approximating algorithm has doubled. Also, there are now four state expectations to be concerned with, and they are in a wider range than for the binary Markov chain prior. These elements that have changed causes other aspects of the model to change as well, and all we can say is that the acceptance rate tells us that the approximation here is not as good as the corresponding binary Markov chain Gaussian case $A1$.

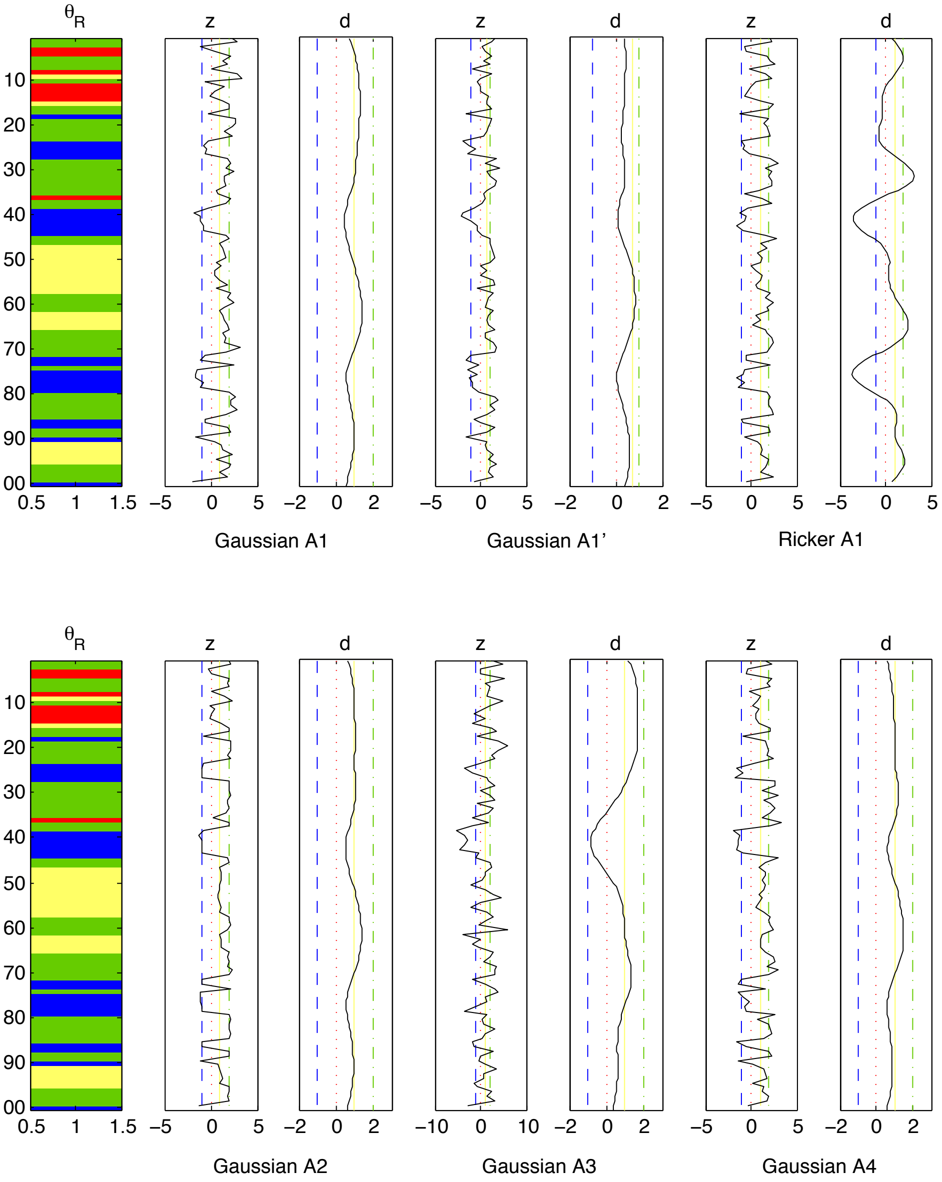
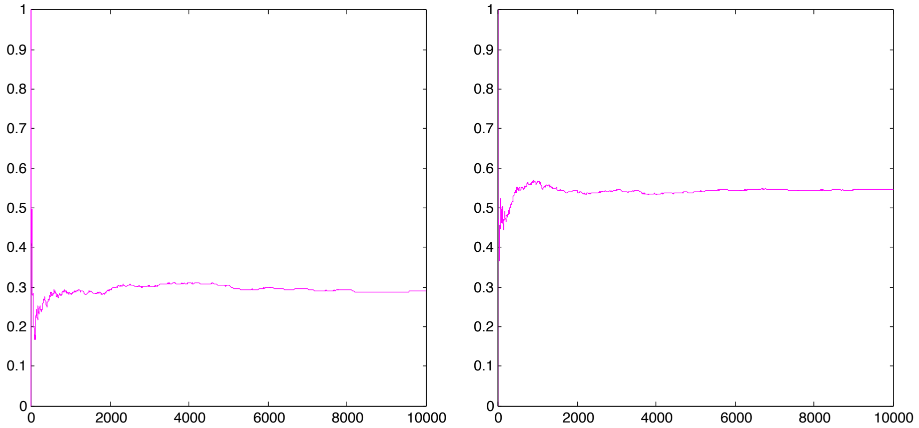


Figure 24: *Generated data* $\mathbf{d}$ *for Gaussian cases* $A\ 1-4$*,* $A1'$ *and Ricker case* $A1$ *plotted with hidden layers* $\mathbf{z}$ *and the reference profile* $\boldsymbol{\theta}_R$*. The expectance of each state are shown as solid, dashed, dotted and dash-dotted lines in their respective colors, which are defined in Figure 23.*



(a) *Acceptance Gaussian case $A1$.* (b) *Acceptance Gaussian case $A1'$.*

Figure 25: *Acceptance rates for Gaussian case $A1$ and $A1'$ using 10 neighbors, where rates ended at 29.06% and 55.05%, respectively.*

4.2.2 Gaussian $A1'$

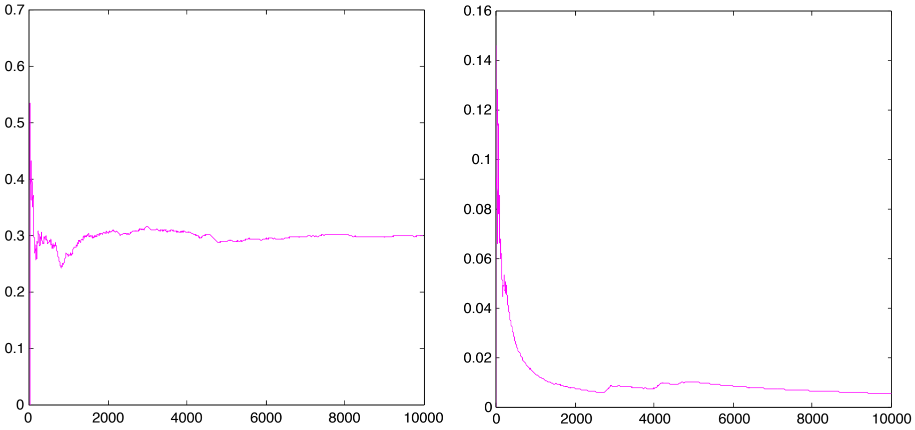
We now consider the Gaussian case $A1$ with the expected values given in (69). The expectations are here in the same range as for the binary Markov chain prior.

The acceptance rate for this case is plotted in Figure 25b, where the acceptance rate ends at 55.05%. This is now very close to the 5 neighbors case in the binary Markov chain Gaussian $A1$ simulation. As mentioned in the previous section, this gave a percentage of 54.10%. Compared to the Gaussian case $A1$ for the four state Markov chain prior, the acceptance has now gone up 25.99%. In Appendix C.3 the simulations for this case can be found together with the Gaussian case $A1$ of the previous section, see Figure 39.

4.2.3 Ricker $A1$

We here consider case 1 in Table 1 with the Ricker wavelet using variance A, and we are again considering the expectations in (68). The data, $\mathbf{d}$, in Figure 24 for the Ricker case $A1$, shows a more fluctuant curve compared to any of the other cases.

In Figure 26a the acceptance rate for this case can be found, which resulted in an acceptance rate of 29.97%. This is close to the acceptance for the Gaussian case $A1$, which was 29.06%. These two cases are closer together in acceptance rate than the corresponding binary Markov chain cases using 5 neighbors, which



(a) *Acceptance Ricker case* $A1$.

(b) *Acceptance Gaussian case* $A2$.

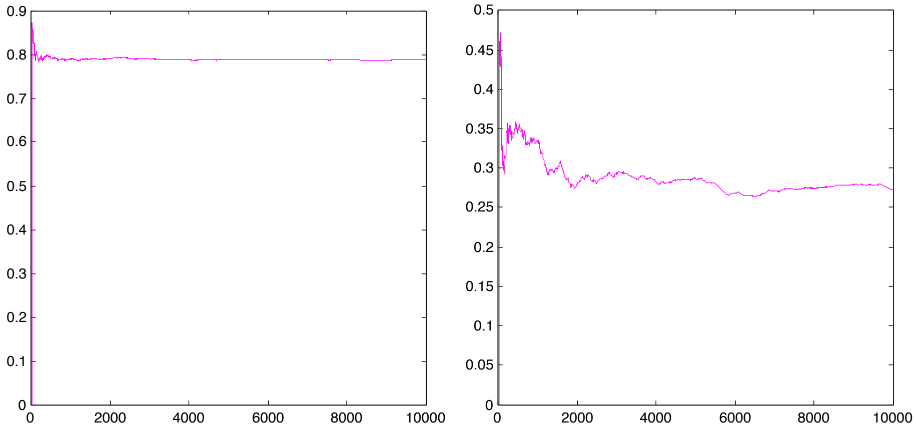
Figure 26: *Acceptance rates for Ricker case* $A1$ *and Gaussian case* $A2$ *using* 10 *neighbors, where the rates ended at* 29.97% *and* 0.53%*, respectively.*

were approximately 11% apart, see Table 3. The binary Markov chain for the Ricker case $A1$ using 5 neighbors gave an acceptance rate of 65.58%, and this case thus results in about half the acceptance rate in comparison. The simulation results for this case can be found in Appendix C.3, Figure 40.

4.2.4 Gaussian $A2$

We now consider case 2 in Table 1 with the Gaussian wavelet using variance A. We are here using the expectations in (68). This is a low noise case, where the model is depending more on the accuracy of the likelihood.

For the binary Markov chain prior the low noise case generated a really low acceptance of only 8.03% for the 5 neighbors case. Here we have an even lower acceptance rate, which is basically zero. In Figure 26b the acceptance has been plotted, and at the end the acceptance rate is 0.53%. The MH algorithm is stuck on a profile, and the Markov chain of simulated profiles does thus not move. To see this, the simulation results has been plotted and is supplemented to Appendix C.3 and can be found in Figure 40. The MH algorithm has because of this not converged to a satisfying degree, and the acceptance rate is assumed to be even lower.



(a) *Acceptance Gaussian case* $A3$.

(b) *Acceptance Gaussian case* $A4$.

Figure 27: *Acceptance rates for Gaussian cases* $A3$ *and* $A4$ *using* 10 *neighbors, where rates ended at* 78.72% *and* 27.27%, *respectively.*

4.2.5 Gaussian $A3$

In this section we are considering case 3 in Table 1. The Gaussian wavelet still has variance A, and we use expected values in (68). Compared to the Gaussian case $A1$, the noise is here increased for the covariance matrix of $\pi(\mathbf{z}|\boldsymbol{\theta})$.

For the binary Markov chain prior this was the case providing the highest acceptance rates, and this is the result here as well. For the Gaussian case $A3$ using 10 neighbors the acceptance rate ended at 78.72%. The acceptance rate has been plotted in Figure 27a. The corresponding binary Markov chain case using 5 neighbors generated an acceptance rate of 89.93%, so we have here a slightly lower result. However, this has so far been a trend for almost all the cases considered. The simulated approximated profiles has been added to Appendix C.3, see Figure 41.

4.2.6 Gaussian $A4$

We here consider case 4 in Table 1 for the Gaussian wavelet with variance A. The expected values used is found in (68). For this case the noise is increased in the covariance matrix of the distribution of $\mathbf{d}|\mathbf{z}$, so basically we increase the white noise in the data.

The acceptance rate that was generated by the MH algorithm here ended at 27.27%, and the acceptance is plotted in Figure 27b. We see that this case is

	5 neighbors binary MC	10 neighbors four state MC
Gaussian $A1$	54.10%	29.06%
Gaussian $A1'$	54.10%	55.05%
Ricker $A1$	65.58%	29.97%
Gaussian $A2$	8.03%	0.53%
Gaussian $A3$	89.93%	78.72%
Gaussian $A4$	56.89%	27.27%

Table 4: *Acceptance rates for the four state Markov chain (MC) prior when using* 10 *neighbors as maximum number of neighbors. The acceptance rates are for the Gaussian A cases* $1-4$*,* $A1'$ *and the Ricker case* $A1$*. For comparison, the binary Markov chain cases when using* 5 *neighbors have been added to the table.*

close in acceptance rate with the base case, which we also saw for the binary Markov chain prior cases. The curve in Figure 27b is unsteady in nature, which is caused by stuck approximated profiles, which can be seen in Figure 41 in Appendix C.3. Compared to the corresponding binary Markov chain case with 5 neighbors, which had an acceptance rate of 56.89%, we again have a lower acceptance.

4.2.7 Review of the Cases

In this section we present the marginal probabilities and MMAP of all the four state Markov chain cases considered. We also discuss the seismic inversion problem, and the quality of the solution of the approximation. Some of the cases we have chosen to approximate without the use of the MH algorithm, and these are included in this section.

In Table 4 the acceptance rates have been presented for the four state Markov chain prior. For comparison, acceptance rates from the corresponding cases of the binary Markov chain prior using 5 neighbors have been added to the table. The four state Markov chain gave lower acceptance rates for almost every case, where the exception is for the Gaussian case $A1'$ which gave a slightly higher acceptance rate. For this particular case the range of the expected values are the same for both priors.

Each case has been simulated for 10000 iterations, which can all be seen in Appendix C.3, and the last 6000 are taken as the resulting realizations. These are used to calculate the MMAP and marginal probabilities found in Figure 28. The Gaussian case $A2$ had an approximately zero acceptance rate, but has been added to the figure even so. The simulation results found in the appendix are hard to interpret as they seem very noisy. The only clearly dominant states seen are bands of state 0 (blue) and a general noisy dominance of state 2 (yellow) and state 3 (green). Here state 0 and state 3 have the lowest and highest expected

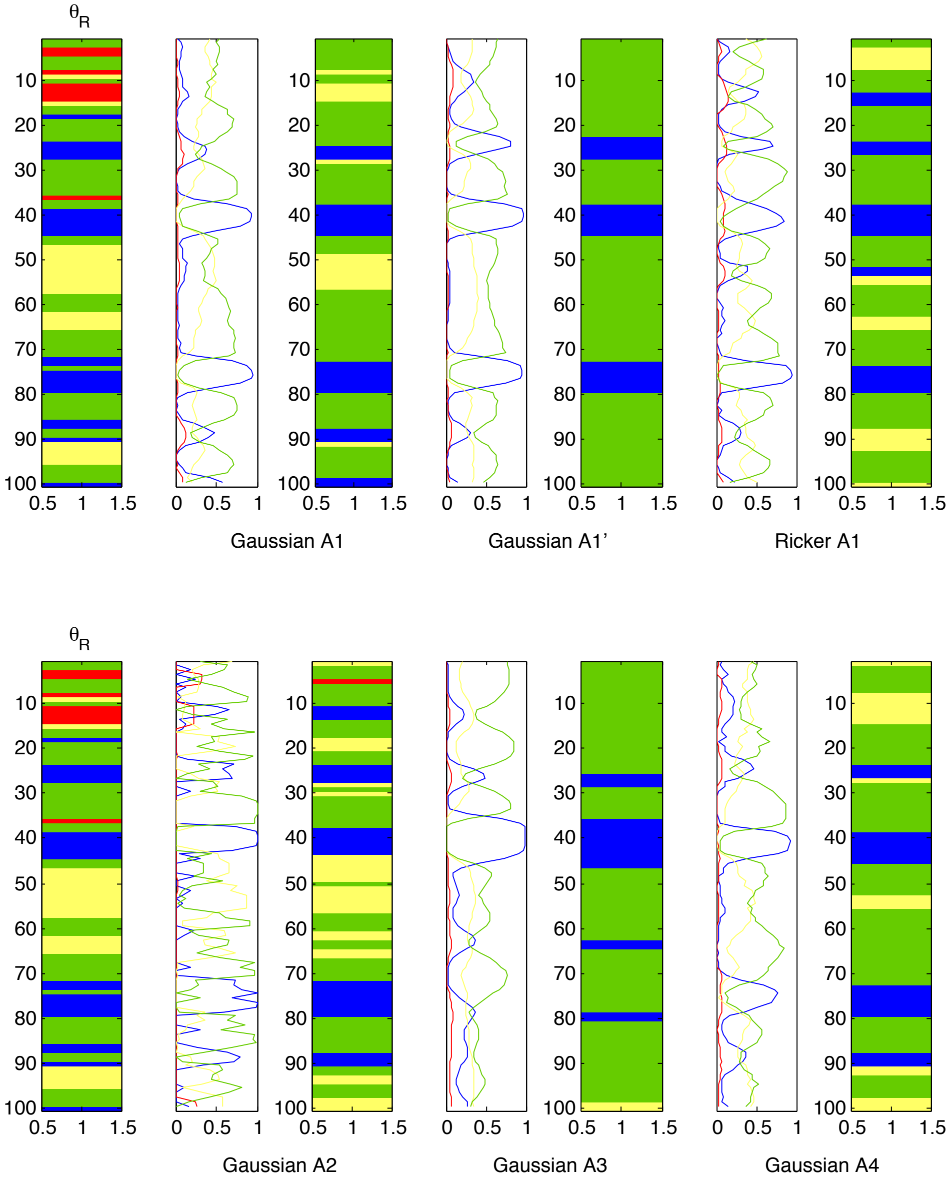


Figure 28: *Marginal probabilities and MMAP for all considered cases for the four state Markov chain prior. The states color coding are found in Figure 23.*

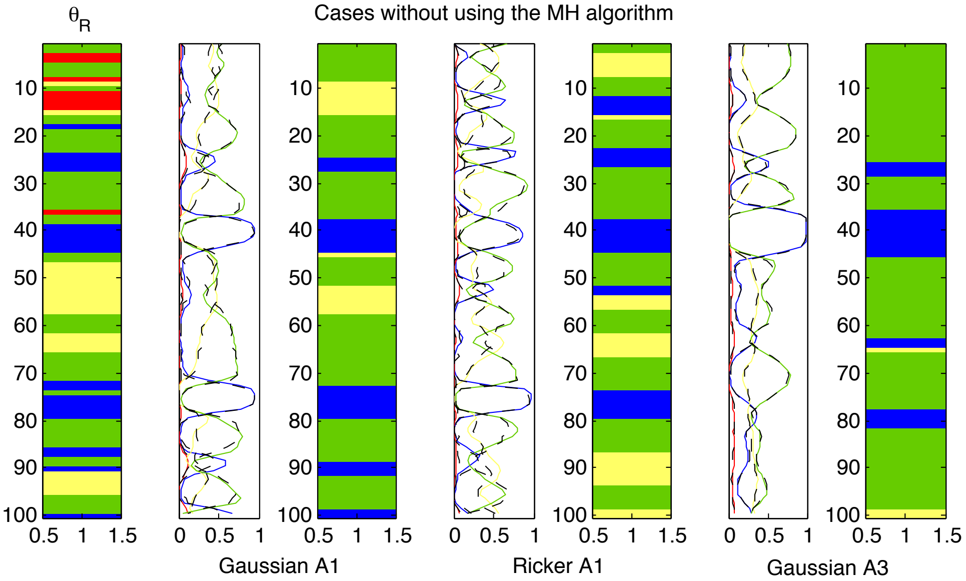


Figure 29: *Marginal probabilities of the approximation plotted next to the maximum marginal approximation for cases without using the MH algorithm. Dashed black lines shows converged marginal probabilities for each corresponding case using the MH algorithm. Reference profile* $\boldsymbol{\theta}_R$ *is found to the left.*

values, respectively. State 1 (red) is also present, but at a low percentage in the plots. This is confirmed by the MMAP and marginal probabilities in Figure 28, where only three states are registered as dominant for the converged cases. For the Gaussian case $A1'$, which has the same range for the expected values as the binary Markov chain prior, only state 0 and state 3 are registered in the MMAP. The Gaussian case $A1$ in comparison has three registered states.

For the Gaussian case $A1$ and $A3$ we apply the approximation using 10 neighbors as maximum number of neighbors, and simulate without using the MH algorithm. This is also done for the Ricker case $A1$. The marginal probabilities and maximum marginal approximations for these cases can be found in Figure 29, where the marginal probabilities for the corresponding case where the MH algorithm has been used have been plotted using dashed lines in black. We here get the same results as for the binary Markov chain prior, where the marginal probabilities are approximately the same for the approximation and the aposteriori, and this regardless of the acceptance rates. The lowest acceptance rate in these cases was found for the Gaussian case $A1$, which had an acceptance of 29.06%. The marginal probabilities shows that the differences from using the MH algorithm and not using it, are small. The same is seen for the Ricker case $A1$. For the Gaussian case $A3$, which generated the highest acceptance rate, the marginal probabilities are here hard to tell apart. As was the case for the corresponding Gaussian $A3$ for the binary Markov chain prior. The approximation and the aposterior are very close in nature for every case we have tested, and we therefore conclude that the approximated forward-backward algorithm is quite viable both for the binary Markov chain prior and the four state Markov chain prior of this convolutional Bayesian model.

5 Closing Remarks

In this thesis an approximate forward-backward algorithm for binary MRFs has been evaluated. We have considered a convolutional Bayesian model for a two-level hidden Markov chain, and transformed the model into a binary MRF. The connection between the field of Bayesian modeling and MRFs has been mentioned in earlier papers, e.g. (Austad, 2011), but case studies for a convolutional model has not been provided before.

The transformation from the Bayesian model into a binary MRF was done by establishing interaction parameters for an energy function of binary variables. The transformation of the posterior distribution depends on the prior of the model, which in this case was a Markov chain. The number of possible states in this Markov chain determined the form of the energy function, and the size and shape of the corresponding DAG of the clique set of the MRF. The approximation provided by Austad (2011) is based on minimizing the SSE for the interaction parameters in the energy function. Given a maximum number of neighbors an approximation is made each time a variable is summed out, and a DAG is built using the approximated nodes for the forward-backward algorithm to use in further calculations. For the evaluation, an independent proposal MH algorithm was implemented to quantify the quality of the approximation. We have applied the approximation both for cases using a binary Markov chain prior and a four state Markov chain prior.

The MH algorithm gave better acceptance rates for the approximation as the number of neighbors are increased, which was to be expected. However, for each added neighbor there is a cost in CPU time, and for many neighbors this cost becomes high. The approximation also operates best when the noise is highest in the categorical values properties, i.e. $\mathbf{z}|\boldsymbol{\theta}$, such that the model is less dependent of the likelihood. Increasing the white noise to the data, i.e. for $\mathbf{d}|\mathbf{z}$, gave approximately the same results in the acceptance rates as the base case using the MH algorithm. Considering the application to the seismic inversion however, shows that smaller areas in between larger areas of states is not registered very well when the noise is increased for $\mathbf{z}|\boldsymbol{\theta}$. This is seen both for the two state case as well as the four state case. One restriction for the four state case was the expected values of the hidden layer. The approximated forward-backward algorithm seemed to have trouble noticing that there were in between states, and majorly found only three of the states both for the approximation and the aposteriori. One state did not even get registered in the MMAP or the maximum marginal approximation. However, comparing the marginal probabilities of the approximation to the aposteriori gave results close in nature. Even the Gaussian case $A2$ using 5 neighbors, which had an acceptance of 8.03% for the binary Markov chain prior, gave approximately the same marginal probabilities from using the MH algorithm and for not using it. Thus, the approximated forward-backward algorithm of Austad (2011) seems viable for this convolutional Bayesian model.

As for further research, different methods for transforming the model would be interesting to consider. Also more simulations would be preferable, perhaps for a lower number of neighbors. Recall that the 2 and 4 neighbors cases gave higher acceptance rates than the 5 neighbors case did, i.e. for the two state Markov chain Gaussian case $A1$. Fewer neighbors would also reduce the CPU time for the four state Markov chain case, making it more efficient to simulate more cases. Further, a study concerning even (symmetric) and odd (non-symmetric) neighbors is compelling, since even numbered neighbors seemed to be giving better acceptance rates than the odd once did. On a different note, in this thesis we only considered constant covariance matrices for the likelihood of the Bayesian model. In the real data tests in Rimstad and Omre (2013) and Ulvmoen and Hammer (2010), the variances depends on the underlying Markov chain and also considers multiple properties in the hidden layer. The real data case presented in Ulvmoen and Hammer (2010) is of even more complex form, where the elements of the hidden layer are not independent of one another. These more complex cases would be a great research areas to explore further for the approximate forward-backward algorithm of Austad (2011).

We conclude this thesis by repeating that the approximated forward-backward algorithm of Austad (2011) seems to be viable for this convolutional Bayesian model, but a more thorough study with more complex test cases and variation in the parameters is recommendable.

References

Austad, H. M. (2011). *Approximations of Binary Markov Random Fields; An approximate forward-backward algorithm applied to binary Markov random fields*. Technical report, Norwegian University of Science and Technology.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B*, **36** 192–236.

Blake, A., Kohli, P. and Rother, C. (2011). *Markov Random Fields for Vision and Image Processing*. MIT Press, Cambridge, MA, USA.

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, **49** 327–335.

Gamerman, D. and Lopes, H. F. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Second edition ed. Chapman & Hall/CRC.

Hamada, M. S., Wilson, A. G. and Reese, C. S. (2008). *Bayesian Reliability*. Springer, New York, NY, USA.

Lee, P. M. (2012). *Bayesian Statistics : An Introduction (4th Edition)*. Wiley, Somerset, NJ, USA.

Rimstad, K. and Omre, H. (2013). Approximate posterior distributions for convolutional two-level hidden Markov models. *Computational Statistics & Data Analysis*, **58** 187–200.

Tjelmeland, H. and Austad, H. M. (2012). Exact and approximate recursive calculations for binary Markov random fields defined on graphs. *Journal of Computational and Graphical Statistics*, **21** 758–780.

Ulvmoen, M. and Hammer, H. (2010). Bayesian lithology/fluid inversion-comparison of two algorithms. *Computational Geosciences*, **14** 357–367.

A Reformulating the Posterior

For the Bayesian posterior distribution given in (35) we will in this appendix section give the thorough calculations on reformulating the distribution to the form of the energy function.

Before we start note that, $\pi(\theta_1|\theta_0) = \pi(\theta_1)$ is the limiting probability distribution of θ_1 and will create a special boundary case. We start by expanding the posterior,

$$\begin{aligned}
\pi(\boldsymbol{\theta}|\mathbf{d}) &\propto \pi(\mathbf{d}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\
&= \frac{|\boldsymbol{\Sigma}_{\mathbf{d}|\boldsymbol{\theta}}|^{-n/2}}{(2\pi)^{n/2}} \exp\big\{-\frac{1}{2}(\mathbf{d}-\mathbf{W}\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}})^T\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}(\mathbf{d}-\mathbf{W}\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}})\big\} \cdot \prod_{i=1}^{n} \pi(\theta_i|\theta_{i-1}) \\
&\propto \exp\big\{-\frac{1}{2}(\mathbf{d}-\mathbf{W}\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}})^T\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}(\mathbf{d}-\mathbf{W}\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}})\big\} \cdot \prod_{i=1}^{n} \pi(\theta_i|\theta_{i-1}) \\
&= \exp\big\{-\frac{1}{2}(\mathbf{d}^T\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}\mathbf{d} - 2\boldsymbol{\mu}^T_{\mathbf{z}|\boldsymbol{\theta}}\mathbf{W}^T\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}\mathbf{d} + \boldsymbol{\mu}^T_{\mathbf{z}|\boldsymbol{\theta}}\mathbf{W}^T\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}\mathbf{W}\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}}\big\} \cdot \prod_{i=1}^{n} \pi(\theta_i|\theta_{i-1}) \\
&\propto \exp\big\{\mathbf{d}^T\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}\mathbf{W}\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}} + \boldsymbol{\mu}^T_{\mathbf{z}|\boldsymbol{\theta}}\mathbf{Q}\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}}\big\} \cdot \prod_{i=1}^{n} \pi(\theta_i|\theta_{i-1}),
\end{aligned} \tag{70}$$

where we have let $\mathbf{Q} = -\frac{1}{2}\mathbf{W}^T\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}\mathbf{W}$. Further, we let $\mathbf{y}^T = \mathbf{d}^T\boldsymbol{\Sigma}^{-1}_{\mathbf{d}|\boldsymbol{\theta}}\mathbf{W}$ such that we may write

$$\pi(\boldsymbol{\theta}|\mathbf{y}) \propto \exp\big\{\mathbf{y}^T\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}} + \boldsymbol{\mu}^T_{\mathbf{z}|\boldsymbol{\theta}}\mathbf{Q}\boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}}\big\} \cdot \prod_{i=1}^{n} \pi(\theta_i|\theta_{i-1}). \tag{71}$$

The next step is to pull the transition probabilities inside the exponential function, and for an easier read of the distribution we also let $\boldsymbol{\mu} = \boldsymbol{\mu}_{\mathbf{z}|\boldsymbol{\theta}}$. These changes gives us the following form to work with

$$\begin{aligned}
\pi(\boldsymbol{\theta}|\mathbf{y}) &\propto \exp\Big\{\mathbf{y}^T\boldsymbol{\mu} + \boldsymbol{\mu}^T\mathbf{Q}\boldsymbol{\mu} + \sum_{i=1}^{n} \ln\big(\pi(\theta_i|\theta_{i-1})\big)\Big\} \\
&= \exp\Big\{\sum_{i=1}^{n} y_i\mu_i + \boldsymbol{\mu}^T\mathbf{Q}\boldsymbol{\mu} + \sum_{i=1}^{n} \ln\big(\pi(\theta_i|\theta_{i-1})\big)\Big\}.
\end{aligned} \tag{72}$$

To break up the second term of the exponential function in (72) and turn them into sums, we look at the structure of the vector-matrix multiplication. This results in the double sum given by

$$
\begin{aligned}
\boldsymbol{\mu}^T\mathbf{Q}\boldsymbol{\mu} &= \begin{bmatrix}\mu_1 & \mu_2 & \dots & \mu_n\end{bmatrix} \begin{bmatrix} Q_{11} & \cdots & Q_{1j} & \cdots & Q_{1n} \\ \vdots & & \vdots & & \vdots \\ Q_{i1} & \cdots & Q_{ij} & \cdots & Q_{in} \\ \vdots & & \vdots & & \vdots \\ Q_{n1} & \cdots & Q_{nj} & \cdots & Q_{nn} \end{bmatrix} \begin{bmatrix}\mu_1 \\ \mu_2 \\ \vdots \\ \mu_n\end{bmatrix} \\
&= \begin{bmatrix}\mu_1 & \mu_2 & \dots & \mu_n\end{bmatrix} \Bigg(\begin{bmatrix} Q_{11} \\ \vdots \\ Q_{i1} \\ \vdots \\ Q_{n1} \end{bmatrix} \cdot \mu_1 + \begin{bmatrix} Q_{12} \\ \vdots \\ Q_{i2} \\ \vdots \\ Q_{n2} \end{bmatrix} \cdot \mu_2 + \dots + \begin{bmatrix} Q_{1j} \\ \vdots \\ Q_{ij} \\ \vdots \\ Q_{nj} \end{bmatrix} \cdot \mu_j + \dots + \begin{bmatrix} Q_{1n} \\ \vdots \\ Q_{in} \\ \vdots \\ Q_{nn} \end{bmatrix} \cdot \mu_n \Bigg) \\
&= \sum_{j=1}^{n}\sum_{i=1}^{n} Q_{ij}\mu_i\mu_j.
\end{aligned}
\tag{73}
$$

We put this into the distribution resulting in the expression

$$
\pi(\boldsymbol{\theta}|\mathbf{y}) \propto \exp\Big\{ \sum_{i=1}^{n} y_i\mu_i + \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}\mu_i\mu_j + \sum_{i=1}^{n} \ln\big(\pi(\theta_i|\theta_{i-1})\big)\Big\}, \tag{74}
$$

such that the energy function is given by

$$
U(\boldsymbol{\theta}) = \sum_{i=1}^{n} y_i\mu_i + \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}\mu_i\mu_j + \sum_{i=1}^{n} \ln\big(\pi(\theta_i|\theta_{i-1})\big). \tag{75}
$$

B Energy Function for Multiple States

The Markov chain of interest often has more than two classes. We let $\boldsymbol{\theta} \in \{0, 1, 2, 3\}^n$, such that we have four possible states in the Markov chain. The energy function in (14) is based on binary variables, so we need to express the variables in $\boldsymbol{\theta}$ with binary variables. To do this we give each θ_i a corresponding binary pair to represent one of the Markov chain states. We let

$$
\theta_i = \begin{cases} 0, & \text{if } \phi_i = 0 \text{ and } \phi_{n+i} = 0, \\ 1, & \text{if } \phi_i = 1 \text{ and } \phi_{n+i} = 0, \\ 2, & \text{if } \phi_i = 0 \text{ and } \phi_{n+i} = 1, \\ 3, & \text{if } \phi_i = 1 \text{ and } \phi_{n+i} = 1, \end{cases} \tag{76}
$$

such that θ_i has corresponding variables $[\phi_i, \phi_{n+i}] \in \{[0, 0], [1, 0], [0, 1], [1, 1]\}, \forall i$.

We start in the same manner as we did for the binary case by looking at the expectation of θ_i, now expressed using $[\phi_i, \phi_{n+i}]$. We have that

$$\begin{aligned}
\mu_i(\theta_i) =&\mu(0)(1-\phi_i)(1-\phi_{n+i}) + \mu(1)\phi_i(1-\phi_{n+i}) \\
&+ \mu(2)(1-\phi_i)x_{n+i} + \mu(3)\phi_i\phi_{n+i} \\
=&\mu(0)(1-\phi_i-\phi_{n+i}+\phi_i\phi_{n+i}) + \mu(1)(\phi_i-\phi_i\phi_{n+i}) \\
&+ \mu(2)(\phi_{n+i}-\phi_i\phi_{n+i}) + \mu(3)\phi_i\phi_{n+i} \\
=&\mu(0) + (\mu(1)-\mu(0))\phi_i + (\mu(2)-\mu(0))\phi_{n+i} \\
&+ (\mu(0)-\mu(1)-\mu(2)+\mu(3))\phi_i\phi_{n+i} \\
=&K_1 + K_2\phi_i + K_3\phi_{n+i} + K_4\phi_i\phi_{n+i},
\end{aligned} \tag{77}$$

where $K_1 = \mu(0)$, $K_2 = (\mu(1) - \mu(0))$, $K_3 = (\mu(2) - \mu(0))$ and $K_4 = (\mu(0) - \mu(1) - \mu(2) + \mu(3))$ are constants. For the quadratic term of the expectation in the energy function in (38), we get the following expression,

$$\begin{aligned}
\mu_i(\theta_i)\mu_j(\theta_j) =& \left(K_1 + K_2\phi_i + K_3\phi_{n+i} + K_4\phi_i\phi_{n+i}\right)\left(K_1 + K_2\phi_j + K_3\phi_{n+j} + K_4\phi_j\phi_{n+j}\right) \\
=&K_1\left(K_1 + K_2\phi_j + K_3\phi_{n+j} + K_4\phi_j\phi_{n+j}\right) \\
&+ K_2\phi_i\left(K_1 + K_2\phi_j + K_3\phi_{n+j} + K_4\phi_j\phi_{n+j}\right) \\
&+ K_3\phi_{n+i}\left(K_1 + K_2\phi_j + K_3\phi_{n+j} + K_4\phi_j\phi_{n+j}\right) \\
&+ K_4\phi_i\phi_{n+i}\left(K_1 + K_2\phi_j + K_3\phi_{n+j} + K_4\phi_j\phi_{n+j}\right) \\
=&K_1^2 + K_1K_2\left(\phi_i + \phi_j\right) + K_1K_3\left(\phi_{n+i} + \phi_{n+j}\right) + K_1K_4\left(\phi_i\phi_{n+i} + \phi_j\phi_{n+j}\right) \\
&+ K_2^2\phi_i\phi_j + K_2K_3\left(\phi_i\phi_{n+j} + \phi_{n+i}\phi_j\right) + K_2K_4\left(\phi_i\phi_j\phi_{n+j} + \phi_i\phi_{n+i}\phi_j\right) \\
&+ K_3^2\phi_{n+i}\phi_{n+j} + K_3K_4\left(\phi_{n+i}\phi_j\phi_{n+j} + \phi_i\phi_{n+i}\phi_{n+j}\right) \\
&+ K_4^2\phi_i\phi_{n+i}\phi_j\phi_{n+j}.
\end{aligned} \tag{78}$$

We now find an expression for the logarithmic transition probabilities between states. Let us again denote the logarithm of the transition probability from state θ_{i-1} to state θ_i by $t_{\theta_{i-1},\theta_i} = \ln\left(\pi(\theta_i|\theta_{i-1})\right)$. We first have to address the boundary case of $i = 1$. The logarithm of the limiting probability of state θ_i is denoted as $t_i = \ln(\pi(\theta_i))$, which yields the expression

$$\begin{aligned}
\ln(\pi(\theta_1)) =& \ln\left(\pi(0)\right)\left(1-\phi_1\right)\left(1-\phi_{n+1}\right) + \ln\left(\pi(1)\right)\phi_1\left(1-\phi_{n+1}\right) \\
&+ \ln\left(\pi(2)\right)\left(1-\phi_1\right)\phi_{n+1} + \ln\left(\pi(3)\right)\phi_1\phi_{n+1} \\
=&t_0\left(1-\phi_1-\phi_{n+1}+\phi_1\phi_{n+1}\right) + t_1\left(\phi_1-\phi_1\phi_{n+1}\right) \\
&+ t_2\left(\phi_{n+1}-\phi_1\phi_{n+1}\right) + t_3\phi_1\phi_{n+1} \\
=&t_0 + \left(t_1-t_0\right)\phi_1 + \left(t_2-t_0\right)\phi_{n+1} \\
&+ \left(t_0-t_1-t_2+t_3\right)\phi_1\phi_{n+1}.
\end{aligned} \tag{79}$$

This has the same build up as the expectation, however, this is not the case for the general expression of the logarithm of the transition probability from state

θ_{i-1} to θ_i, for $i = 2, 3, \ldots, n$. For the transition between these states we have that

$$\begin{aligned}
\ln(\pi(\theta_i|\theta_{i-1})) = &\ln(\pi(0|0))(1-\phi_{i-1})(1-\phi_{n+i-1})(1-\phi_i)(1-\phi_{n+i}) \\
&+ \ln(\pi(1|0))(1-\phi_{i-1})(1-\phi_{n+i-1})\phi_i(1-\phi_{n+i}) \\
&+ \ln(\pi(2|0))(1-\phi_{i-1})(1-\phi_{n+i-1})(1-\phi_i)\phi_{n+i} \\
&+ \ln(\pi(3|0))(1-\phi_{i-1})(1-\phi_{n+i-1})\phi_i\phi_{n+i} \\
&+ \ln(\pi(0|1))\phi_{i-1}(1-\phi_{n+i-1})(1-\phi_i)(1-\phi_{n+i}) \\
&+ \ln(\pi(1|1))\phi_{i-1}(1-\phi_{n+i-1})\phi_i(1-\phi_{n+i}) \\
&+ \ln(\pi(2|1))\phi_{i-1}(1-\phi_{n+i-1})(1-\phi_i)\phi_{n+i} \\
&+ \ln(\pi(3|1))\phi_{i-1}(1-\phi_{n+i-1})\phi_i\phi_{n+i} \\
&+ \ln(\pi(0|2))(1-\phi_{i-1})\phi_{n+i-1}(1-\phi_i)(1-\phi_{n+i}) \\
&+ \ln(\pi(1|2))(1-\phi_{i-1})\phi_{n+i-1}\phi_i(1-\phi_{n+i}) \\
&+ \ln(\pi(2|2))(1-\phi_{i-1})\phi_{n+i-1}(1-\phi_i)\phi_{n+i} \\
&+ \ln(\pi(3|2))(1-\phi_{i-1})\phi_{n+i-1}\phi_i\phi_{n+i} \\
&+ \ln(\pi(0|3))\phi_{i-1}\phi_{n+i-1}(1-\phi_i)(1-\phi_{n+i}) \\
&+ \ln(\pi(1|3))\phi_{i-1}\phi_{n+i-1}\phi_i(1-\phi_{n+i}) \\
&+ \ln(\pi(2|3))\phi_{i-1}\phi_{n+i-1}(1-\phi_i)\phi_{n+i} \\
&+ \ln(\pi(3|3))\phi_{i-1}\phi_{n+i-1}\phi_i\phi_{n+i}.
\end{aligned} \tag{80}$$

We replace the constants with $t_{\theta_{i-1},\theta_i} = \ln(\pi(\theta_i|\theta_{i-1}))$, and multiply out all the parenthesis products. This becomes

$$\begin{aligned}
\ln(\pi(\theta_i|\theta_{i-1})) = &t_{00}(1-\phi_{i-1}-\phi_{n+i-1}+\phi_{i-1}\phi_{n+i-1})(1-\phi_i-\phi_{n+i}+\phi_i\phi_{n+i}) \\
&+ t_{01}(1-\phi_{i-1}-\phi_{n+i-1}+\phi_{i-1}\phi_{n+i-1})\phi_i(1-\phi_{n+i}) \\
&+ t_{02}(1-\phi_{i-1}-\phi_{n+i-1}+\phi_{i-1}\phi_{n+i-1})(1-\phi_i)\phi_{n+i} \\
&+ t_{03}(1-\phi_{i-1}-\phi_{n+i-1}+\phi_{i-1}\phi_{n+i-1})\phi_i\phi_{n+i} \\
&+ t_{10}\phi_{i-1}(1-\phi_{n+i-1})(1-\phi_i-\phi_{n+i}+\phi_i\phi_{n+i}) \\
&+ t_{11}\phi_{i-1}(1-\phi_{n+i-1})\phi_i(1-\phi_{n+i}) \\
&+ t_{12}\phi_{i-1}(1-\phi_{n+i-1})(1-\phi_i)\phi_{n+i} \\
&+ t_{13}\phi_{i-1}(1-\phi_{n+i-1})\phi_i\phi_{n+i} \\
&+ t_{20}(1-\phi_{i-1})\phi_{n+i-1}(1-\phi_i-\phi_{n+i}+\phi_i\phi_{n+i}) \\
&+ t_{21}(1-\phi_{i-1})\phi_{n+i-1}\phi_i(1-\phi_{n+i}) \\
&+ t_{22}(1-\phi_{i-1})\phi_{n+i-1}(1-\phi_i)\phi_{n+i} \\
&+ t_{23}(1-\phi_{i-1})\phi_{n+i-1}\phi_i\phi_{n+i} \\
&+ t_{30}\phi_{i-1}\phi_{n+i-1}(1-\phi_i-\phi_{n+i}+\phi_i\phi_{n+i}) \\
&+ t_{31}\phi_{i-1}\phi_{n+i-1}\phi_i(1-\phi_{n+i}) \\
&+ t_{32}\phi_{i-1}\phi_{n+i-1}(1-\phi_i)\phi_{n+i} \\
&+ t_{33}\phi_{i-1}\phi_{n+i-1}\phi_i\phi_{n+i}.
\end{aligned} \tag{81}$$

Finally all constants are collected for combinations of linear, quadratic, cubic and quartic terms of ϕ_{i-1}, ϕ_{n+i-1}, ϕ_i and ϕ_{n+i}. These constants are then renamed for the quadratic, cubic and quartic constants to simplify the equation. The logarithm to the transition probability from state θ_{i-1} to state θ_i becomes

$$
\begin{aligned}
\ln(\pi(\theta_i|\theta_{i-1})) =& t_{00} + (t_{10} - t_{00})\,\phi_{i-1} + (t_{20} - t_{00})\,\phi_{n+i-1} \\
&+ (t_{01} - t_{00})\,\phi_i + (t_{02} - t_{00})\,\phi_{n+i} \\
&+ (t_{00} - t_{01} - t_{10} + t_{11})\,\phi_{i-1}\phi_i \\
&+ (t_{00} - t_{10} - t_{20} + t_{30})\,\phi_{i-1}\phi_{n+i-1} \\
&+ (t_{00} - t_{02} - t_{10} + t_{12})\,\phi_{i-1}\phi_{n+i} \\
&+ (t_{00} - t_{01} - t_{20} + t_{21})\,\phi_i\phi_{n+i-1} \\
&+ (t_{00} - t_{01} - t_{02} + t_{03})\,\phi_i\phi_{n+i} \\
&+ (t_{00} - t_{02} - t_{20} + t_{22})\,\phi_{n+i-1}\phi_{n+i} \\
&+ (t_{01} + t_{10} - t_{11} + t_{20} - t_{21} - t_{30} + t_{31} - t_{00})\,\phi_{i-1}\phi_i\phi_{n+i-1} \\
&+ (t_{01} + t_{02} - t_{03} + t_{10} - t_{11} - t_{12} + t_{13} - t_{00})\,\phi_{i-1}\phi_i\phi_{n+i} \\
&+ (t_{02} + t_{10} - t_{12} + t_{20} - t_{22} - t_{30} + t_{32} - t_{00})\,\phi_{i-1}\phi_{n+i-1}\phi_{n+i} \\
&+ (t_{01} + t_{02} - t_{03} + t_{20} - t_{21} - t_{22} + t_{23} - t_{00})\,\phi_i\phi_{n+i-1}\phi_{n+i} \\
&+ (t_{00} - t_{01} - t_{02} + t_{03} - t_{10} + t_{11} + t_{12} - t_{13} - t_{20} + t_{21} \\
&+ t_{22} - t_{23} + t_{30} - t_{31} - t_{32} + t_{33})\phi_i\phi_{i-1}\phi_{n+i-1}\phi_{n+i} \\
=& t_{00} + (t_{10} - t_{00})\,\phi_{i-1} + (t_{20} - t_{00})\,\phi_{n+i-1} \\
&+ (p_{01} - p_{00})\,\phi_i + (p_{02} - p_{00})\,\phi_{n+i} \\
&+ G_1\phi_{i-1}\phi_i + G_2\phi_{i-1}\phi_{n+i-1} + G_3\phi_{i-1}\phi_{n+i} \\
&+ G_4\phi_i\phi_{n+i-1} + G_5\phi_i\phi_{n+i} + G_6\phi_{n+i-1}\phi_{n+i} \\
&+ H_1\phi_{i-1}\phi_i\phi_{n+i-1} + H_2\phi_{i-1}\phi_i\phi_{n+i} \\
&+ H_3\phi_{i-1}\phi_{n+i-1}\phi_{n+i} + H_4\phi_i\phi_{n+i-1}\phi_{n+i} \\
&+ J_1\phi_i\phi_{i-1}\phi_{n+i-1}\phi_{n+i},
\end{aligned}
\tag{82}
$$

which is where the constants $G_1 - G_6$, $H_1 - H_4$ and J_1 in (51) comes from. The expressions in (77), (78), (79) and (82) are finally put into the energy function in (38), resulting in the final form of the energy equation in (53).

C Simulation Plots

C.1 Two State Markov Chain - using MH

In this section the simulation plots for the two state Markov chain cases are found. The plots show the simulations for 5 and for 10 neighbors in each case. Cases are simulated with the use of the MH algorithm.

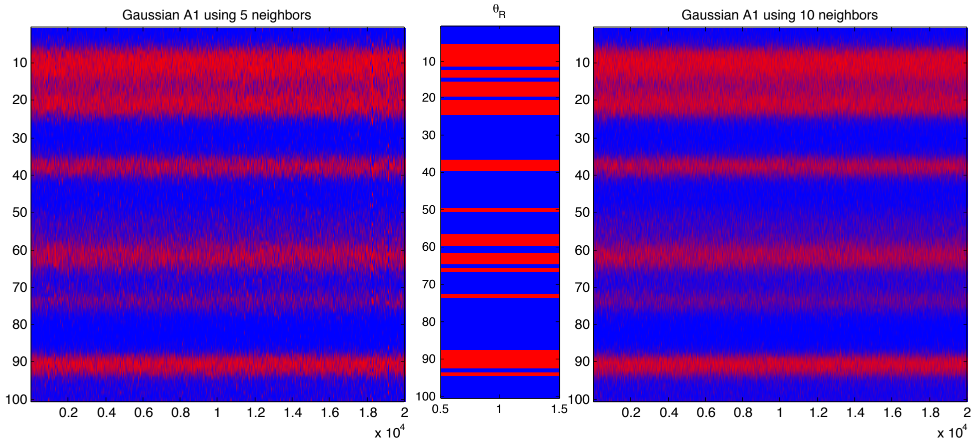


Figure 30: *Simulation results for Gaussian case A1 using* 5 *and* 10 *neighbors, where the acceptance rates are* 54.10% *and* 78.19%*, respectively.*

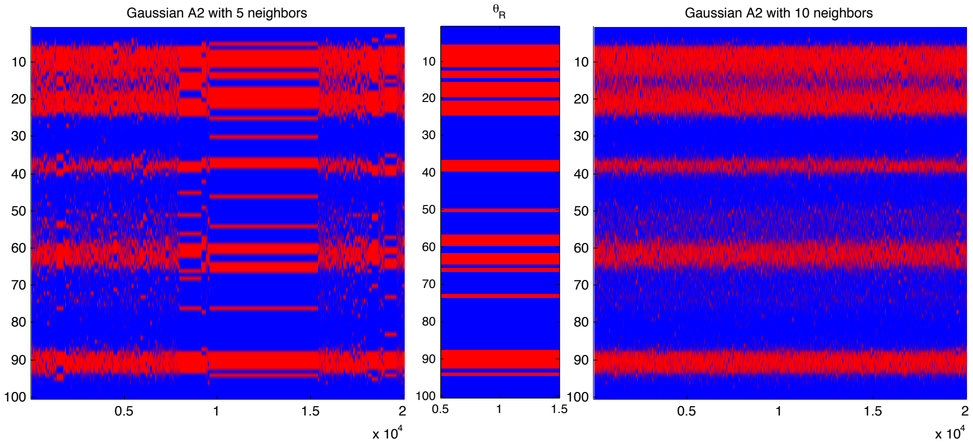


Figure 31: *Simulation results for Gaussian case A2 using* 5 *and* 10 *neighbors, where the acceptance rates are* 8.03% *and* 33.49%*, respectively.*

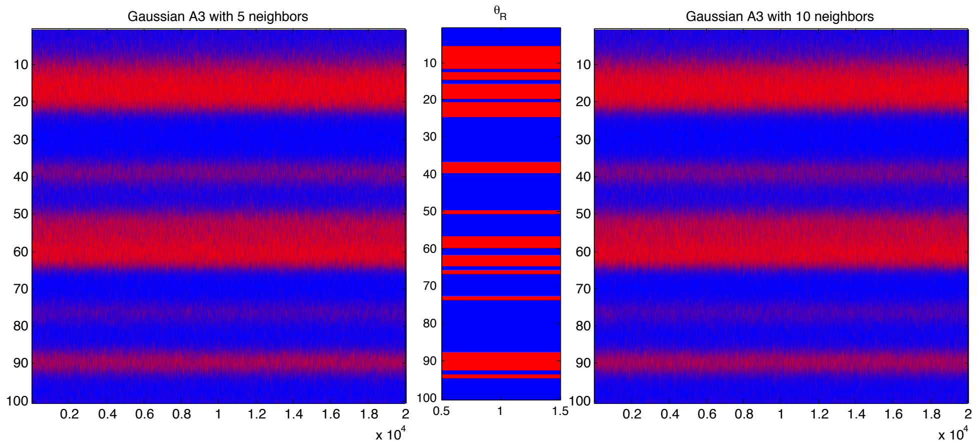


Figure 32: *Simulation results for Gaussian case A3 using* 5 *and* 10 *neighbors, where the acceptance rates are* 89.93% *and* 94.95%, *respectively.*

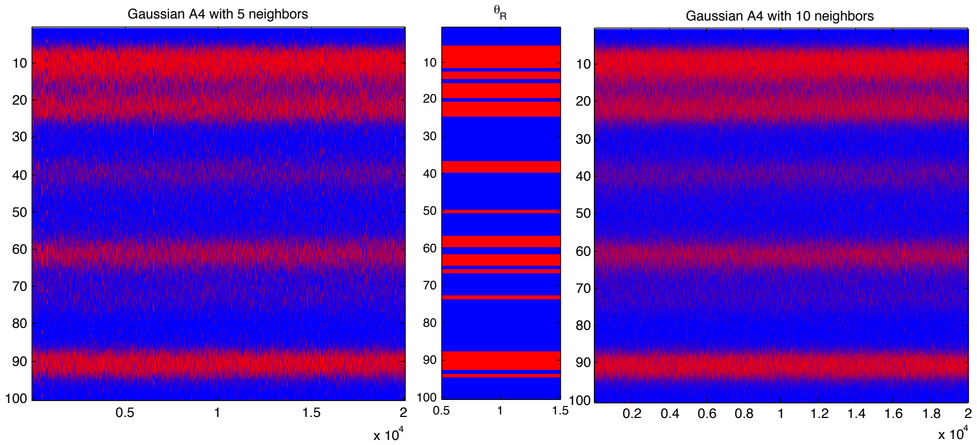


Figure 33: *Simulation results for Gaussian case A4 using* 5 *and* 10 *neighbors, where the acceptance rates are* 56.89% *and* 78.61%, *respectively.*

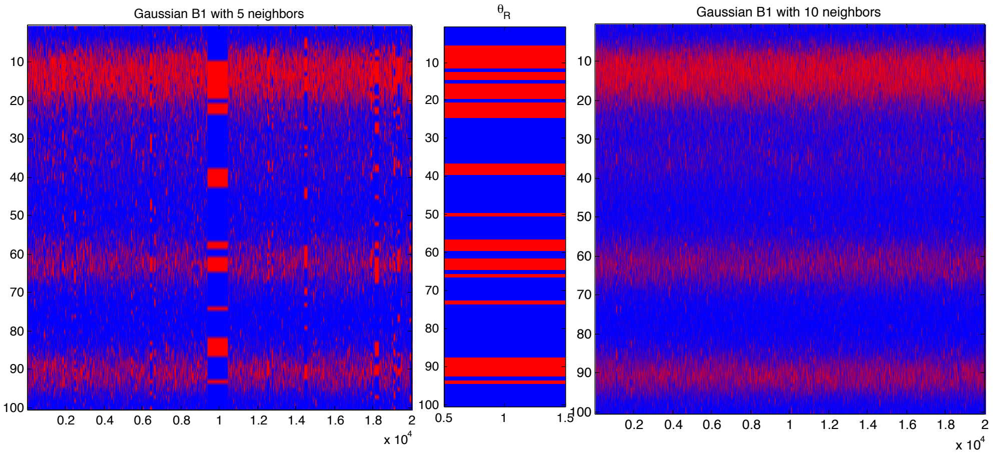


Figure 34: *Simulation results for Gaussian case* $B1$ *using* 5 *and* 10 *neighbors, where the acceptance rates are* 29.20% *and* 55.91%, *respectively.*

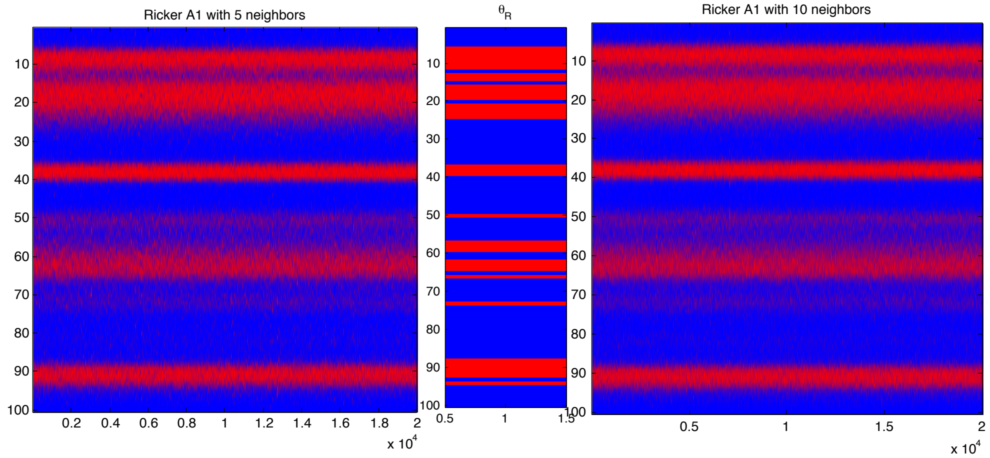


Figure 35: *Simulation results for Ricker case* $A1$ *using* 5 *and* 10 *neighbors, where the acceptance rates are* 65.58% *and* 77.35%, *respectively.*

C.2 Two States - without MH

Here the simulations without the use of the MH algorithm are supplemented. The plots in this section are for Gaussian case $A1$ and $A2$ using 5 neighbors, and for Gaussian case $A3$ using 10 neighbors.

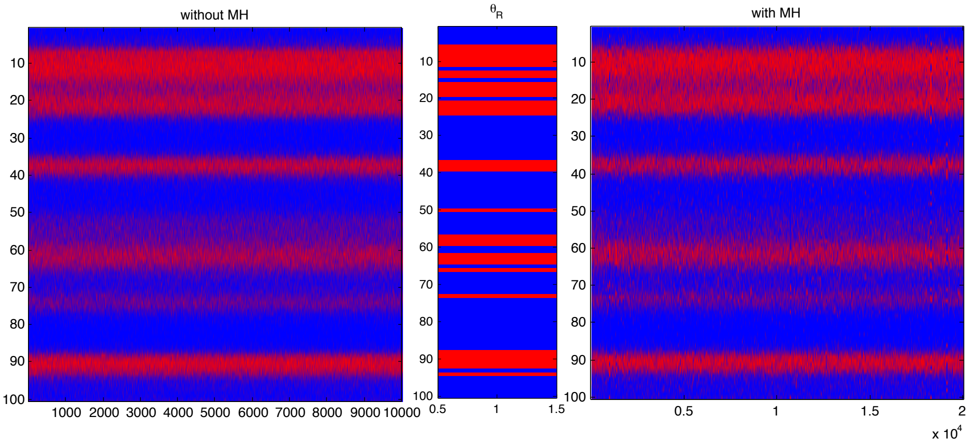


Figure 36: *Gaussian case $A1$ using 5 neighbors, with and without the use of the MH algorithm. Acceptance when using the MH algorithm is* 54.10%.

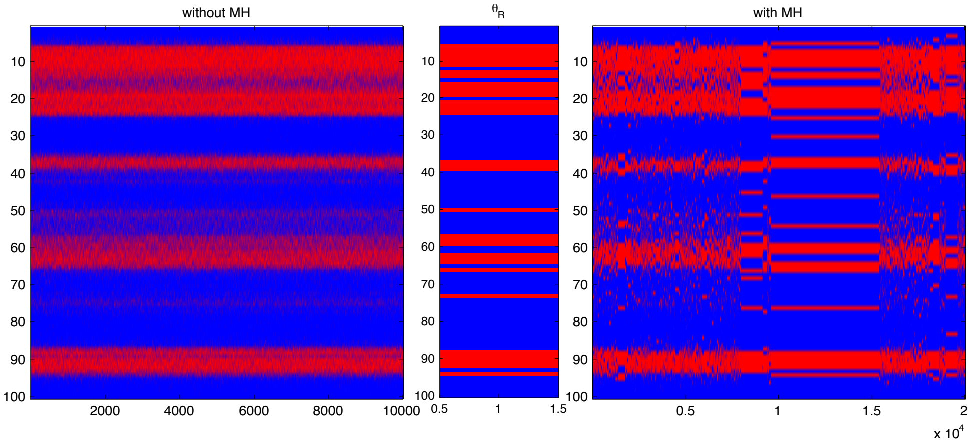


Figure 37: *Gaussian case $A2$ using 5 neighbors, with and without the use of the MH algorithm. Acceptance rate when using the MH algorithm is* 8.03%.

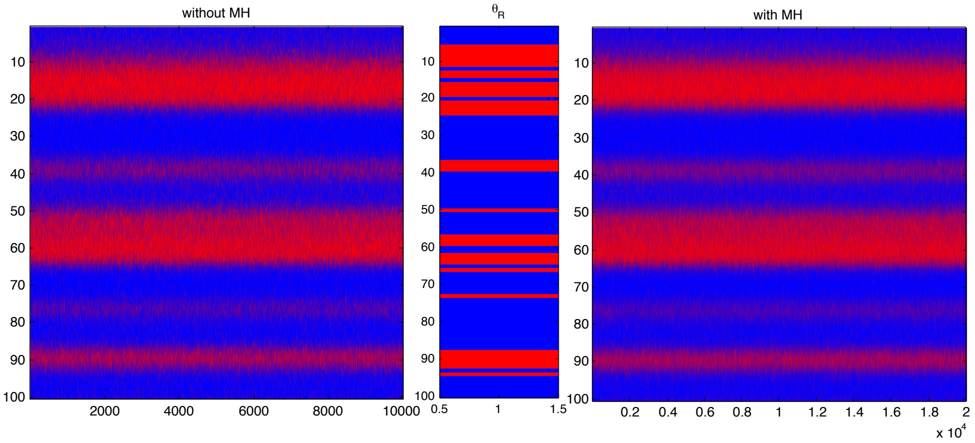


Figure 38: *Gaussian case A3 using* 10 *neighbors, with and without the use of the MH algorithm. Acceptance rate when using the MH algorithm is* 94.95%.

C.3 Four State Markov Chain - using MH

We here present the simulations for the four state Markov chain prior, which is done in the form of plots. The simulations are generated with use of the MH algorithm.

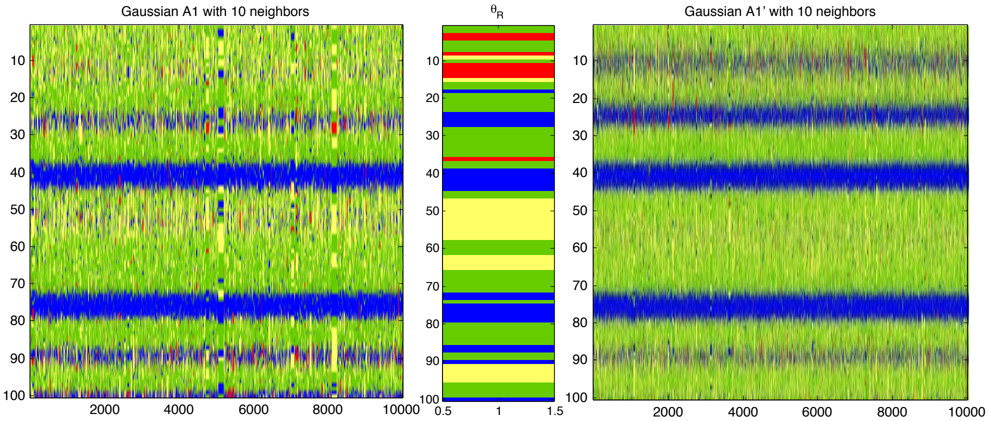


Figure 39: *Simulation results for Gaussian case A*1 *and case A*1′*, with acceptance rates* 29.06% *and* 55.05%*, respectively.*

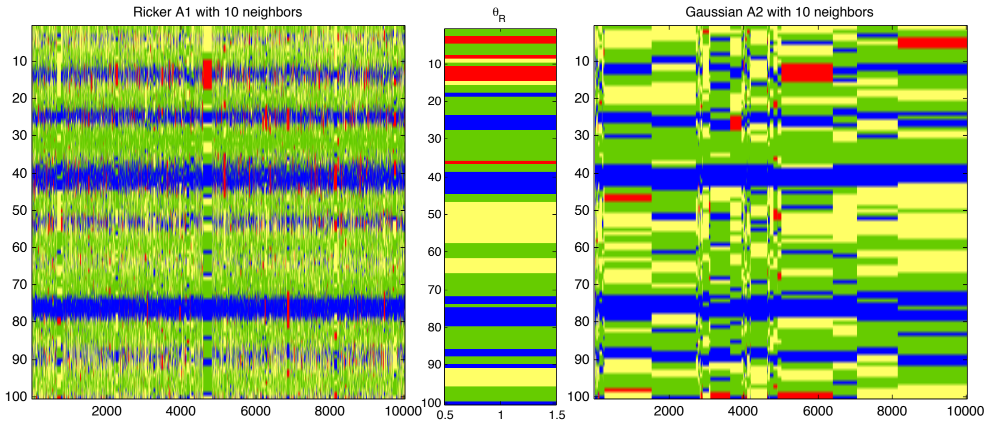


Figure 40: *Simulation results for Ricker case A1 and Gaussian case A2, with acceptance rates* 29.97% *and* 0.53%*, respectively.*

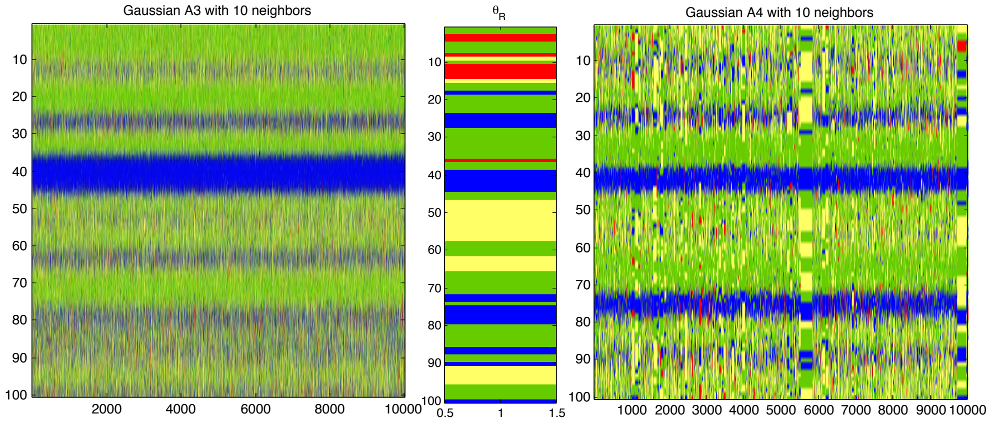


Figure 41: *Simulation results for Gaussian cases A3 and A4, with acceptance rates* 78.72% *and* 27.27%*, respectively.*

C.4 Four State Markov Chain - without MH

In this section simulations without the use of the MH algorithm are supplemented for the four state Markov chain prior. The plots in this section are for Gaussian case $A1$ and $A3$ using 10 neighbors, and for Ricker case $A1$ using 10 neighbors. For each case the simulations where the MH algorithm has been used are also presented for comparison.

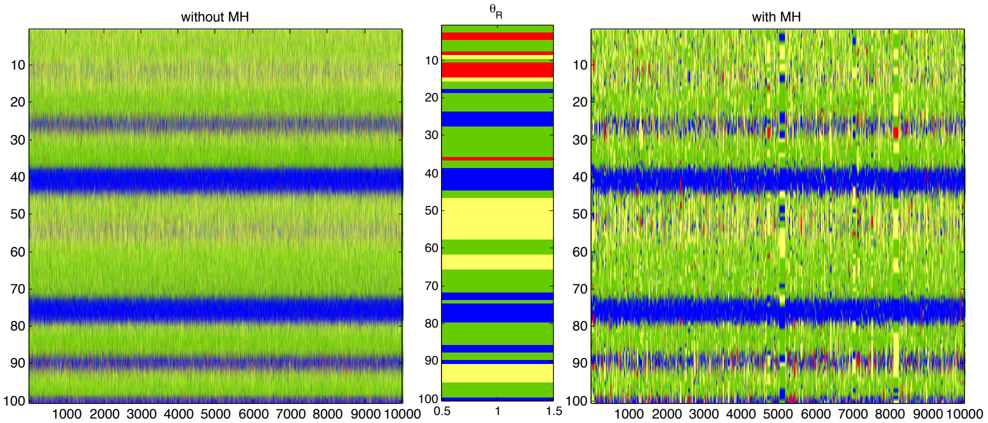


Figure 42: *Gaussian case* $A1$ *using* 10 *neighbors, without and with the use of the MH algorithm. Acceptance when using the MH algorithm is* 29.06%.

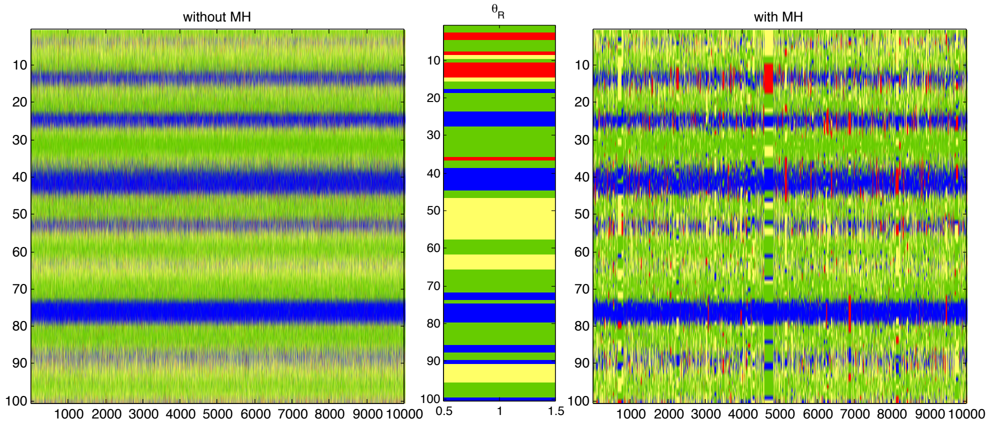


Figure 43: *Ricker case A*1 *using* 10 *neighbors, without and with the use of the MH algorithm. Acceptance when using the MH algorithm is* 29.97%.

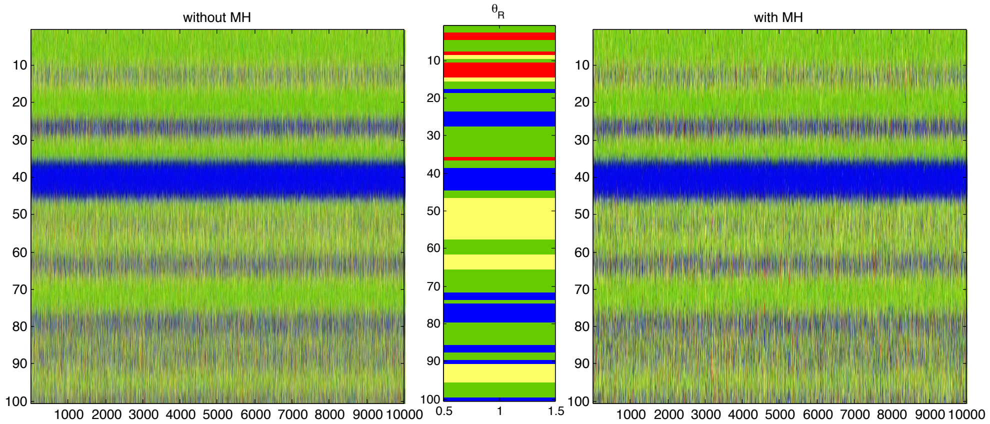


Figure 44: *Gaussian case A*3 *using* 10 *neighbors, without and with the use of the MH algorithm. Acceptance when using the MH algorithm is* 78.72%.