Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Mathematical Sciences

Karen Stormark Auestad

Seismic AVO Inversion with Fast Approximate Markov Chain Monte Carlo with Application to the Alvheim Field

Master's thesis in Applied Physics and Mathematics
Supervisor: Jo Eidsvik
Co-supervisor: Mina Spremić and The Tien Mai
January 2024

NTNU
Norwegian University of
Science and Technology

Karen Stormark Auestad

Seismic AVO Inversion with Fast Approximate Markov Chain Monte Carlo with Application to the Alvheim Field

Master's thesis in Applied Physics and Mathematics
Supervisor: Jo Eidsvik
Co-supervisor: Mina Spremić and The Tien Mai
January 2024

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Preface

This master's thesis was written for the course "TMA4900 - Industrial Mathematics, Master's Thesis", as part of the Master of Science degree in Applied Physics and Mathematics (MTFYMA) at the Norwegian University of Science and Technology (NTNU).

The thesis concerns Bayesian inversion with Markov chain Monte Carlo. The thesis also considers two supervised learning methods which are used to simplify parts of the Bayesian inversion model, and therefore reduce the computation time of the Markov chain Monte Carlo calculations. The fast and approximate MCMC is tested for seismic amplitude-variation-with-offset inversion at the Alvheim field. It is assumed that the reader has fundamental knowledge of statistics and that the reader has a basic idea of the concept of statistical learning. The inverse problem in question will be explained, however, some acquaintance with spatial statistics is recommended.

I would like to thank my supervisor Professor Jo Eidsvik and co-supervisors PhD Candidate Mina Spremić and Postdoctoral Fellow The Tien Mai for supervising me while writing this thesis.

Abstract

Inverse problems can be solved in a Bayesian framework by combining a priori knowledge about the variables of interest with a likelihood model for the observed data, to form a posterior distribution for the variables of interest. Markov chain Monte Carlo (MCMC) can be used to draw samples from the posterior distribution and is a popular approach when faced with high-dimensional inverse problems. However, a drawback of the approach is that it can be slow. Time-consuming calculation of non-linear forward models in high dimensions is often the reason for this. Substituting the forward model with an approximation to the forward model is a strategy for reducing the computation time of the algorithm. In this thesis, the complicated and non-linear forward model in the likelihood model for seismic amplitude-variation-with-offset (AVO) data is approximated by a multivariate adaptive regression spline (MARS). The MARS model is on average 32 times faster than the exact forward model. This reduction in computation time enables fast approximate MCMC for seismic AVO inversion at the Alvheim field. A comparison between MCMC samples with the exact forward model and MCMC samples with the MARS model on a small area in the Alvheim field shows great similarities. Using the MARS model also enables extensive comparison between four MCMC algorithms in a small area. The most efficient algorithm is used for approximate MCMC over the total area of the Alvheim field. The approximate MCMC samples from the Alvheim field contain more oil and clay compared to the results in Spremić et al. (2024), which used a localized ensemble-based approach to approximate the posterior in the Alvheim field.

Sammendrag

Inverse problemer kan løses i et Bayesiansk rammeverk. Da kombineres a priori kunnskap om modellparametre med en rimelighetsmodell for observerte data, for å konstruere en posterior-fordeling for modellparameterne. Markov chain Monte Carlo (MCMC) kan brukes til å trekke realisasjoner fra posterior-fordelingen, og den er en populær metode for å håndtere høydimensjonale inverse problemer. En ulempe med metoden er imidlertid at den kan være treg. Tidkrevende beregning av ikke-lineære forovermodeller i høye dimensjoner er ofte årsaken til dette. For å redusere kjøretiden til algoritmen kan forovermodellen erstattes av en enklere tilnærming. I denne oppgaven tilnærmes den kompliserte og ikke-lineære forovermodellen i rimelighetsmodellen for seismisk amplitude-variasjon-med-offset (AVO) data med en multivariat adaptiv regresjonsspline (MARS). MARS-modellen er i gjennomsnitt 32 ganger raskere enn den eksakte forovermodellen. Denne reduksjonen i beregningstiden muliggjør rask approksimativ MCMC for seismisk AVO-inversjon ved Alvheim-feltet. Sammenligning av resultater fra MCMC med den eksakte forovermodellen og med MARS-modellen på et lite område av Alvheimfeltet, viser store likheter. Bruk av MARS-modellen muliggjør også en omfattende sammenligning av fire MCMC-algoritmer på et lite område. Den mest effektive algoritmen brukes til å trekke approksimerte MCMC realisasjoner fra hele Alvheim-feltet. De approksimerte MCMC-resultatene fra Alvheim-feltet inneholder mer olje og leire sammenlignet med resultatene fra Spremić et al. (2024), som brukte et ensemblebasert Kalman-filtrer for å tilnærme a posterior-fordelingen.

Table of Contents

1

Introduction

A frequent scientific problem is that of estimating unobservable quantities of interest from observed data. There is often a known physical or mathematical relation between the quantities of interest and the data, known as the forward model. However, the inverse problem of going from data to the quantity of interest is much more difficult. An inverse problem can be solved in the Bayesian framework by combining a priori knowledge about the quantities of interest with a likelihood model for the observed data to form a posterior distribution for the quantities of interest (Mosegaard & Tarantola, 1995). Bayesian inversion is suitable for geophysical inverse problems, because these problems often require a priori knowledge of the earth parameters, as the information from the measurements can be sparse. A priori knowledge is represented by a prior distribution, which enables us to account for the trends and uncertainties in the prior beliefs (Malinverno & Briggs, 2004). A likelihood model, built on geophysical principles, expresses which earth parameter values are compatible with the observed data. A real-life example of an inverse problem is that of estimating reservoir variables from seismic amplitude-variation-with-offset (AVO) data in the Alvheim field, located in the North Sea off the Norwegian west coast.

As pointed out by van Ravenzwaaij et al. (2018), posterior densities in Bayesian inversion are often challenging to examine analytically. In low dimensions, posterior distributions can be investigated by evaluating the posterior systematically throughout the parameter space. However, such an exhaustive search of the parameter space becomes infeasible in high dimensions (Mosegaard & Tarantola, 1995). For high-dimensional inverse problems, Markov chain Monte Carlo (MCMC) is a popular way of estimating posterior densities in Bayesian inversion (Robert & Casella, 2011). MCMC draws dependent samples from the posterior distribution by constructing a Markov chain that converges to the posterior distribution. These samples are used to estimate population means with Monte Carlo integration (Givens & Hoeting, 2013). MCMC was first introduced in the early 1950s. As computers advanced in the 1990s, the algorithm became increasingly popular. With MCMC, it was possible to analyse models that were too complex for previous methods to handle satisfactorily. However, MCMC is as of now too slow for many high-dimensional spatial problems. The increasing demand for solving inverse problems with slow calculations of the forward model has shown that straightforward use of MCMC algorithms is unfeasible (Khoshkholgh et al., 2021).

To improve the computational efficiency of MCMC, time-consuming forward model calculations can be replaced with faster surrogate model calculations. A surrogate model aims to approximate a complicated model with fewer parameters (Chen et al., 2014). In Holm-Jensen and Hansen (2019), a linear approximation to the forward model was constructed using ridge regression on data sampled from the prior, to speed up a linear waveform tomography inversion. A surrogate system based on sparse grid interpolation

was used by Zeng et al. (2012) to accelerate forward calculations in a contaminant source identification problem. Chen et al. (2013) compared using several polynomial regression models, a Gaussian process regression and a multivariate adaptive regression spline (MARS) as surrogates in the non-linear process of hydraulic fracturing. Among the surrogates, MARS showed the best predicting performance. MARS was also used by Chen et al. (2014) as a surrogate for the time-consuming forward function in a hydrothermal model.

In this thesis, a surrogate model for the complicated and non-linear forward model in the likelihood model for the AVO data from the Alvheim field is constructed. Inspired by Chen et al. (2013) and Chen et al. (2014), a MARS model is trained on data sampled from the prior distribution. The main result of this thesis is that evaluating earth parameters using the MARS approximation to the forward model is about 32 times faster than using the exact forward model. The MARS model is compared to a more complex MARS model and two non-parametric kernel regression (NPKR) models. The models are compared in terms of correlation, means square error (MSE) and computation time. The gradient of the MARS and NPKR models is used to approximate the gradient of the forward model. The approximations are compared to a MARS model trained directly to the partial derivatives of the forward model.

The MARS model is used to sample approximate posterior samples with MCMC at the Alvheim field. The samples are compared to posterior samples obtained using the exact forward model on a small area in the Alvheim field. Using the MARS model, the efficiency of four MCMC algorithms is compared on a small area in the Alvheim field. The four methods are the Metropolis-Hastings (MH) algorithm with a standard random walk as proposal distribution, the MH algorithm with a random walk with covariance as proposal distribution, the MH algorithm with the preconditioned Crank Nicolson (pCN) defined in (Pinski et al., 2015) as proposal distribution, and the Metropolis-adjusted Langevin algorithm (MALA). MALA uses the approximation to the gradient of the forward model to reduce the computation time. The most efficient algorithm among the four is used to sample approximate posterior samples over the entire Alvheim field. The results are compared to the results of Spremić et al. (2024), who used a localized ensemble-based approach for approximate and efficient seismic AVO inversion at the Alvheim field. The method of Spremić et al. (2024) gives approximate posterior samples, whereas MCMC in principle can draw exact samples from the posterior. However considering that the forward model is approximated, the MCMC samples are from an approximate posterior distribution. To partially adjust for the error caused by approximating the forward model, additional variance and covariance are added to the likelihood model.

Monte Carlo approximation of the MCMC samples provides mean saturations of oil and gas from the Bayesian posterior model at the Alvheim field. In Norway, the oil fields currently have an average oil recovery rate of about 47 %. To maximize the use of established infrastructure, oil companies seek to increase the oil recovery near existing wells. Moreover, smaller deposits are only profitable if existing infrastructure is used[1]. AVO inversion could be used to increase oil recovery. Oil and gas extraction is debatable. On one hand, the climate crisis requires fossil fuel energy to be phased out and replaced by sustainable energy resources. Goal 7 of the United Nations 17 Sustainable Development Goals is to "ensure access to affordable, reliable, sustainable and modern energy for all"[2]. Fossil fuel energy is neither sustainable nor modern. However, in 2021, 61.5% of the world's electricity production came from fossil energy sources (IEA, 2022). An argument for extracting oil and gas is therefore that sustainable and renewable energy sources do not supply enough or affordable energy for everyone yet.

[1] https://www.norskpetroleum.no/en/developments-and-operations/resource-management-in-mature-areas/. Accessed 16.01.2024.

[2] https://sdgs.un.org/goals/goal7 Accessed 25.01.2024

The structure of this thesis is as follows: In Chapter 2, Bayes inversion and MCMC are presented. A simple bivariate example is used to demonstrate Bayesian inversion and basic MCMC concepts. Section 2.2 is devoted to giving the reader an understanding of how the MCMC algorithm MH works. Different types of proposal distributions used by the MH algorithm are introduced. Several techniques for analysing MCMC results are also presented. In Chapter 3, details of AVO inversion at the Alvheim field are introduced. Prior and likelihood models are presented and necessary notation is given. Chapter 4 discusses approximating the forward function in the likelihood model for the AVO data. In particular, MARS and NPKR models are considered for this task. In Chapter 5, MCMC results from the Alvheim field are presented and compared to the results obtained by Spremić et al. (2024). The results are also discussed in Chapter 5. Chapter 6 contains some closing remarks.

The experiments in this thesis were performed using R 4.3.2 on x86 Ubuntu 20.04 on an Intel Xeon E5-2690 v4 2.6 GHz CPU. The function `earth` in the package `earth` (https://cran.r-project.org/web/packages/earth/earth.pdf) and the function `npreg` in the package `np` (https://cran.r-project.org/web/packages/np/np.pdf) were used to create the two supervised learning models MARS and NPKR, presented in Chapter 4.

Matrices, vectors and vector-evaluated functions are written in bold. Capital letters are used for matrices.

2

Bayesian inversion

This chapter concerns Bayesian inversion. Components of Bayesian inversion such as priors, likelihoods and posteriors are introduced. A simple two-dimensional example is used to illustrate Bayesian inversion. In Section 2.2, the MH algorithm is presented. Also introduced in this section are the proposal distributions random walk, preconditioned Crank-Nicolson (pCN) and the proposal of the Metropolis adjusted Langevin algorithm (MALA). The two-dimensional example is revisited to illustrate how to tune the proposal distributions. In the last part of this chapter, the example is used to demonstrate how to analyse the convergence and mixing of MCMC results. Monte Carlo approximation is presented briefly at the end of this chapter.

2.1 The Bayesian inversion model

The inverse problem arises when it is possible to measure data, denoted $\boldsymbol{y}$, that are related to an unobservable variable of interest, denoted by $\boldsymbol{x}$. A relationship between the variable of interest and the observed data is expressed as

$$\boldsymbol{y} = h(\boldsymbol{x}) + \boldsymbol{\varepsilon}, \tag{2.1}$$

where h represents a known functional relationship and $\boldsymbol{\varepsilon}$ is random noise. When the objective $\boldsymbol{x} = h^{-1}(\boldsymbol{y})$ is non-unique or difficult to find, Bayesian inversion can be employed to do inversion (Ulrych et al., 2001).

In Bayesian inversion, a likelihood model for the observed data and a priori knowledge about the variable of interest, are combined to form a posterior distribution $f(\boldsymbol{x}|\boldsymbol{y})$. A priori knowledge is expressed through a prior distribution, denoted $f(\boldsymbol{x})$. The likelihood probability density function, $f(\boldsymbol{y}|\boldsymbol{x})$, expresses how likely the observed data are given a value of $\boldsymbol{x}$. The posterior distribution function is given by Bayes' theorem:

$$f(\boldsymbol{x}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{x})f(\boldsymbol{x})}{f(\boldsymbol{y})} \propto f(\boldsymbol{y}|\boldsymbol{x})f(\boldsymbol{x}), \tag{2.2}$$

(Robert & Casella, 2004). The denominator, $f(\boldsymbol{y})$, in equation (2.2) is a normalising constant independent of the desired variable $\boldsymbol{x}$ and will therefore be neglected. The posterior $f(\boldsymbol{x}|\boldsymbol{y})$ will be denoted by $\pi(\boldsymbol{x})$.

A priori knowledge used to create the prior can be subjective knowledge about the variable $\boldsymbol{x}$. This can be beneficial as you can incorporate domain or expert knowledge into the model. Bayesian inversion is reasonable for solving an inverse geophysical problem where a priori information is necessary as the information from the measurements alone is insufficient (Malinverno & Briggs, 2004). On the other hand, one might argue that however reasonable a priori information is, the posterior model is nevertheless built

on beliefs and not objective data.

Selecting a prior is an important and often difficult part of Bayesian inversion. However, the focus of this thesis is not to find a prior, but on using Bayesian inversion and MCMC to draw samples from a posterior distribution $\pi(\boldsymbol{x})$. For the real-world example in this thesis, the same prior as in Spremić et al. (2024) is used.

A two-dimensional example is now introduced to illustrate Bayesian inversion (see Figure 2.1). A similar example is found in Auestad (2023). Suppose a wave is sent from a source to a receiver through two layers of different substances (different rocks, liquids, etc.). The objective is to find the slowness in the first and second layers, denoted by x_1 and x_2 respectively. Slowness is the inverse of the speed. Hence $x_1 = v_1^{-1}$ and $x_2 = v_2^{-1}$, where v_1 and v_2 are the speed of the wave through the first and second layers respectively. Let d_1 and d_2 be the distances along the ray of the wave in the first and second layers respectively. The total travel time is the sum of the travel times in the two layers, t_1 and t_2. Using Pythagoras' theorem and the equations of motion, the total travel time the ray uses to travel from the source located at (w_s, z_s) to the receiver at (w_r, z_r) is

$$t = t_1 + t_2 = \frac{d_1}{v_1} + \frac{d_2}{v_2} = x_1 d_1 + x_2 d_2 = x_1\sqrt{(w_s - w_c)^2 + (z_s - z_c)^2} + x_2\sqrt{(w_c - w_r)^2 + (z_s - z_c)^2}, \quad (2.3)$$

The location of the source (w_s, z_s) and receiver (w_r, z_r) are known. Figure 2.1 illustrates the example set-up. The source and receiver locations are set to $(w_s, z_s) = (2, 2)$ and $(w_r, z_r) = (0, 0)$ in the example. Moreover, the transition between the two layers is at $z_c = 1$. Then (2.3) reduces to

$$t = x_1\sqrt{(2 - w_c)^2 + 1} + x_2\sqrt{w_c^2 + 1}. \quad (2.4)$$

According to Fermat's principle, the ray travels the fastest route (Ammon et al., 2021). Therefore w_c is the solution to

$$\frac{\partial t}{\partial w_c} = 0, \quad (2.5)$$

which can be found numerically by using Newton's method (Nocedal & Wright, 2006).

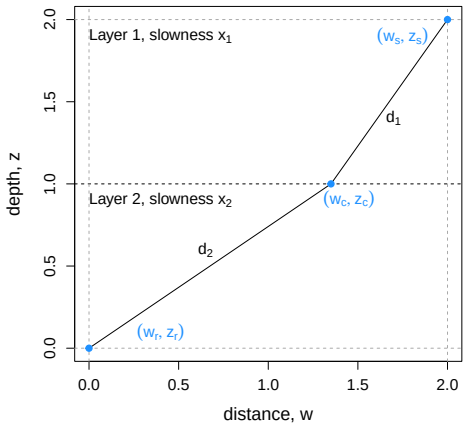


Figure 2.1: Illustration of the situation in the ray tracing example. Both layers have the same depth. A wave is sent from the source at $(w_s, z_s) = (2, 2)$ and is received at $(w_r, z_r) = (0, 0)$. The transition between the two layers is at depth $z_c = 1$. The distances travelled in the layers by the shortest ray path are denoted d_1 and d_2. The slowness of layer 1 is denoted x_1 and the slowness of layer 2 is denoted x_2.

Suppose an observation of the total travel time, y, deviates from the true travel time due to noise in the

measurement. That is,

$$y = t + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \gamma^2). \tag{2.6}$$

The forward model is thus $h(x_1, x_2) = x_1\sqrt{(2 - w_c)^2 + 1} + x_2\sqrt{w_c^2 + 1}$, where w_c is the solution to equation (2.5). In this example, the noise is $\gamma^2 = 0.05^2$, which gives the likelihood

$$y|x_1, x_2 \sim \mathcal{N}(t, 0.05^2). \tag{2.7}$$

The likelihood for an observation $y = 3$ is visualised in Figure 2.2. For a measured travel time, there are two unknowns x_1 and x_2, and so from a single measurement, y, there is no unique solution for $[x_1, x_2]$ even if the noise variance γ^2 approaches zero.

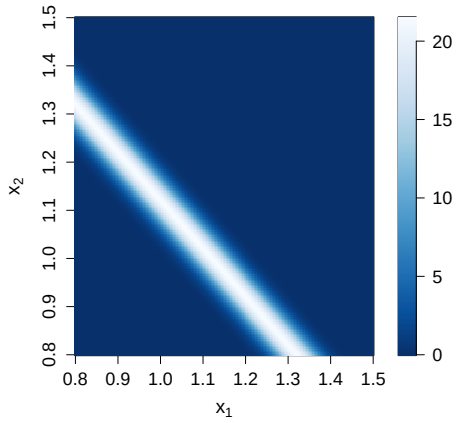


Figure 2.2: Graphical description of the likelihood $f(y|x_1, x_2)$ in the ray tracing example for data $y = 3$.

A log-normal distribution is assumed as prior for the slowness parameters:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \text{log-normal}\left(\boldsymbol{\mu}, \sigma^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right), \tag{2.8}$$

with $\boldsymbol{\mu} = [0.1, 0.1]^T$ and $\sigma^2 = 0.1^2$. This prior is symmetric in $(\log(x_1), \log(x_2))$-space. An illustration of the prior in (x_1, x_2)-space is shown in Figure 2.3.

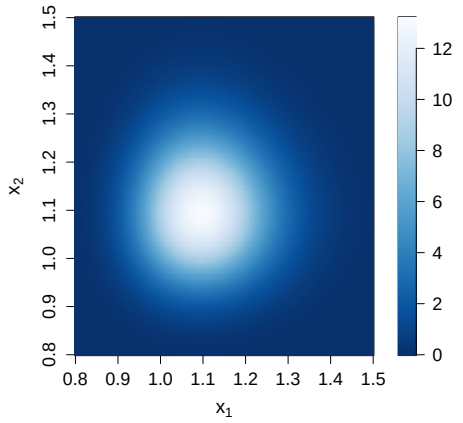


Figure 2.3: Graphical description of the prior $f(x_1, x_2)$ for the slowness parameters in the ray tracing example.

It is in practice impossible to find an analytic expression for this posterior, but as it is only bivariate, it is possible to visualize a numerical approximation to the posterior. To do this, the likelihood and prior are calculated for 200^2 evenly spaced discrete values in the area $[0.8, 1.5] \times [0.8, 1.5]$. That is $(x_1, x_2) \in \mathcal{D} = \{[0.8, 0.8 + \delta, ..., 1.5 - \delta, 1.5] \times [0.8, 0.8 + \delta, ..., 1.5 - \delta, 1.5]\}$ for $\delta = \frac{1.5-0.8}{200} \approx 0.0035$. At each point on the grid, the product of the likelihood and the prior is computed according to

$$\pi(x_1, x_2) \propto f(y|x_1, x_2) f(x_1, x_2). \tag{2.9}$$

This gives values proportional to the posterior, which in turn are normalised such that

$$\sum_{x_1, x_2 \in \mathcal{D}} \pi(x_1, x_2)\delta^2 = 1. \tag{2.10}$$

The numerical approximation to the posterior is visualised in Figure 2.4.

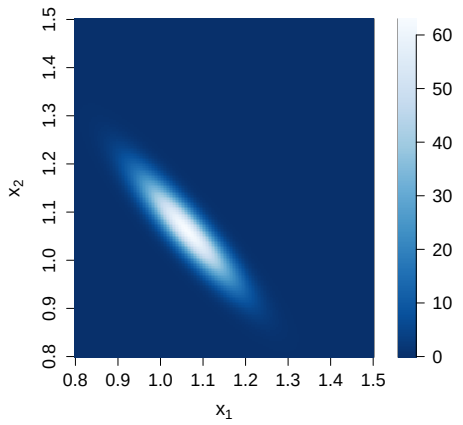


Figure 2.4: Graphical description of the posterior $\pi(x_1, x_2)$ for the slowness parameters in the ray tracing example.

When the dimension is higher, it is infeasible to calculate, normalize and visualize the posterior in this way. In these cases, MCMC can be used to do Bayesian inversion and MCMC is the topic of the next section.

2.2 Markov chain Monte Carlo

MCMC is a method that can be used to draw samples from a posterior distribution. The idea of MCMC is to construct a Markov chain that converges to a desired posterior distribution (Givens & Hoeting, 2013). Robert and Casella (2004) define an MCMC method as any method producing an ergodic Markov chain whose stationary distribution is the posterior π. This section attempts to briefly understand this definition by presenting Markov chains, in particular ergodic Markov chains, and stationary distributions. The MCMC methods in this thesis construct discrete-time Markov chains on the state space $\mathbb{R}^N$. For simplicity, the following paragraphs regard discrete-time Markov chains on finite state space, though, the results for continuous state spaces are similar to the results presented in the next paragraphs (Givens & Hoeting, 2013). After establishing a notion of an ergodic Markov chain and stationary distributions, the MH algorithm is presented. Proposal distributions are important in the MH algorithm and four

proposal distributions are presented in Section 2.2.3. Convergence and mixing analysis tools are presented. The two-dimensional ray tracing example introduced in Section 2.1 is revisited to illustrate how to tune proposal distributions and how to examine the convergence and mixing. Finally, the Monte Carlo approximation is described in brief.

2.2.1 Markov chains

To define a Markov chain, let $\{x_t\}_{t=0}^{m}$ be a stochastic process, that is a sequence of random variables, with subscripts t indicating the time iteration. Moreover, let $\mathcal{S}$ denote the state space, that is the collection of all possible values x_t can take. If $a \in \mathcal{S}$, then a is called a state. A Markov chain is a stochastic process where the probability of transitioning from one state to another state in one time step is only dependent on the current state. This is called the Markov property. By denoting the one-step transition probability from state a to state b as $p(b|a)$, the Markov property is mathematically written as

$$p(b|a) := P\left(x_{t+1} = b | x_t = a, x_{t-1} = a_{t-1}, \ldots, x_0 = a_0\right) = P\left(x_{t+1} = b | x_t = a\right), \tag{2.11}$$

for states $b, a, a_{t-1}, \ldots, a_0 \in \mathcal{S}$ (Givens & Hoeting, 2013).

Some Markov chains have stationary distributions. That is a distribution $\boldsymbol{\pi} = [..., \pi_a, \pi_b, ...]$ such that if x_t follows this distribution, so does x_{t+1} (Robert & Casella, 2004). If a Markov chain converges to its stationary distribution, the element π_b of the stationary distribution $\boldsymbol{\pi}$, approaches the probability of finding the Markov chain in state b as $t \to \infty$. Another interpretation is that π_b gives the long-run mean fraction of time the Markov chain spends in state b (Pinksy & Karlin, 2011).

For a Markov chain to have and converge to a unique stationary distribution some requirements need to be met. If a Markov chain is recurrent, aperiodic and irreducible it has a unique stationary distribution to which it converges (Pinksy & Karlin, 2011). According to Givens and Hoeting (2013), such Markov chains (irreducible, aperiodic and recurrent) are called ergodic. This paragraph explains intuitively in turn what these properties mean. In an irreducible Markov chain, it is possible to transition between any two states in the state space in a finite number of time steps. If, in addition, it is possible to stay in the same state over two or more time iterations, the Markov chain is called aperiodic. A state is recurrent if the probability that the chain returns to the state is 1 (Givens & Hoeting, 2013).

Fundamental convergence results are elemental for the motivation of MCMC methods (Robert & Casella, 2004). However, showing that these properties hold for a general state space is outside the scope of this thesis. In this thesis, the transition probabilities are Gaussian proposals, which have support on $\mathbb{R}^N$, multiplied by an acceptance probability. For these proposals, the requirements suffice (Robert & Casella, 2004). Intuitively this seems reasonable as one can draw the same sample twice from a Gaussian distribution. It is also possible to reach anywhere from anywhere in a finite number of steps. And as $t \to \infty$, the Markov chain eventually visits every state. Moving on it is assumed that the chains created by the MCMC scheme are ergodic Markov chains as described in Robert and Casella (2004).

2.2.2 The Metropolis-Hastings algorithm

The definition of an MCMC method by Robert and Casella (2004) encloses different types of MCMC methods such as Gibbs sampling and MH (Givens & Hoeting, 2013), however, this thesis only concerns the MH algorithm. The MH algorithm uses the detailed balance equation:

$$\pi_a p(b|a) = \pi_b p(a|b) \tag{2.12}$$

to ensure that the stationary distribution is the posterior (Givens & Hoeting, 2013). If there exists a function $\boldsymbol{\pi}$ that satisfies the equation (2.12) then $\boldsymbol{\pi}$ is a stationary distribution to the Markov chain defined by the transition probabilities $p(b|a)$ for all $a, b \in \mathcal{S}$. A Markov chain that satisfies the detailed balance equation is reversible (Robert & Casella, 2004). This means that, in the long run, the chain transitions equally many times from a to b as from b to a. For $\boldsymbol{\pi}$ to be a stationary distribution to the Markov chain associated with the transition probability $p(b|a)$, it is not required that the chain is reversible, but it is a sufficient condition which is easily checked, and therefore used by many MCMC methods (Robert & Casella, 2004).

To exploit the detailed balance equation (2.12) such that the posterior distribution $\boldsymbol{\pi}$ is the stationary distribution, the transition probability $p(b|a)$ is expressed as the product

$$p(b|a) = q(b|a)\alpha(b|a), \tag{2.13}$$

where $q(b|a)$ is the probability of proposing a move to state b when the current state is a and $\alpha(b|a)$ is the probability of accepting this move. Proposal distributions are the subject of the next section, for now, the attention is directed to the acceptance probability $\alpha(b|a)$. The key is to select this acceptance probability such that the detailed balance equation (2.12) is fulfilled. Inserting equation (2.13) into equation (2.12) gives

$$\pi_a q(b|a)\alpha(b|a) = \pi_b q(a|b)\alpha(a|b), \tag{2.14}$$

which can be written as

$$\frac{\alpha(b|a)}{\alpha(a|b)} = \frac{\pi_b q(a|b)}{\pi_a q(b|a)} \tag{2.15}$$

By choosing

$$\alpha(b|a) = \min\left\{1, \frac{\pi_b}{\pi_a}\frac{q(a|b)}{q(b|a)}\right\}, \tag{2.16}$$

then in the case where $\pi_a q(b|a) \leq \pi_b q(a|b)$

$$\alpha(b|a) = 1 \tag{2.17}$$

and

$$\alpha(a|b) = \frac{\pi_a}{\pi_b}\frac{q(b|a)}{q(a|b)}. \tag{2.18}$$

Thus

$$\frac{\alpha(b|a)}{\alpha(a|b)} = \frac{\pi_b}{\pi_a}\frac{q(a|b)}{q(b|a)} \tag{2.19}$$

which satisfies equation (2.15). In the other case when $\pi_a q(b|a) > \pi_b q(a|b)$

$$\alpha(a|b) = 1 \tag{2.20}$$

and thus

$$\frac{\alpha(b|a)}{\alpha(a|b)} = \alpha(b|a) = \frac{\pi_b}{\pi_a}\frac{q(a|b)}{q(b|a)}, \tag{2.21}$$

which also fulfils equation (2.15). Consequently, by choosing $\alpha(a|b)$ according to (2.16) the detailed balance given in equation (2.12) is satisfied and $\boldsymbol{\pi}$ is the stationary distribution of the Markov chain associated with the transition probability $q(b|a)\alpha(b|a)$, for all $a, b \in \mathcal{S}$. Moreover, as it is assumed that the Markov chain produced by the MH scheme is ergodic, the Markov chain converges to this stationary distribution.

Moving on, the state of a Markov chain produced by the MH algorithm at time t, is referred to as a sample instead of a state and denoted by $\boldsymbol{x}_t$.

The MH algorithm works as follows. At time iteration $t+1$, a sample $\boldsymbol{x}^p$ is proposed from a proposal distribution $q(\boldsymbol{x}^p|\boldsymbol{x}_t)$. The proposed sample is accepted with probability

$$\alpha(\boldsymbol{x}^p|\boldsymbol{x}_t) = \min\left\{1, \frac{\pi(\boldsymbol{x}^p)}{\pi(\boldsymbol{x}_t)}\frac{q(\boldsymbol{x}_t|\boldsymbol{x}^p)}{q(\boldsymbol{x}^p|\boldsymbol{x}_t)}\right\}. \tag{2.22}$$

If $\boldsymbol{x}^p$ is accepted, then $\boldsymbol{x}_{t+1} = \boldsymbol{x}^p$, otherwise the current sample is kept $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$. As the MH algorithm depends on the posterior density ratio,

$$\frac{\pi(\boldsymbol{x}^p)}{\pi(\boldsymbol{x}_t)} = \frac{f(\boldsymbol{y}|\boldsymbol{x}^p)f(\boldsymbol{x}^p)f(\boldsymbol{y})}{f(\boldsymbol{y})f(\boldsymbol{y}|\boldsymbol{x}_t)f(\boldsymbol{x}_t)} = \frac{f(\boldsymbol{y}|\boldsymbol{x}^p)f(\boldsymbol{x}^p)}{f(\boldsymbol{y}|\boldsymbol{x}_t)f(\boldsymbol{x}_t)}, \tag{2.23}$$

normality constant of the posterior, $f(\boldsymbol{y})$, and the normality constant of the proposal distribution, are not needed.

2.2.3 Proposal distributions

By defining the acceptance rate as in equation (2.22), almost any proposal distribution, as long as the produced Markov chain is ergodic, can be used in the MH algorithm. Selecting a good proposal distribution, q, is an important and often difficult task. A good proposal distribution covers the posterior at its tails and is also similar to the posterior (Givens & Hoeting, 2013). Moreover, the proposal distribution should be easy to sample from. The proposal distribution influences the efficiency of the MH algorithm. Now, four different types of proposal distributions are presented, denoted by q_1, q_2, q_3 and q_4 respective to the order they are presented. All these proposal distributions are dependent proposal distributions because they are dependent on the current sample. If the proposal distribution is independent of the current sample it is called an independent proposal distribution (Givens & Hoeting, 2013).

Random walk

The first proposal distribution is the standard random walk. It is a simple and intuitive proposal. In random walk, a sample is proposed according to

$$\boldsymbol{x}^p = \boldsymbol{x}_t + \boldsymbol{\varepsilon}_t, \quad \boldsymbol{\varepsilon}_{t+1} \sim \mathcal{N}(\boldsymbol{0}, s^2\boldsymbol{I}). \tag{2.24}$$

Here $\boldsymbol{x}_t$ is the current sample and $\boldsymbol{\varepsilon}_t$ are i.i.d. noise with standard deviation s. Hence the proposal distribution is a Gaussian distribution,

$$q_1(\boldsymbol{x}^p|\boldsymbol{x}_t) = \varphi(\boldsymbol{x}_t, s^2\boldsymbol{I}), \tag{2.25}$$

where $\varphi(\boldsymbol{x}_t, s^2\boldsymbol{I})$ denotes the normal probability density function with $\boldsymbol{x}_t$ as expected value and variance s^2. The standard deviation, s, is a tuning parameter which should be tuned to make the algorithm more efficient. Approaches for tuning s are discussed after the four proposal distributions have been presented.

Due to the simplicity and generality of this proposal distribution, the random walk is widely used (Kamatani, 2020). An advantage of the random walk is that it is symmetric. This means that $q_1(\boldsymbol{x}^p|\boldsymbol{x}_t) = q_1(\boldsymbol{x}_t|\boldsymbol{x}^p)$. An MH algorithm with a symmetric proposal is called a Metropolis algorithm (Givens & Hoeting, 2013). This is convenient as the acceptance rate in equation (2.22) reduces to

$$\alpha = \min\left\{1, \frac{\pi(\boldsymbol{x}^p)}{\pi(\boldsymbol{x}_t)}\right\}. \tag{2.26}$$

At each iteration, only the likelihood $f(\boldsymbol{y}|\boldsymbol{x}^p)$ and prior $f(\boldsymbol{x}^p)$ needs to be computed. Proposing a sample has a relatively low computational cost as the covariance matrix has zero elements everywhere but the diagonal.

The efficiency of this proposal decreases with increasing dimension. When the posterior is light-tailed and similar to a Gaussian distribution, the number of iterations m should be proportional to the dimension N (Kamatani, 2020). However, Kamatani (2020) found that when the posterior is not light-tailed and further from a Gaussian distribution, the number of iterations should be proportional to N^2 for the standard random walk.

Random walk with covariance

Another similar and symmetric proposal distribution is a random walk with covariance. That is

$$q_2(\boldsymbol{x}^p|\boldsymbol{x}_t) = \varphi(\boldsymbol{x}_t, s^2\boldsymbol{\Sigma}), \tag{2.27}$$

where $\boldsymbol{\Sigma}$ is a covariance matrix and s is a tuning parameter.

The acceptance probability is reduced to (2.26) because it is symmetric. As for q_1, only the likelihood $f(\boldsymbol{y}|\boldsymbol{x}^p)$ and prior $f(\boldsymbol{x}^p)$ needs to be computed at each iteration. However, because this proposal distribution draws samples with correlation, proposing a sample requires a matrix multiplication.

As the covariance structure of the posterior is unknown, it is not clear which covariance matrix to use. In this thesis, the covariance matrix of the prior is used. In this way, this proposal distribution uses more information than the first proposal. However, at the cost of being less computationally efficient compared to the standard random walk. Especially when the dimension becomes very high, the matrix multiplications with the covariance matrix become very computationally expensive.

Preconditioned Crank-Nicolson

A modified version of the random walk is the pCN proposal distribution. Suppose the prior distribution in a Bayesian inversion framework has a Gaussian distribution $\varphi(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with non-zero mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Then the pCN proposal distribution

$$q_3(\boldsymbol{x}^p|\boldsymbol{x}_t) = \varphi\left(\boldsymbol{\mu} + \sqrt{1-s^2}(\boldsymbol{x}_t - \boldsymbol{\mu}), s^2\boldsymbol{\Sigma}\right), \tag{2.28}$$

with tuning parameter s is reversible with respect to the prior distribution (Pinski et al., 2015). This means that

$$f(\boldsymbol{x}_t)q_3(\boldsymbol{x}^p|\boldsymbol{x}_t) = f(\boldsymbol{x}^p)q_3(\boldsymbol{x}_t|\boldsymbol{x}^p) \tag{2.29}$$

which in turn means that the acceptance probability given in equation (2.22) reduces to the likelihood ratio

$$\alpha = \min\left\{1, \frac{f(\boldsymbol{y}|\boldsymbol{x}^p)}{f(\boldsymbol{y}|\boldsymbol{x}_t)}\right\}. \tag{2.30}$$

This leads to significant speed-ups for problems which are discretized on a grid of dimension N. This proposal only requires evaluating the likelihood function and performing a matrix multiplication when proposing a new sample.

As opposed to the standard random walk q_1, the number of steps required by using pCN does not increase with the dimension N. Hence using pCN, the support of the posterior is explored more efficiently than by using q_1, and the difference in efficiency increases with the dimension (Rudolf & Sprungk, 2016).

Metropolis-adjusted Langevin algorithm

Finally, the last proposal is on the form

$$q_4(\boldsymbol{x}^p|\boldsymbol{x}_t) = \varphi\left(\boldsymbol{x}_t + \frac{s^2}{2}\nabla \log\left(\pi(\boldsymbol{x}_t)\right), s^2 I\right). \tag{2.31}$$

Here $\nabla \log\left(\pi(\boldsymbol{x}_t)\right)$ is the gradient of the posterior evaluated at the current sample and s is a tuning parameter. An MH algorithm with the distribution in equation (2.31) as proposal distribution is called MALA. One can deduct MALA from the Langevin diffusion equation (Roberts & Rosenthal, 1998).

This proposal distribution exploits the form of the posterior by shifting the expected value in the direction of higher posterior density. Using information about the posterior built into the proposal gives a higher chance of creating an efficient algorithm (Khoshkholgh et al., 2021). MALA is the most computationally expensive of the four proposal distributions. To calculate the acceptance probability, the prior, likelihood, gradient of the prior and gradient of the likelihood need to be evaluated. Even though each iteration is more computationally expensive, one probably does not need to compute as many iterations to obtain the same information as one would by using q_1, because MALA can be considerably more efficient than the MH algorithm with the standard random walk as proposal distribution. What is meant by efficient is described in Section 2.2.4. Roberts and Rosenthal (1998) found that for a lightly-tailed posterior similar to a Gaussian distribution, the number of iterations, m, needed is proportional to $N^{\frac{1}{3}}$ compared to N for the random walk q_1. Unfortunately, if the posterior has many modes, using the MALA may lead to only exploring one of the posterior's modes and sometimes lead to slow convergence (Givens & Hoeting, 2013).

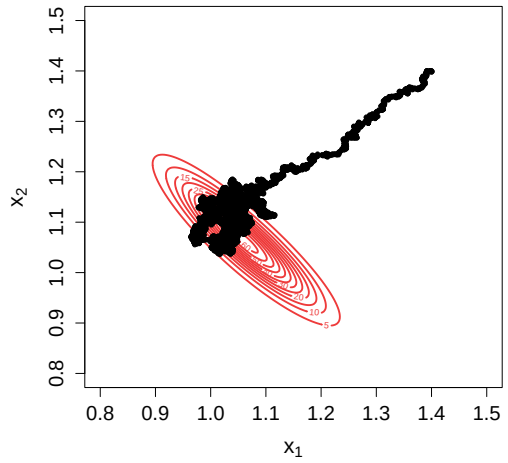
Scaling the proposal distributions

The tuning parameter, s, in the four proposal distributions, should be tuned to make the MH algorithms more efficient. Tuning can be time-consuming (Eidsvik & Tjelmeland, 2006), however, studies where the MH algorithms are tuned in general cases can serve as guidelines when using MH algorithms for specific problems. Under quite general conditions, Roberts et al. (1997) found that scaling s such that the asymptotic acceptance rate is 23.4%, maximizes the efficiency of a multidimensional random walk Metropolis algorithm. In Cotter et al. (2013) the pCN is also tuned such that the acceptance rate is about 23.4%. The optimal asymptotic acceptance rate for MALA in high dimension is approximately 57.4% (Roberts & Rosenthal, 1998). These acceptance rates are used as pointers to values of s which give efficient MCMC sampling in high dimension. In this thesis, s will be tuned such that the acceptance rates of q_1, q_2 and q_3 are approximately 23.4% and q_4 such that the acceptance rate is near 57.4%.

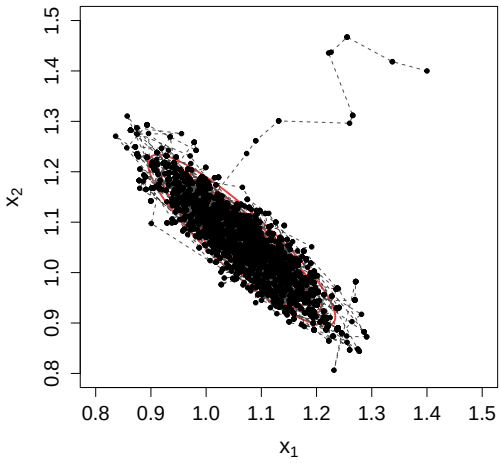
The ray tracing example is employed to illustrate why these acceptance rates are favoured by demonstrating what happens when s is chosen either too low or too high. A description of the prior and likelihood in the ray tracing example is given in Section 2.1. MCMC is performed for $m = 5000$ iterations using three different values for s. For this part, only the proposal q_1 is used. Figure 2.5a, Figure 2.5b and Figure 2.5c show the values of the produced Markov chains for standard deviations $s = 0.002$, $s = 0.1$ and $s = 0.8$, respectively. Black circles indicate the values of $\{[x_1, x_2]_t\}_{t=0}^{m=5000}$ at time iteration t, while grey, the dashed line shows the path. The red contour curves are from the numerically calculated posterior distribution corresponding to Figure 2.4 in Section 2.1. The Markov chains start at $\boldsymbol{x}_0 = [1.4, 1.4]_0$.

Figure 2.5a illustrate the consequence of selecting a too low standard deviation s. Since s is very small, the proposed sample is likely very close to the current sample, and consequently, the steps of the random walk are small. The grey dashed lines are not visible as the black circles are so close to each other. The

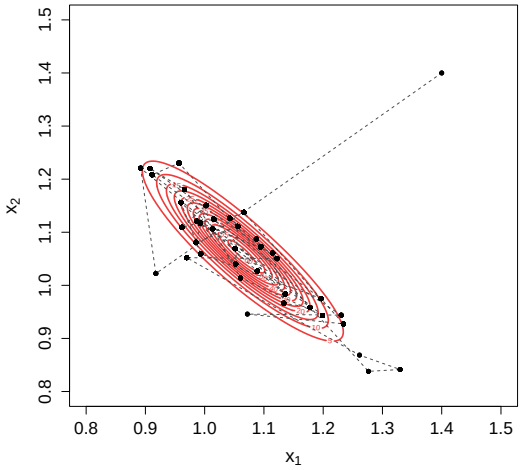
sample values are similar and strongly influenced by previous sample values. This is called poor mixing. Eventually, the Markov chain arrives to the high posterior density region, but it is unable to explore much of the posterior in the remaining time iterations. In Figure 2.5a, one can see essential parts of the posterior left unexplored. This scenario gives a high acceptance rate, beacause the posterior ratio is close to one. The acceptance rate from this MCMC run was 96.3%.



(a) Path of $\{[x_1, x_2]_t\}_{t=0}^{m=5000}$ sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.002$. The acceptance rate was 96.3%.



(b) Path of $\{[x_1, x_2]_t\}_{t=0}^{m=5000}$ sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.1$. The acceptance rate was 23.6%.



(c) Path of $\{[x_1, x_2]_t\}_{t=0}^{m=5000}$ sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.8$. The acceptance rate was 0.6%.

Figure 2.5: Path of Markov chains from the ray tracing example described in Section 2.1. Values of $\{[x_1, x_2]_t\}_{t=0}^{m=5000}$ are marked as black circles, while grey dashed lines mark the path. Red contour curves are numerical approximations of the posterior distribution also shown in Figure 2.4. The Markov chains started in $\boldsymbol{x}_0 = [1.4, 1.4]_0$.

If, on the other hand, the standard deviation is too large, the acceptance rate becomes very low. In Figure 2.5c, the tuning parameter was 0.8 and the acceptance rate was only 0.6%. In this path, the proposed sample is far away from the current sample and therefore seldom in the high posterior density region. Hence many proposed samples are not accepted and the process stays in the same sample over many time iterations before moving. This is an inefficient and time-consuming way to explore the region of the posterior.

The acceptance rate of 23.4% presented in Roberts et al. (1997) give a trade-off between accepting a sufficient amount of the proposed samples, while also exploring most of the posterior. Figure 2.5b illustrates MCMC results when s was tuned according to Roberts et al. (1997). The standard deviation was 0.1 which gave an acceptance rate of 23.6%. More or less every part of the posterior is visited. Between the three MCMC runs, this is clearly the most efficient way to explore the posterior in the 5000 given time iterations. Figure 2.5b demonstrates good mixing and fast convergence, which is discussed in the next section.

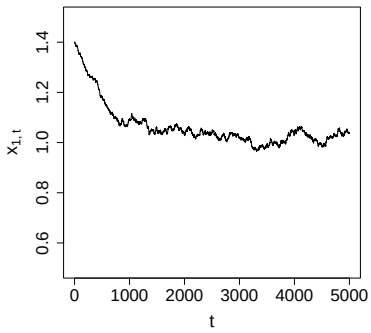
2.2.4 Convergence and mixing diagnostics

In most situations where MCMC is performed, there are unfortunately no contour plots of the posterior to compare the MCMC samples with. However, tools are available for assessing convergence and mixing. Convergence means that the Markov chain has approximately reached its stationary distribution. Mixing is how quickly the Markov chain explores the support of the posterior and how fast it forgets its initial value (Givens & Hoeting, 2013). Mixing and convergence are two related concepts and many of the same diagnostics apply to investigate both properties. This section concerns trace plots, autocorrelation plots and effective sample size (ESS). Note that no diagnostics can assure convergence, only indicate convergence of an MCMC algorithm. Givens and Hoeting (2013) therefore recommend using a variety of diagnostics techniques. Again, the ray tracing example is employed to illustrate how to use these techniques.

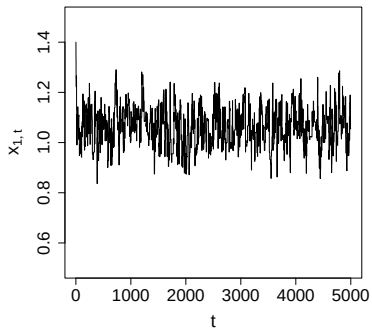
Trace plots

Trace plots display the value of a stochastic process $\{\boldsymbol{x}_t\}_{t=0}^m$ as a function of the time iteration t. Figure 2.6a, Figure 2.6b and Figure 2.6c show trace plots for $\{x_{1,t}\}_{t=0}^m$ in the ray tracing example. The trace plots correspond to values in Figure 2.5a, Figure 2.5b and Figure 2.5c, that is, they are from MCMC runs using the standard random walk proposal, q_1, with standard deviations $s = 0.002$, $s = 0.1$ and $s = 0.8$, respectively. The three scenarios from the ray tracing example will serve as objects of study to understand how trace plots can be used to determine convergence and mixing properties. Only trace plots for $\{x_{1,t}\}_{t=0}^m$ are considered here, however, one could have regarded trace plots of $\{x_{2,t}\}_{t=0}^m$ instead.

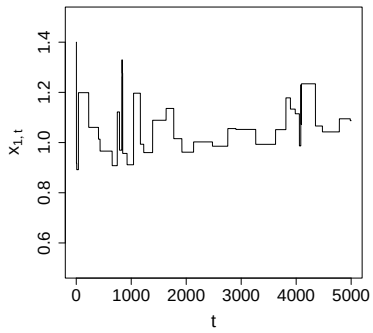
Identifying a clear burn-in period in a trace plot indicates convergence. The burn-in period are the first iterations, where the Markov chain moves from the starting value to the high posterior density region (Givens & Hoeting, 2013). A shorter burn-in period indicates faster convergence. How many iterations are required for convergence depends on where you start and which proposal you use. In the ray tracing example, all three MCMC runs started at the same initial sample, however, using a low value for s required more iterations before reaching convergence as shown in Figure 2.6a. Approximately the first 1000 samples make up the burn-in in Figure 2.6a, which is significantly more than in Figure 2.6b and Figure 2.6c, where the burn-in appears to be about $10 - 20$ iterations. It is a bit difficult to see the burn-in in Figure 2.6b and especially in Figure 2.6c.



(a) Trace plot of MCMC samples sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.002$. The acceptance rate was 96.3%.

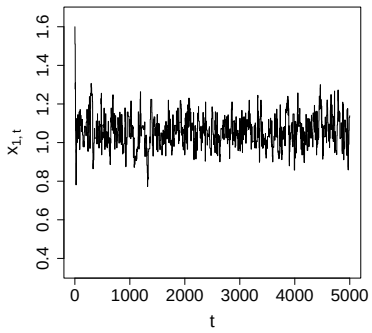


(b) Trace plot of MCMC samples sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.1$. The acceptance rate was 23.6%.

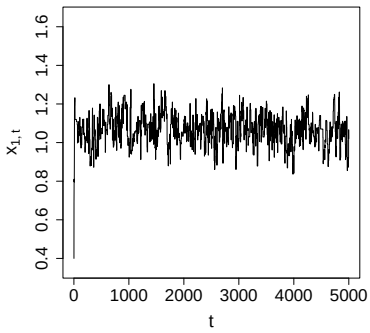


(c) Trace plot of MCMC samples sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.8$. The acceptance rate was 0.6%.

Figure 2.6: Trace plots for $\{x_{1,t}\}_{t=0}^{m}$ in the ray tracing example described in Section 2.1. Values for $\{x_{1,t}\}_{t=0}^{m}$ are plotted against the time iteration t. The Markov chain started in $\boldsymbol{x}_0 = [1.4, 1.4]_0$.



(a) Trace plot of MCMC samples sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.1$ starting in $\boldsymbol{x}_0 = [1.6, 1.6]_0$.



(b) Trace plot of MCMC samples sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.1$ starting in $\boldsymbol{x}_0 = [0.4, 0.4]_0$.

Figure 2.7: Trace plots for $\{x_{1,t}\}_{t=0}^{m}$ in the ray tracing example described in Section 2.1. The Markov chain was started in two different initial samples.

The Markov chain should be started with several different initial values to ensure that the Markov chain has converged. If the Markov chain converges to the same region this indicates that it has reached its stationary distribution (Givens & Hoeting, 2013). Figure 2.7 shows two trace plots from the ray tracing example with initial samples $[1.6, 1.6]_0$ and $[0.4, 0.4]_0$. Both paths end up in the same region, indicating that the samples after burn-in are sampled from the posterior distribution. It is especially important to try several starting values if the posterior is multi-modal, as the Markov chain can stay in one of the high-density regions for a long time before moving to another high-density region. If the trace plot only captures the burn-in and samples from one of the high-density regions, it gives the illusion of convergence,

while in fact the samples only represent parts of the stationary distribution. Before estimating distribution properties from MCMC samples, the burn-in is removed.

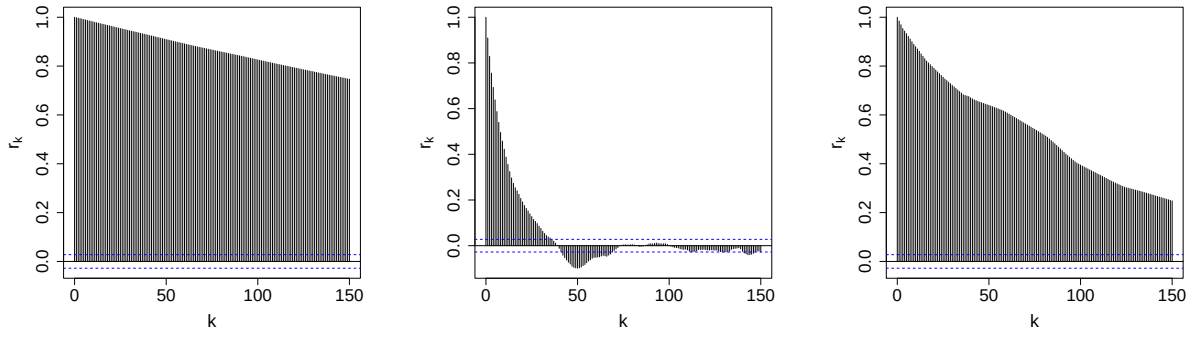
Trace plots can also be used to assess the mixing properties of an MCMC method. If the MCMC algorithm is mixing poorly, the Markov chain will stay close to or at the same value for many iterations (Givens & Hoeting, 2013). The path in Figure 2.6c stays at the same value over many iterations and the values of the path in Figure 2.6a are close to each other. Hence these are examples of bad mixing. Ideally, a trace plot similar to Figure 2.6b is desired. The stable, horizontal band indicates good mixing (Givens & Hoeting, 2013).

Autocorrelation plots

Autocorrelation plots are good for assessing mixing. Mixing is closely related to the correlation between samples at different lags. In an autocorrelation plot, the autocorrelation function (ACF) is shown on the vertical axis and lag, k, on the horizontal axis. The ACF, denoted by r_k, gives the sample correlation between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+k}$ at lag k:

$$r_k = \frac{\sum_{t=0}^{m}(\boldsymbol{x}_{t+k} - \bar{\boldsymbol{x}})(\boldsymbol{x}_t - \bar{\boldsymbol{x}})}{\sum_{t=0}^{m}(\boldsymbol{x}_t - \bar{\boldsymbol{x}})^2} \tag{2.32}$$

(Venables & Ripley, 2002). Values of r_k close to 1 signify that the values of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+k}$ are similar. Conversely, if r_k is close to minus 1 the values of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+k}$ are dissimilar (Walpole et al., 2012). Figure 2.8 shows autocorrelation plots for the three different MCMC runs in the ray tracing example discussed so far. Ideally, the autocorrelation plot decreases quickly because this signifies that the samples are less correlated. That is, the Markov chain forgets its initial value quickly and thus mixes well. Using a relatively low or high value for s, gives autocorrelation plots that decay slowly. When s is too low, the MCMC samples are very similar and therefore highly correlated. On the other hand, when s is too high, the same sample is kept over many time iterations which also makes the samples correlated.



(a) Autocorrelation plot of MCMC samples sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.002$. The acceptance rate was 96.3%.

(b) Autocorrelation plot of MCMC samples sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.1$. The acceptance rate was 23.6%.

(c) Autocorrelation plot of MCMC samples sampled with the MH algorithm using the proposal distribution q_1 with standard deviation $s = 0.8$. The acceptance rate was 0.6%.

Figure 2.8: Autocorrelation plots for $\{x_{1,t}\}_{t=0}^{m}$ in the ray tracing example described in Section 2.1. The ACF r_k is on the vertical axis and the lag k is on the horizontal axis.

Effective sample size

To find the trade-off value between low and high s one can use the ESS. ESS is how many independent samples the information from the dependent MCMC samples represents (Givens & Hoeting, 2013). That is, the higher ESS, the less correlated samples. For $\widetilde{m}$ MCMC samples after burn-in, ESS is defined as

$$\text{ESS} = \frac{\widetilde{m}}{\tau}, \tag{2.33}$$

where τ is the integrated autocorrelation time (IAT) defined as

$$\tau = 1 + 2\sum_{k=1}^{\infty} r_k. \tag{2.34}$$

It is common to truncate the sum in equation (2.34) when $r_k < 0.05$ (Kass et al., 1998). The IAT is proportional to the area under the ACF curve. IAT is roughly the number of iterations per independent sample (Christen & Fox, 2010). Hence, a low IAT is good. As ESS is inversely proportional to IAT, it is clear that a high value of ESS is better. An MCMC algorithm with a larger ESS is likely to converge in fewer time iterations compared to an MCMC algorithm with a smaller ESS (Givens & Hoeting, 2013).

The ESSs from the ray tracing example are presented in Table 2.1. The ESSs are calculated on 3000 iterations after burn-in. By now, it is probably no surprise that using $s = 0.1$ obtains the highest ESS. The 3000 MCMC samples contain as much information as approximately 139 i.i.d. samples.

proposal	s	acceptance rate [%]	ESS [#]
q_1	0.002	96.3	6.081
q_1	0.1	23.6	139.410
q_1	0.8	0.6	5.869

Table 2.1: Comparison of acceptance rates and ESS for three MH algorithms with proposal distribution q_1 in the ray tracing example described in Section 2.1. The three proposal distributions in the algorithms had different standard deviations, as reported in the table. ESSs are measured for $\{x_{1,t}\}_{t=2000}^{5000}$.

An attempt to illustrate how ESS relates to the acceptance rate is shown in Figure 2.9. The ESS of $200,000$ MCMC samples after burn-in of the stochastic process $\{x_{1,t}\}_t$ is calculated for different values of s. The ESSs are divided by the max ESS such that they can be plotted on top of the acceptance rate. The black dashed line shows the parameter value, $s = 0.1$, used in the ray tracing example. The figure shows that this value approximately corresponds to an acceptance rate of 23.4% marked by the dashed, orange line and that this is close to the maximum ESS. In high dimension, tuning the standard deviation such that the asymptotic acceptance rate is 23.4% maximizes the ESS for the same number of iterations (Roberts & Rosenthal, 2001). As the dimension of $\boldsymbol{x}$ is only 2, it is reasonable that the optimal acceptance rate is different to 23.4%. Roberts and Rosenthal (2001) finds that for dimension around 5 and higher, the optimal acceptance rate approaches 23.4%.

For a fixed number of samples, a higher ESS is better, however, to measure the efficiency of different MH algorithms, the time used to produce those samples must also be accounted for. This is because the overall goal is to produce the least correlated samples in the shortest amount of time. Eidsvik and Tjelmeland (2006) used the number of evaluations per independent sample to measure the efficiency. In

this thesis, the measure ESS per computation time will determine the efficiency of the algorithm. The algorithm with the highest ESS per time will serve as the most efficient.

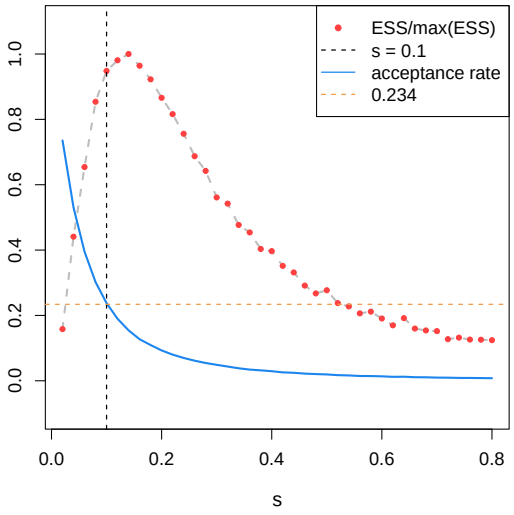


Figure 2.9: ESS of $\{x_{1,t}\}_{t=5000}^{205000}$ plotted as a function of the standard deviation s. The ESSs are divided by the maximum ESS such that they can be compared to the acceptance rates. The blue line shows the acceptance rate as a function of s. The black dashed line marks the parameter value $s = 0.1$ used in the ray tracing example. The acceptance rate of 23.4% is marked by the orange dashed line.

2.2.5 Monte Carlo approximations

Monte Carlo approximation is a procedure to approximate important distribution quantities from a sample set. Given a set of $\widetilde{m}$ samples $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{\widetilde{m}}$ from a distribution, $f(\boldsymbol{x})$, not to be confused with the prior distribution, many interesting quantities can be expressed as an expected value of a function $E[u(\boldsymbol{x})]$. The Monte Carlo approximation estimates this quantity by the sample average

$$E[u(\boldsymbol{x})] \approx \frac{1}{\widetilde{m}} \sum_{t=1}^{\widetilde{m}} u(\boldsymbol{x}_t). \tag{2.35}$$

If the samples $\boldsymbol{x}$ are i.i.d. then

$$E[u(\boldsymbol{x})] \approx \frac{1}{\widetilde{m}} \sum_{t=1}^{\widetilde{m}} u(\boldsymbol{x}_t) \to \int u(\boldsymbol{x}) f(\boldsymbol{x}) dx = E[u(\boldsymbol{x})] \tag{2.36}$$

as $\widetilde{m} \to \infty$ by the strong law of large numbers (Givens & Hoeting, 2013). As mentioned, MCMC does not draw independent samples. However, a sequence of samples produced by an MCMC scheme is sufficient to ensure proper approximations through approximations like (2.35) see e.g. (Robert & Casella, 2004). The burn-in samples must be removed before performing Monte Carlo approximations.

3

Seismic AVO inversion at the Alvheim field

This chapter is devoted to describing the inverse problem at the Alvheim oil and gas field. First, an orientation of the situation and necessary notation are given. Then the choices regarding the prior distribution are presented. Lastly, seismic AVO data, well-log data, and the likelihood functions are described. The information in this chapter is based on the article of Spremić et al. (2024). It is very similar to the information provided and the notation given in the specialization project by Auestad (2023), which also was based on the article of Spremić et al. (2024).

3.1 The Alvheim field

The Alvheim field is situated in the North Sea off the Norwegian west coast. Figure 3.1 shows the location of the field. It is a producing oil and gas field. As stated in Spremić et al. (2024), conducting seismic reservoir characterization is challenging as the distribution of lithology and fluids is complex. Further details regarding the geology and lithology of the Alvheim field can be found in for example (Rimstad et al., 2012b).

Figure 3.1: Location of the Alvheim field in the North Sea [1].

Seismic AVO inversion is performed at the top-reservoir in the Alvheim field. Following the framework of Spremić et al. (2024), the horizontal axis represents the inline direction and the vertical axis represents

[1] Map from : https://factmaps.npd.no. Accessed 13.12.23.

the crossline direction at the top-reservoir. The area is approximately 109.3 square kilometres, roughly 12.35 km in the inline direction and 8.85 km in the crossline direction. Seismic AVO data are available at N evenly spaced locations in the field, denoted by l, $l = 1, 2, ..., N$. Also available at each location, l, is depth, calculated from measured travel times. The depth of the top-reservoir in the field is shown in Figure 3.2. Already drilled wells in the area are marked as red and violet circles. Red circles mark wells where primarily gas was found. Mainly oil was retrieved in the well marked by the violet circle. As shown in Figure 3.2, all four wells are located at relatively shallow locations.

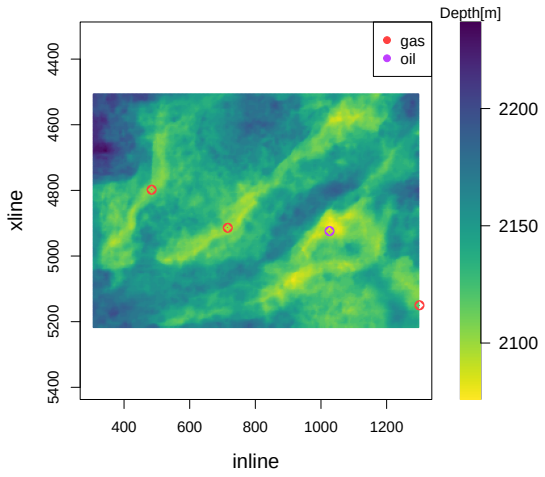


Figure 3.2: The depth of the top-reservoir of the Alvheim field. Four wells are marked by coloured circles, where the colour indicates the majority type of hydrocarbons found at that well. The wells are located at (inline,crossline)-locations $(484, 4798)$, $(716, 4914)$, $(1026, 4924)$ and $(1300, 5150)$.

The variable of interest, $\boldsymbol{x}$, consists of the transformed reservoir variables. That is $\boldsymbol{x}^l = [x_g^l, x_o^l, x_c^l]$, where x_g^l, x_o^l and x_c^l are transformed values for gas, oil and clay respectively at location l. Figure 3.3 illustrates the variable $\boldsymbol{x} = [\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^N]$ discretized on a grid, where the location l correspond to the location where the seismic AVO data and depth measurement is collected. The area consists of 248 data points in the inline direction and 178 data points in the crossline direction. This leaves $N = 44,144$ equally spaced discrete locations with approximately 50 meters apart.

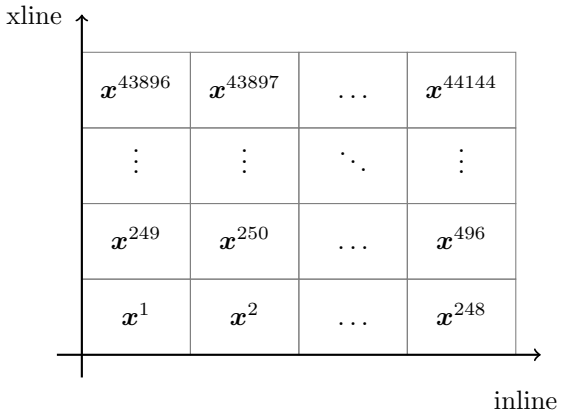


Figure 3.3: Spatial representation of the random variable $\boldsymbol{x} \in \mathbb{R}^N$ with $N = 44,144$ in the inline,crossline coordinate system.

In this thesis, the random variables of interest $\boldsymbol{x}_g$, $\boldsymbol{x}_o$ and $\boldsymbol{x}_c$ are represented as vectors

$$\boldsymbol{x}_g = [x_g^1, x_g^2, \ldots, x_g^N], \tag{3.1}$$

$$\boldsymbol{x}_o = [x_o^1, x_o^2, \ldots, x_o^N], \tag{3.2}$$

and

$$\boldsymbol{x}_\mathrm{c} = [x_\mathrm{c}^1, x_c^2, \ldots, x_c^N], \tag{3.3}$$

with $x_g^l, x_o^l, x_c^l \in \langle -\infty, \infty \rangle$. Motivated by insight gained from the Alvheim field, Spremić et al. (2024) chose to model these variables as continuous variables because it allows for sand-clay mixtures and partial saturations in the sample space and avoids unrealistically strong classification tendencies and accuracies. Following the notation of Spremić et al. (2024), logistic transformations,

$$S_g^l(\boldsymbol{x}^l) = \frac{e^{x_g^l}}{1 + e^{x_g^l} + e^{x_o^l}} \tag{3.4}$$

and

$$S_o^l(\boldsymbol{x}^l) = \frac{e^{x_o^l}}{1 + e^{x_g^l} + e^{x_o^l}} \tag{3.5}$$

are applied to x_g^l and x_o^l at each location l. Brine is used for reference saturation. Hence

$$S_b^l(\boldsymbol{x}^l) = \frac{1}{1 + e^{x_g^l} + e^{x_o^l}} \tag{3.6}$$

such that $S_g^l + S_o^l + S_b^l = 1$ for all locations $l = 1, ..., N$. For the clay content, the logistic transformation

$$V_c^l(\boldsymbol{x}^l) = \frac{e^{x_c^l}}{1 + e^{x_c^l}}, \tag{3.7}$$

is applied at each location l, such that $V_c^l \in \langle 0, 1 \rangle$. For efficient calculations, the algorithms work with the variables x_g^l, x_o^l, x_c^l instead of the modelled saturations S_g^l and S_o^l and clay content V_c^l.

3.2 Seismic AVO inversion

The posterior, $\pi(\boldsymbol{x})$, in the Alvheim case consists of two likelihood functions and a prior:

$$\pi(\boldsymbol{x}) = \text{const} \cdot f(\boldsymbol{y}_s|\boldsymbol{x}) \cdot f(\boldsymbol{y}_w|\boldsymbol{x}) \cdot f(\boldsymbol{x}). \tag{3.8}$$

The likelihood $f(\boldsymbol{y}_s|\boldsymbol{x})$ is the contribution from the seismic AVO data and $f(\boldsymbol{y}_w|\boldsymbol{x})$ is the contribution from the well-log data. As in Chapter 2, $f(\boldsymbol{x})$ denotes the prior. It follows an elaboration of the prior, the likelihoods and the data. The gradient of the prior and likelihoods are also presented. The gradients are used by the MALA presented in Chapter 2.

3.2.1 The prior

As briefly mentioned in Chapter 2, the prior used in this thesis is the same as in Spremić et al. (2024). This is an informed type of prior, constructed off the well-log data and expert knowledge of geological trends, spatial smoothness and fluid sorting (Spremić et al., 2024). Depth, shown in Figure 3.2 is incorporated into the model by letting the variables $\boldsymbol{x}_g$, $\boldsymbol{x}_o$ and $\boldsymbol{x}_c$ depend on the depth (Spremić et al., 2024). The expected values of the prior, $\boldsymbol{\mu}_g$, $\boldsymbol{\mu}_g$ and $\boldsymbol{\mu}_c$, represent depth trends estimated from well-log data. As

described in (Spremić et al., 2024), $\boldsymbol{x}_g$, $\boldsymbol{x}_o$ and $\boldsymbol{x}_c$ are assumed independent Gaussian random fields. That is,

$$\boldsymbol{x}_g \sim \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g), \tag{3.9}$$

$$\boldsymbol{x}_o \sim \mathcal{N}(\boldsymbol{\mu}_o, \boldsymbol{\Sigma}_o), \tag{3.10}$$

and

$$\boldsymbol{x}_c \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c). \tag{3.11}$$

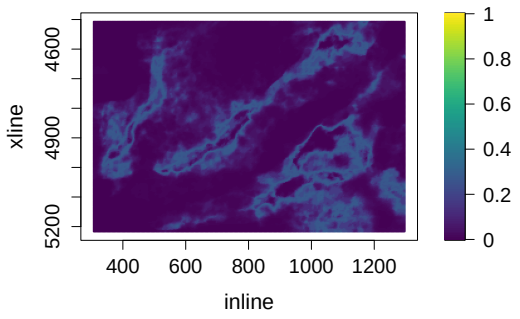
Hence the log prior is

$$\log(f(\boldsymbol{x})) = \text{const} - \frac{1}{2}(\boldsymbol{x}_g - \boldsymbol{\mu}_g)^T \boldsymbol{\Sigma}_g^{-1}(\boldsymbol{x}_g - \boldsymbol{\mu}_g) - \frac{1}{2}(\boldsymbol{x}_o - \boldsymbol{\mu}_o)^T \boldsymbol{\Sigma}_o^{-1}(\boldsymbol{x}_o - \boldsymbol{\mu}_o) - \frac{1}{2}(\boldsymbol{x}_c - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}_c - \boldsymbol{\mu}_c), \tag{3.12}$$

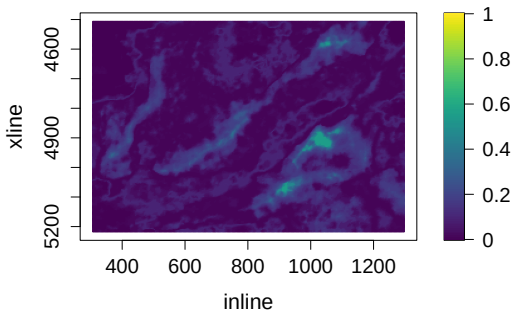
with gradient

$$\boldsymbol{\nabla} \log\left(f(\boldsymbol{x})\right) = -\left[\boldsymbol{\Sigma}_g^{-1}(\boldsymbol{x}_g - \boldsymbol{\mu}_g), \boldsymbol{\Sigma}_o^{-1}(\boldsymbol{x}_o - \boldsymbol{\mu}_o), \boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}_c - \boldsymbol{\mu}_c)\right]. \tag{3.13}$$

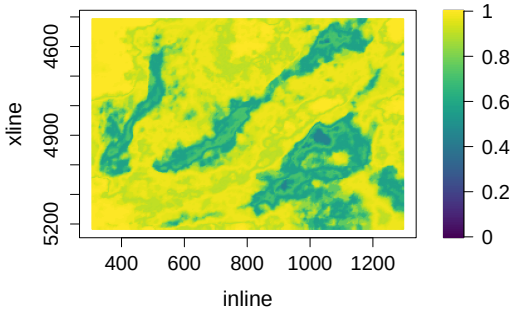
Logistic transformations, given in equations (3.4), (3.5) and (3.6), applied to the prior means $\boldsymbol{\mu}_g$ and $\boldsymbol{\mu}_o$ are shown in Figure 3.4. Generally, Figure 3.4a and 3.4b show that the prior mean gas and oil saturation are higher in shallow areas. The prior mean saturation of oil is also relatively high around the area of the well marked by a violet circle in Figure 3.2, whereas the prior mean saturation of gas is low in this area. In the deepest areas, such as the top left corner, both saturations are low. The mean saturation of oil is slightly higher in the relatively deep area in the top middle. Between the middle ridge and the high area where the well that contained most oil is located, the prior mean saturation for oil is higher than the prior mean saturation of gas. Overall, in the areas where both the mean saturations are relatively high, the gas saturation is slightly higher.



(a) Prior mean saturation of gas.



(b) Prior mean saturation of oil.



(c) Prior mean saturation of brine.

Figure 3.4: Saturation of the prior means $\boldsymbol{\mu}_g$ and $\boldsymbol{\mu}_g$. Brine is used for reference saturation.

Figure 3.5 shows the prior mean clay content. That is the logistic transformation, given in equation (3.7), of $\boldsymbol{\mu}_c$. The prior mean clay content is generally very low, however, it is very high in the deep area at the top left corner.

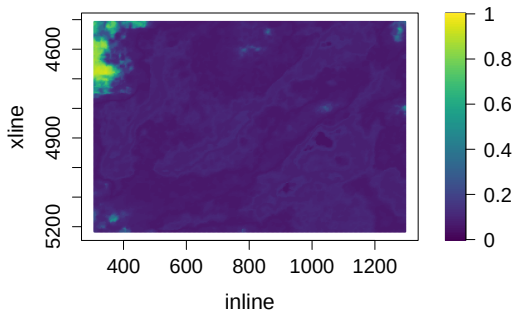


Figure 3.5: Prior mean of the clay content.

Spatial correlation between the variables $\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^N$ is included as a Gaussian correlation function with an effective correlation length of 15 grid cells (Spremić et al., 2024). This means that the element at row l and column k of the covariance matrix $\boldsymbol{\Sigma}_j$, $j = g, o, c$, is

$$\sigma_j^2 e^{-\frac{1}{2}\left(\frac{d_{lk}}{15}\right)^2}, \tag{3.14}$$

with standard deviations $\sigma_g = 2.83$, $\sigma_o = 2.68$ and $\sigma_c = 1.64$. Here d_{lk} is the euclidean distance between the location of $\boldsymbol{x}^l$ and the location of $\boldsymbol{x}^k$,

$$d_{lk} = \sqrt{(e^l - e^k)^2 + (n^l - n^k)^2}, \tag{3.15}$$

where east and north coordinates are

$e^l = (l-1)$ modulo 178,

$e^k = (k-1)$ modulo 178,

$n^l = (l-1)$ integer division 178,

$n^k = (k-1)$ integer division 178.

The covariance matrices $\boldsymbol{\Sigma}_o$, $\boldsymbol{\Sigma}_o$ and $\boldsymbol{\Sigma}_o$ are each of size $N \times N$. Fast Fourier transform (FFT) will be used to compute the matrix multiplications in equation (3.12) (Gray, 2006).

3.2.2 The data and likelihood models

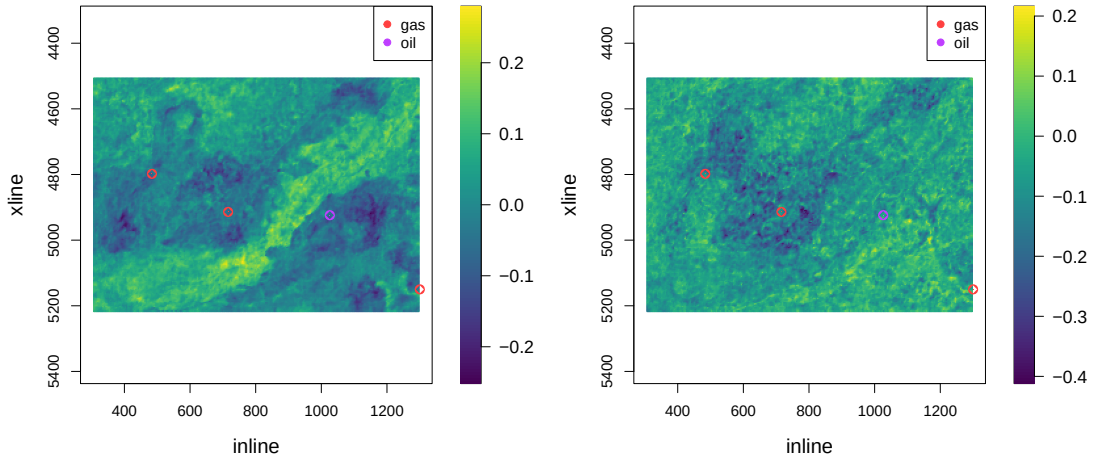
There are two sorts of data available. Seismic AVO data is denoted $\boldsymbol{y}_s$ and well-log data from the four wells marked as red and violet circles in Figure 3.2 is denoted by $\boldsymbol{y}_w$. The data are collectively denoted as

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_s \\ \boldsymbol{y}_w \end{bmatrix}. \tag{3.16}$$

As there are two types of data, there are also two sources of contribution to the likelihood, $f(\boldsymbol{y}_s|\boldsymbol{x})$ and $f(\boldsymbol{y}_w|\boldsymbol{x})$, which will be described in turn.

Seismic AVO data

The seismic AVO data consists of the measurements zero-offset intercept (R_0) and AVO gradient (G) from the top-reservoir. The measurements, shown in Figure 3.6, are available at each location l. A description of the processing procedure for making these data appropriate for quantitative analysis is given in Rimstad et al., 2012a.



(a) Zero-offset intercept (R_0).

(b) AVO gradient (G).

Figure 3.6: Presentation of the seismic AVO data in the crossline,inline coordinate system.

Seismic AVO data at location l is denoted $\boldsymbol{y}_s^l = \left[R_0^l, G^l\right]$. All $2N$ seismic AVO measurements are collected in the vector

$$\boldsymbol{y}_s = \left[\boldsymbol{y}_s^1, \ldots, \boldsymbol{y}_s^N\right]^T = \left[R_0^1, G^1, R_0^2, G^2, ..., R_0^N, G^N\right]^T. \tag{3.17}$$

Following Spremić et al. (2024), a relationship on the form

$$\boldsymbol{y}_s^l = \boldsymbol{h}(\boldsymbol{x}^l) + \boldsymbol{\omega}, \quad \boldsymbol{\omega} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Omega}_0), \tag{3.18}$$

with

$$\boldsymbol{\Omega}_0 = \begin{bmatrix} \text{Var}[R_0] & \text{Cov}[R_0, G] \\ \text{Cov}[R_0, G] & \text{Var}[G] \end{bmatrix} = \begin{bmatrix} 0.003 & -0.6\sqrt{\text{Var}[R_0]\text{Var}[G]} \\ -0.6\sqrt{\text{Var}[R_0]\text{Var}[G]} & 0.03 \end{bmatrix}, \tag{3.19}$$

is assumed. The non-linear functional relationship,

$$\boldsymbol{h}(\boldsymbol{x}) = \left[h_{R_0^1}, h_{G^1}, h_{R_0^2}, h_{G^2}, ..., h_{R_0^N}, h_{G^N}\right], \tag{3.20}$$

builds on rock physics relations between the seismic AVO data and the variables of interest $\boldsymbol{x}_g$, $\boldsymbol{x}_o$ and $\boldsymbol{x}_c$ (Spremić et al., 2024). Depth is also an input variable to the forward model. In this thesis, this function is treated as a black box. The function uses a relatively long time to map a sample $\boldsymbol{x}^l$ and the corresponding depth to $[h_{R_0^l}, h_{G^l}]$. In the next chapter, an approximation to the forward model is described, to reduce the overall computation time of the MCMC algorithm. The seismic AVO log-likelihood is

$$\log\left(f(\boldsymbol{y}_s|\boldsymbol{x})\right) = \text{const} - \frac{1}{2}\left(\boldsymbol{y}_s - \boldsymbol{h}(\boldsymbol{x})\right)^T \boldsymbol{\Omega}^{-1}\left(\boldsymbol{y}_s - \boldsymbol{h}(\boldsymbol{x})\right), \tag{3.21}$$

where

$$\boldsymbol{\Omega} = \begin{bmatrix} [\boldsymbol{\Omega}_0] & 0 & \dots & 0 & 0 \\ 0 & [\boldsymbol{\Omega}_0] & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & [\boldsymbol{\Omega}_0] & 0 \\ 0 & 0 & \dots & 0 & [\boldsymbol{\Omega}_0] \end{bmatrix}. \tag{3.22}$$

The gradient of $\log\left(f(\boldsymbol{y}_s|\boldsymbol{x})\right)$ is

$$\left(\boldsymbol{\Omega}^{-1}\left(\boldsymbol{y}_s - \boldsymbol{h}(\boldsymbol{x})\right)\right)^T \boldsymbol{\nabla}\boldsymbol{h}(\boldsymbol{x}). \tag{3.23}$$

As there are 3 random variables at all N locations, there are essentially $3N$ variables $x_g^1, x_o^1, x_c^1, x_g^2, x_o^2, x_c^2, ..., x_g^N, x_o^N, x_c^N$. The gradient is thus also of length $3N$

$$\boldsymbol{\nabla} = \left[\frac{\partial}{\partial x_g^1}, \frac{\partial}{\partial x_o^1}, \frac{\partial}{\partial x_c^1}, \frac{\partial}{\partial x_g^2}, \frac{\partial}{\partial x_o^2}, \frac{\partial}{\partial x_c^2}, ..., \frac{\partial}{\partial x_g^N}, \frac{\partial}{\partial x_o^N}, \frac{\partial}{\partial x_c^N}\right], \tag{3.24}$$

and $\boldsymbol{\nabla}\boldsymbol{h}(\boldsymbol{x})$ has $2N$ rows and $3N$ columns:

$$\begin{bmatrix} \frac{\partial}{\partial x_g^1}h_{R_0^1} & \frac{\partial}{\partial x_o^1}h_{R_0^1} & \frac{\partial}{\partial x_c^1}h_{R_0^1} & \frac{\partial}{\partial x_g^2}h_{R_0^1} & \frac{\partial}{\partial x_o^2}h_{R_0^1} & \frac{\partial}{\partial x_c^2}h_{R_0^1} & \dots & \frac{\partial}{\partial x_g^N}h_{R_0^1} & \frac{\partial}{\partial x_o^N}h_{R_0^1} & \frac{\partial}{\partial x_c^N}h_{R_0^1} \\ \frac{\partial}{\partial x_g^1}h_{G^1} & \frac{\partial}{\partial x_o^1}h_{G^1} & \frac{\partial}{\partial x_c^1}h_{G^1} & \frac{\partial}{\partial x_g^2}h_{G^1} & \frac{\partial}{\partial x_o^2}h_{G^1} & \frac{\partial}{\partial x_c^2}h_{G^1} & \dots & \frac{\partial}{\partial x_g^2}h_{G^1} & \frac{\partial}{\partial x_o^N}h_{G^1} & \frac{\partial}{\partial x_c^N}h_{G^1} \\ \frac{\partial}{\partial x_g^1}h_{R_0^2} & \frac{\partial}{\partial x_o^1}h_{R_0^2} & \frac{\partial}{\partial x_c^1}h_{R_0^2} & \frac{\partial}{\partial x_g^2}h_{R_0^2} & \frac{\partial}{\partial x_o^2}h_{R_0^2} & \frac{\partial}{\partial x_c^2}h_{R_0^2} & \dots & \frac{\partial}{\partial x_g^N}h_{R_0^2} & \frac{\partial}{\partial x_o^N}h_{R_0^2} & \frac{\partial}{\partial x_c^N}h_{R_0^2} \\ \frac{\partial}{\partial x_g^1}h_{G^2} & \frac{\partial}{\partial x_o^1}h_{G^2} & \frac{\partial}{\partial x_c^1}h_{G^2} & \frac{\partial}{\partial x_g^2}h_{G^2} & \frac{\partial}{\partial x_o^2}h_{G^2} & \frac{\partial}{\partial x_c^2}h_{G^2} & \dots & \frac{\partial}{\partial x_g^N}h_{G^2} & \frac{\partial}{\partial x_o^N}h_{G^2} & \frac{\partial}{\partial x_c^N}h_{G^2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial}{\partial x_g^1}h_{R_0^N} & \frac{\partial}{\partial x_o^1}h_{R_0^N} & \frac{\partial}{\partial x_c^1}h_{R_0^N} & \frac{\partial}{\partial x_g^2}h_{R_0^N} & \frac{\partial}{\partial x_o^2}h_{R_0^N} & \frac{\partial}{\partial x_c^2}h_{R_0^N} & \dots & \frac{\partial}{\partial x_g^N}h_{R_0^N} & \frac{\partial}{\partial x_o^N}h_{R_0^N} & \frac{\partial}{\partial x_c^N}h_{R_0^N} \\ \frac{\partial}{\partial x_g^1}h_{G^N} & \frac{\partial}{\partial x_o^1}h_{G^N} & \frac{\partial}{\partial x_c^1}h_{G^N} & \frac{\partial}{\partial x_g^2}h_{G^N} & \frac{\partial}{\partial x_o^2}{}^2h_{G^N} & \frac{\partial}{\partial x_c^2}h_{G^N} & \dots & \frac{\partial}{\partial x_g^N}h_{G^N} & \frac{\partial}{\partial x_o^N}h_{G^N} & \frac{\partial}{\partial x_c^N}h_{G^N} \end{bmatrix} \tag{3.25}$$

$$= \begin{bmatrix} \frac{\partial}{\partial x_g^1}h_{R_0^1} & \frac{\partial}{\partial x_o^1}h_{R_0^1} & \frac{\partial}{\partial x_c^1}h_{R_0^1} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \frac{\partial}{\partial x_g^1}h_{G^1} & \frac{\partial}{\partial x_o^1}h_{G^1} & \frac{\partial}{\partial x_c^1}h_{G^1} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial}{\partial x_g^2}h_{R_0^2} & \frac{\partial}{\partial x_o^2}h_{R_0^2} & \frac{\partial}{\partial x_c^2}h_{R_0^2} & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial}{\partial x_g^2}h_{G^2} & \frac{\partial}{\partial x_o^2}h_{G^2} & \frac{\partial}{\partial x_c^2}h_{G^2} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \frac{\partial}{\partial x_g^N}h_{R_0^N} & \frac{\partial}{\partial x_o^N}h_{R_0^N} & \frac{\partial}{\partial x_c^N}h_{R_0^N} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \frac{\partial}{\partial x_g^N}h_{G^N} & \frac{\partial}{\partial x_o^N}h_{G^N} & \frac{\partial}{\partial x_c^N}h_{G^N} \end{bmatrix}. \tag{3.26}$$

The elements of matrix (3.26) are not known analytically. The numerical gradients are available, but computationally expensive to calculate. This thesis proposes using an approximation to the numerical gradient, which will be denoted $\boldsymbol{\nabla}\boldsymbol{h}(\boldsymbol{x})$. This is discussed in greater detail in the next chapter.

The well-log data

In addition to being used in the construction of the prior and forward model, the well-log data is incorporated into the expression for the posterior given in equation (3.8) as a likelihood function. Let l_{well} denote the set of the locations of the four wells, marked by coloured circles in Figure 3.2. The likelihood $f(\boldsymbol{y}_w|\boldsymbol{x})$ is constructed to impose values of $\boldsymbol{x}_g$ and $\boldsymbol{x}_o$ that agree with the type of hydrocarbons found in the four wells. The log-likelihood of the well-log data is

$$\log\left(f(\boldsymbol{y}_w|\boldsymbol{x})\right) = \text{const} - \frac{1}{2\sigma^2}\sum_{l\in l_{\text{well}}}\left(y^l_{w,g} - x^l_g\right)^2 - \frac{1}{2\sigma^2}\sum_{l\in l_{\text{well}}}\left(y^l_{w,o} - x^l_o\right)^2, \tag{3.27}$$

where σ^2 is set to 0.1^2. At the locations of the wells that primarily contained gas, $y^l_{w,g}$ is set to the relatively high value 4, while $y^l_{w,o}$ is set to the relatively low value -4. Conversely, at the location of the well that mainly contained oil, $y^l_{w,g} = -4$ and $y^l_{w,o} = 4$. The gradient of the log-likelihood is zero for all locations $l \notin l_{\text{well}}$. At each location $l \in l_{\text{well}}$, the partial derivative of $\log\left(f(\boldsymbol{y}_w|\boldsymbol{x})\right)$ with respect to x^l_g is

$$\frac{y^l_{w,g} - x^l_g}{\sigma^2}, \tag{3.28}$$

the partial derivative of $\log\left(f(\boldsymbol{y}_w|\boldsymbol{x})\right)$ with respect to x^l_o is

$$\frac{y^l_{w,o} - x^l_o}{\sigma^2}, \tag{3.29}$$

and the partial derivative of $\log\left(f(\boldsymbol{y}_w|\boldsymbol{x})\right)$ with respect to x^l_c is 0.

4

Approximation of the forward model

Evaluating the forward model in the Alvheim case from Chapter 3 at a value $\boldsymbol{x}$ takes a relatively long time. At each iteration in the MCMC algorithm, $\boldsymbol{h}(\boldsymbol{x}^p)$ needs to be calculated to find the likelihood at the proposed sample $\boldsymbol{x}^p$. In this chapter, substitutes $\widehat{\boldsymbol{h}}$ are explored with the goal of speeding up the MCMC algorithm. The goal is also that the approximation has a small error, such that the MCMC samples obtained with $\widehat{\boldsymbol{h}}$ are similar to the MCMC samples from using $\boldsymbol{h}$. By sampling from the prior, a data set with the forward model as the response is created. Two supervised learning methods, MARS and NPKR, are attempted as surrogates for $\boldsymbol{h}$. In addition to finding a substitute for $\boldsymbol{h}$, an approximation to $\boldsymbol{\nabla h}$ is found. This is used in the MALA proposal distribution, described in Chapter 2.

There are numerous supervised learning techniques. At the beginning of this work, candidates for approximating the forward model included polynomial regression, random forest, MARS and NPKR. An advantage of MARS and NPKR is that they do not assume a specific functional form of the response. In polynomial regression, the degree of the model is typically set manually or found using cross-validation (James et al., 2021). This requires some insight into the function $\boldsymbol{h}$, however, this function is treated as a black box in this thesis, and non-parametric methods were therefore favoured. Another advantage of MARS and the NPKR model used in this thesis is that is possible to calculate the gradient of both models. These could possibly serve as substitutes for $\boldsymbol{\nabla h}$. A random forest does not produce a model which has a gradient. In this thesis, MARS and NPKR are tested for approximating the forward model as they have qualities suitable for this situation. However, other supervised learning methods, including polynomial regression and random forest, could be explored for approximating a non-linear and computationally inefficient forward model.

This chapter starts with presenting MARS and its gradient. Then the NPKR used in this thesis is presented, as well as its gradient. Afterwards, a 2D example is introduced to illustrate the properties of the models. The models are then used to approximate the forward model in the Alvheim case, and its gradient. Finally, a brief comment on how the error made by the approximation can be adjusted for is given.

There are some common notations in Section 4.3 and Section 4.4. In both the 2D example and the Alvheim case, a data set is used to train the models. In both cases, the number of data points in the training set is denoted by n and the type of covariate is denoted by j. The number of covariates is p. In the 2D example, the response is denoted by f. An observation in the data set is then $(\boldsymbol{x}_i, f_i)$ with $\boldsymbol{x}_i = [x_{i1}, x_{i2}, ..., x_{ip}]$. In Section 4.4, when approximating the forward model, the responses are denoted h_{R_0} and h_G.

4.1 Multivariate adaptive regression spline

A MARS model combines piecewise linear functions to approximate a response f. The theory in this section is based on the theory in Hastie et al. (2009). A MARS model consists of linear basis functions $(x_j - c_j)_+$ and $(c_j - x_j)_+$, where x_j is a covariate and c_j is an observed value of that covariate. The subscript $+$ means the positive part, hence

$$(x_j - c_j)_+ = \begin{cases} x_j - c_j, & \text{for } x_j > c_j, \\ 0, & \text{otherwise,} \end{cases} \tag{4.1}$$

and

$$(c_j - x_j)_+ = \begin{cases} c_j - x_j, & \text{for } c_j > x_j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

Figure 4.1 shows the functions $(x_j - 1)_+$ and $(1 - x_j)_+$ for $x_j \in [-2, 4]$. Hastie et al. (2009) call $(x_j - 1)_+$ and $(1 - x_j)_+$ a reflective pair.

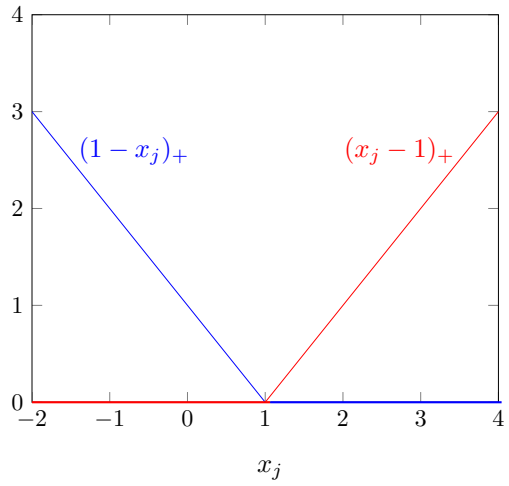


Figure 4.1: Reflective pair $(x_j - 1)_+$ and $(1 - x_j)_+$ for $x_j \in [-2, 4]$.

The set $\mathcal{C}$ consist of all functions $(x_j - c_j)_+$ and $(c_j - x_j)_+$ for every covariate x_j and observed value x_{ij} in the data set, that is

$$\mathcal{C} = \{(x_j - c_j)_+, (c_j - x_j)_+\}_{c_j \in \{x_{1j}, x_{2j}, \ldots x_{nj}\}}. \tag{4.3}$$

The MARS model is on the form

$$\widehat{f}(\boldsymbol{x}) = \beta_0 + \sum_{d=1}^{D} \beta_d g_d(\boldsymbol{x}), \tag{4.4}$$

where $\beta_1, \beta_2, ..., \beta_D$ are coefficients jointly estimated by minimizing the residual sum of squares, and $g_d(\boldsymbol{x})$ are functions or products of functions from the set $\mathcal{C}$. Even though g_d only depends on one or more covariates it is considered a function on all of the covariates. The number of terms in the model is D.

Building a MARS model consists of two main steps, a forward model building procedure and a backward pruning procedure. In the forward step, functions $g_d(\boldsymbol{x})$ are chosen successively until a large, often overfitted model is obtained, which then is pruned to a smaller model. To go through these steps in detail, let $\mathcal{D}$ be the set of chosen combinations of basis functions for the model. Initially, $\mathcal{D} = \{g_0\}$, with $g_0 = 1$. Next, all functions in $\mathcal{D}$ are successfully multiplied by the pairs $\mathcal{C}$ and the term

$$\widehat{\beta}_{D+1} g_r(\boldsymbol{x})(x_j - c_j)_+ + \widehat{\beta}_{D+2} g_r(\boldsymbol{x})(c_j - x_j)_+, \quad g_r(x) \in \mathcal{D} \tag{4.5}$$

that produces the largest decrease in the training error is added to the model. This greedy step is repeated until a maximum number of terms is reached or some other stopping criterion is fulfilled.

After the stopping criterion is met, the model is pruned by removing the term that leads to the smallest increase in residual square error. This produces an estimated best model $\widetilde{f}_\lambda$, for each number of terms λ. For each $\widetilde{f}_\lambda$ the generalized cross-validation

$$GCV_f(\lambda) = \frac{\sum_{i=1}^{n}(f_i - \widetilde{f}_\lambda(\boldsymbol{x}_i))^2}{\left(1 - \frac{D(\lambda)}{n}\right)^2}, \tag{4.6}$$

is calculated, where f_i is the response at data point i. $\mathcal{D}(\lambda)$ is the effective number of parameters as described in Hastie et al. (2009). Finally, the model that minimises $GCV_h(\lambda)$ is chosen to be $\widehat{f}$.

Interactions between the covariates are allowed. However, $g_d(\boldsymbol{x})$ can only include first-order terms of one covariate. This is because higher-order powers in the response can be modelled by piecewise linear functions, and allowing for higher-order powers in the model can lead to unwanted increases or decreases near the boundaries of the feature space. Interactions are built up from products with terms already in the model. The idea is that a high-order interaction probably only exists if a factor of the interaction exists as a lower-order as well. It is possible to specify the order of interactions. By specifying no interactions, the MARS model, which is then an additive model, is easier to interpret. In this case, however, the response $\boldsymbol{h}$ is treated as a black box, and the interpretability of $\widehat{\boldsymbol{h}}$ is not important.

There are several advantages to MARS. For instance, the produced model is continuous and differentiable (Friedman & Roosen, 1995). Another advantage is that a MARS model does not assume a functional form between the data $\boldsymbol{x}$ and response f. In this way, a MARS model can model responses of arbitrary shape. This is especially important in high-dimensional data, as one is unable to visualise the relationship between the variables and the response. MARS is also good in high dimensions because it consists of functions that are zero over many parts of the feature space, such that parameters are only used where they are needed. Hastie et al. (2009) state that the use of other basis functions, such as polynomials, which would produce nonzero products almost everywhere, would not work as well in high dimensions. Another advantage of MARS is that the number of knots and the number of terms are automatically selected by the algorithm, which makes the method very easy to use.

In both the 2D example and when approximating the forward function in the Alvheim case, two MARS models are built. Both models use the same forward model building procedure. The `earth` package (https://cran.r-project.org/web/packages/earth/earth.pdf) in R is used to fit these models. Functionalities provided by the `earth` function are exploited. The parameter `degree` is set to the number of covariates, p, allowing interactions between all covariates. The maximum number of terms before the forward step terminates, `nk`, is set to 50. The parameter `thresh` determines how many terms are added in the forward pass by stopping the forward pass if adding a term changes R^2 by less than `thresh`. It is set very low to 10^{-9}, such that `nk` determines when the forward model procedure stops.

After the forward pass, the first MARS model is finished. This Mars model will be denoted by $\widehat{f}_{\text{MARS}_{\text{OF}}}$ in the 2D example and $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$ in the Alvheim case. The possibly overfitted model does not necessarily have 50 terms because `earth` automatically removes linearly dependent terms at the end of the forward pass. The motivation for trying a large, possibly overfitted, model, is that the derivative of the model might be closer to the derivative of the response because a larger MARS model has more cuts. However, an overfitted model generally leads to a larger test error (James et al., 2021). This is particularly an issue if there are few data points in the training data set. When approximating the forward function, the training data are sampled from the prior distribution, such that the amount of training data is unlimited.

The second MARS model, denoted by $\widehat{f}_{\text{MARS}}$ and later $\widehat{\boldsymbol{h}}_{\text{MARS}}$, has a subset of the larger MARS models's terms. The option for the backward pruning method, `pmethod`, is set to "exhaustive". This functionality uses the function `leaps` from the package `leaps` (https://cran.r-project.org/web/packages/leaps/leaps.pdf) which performs an exhaustive subset selection. The maximum number of terms in the final model is set to 25.

4.1.1 Gradient of multivariate adaptive regression spline

The gradient of a MARS model can be found by taking partial derivatives of the model in (4.4) directly

$$\frac{\partial}{\partial x_j} f(\boldsymbol{x}) = \frac{\partial}{\partial x_j} \left[\beta_0 + \sum_{d=1}^{D} \beta_d g_d(\boldsymbol{x}) \right] \tag{4.7}$$

$$= \sum_{d=1}^{D} \beta_d \frac{\partial}{\partial x_j} g_d(\boldsymbol{x}). \tag{4.8}$$

The partial derivative of $g_d(\boldsymbol{x})$ with respect to x_j is simple because $g_d(\boldsymbol{x})$ is just a product of functions from $\mathcal{C}$. The tedious part is finding out the form of $g_d(\boldsymbol{x})$ such that the gradient can be calculated. After finding the derivative of $g_d(\boldsymbol{x})$ this must be multiplied with the corresponding coefficient β_d and summed according to (4.8).

In order to find the partial derivative of $g_d(\boldsymbol{x})$ with respect to x_j, one needs to decide the form of $g_d(\boldsymbol{x})$. That includes which covariates are present, which of the reflective pairs $(x_j - c_j)_+$ or $(c_j - x_j)_+$ are present for each covariate and for each present covariate and check if the positive criterion is satisfied. Terms without x_j gives $\frac{\partial}{\partial x_j} g_d(\boldsymbol{x}) = 0$. Terms on the form $(x_j - c_j)_+$ gives

$$\frac{\partial}{\partial x_j} g_d(\boldsymbol{x}) = \begin{cases} 1, & x_j > c_j \\ 0, & \text{otherwise.} \end{cases} \tag{4.9}$$

On the other hand, terms on the form $(c_j - x_j)_+$ gives

$$\frac{\partial}{\partial x_j} g_d(\boldsymbol{x}) = \begin{cases} -1, & c_j > x_j \\ 0, & \text{otherwise} \end{cases} \tag{4.10}$$

Interactions terms are more complicated. First of all, the positive condition needs to be checked for all covariates present in the term. That is, depending on the function, the condition $x_j > c_j$ or $x_j < c_j$ must be checked for all covariates j in the term. If all the factors in the term are positive, the first factor of the partial derivative is either 1 or -1, depending on the form of the term. This is then multiplied by the other factors which are either $x_{j'} - c_{j'}$ or $c_{j'} - x_{j'}$, $j \neq j^{'}$. Checking conditions for non-interactions terms is done relatively fast by exploiting binary search. This is possible because cuts c_j can be sorted in advance. However, when g_d had many interactions, it was harder to check all the necessary conditions in a fast way in order to compute the partial derivative of g_d with respect to x_j.

4.2 Non-parametric kernel regression

NPKR approximates the response f at a new data point by a weighted average of the data points in the training set. There are numerous ways to determine these weights. In this thesis, the Nadaraya-Watson

estimator

$$\widehat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} w_i(\boldsymbol{x}) f_i \tag{4.11}$$

will be used with weights

$$w_i(\boldsymbol{x}) = \frac{K_{\boldsymbol{b}}(\boldsymbol{x} - \boldsymbol{x}_i)}{\sum_{l=1}^{n} K_{\boldsymbol{b}}(\boldsymbol{x} - \boldsymbol{x}_l)} \quad i = 1, 2, ..., n. \tag{4.12}$$

The function $K_{\boldsymbol{b}}(\boldsymbol{x} - \boldsymbol{x}_i)$ is a Gaussian kernel with fixed bandwidths for each predictor $\boldsymbol{b} = [b_1, b_2, ..., b_p]$. When using the Gaussian kernel, the bandwidths are the standard deviation of the density. The weight in equation (4.12) decrease with the distance between the new data point and the data points in the training set. For p covariates, $K_{\boldsymbol{b}}$ is chosen to be the product of p one dimensional Gaussian kernels:

$$K_{\boldsymbol{b}}(\boldsymbol{x} - \boldsymbol{x}_i) = \prod_{j=1}^{p} k_{b_j}(x_j - x_{ji}) \tag{4.13}$$

$$= \prod_{j=1}^{p} \frac{1}{b_j} k\left(\frac{x_j - x_{ji}}{b_j}\right) \tag{4.14}$$

$$= \prod_{j=1}^{p} \frac{1}{\sqrt{2\pi} b_j} e^{-\frac{(x_j - x_{ji})^2}{2b_j^2}}. \tag{4.15}$$

The Gaussian kernel is a popular non-compact kernel (Hastie et al., 2009). The function `npreg` from the package `np` (https://cran.r-project.org/web/packages/np/np.pdf) in R is used to fit the NPKR model. Racine and Li (2004) describe this model in detail.

The bias-variance tradeoff asserts itself when selecting the bandwidths. Selecting small bandwidths tends to lower the bias and increase the variance as the nearest points are valued a lot. On the other hand, selecting too large bandwidths leads to a high bias and low variance, since this tends to include values far away from the new data points in the estimate. For the NPKR model used in this thesis, the bandwidths are found using least squares cross-validation. This is the only predetermined property of an NPKR model because the weights need to be determined every time a prediction of a new data point is made. Much of the work is therefore done at evaluation time (Hastie et al., 2009).

There are advantages and disadvantages of using an NPKR model. An advantage of NPKR models is that they do not assume a functional form of the response, which as mentioned in the previous section, is important in high dimension. Moreover, averaging nearby data points is quite intuitive. The model described with equation (4.11) and (4.12) is a linear combination of smooth functions, and therefore the model is also smooth. A disadvantage of NPKR is that it can suffer from being biased at the boundaries (Hastie et al., 2009; Efron & Hastie, 2016). A weighted average near a boundary will use more data points away from the boundary compared to towards the boundary to predict the response near the boundary, because once the boundary is reached there are no more data points in that direction. Predictions near the boundaries are therefore biased towards data points away from the boundary. This can result in predicting an increasing function towards the boundaries, while in fact, the response is decreasing. This is more problematic in higher dimension because the fraction of data points near the boundary is larger (Hastie et al., 2009). To adjust for the boundary bias, one could fit a locally weighted linear regression instead of a locally weighted average (Hastie et al., 2009). However, this is not tested in this thesis. The logic of the boundary bias also applies inside the boundaries if the training data are spread very unevenly (Hastie et al., 2009). Suppose a new data points is to be predicted. If there are many more close data points in one direction compared to the opposite direction, the prediction will be biased towards the response values of the data points in the direction with the most data points. As the NPKR model is a weighted average of response values at the training data, predictions are dependent on which data points are in the training data. Evenly spaced training data could perhaps improve the NPKR model.

4.2.1 Gradient of a non-parametric kernel regression model

To find the derivative of an NPKR model, two approaches are tried. First analytical derivatives are calculated. Second, the R function `npreg` used to fit the model offer gradients at the training data, and the gradient at close training data points are averaged to approximate the gradient at a new data point. The two approaches are described in turn.

The analytical gradients are found by taking derivatives of the expression (4.11), which gives

$$\frac{\partial}{\partial x_j}\widehat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} f_i \frac{\partial}{\partial x_j} w_i(\boldsymbol{x}). \tag{4.16}$$

In order to find the partial derivative of the weight $w_i(\boldsymbol{x})$ given in equation (4.12) with respect to the predictor x_j, the quotient rule is applied:

$$\frac{\partial}{\partial x_j} w_i(\boldsymbol{x}) = \frac{\partial}{\partial x_j} \frac{K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_i)}{\sum_{l=1}^{n} K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_l)} \tag{4.17}$$

$$= \frac{\left(\sum_{l=1}^{n} K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_l)\right) \frac{\partial}{\partial x_j} K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_i) - K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_i) \frac{\partial}{\partial x_j}\left(\sum_{l=1}^{n} K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_l)\right)}{\left(\sum_{l=1}^{n} K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_l)\right)^2}. \tag{4.18}$$

The partial derivative of $K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_i)$ with respect to x_j is found directly by differentiating the expression in (4.15)

$$\frac{\partial}{\partial x_j} K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_i) = -\left(\frac{x_j - x_{ji}}{b_j^2}\right) K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_i). \tag{4.19}$$

which inserted into equation (4.18) gives

$$\frac{\partial}{\partial x_j} w_i(\boldsymbol{x}) = -\frac{\left(\frac{x_j - x_{ji}}{b_j^2}\right) K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_i)}{\sum_{l=1}^{n} K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_l)} + \frac{K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_i)\left(\sum_{l=1}^{n}\left(\frac{x_j - x_{jl}}{b_j^2}\right) K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_l)\right)}{\left(\sum_{l=1}^{n} K_{\boldsymbol{b}}(\boldsymbol{x}-\boldsymbol{x}_l)\right)^2} \tag{4.20}$$

All quantities in equation (4.20) are available, hence the partial derivative of the NPKR model with respect to covariate x_j is found by inserting equation (4.20) into equation (4.16).

The analytical gradient of the NPKR model given in equation (4.20) and equation (4.16), can be approximated by using available gradient values provided by the function `npreg` in R. The R function `npreg` gives the gradient value at the data points used to train the NPKR model. These values are used in the following way. The data points are first divided into K groups using K-means, that is they are divided into K groups such that the within-cluster sum of squares is minimized (Hartigan & Wong, 1979). A new data point is compared to the centres of the K clusters. The closest cluster has the minimum distance between the new data point and the cluster mean. Finally, the mean of the gradients at the closest neighbouring data points in the nearest cluster are used to approximate the gradient at the new data point.

The intention for trying to approximate the gradient with the K-means approach is to use the available gradients at the training data in a smart way. A natural way to use these gradients is to approximate the gradient at a new data point as the average of the gradient at close data points in the training data. Because these gradients will be evaluated many times in the MCMC algorithm MALA, introduced in Chapter 2, the computation time is important. K-means is a way to avoid searching through all the training data, to find the closest training data points.

There are however issues with this approach. The first issue is that K-means might not find the closest data points, for example, if the new data point is at the outskirts of a cluster. However, this is a price

to pay to reduce the computation time. Another problem with K-means is that the number of clusters is set manually (Hastie et al., 2009). For the procedure described above, the number of nearest neighbours in the cluster which are averaged, also needs to be set manually. In both the 2D example and the Alvheim case the number of clusters and the number of neighbours are found using a simple grid search of different combinations of these values. However, the parameter grid was relatively small and could have been done more thoroughly. In addition, there were multiple metrics for assessing the algorithms, such that the optimal parameters were ambiguous. In Section 4.4 the metrics correlation, mean square error (MSE) and computation time are considered, while in Section 4.3, only correlation and MSE were evaluated. To select the parameters, computation time was valued most, thereafter the correlation and MSE. An alternative, which was not tested in this thesis, is using hierarchical clustering instead of K-means clustering. The number of clusters does not need to be set manually in hierarchical clustering (Hastie et al., 2009). The last issue is whether or not the covariates should be scaled before dividing the data. By not scaling the data, the covariate on the highest scale essentially dominates the distance between the data points (James et al., 2021). However, by scaling the data points, the input data would need to be scaled at each iteration, which adds additional computation time. In the end, the data was not scaled.

4.3 2D example

The following toy example was created to examine MARS and NPKR models and their gradients. The surface

$$f(x_1, x_2) = -\sqrt{x_2}(x_1 - 2)^3 + \sqrt{x_1} - (x_2 - 2)^2 \tag{4.21}$$

illustrated in Figure 4.2 is chosen as the response. It is non-linear and has interactions between x_1 and x_2. The function in equation (4.21) has gradient

$$\boldsymbol{\nabla} f(x_1, x_2) = \left[\frac{1}{2\sqrt{x_1}} - 3(x_1 - 2)^2\sqrt{x_2}, -\frac{(x_1 - 2)^3}{2\sqrt{x_2}} - 2x_2 + 4 \right], \tag{4.22}$$

illustrated in Figure 4.3. In this section, both the function in equation (4.21) and its gradient in equation (4.22) are approximated by models described so far in this chapter.

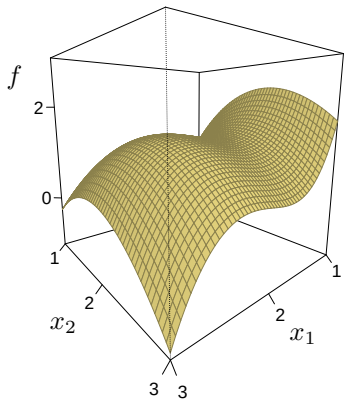
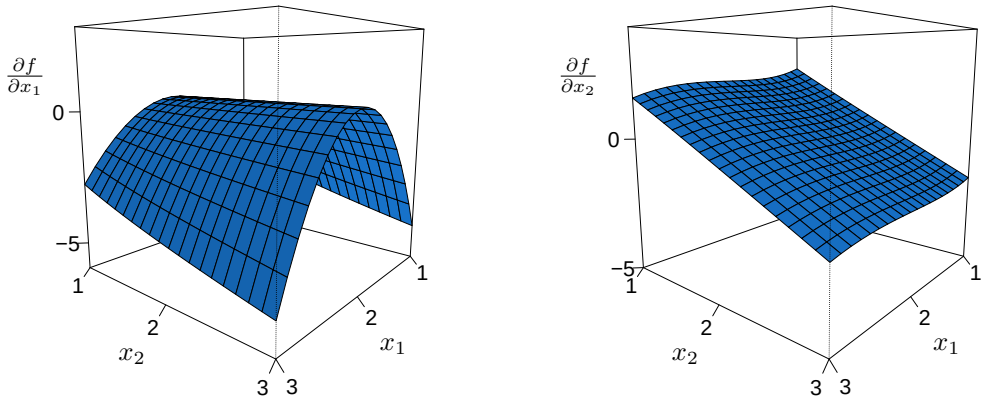


Figure 4.2: The surface $f(x_1, x_2) = -\sqrt{x_2}(x_1 - 2)^3 + \sqrt{x_1} - (x_2 - 2)^2$ on $x_1, x_2 \in [1, 3] \times [1, 3]$.



(a) Partial derivative of f with respect to x_1.

(b) Partial derivative of f with respect to x_2.

Figure 4.3: Partial derivative of the function in equation (4.21) with respect to x_1 (left) and x_2 (right) on $x_1, x_2 \in [1, 3] \times [1, 3]$.

Training data of x_1 and x_2 are sampled independently from a uniform distribution on the range $[1, 3]$. The number of data points is $n = 100$. The response is added to the training set by evaluating equation (4.21) at every data point in the training set. The three models $\widehat{f}_{\text{MARS}}$, $\widehat{f}_{\text{MARS}_{\text{OF}}}$ and $\widehat{f}_{\text{NPKR}}$, described in Section 4.1 and in Section 4.2, are fitted to the response. The model $\widehat{f}_{\text{MARS}}$ has 21 terms of which 6 are interaction terms, while the larger MARS model has 30 terms of which 8 are interaction terms. The bandwidths of $\widehat{f}_{\text{NPKR}}$ for x_1 and x_2 are approximately 0.087 and 0.106 respectively.

The test data are 100 evenly spaced data points on $x_1, x_2 \in [1, 3] \times [1, 3]$. Predictions of f are shown in Figure 4.4. The two MARS models look very similar to each other and also to the response f. The surface predicted by $\widehat{f}_{\text{NPKR}}$ looks worse than the predictions made by the MARS models. The surface in Figure 4.4c varies very locally, whereas the surfaces from the MARS models can capture the form of the response much better. The boundary bias of the $\widehat{f}_{\text{NPKR}}$ model is also visible in Figure 4.4c. At for example $x_1 \approx 1$ and $x_2 \to 3$ the prediction seems to be flat or even slightly increasing, whereas the response surface is actually decreasing as $x_2 \to 3$.

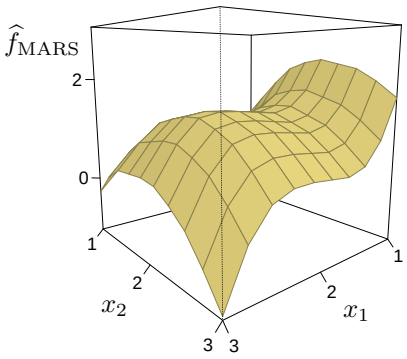
The accuracy of the predictions are examined by evaluating the sample correlation

$$\text{Corr}\left(f, \widehat{f}\right) = \frac{\sum_{i=1}^{n} \left(f - \bar{f}\right)\left(\widehat{f}_i - \bar{\widehat{f}}\right)}{\sqrt{\sum_{i=1}^{n} \left(f_i - \bar{f}\right)^2} \sqrt{\sum_{i=1}^{n} \left(\widehat{f}_i - \bar{\widehat{f}}\right)^2}}. \tag{4.23}$$

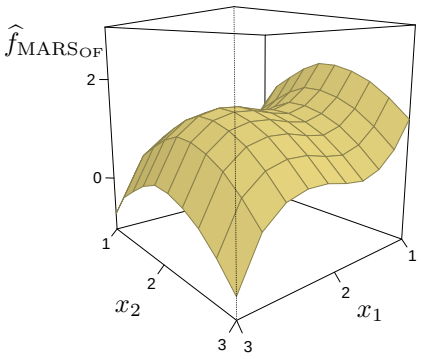
and MSE

$$\text{MSE}\left(f, \widehat{f}\right) = \frac{1}{n} \sum_{i=1}^{n} \left(f_i - \widehat{f}_i\right)^2. \tag{4.24}$$

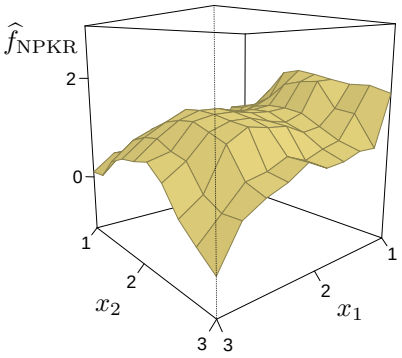
The mean of the predictions is denoted by $\bar{\widehat{f}}$, while the mean of the responses is denoted by $\bar{f}$. A high sample correlation is better, while the best MSE is the smallest. Computation time is not considered in this simple example but will be considered in the next section. The metrics are calculated on the test data.



(a) Fitted surface $\widehat{f}_{\text{MARS}}$.



(b) Fitted surface $\widehat{f}_{\text{MARS}_{\text{OF}}}$.



(c) Fitted surface $\widehat{f}_{\text{NPKR}}$.

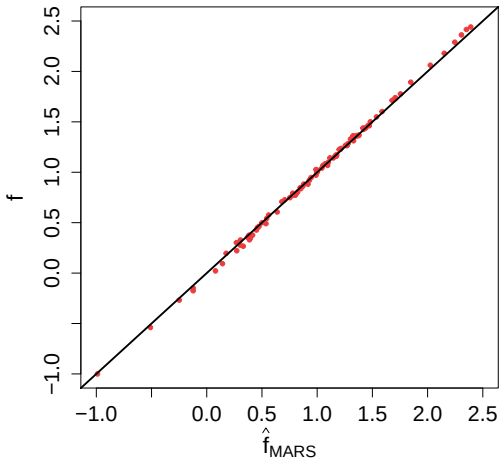
Figure 4.4: Approximations, $\widehat{f}_{\text{MARS}}$, $\widehat{f}_{\text{MARS}_{\text{OF}}}$ and $\widehat{f}_{\text{NPKR}}$ to the surface f shown in Figure 4.2. The response is predicted at 100 equally distanced values in $x_1, x_2 \in [1, 3] \times [1, 3]$.

Table 4.1 reports the correlation and MSE for the three models. The two MARS models have very similar correlation and MSE. They have a higher correlation and much lower MSE compared to the NPKR model, and are therefore better at predicting the response f. The large MARS model is slightly better than the pruned MARS model, however the large MARS model has 9 more terms. This suggests that $\widehat{f}_{\text{MARS}}$ is slightly more biased than $\widehat{f}_{\text{MARS}_{\text{OF}}}$ in this example, however, the differences in precision are minor.

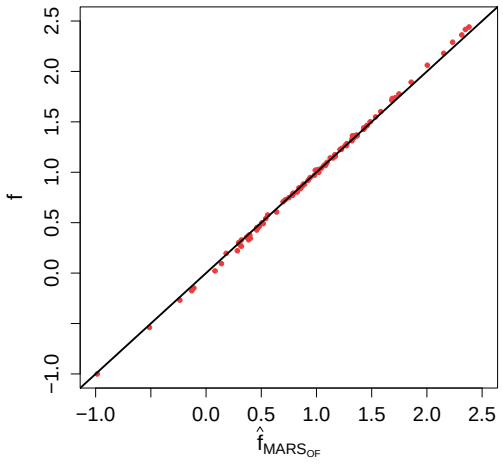
$\widehat{f}$	correlation	MSE
$\widehat{f}_{\text{MARS}}$	0.99951	0.00068
$\widehat{f}_{\text{MARS}_{\text{OF}}}$	0.99967	0.00061
$\widehat{f}_{\text{NPKR}}$	0.96368	0.03695

Table 4.1: Correlation and MSE for the three approximations $\widehat{f}_{\text{MARS}}$, $\widehat{f}_{\text{MARS}_{\text{OF}}}$ and $\widehat{f}_{\text{NPKR}}$ fitted to the function f in the 2D example.

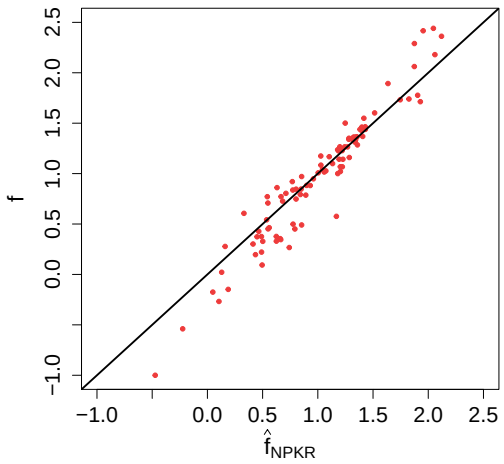
The true values f plotted against predicted values are shown in Figure 4.5a, 4.5b and 4.5c. Ideally, the red circles should align with the black line, because the black line is where the predicted value is equal to the true value. Figure 4.5 shows that the two MARS models perform very well and have no systematic errors. On the other hand, the NPKR model performs especially poorly for low and high function values. The error is also generally larger for the NPKR model compared to the MARS models as the red circles are further away from the black line.



(a) Red circles mark true function value f plotted against predicted value $\widehat{f}_{\text{MARS}}$ for test data marked.



(b) Red circles mark true function value f plotted against predicted value $\widehat{f}_{\text{MARS}_{\text{OF}}}$ for test data.



(c) Red circles mark true function value f plotted against predicted value $\widehat{f}_{\text{NPKR}}$ for test data.

Figure 4.5: Red circles mark true response value f plotted against predicted value $\widehat{f}$ for test data. The black line is $f = \widehat{f}$.

The gradients of the three models are tested for approximating the gradient of the response shown in Figure 4.3. How to extract the partial derivatives from the MARS and NPKR models is explained in

Sections 4.1 and 4.2 respectively. In addition, two MARS models are trained with one gradient component as a response each. That is the partial derivatives of f with respect to x_1 and x_2, from equation (4.22), are evaluated at the training values and added to the training set. These MARS models will be collectively denoted as $\text{MARS}_{\boldsymbol{\nabla}}$. For the K-means approximation to the gradient of $\widehat{f}_{\text{NPKR}}$, the number of clusters and number of closest data points to average was found by searching a simple parameter grid with the number of clusters $\in [2, ..., 9]$ and the number of closest neighbours to average $\in [2, .., 6]$. The combination that gave the lowest MSE and highest correlation was 5 clusters and 5 neighbours. Figure 4.6 shows the training data divided into 5 clusters using K-means.

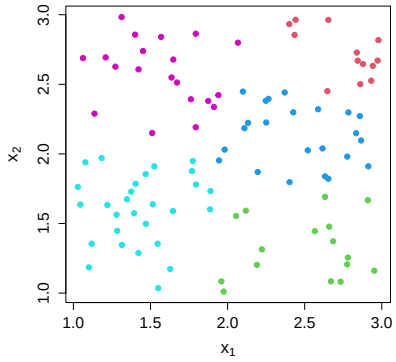


Figure 4.6: Training data in the 2D example divided into 5 clusters with K-means clustering.

The accuracy of the approximations to the gradient is the mean correlation

$$\overline{\text{Corr}\left(\boldsymbol{\nabla} f, \boldsymbol{\nabla}\widehat{f}\right)} = \frac{1}{2}\sum_{j=1}^{2} \text{Corr}\left(\frac{\partial f}{\partial x_j}, \frac{\widehat{\partial f}}{\partial x_j}\right), \tag{4.25}$$

and MSE

$$\overline{\text{MSE}\left(\boldsymbol{\nabla} f, \boldsymbol{\nabla}\widehat{f}\right)} = \frac{1}{2}\sum_{j=1}^{2} \text{MSE}\left(\frac{\partial f}{\partial x_j}, \frac{\widehat{\partial f}}{\partial x_j}\right), \tag{4.26}$$

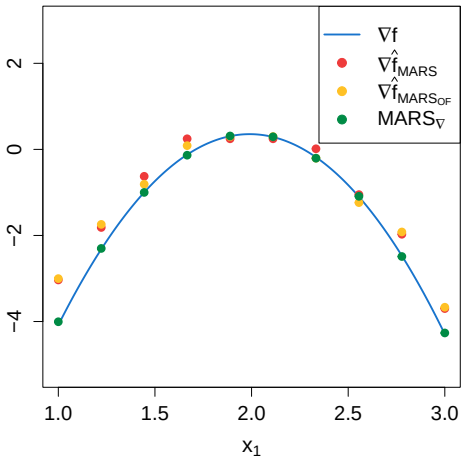
for the two covariates. The metrics are calculated on the test data.

For ease of visualisation, the approximations to the gradient are illustrated on slices of the surfaces in Figure 4.3. Figure 4.7 shows a slice of the partial derivative of f with respect to x_1 for fixed value $x_2 \approx 2.33$, while the slice of the partial derivative with respect to x_2 for fixed value $x_1 \approx 1.67$ is shown in Figure 4.8. The blue lines in these figures are the true partial derivatives, while coloured circles are predicted values.

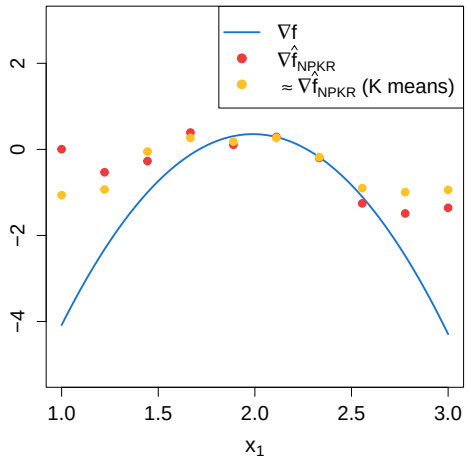
The plots in Figure 4.7a and 4.8a show that the $\text{MARS}_{\boldsymbol{\nabla}}$ is a very good approximation. It appears to predict the gradient at the fixed values perfectly. Moreover, these figures show that the approximations $\boldsymbol{\nabla}\widehat{f}_{\text{MARS}}$ and $\boldsymbol{\nabla}\widehat{f}_{\text{MARS}_{\text{OF}}}$ performs quite good, even though there are some gaps between the true gradient and the predictions, especially for the partial derivative with respect to x_1 shown in Figure 4.7a. The gradient of the large MARS model performs slightly better as the yellow circles are closer to the true gradient.

According to Figure 4.7b and Figure 4.8b, the gradient of model $\widehat{f}_{\text{NPKR}}$ and the K-means approximation to the gradient of $\widehat{f}_{\text{NPKR}}$ are poor at predicting the partial derivatives of f. From the figures, it is not clear if the gradient of the NPKR model or the K-means approach is worse. Both predictions behave

similarly. The predictions are good for values of x_1 close to 2 in Figure 4.7b, and values of x_2 close to 2 in Figure 4.8b. As discussed, NPKR can suffer from boundary bias. If $\widehat{f}_{\text{NPKR}}$ is very different to f at the boundaries then so are their gradients, which is clear from Figure 4.7b and Figure 4.8b.

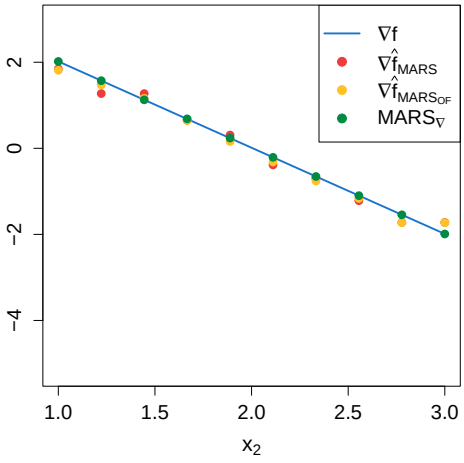


(a) The red and yellow circles are the partial derivative of $\widehat{f}_{\text{MARS}}$ and $\widehat{f}_{\text{MARS}_{\text{OF}}}$ respectively, with respect to x_1. The green circles are the predictions of MARS_{∇}.

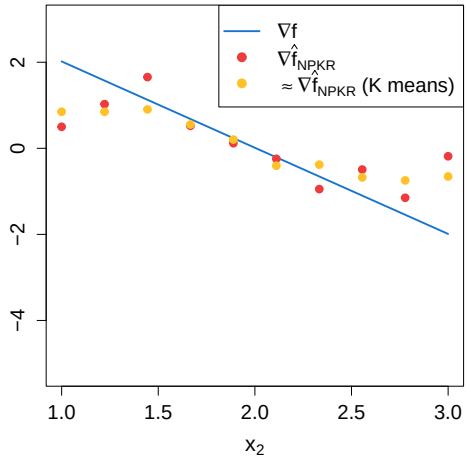


(b) The red circles are the partial derivative of the NPKR model. The yellow circles are the K-means approximation to the derivative of the NPKR model.

Figure 4.7: Approximation of partial derivative of f. The blue curve is the partial derivative of f with respect to x_1 at $x_2 \approx 2.33$. Coloured circles are approximations to the partial derivative.



(a) The red and yellow circles are the partial derivative of $\widehat{f}_{\text{MARS}}$ and $\widehat{f}_{\text{MARS}_{\text{OF}}}$ respectively, with respect to x_2. The green circles are the predictions of MARS_{∇}.



(b) The red circles are the partial derivative of the NPKR model. The yellow circles are the K-means approximation to the derivative of the NPKR model.

Figure 4.8: Approximation of partial derivative of f. The blue curve is the partial derivative of f with respect to x_2 at $x_1 \approx 1.67$. Coloured circles are approximations to the partial derivative.

The mean correlation and mean MSE of the approximations to the gradient are reported in Table 4.2. The MARS model fitted directly to the gradient components has the highest correlation and lowest MSE. This agrees with the good predictions in Figure 4.7a and Figure 4.8a. The gradients of the two MARS models have high correlations, however, lower than $\text{MARS}_{\boldsymbol{\nabla}}$. They also have much higher MSE compared to $\text{MARS}_{\boldsymbol{\nabla}}$. The large MARS model is slightly better than the smaller MARS model. This agrees with the motivation for creating the large MARS model; more cuts might lead to better prediction of the gradient. However, as presented in Table 4.1, $\widehat{f}_{\text{MARS}_{\text{OF}}}$ was slightly better at predicting f compared to $\widehat{f}_{\text{MARS}}$. This could be the reason for the gradient of the large MARS model predicting the gradient of the response better. The gradient of the NPKR model has a very low correlation compared to the other approximations. This approximation also had the highest MSE. Table 4.2 shows that the K-means approximation is better than the gradient of $\widehat{f}_{\text{NPKR}}$, though this is not clear from Figure 4.7b and Figure 4.8b. The correlation and MSE of the gradient of the NPKR model and the K-means approximation to the gradient of the NPKR model are much worse compared to the three other approximations.

$\widehat{\boldsymbol{\nabla} f}$	correlation	MSE
$\text{MARS}_{\boldsymbol{\nabla}}$	0.9999	0.0052
$\boldsymbol{\nabla}\ \widehat{f}_{\text{MARS}}$	0.9726	2.0117
$\boldsymbol{\nabla}\ \widehat{f}_{\text{MARS}_{\text{OF}}}$	0.9758	1.8087
$\boldsymbol{\nabla}\ \widehat{f}_{\text{NPKR}}$	0.4786	19.8937
$\approx \boldsymbol{\nabla}\ \widehat{f}_{\text{NPKR}}$ (K-means)	0.8850	13.4772

Table 4.2: Mean correlation and MSE of approximations to the partial derivative of f with respect to x_1 and x_2.

4.4 Approximating the forward model in the Alvheim case

To approximate the forward model in the Alvheim case, a training data set is created. In this section, covariates are denoted by letters instead of numbers. The data set has $20,000$ observations of covariates x_g, x_o, x_c and x_d. The value of data point i at covariate j is denoted by x_{ij}, where $j = g, o, c, d$ corresponds to gas, oil, clay and depth respectively. Data point i is sampled as follows. First, a value for x_{id} is sampled from a uniform distribution on the interval between the minimum and maximum depth at the top reservoir of the Alvheim field shown in Figure 3.2. Next x_{id} is used to find the expected value of the prior and draw x_{ig}, x_{io} and x_{ic} from the prior distribution. The test data are created in the same manner as the training data. It has 44144 test data points, which is the size of one sample $\boldsymbol{x}$ at the Alvheim field.

Recall from Chapter 3 that there are two types of seismic AVO data, the zero offset (R_0) and the seismic gradient (G). That is, $\boldsymbol{h} : \mathbb{R}^4 \to \mathbb{R}^2$. To approximate $\boldsymbol{h}$, two models are created. A model for the zero offset, $\widehat{h_{R_0}}$, and one for the seismic gradient $\widehat{h_G}$ such that $\widehat{\boldsymbol{h}}$ collectively denotes the two models $\widehat{h_{R_0}}$ and $\widehat{h_G}$ with $\widehat{h_{R_0}} : \mathbb{R}^4 \to \mathbb{R}$ and $\widehat{h_G} : \mathbb{R}^4 \to \mathbb{R}$.

As candidates for approximating $\boldsymbol{h}$, the MARS models $\widehat{\boldsymbol{h}}_{\text{MARS}}$ and $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$ are trained on the $20,000$ data points. The procedure of how these models are made is described in Section 4.1. The large MARS models for the two AVO properties h_{R_0} and h_G have 39 and 40 terms of which 26 and 24 are interaction

terms respectively. The pruned MARS model for h_{R_0} has 25 terms of which 17 are interaction terms, while the pruned MARS model for h_G has 25 of which 13 are interaction terms. In addition the models $\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$ and $\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$ are trained as described in Section 4.2, with 1000 and 4000 of the training data respectively. They are trained on subsets of the training data set because of computation time. As the NPKR models are weighted averages of the training data, the number of training data affects the prediction time. The subsets of the training data are random subsets, however one could try to select these points in a more optimal way to improve the NPKR models as discussed in Section 4.2.

Precision metrics, such as correlation and MSE are the mean of these metrics for the two models. That is, the correlation between a model $\widehat{\boldsymbol{h}}$ and the response $\boldsymbol{h}$ is the mean of Corr $\left(h_{R_0}, \widehat{h_{R_0}}\right)$ and Corr $\left(h_G, \widehat{h_G}\right)$ calculated using equation (4.23). Similarly, the MSE is the mean of MSE $\left(h_{R_0}, \widehat{h_{R_0}}\right)$ and MSE $\left(h_G, \widehat{h_G}\right)$ from equation (4.24). Moving on $\widehat{\boldsymbol{h}}_{\text{MARS}}$, $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$, $\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$ and $\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$ are referred to as one model, although they are two models, as described in the second paragraph of this section.

The computation time for predicting the test data, correlation and MSE are reported in Table 4.3. The computation time reported in the table is the average of predicting the test data 50 times. For comparison, the computation time of the true forward model used on average 2.059 seconds to compute $\boldsymbol{h}(\boldsymbol{x})$. The fastest model was $\widehat{\boldsymbol{h}}_{\text{MARS}}$, which was on average ≈ 32 times faster than the true forward model. The larger MARS model was also very fast, although slower than the smaller MARS model. The larger MARS model is marginally better than $\widehat{\boldsymbol{h}}_{\text{MARS}}$ when considering the correlation and MSE, however, it is much more complex. By setting the maximum number of terms in the MARS model after pruning to a lower number, the computation time is reduced. This is at a cost of increasing MSE and decreasing correlation.

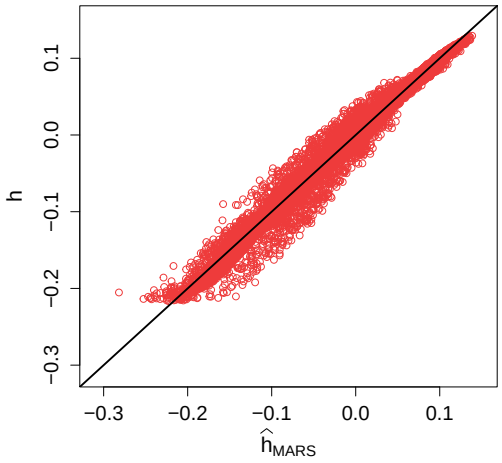
The correlation and MSE are better for $\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$ compared to $\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$. The correlation of $\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$ is almost as good as the correlation of the MARS models, however, the MSE of the MARS models are half the MSE of $\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$ and the computation time is on average over 200 times faster. The model with the worst MSE and correlation is $\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$. Table 4.3 shows that the NPKR models were much slower than the MARS models, and even the forward function $\boldsymbol{h}$. The computation time also increased with the number of data points in the NPKR model as $\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$ was much slower than $\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$. In addition to the computation times reported in Table 4.3, the computation time of an NPKR model created from 100 training data was tested. It used on average 0.674 seconds to predict the test data. Even though that was considerably less than the computation time of $\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$ and $\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$, it is still much longer than the computation times of the MARS models.

$\widehat{\boldsymbol{h}}$	correlation	MSE [10^{-5}]	computation time [sec]
$\widehat{\boldsymbol{h}}_{\text{MARS}}$	0.9954	3.8	0.064
$\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$	0.9958	3.4	0.107
$\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$	0.9833	14.0	5.694
$\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$	0.9914	7.2	22.674

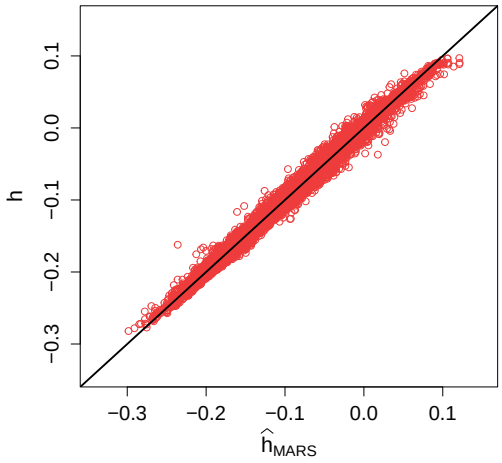
Table 4.3: Correlation, MSE and computation time for $\widehat{\boldsymbol{h}}_{\text{MARS}}$, $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$, $\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$ and $\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$.

Figure 4.9 shows a scatter plot of the $\boldsymbol{h}$ versus $\widehat{\boldsymbol{h}}_{\text{MARS}}$ on the test data, while Figure 4.10 depicts the same for $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$. Overall the figures indicate that the errors are slightly smaller for the predictions of the seismic gradient than for the zero offset. Neither $\widehat{\boldsymbol{h}}_{\text{MARS}}$ nor $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$ seems to make grave systematic

errors, except perhaps for the lowest values of h_{R_0}.

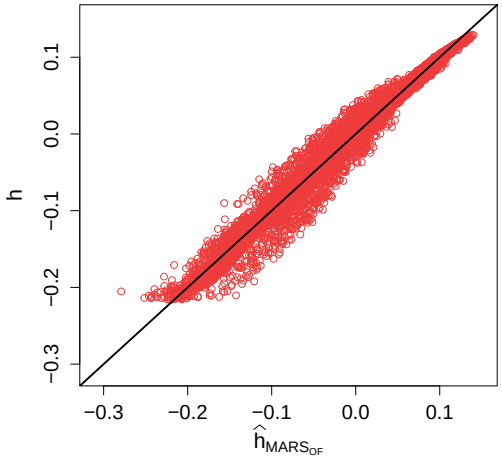


(a) True forward model values for h_{R_0} plotted as function of predicted values from the model $\widehat{\boldsymbol{h}}_{\text{MARS}}$ for the test data.

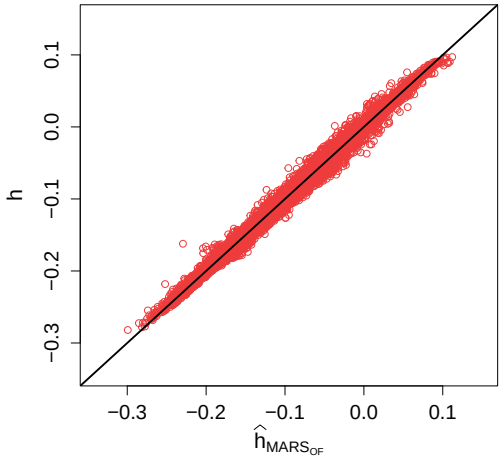


(b) True forward model values for h_G plotted as function of predicted values from the model $\widehat{\boldsymbol{h}}_{\text{MARS}}$ for the test data.

Figure 4.9: True forward model plotted as function of predicted forward model on test data using $\widehat{\boldsymbol{h}}_{\text{MARS}}$.



(a) True forward model values for h_{R_0} plotted as function of predicted values from the model $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$ for the test data.



(b) True forward model values for h_G plotted as function of predicted values from the model $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$ for the test data.

Figure 4.10: True forward model plotted as a function of predicted forward model on test data using $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$.

The predictions in Figures 4.9a and 4.10a look almost identical. This is confirmed by Figure 4.11, where the predictions of $\widehat{\boldsymbol{h}}_{\text{MARS}}$ are plotted against the predictions of $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$. The 14 more terms in model $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$ seems excessive when predictions are close to identical.

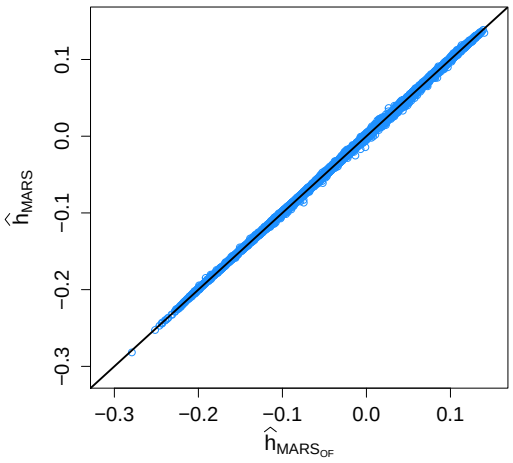


Figure 4.11: Predictions of h_{R_0} from $\widehat{\boldsymbol{h}}_{\text{MARS}}$ plotted as a function of predictions of h_{R_0} from $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$. The black line is $\widehat{\boldsymbol{h}}_{\text{MARS}} = \widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$.

Chapter 5 gives results from MCMC simulations at the Alvheim field using $\widehat{\boldsymbol{h}}_{\text{MARS}}$ as a replacement of $\boldsymbol{h}$. This model was chosen because it was the fastest, and had marginally higher MSE and lower correlation than $\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$. Considering correlation and MSE, both MARS models would serve as good substitutes for the forward model, while reducing the computation time considerably.

As the goal is to reduce computation time, the NPKR models tested in this thesis are not good substitutes for the forward model. A disadvantage of the NPKR model is that more training data makes the model slower. Reducing the number of training data reduces the computation time of the prediction. Predicting the test data 50 times with an NPKR model trained on 100 data points took on average 0.674 seconds which is faster than $\boldsymbol{h}$. However, this is still much slower than the MARS models, and it is also at the cost of increasing the MSE and lowering the correlation. The 100 data points were sampled randomly from the prior, as described at the beginning of the section. Sampling the data in a smarter way, could increase the correlation and lower the MSE, however, the model built on 4000 data points had a correlation close to the MARS model's but a much higher MSE. For an NPKR model to compete with MARS, one would need to find less than 100 training data, that gives a better NPKR model, in terms of correlation and MSE, compared to the NPKR model trained on 4000 training data. This sounds difficult, however, other similar models to the NPKR model used in this thesis could possibly be better. In Section 4.2, issues with NPKR were discussed.

4.4.1 Approximating the gradient of the forward model

To use the MALA introduced in Section 2.2.3, the gradient $\boldsymbol{\nabla h}$ is needed. However, the forward model is treated as a black box. Consequently, the partial derivatives are unavailable. A numerical approximation to the gradient can be found by adding small perturbations, ε, to only one of the covariates while the rest are fixed. The difference in the forward model with and without the perturbation is divided by ε. For the numerical partial derivative of h_{R_0} with respect to the covariate x_g at data point $\boldsymbol{x}_i = [x_{ig}, x_{io}, x_{ic}, x_{id}]$ this is

$$\frac{\partial h_{R_0}}{\partial x_g}(\boldsymbol{x}_i) = \frac{h_{R_0}(\boldsymbol{x}_i + [\varepsilon, 0, 0, 0]) - h_{R_0}(\boldsymbol{x}_i)}{\varepsilon}, \tag{4.27}$$

with $\varepsilon = 0.0001$. However, the forward function needs to be evaluated to calculate the numerical partial derivatives, which require a lot of computation time. In this thesis, the gradients of the four models discussed so far are tested for approximating the gradient of the forward model, $\boldsymbol{\nabla h}$, in order to reduce the computation time. The gradient of the MARS models and NPKR models are described in Sections 4.1 and 4.2 respectively. Because the analytical gradient is unavailable, the approximations are compared to the numerical approximations of the gradient instead. For simplicity, the numerical approximation to the gradient is denoted by $\boldsymbol{\nabla h}$ and referred to as the gradient of the forward model throughout this section.

In addition, as in Section 4.3, MARS models are created with partial derivatives as responses. That is, six MARS models are fitted with each of the partial derivatives $\frac{\partial h_{R_0}}{\partial x_g}$, $\frac{\partial h_{R_0}}{\partial x_o}$, $\frac{\partial h_{R_0}}{\partial x_c}$, $\frac{\partial h_G}{\partial x_g}$, $\frac{\partial h_G}{\partial x_o}$, and $\frac{\partial h_G}{\partial x_c}$ as responses. As MCMC is only performed with $\boldsymbol{x} = [\boldsymbol{x}_g, \boldsymbol{x}_o, \boldsymbol{x}_c]$ the partial derivative of $\boldsymbol{h}$ with respect to the depth is not needed. The six MARS models fitted to the partial derivatives are collectively denoted as $\text{MARS}_{\boldsymbol{\nabla}}$.

The K-means approximation to the gradient of an NPKR model is again tested. The procedure is described in Section 4.2. A simple grid search of the number of clusters $\in [3, 5, 10]$ and the number of closest neighbours to average $\in [3, 5, 10]$ was performed. The computation time decreased as the number of clusters increased. Therefore, the number of clusters was set to 10. When the number of clusters was 10, the lowest MSE and the highest correlation was achieved when using 5 neighbours for both $\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$ and $\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$.

The accuracy of the approximations to the gradient is the mean correlation

$$\text{Corr}\left(\boldsymbol{\nabla h}, \widehat{\boldsymbol{\nabla h}}\right) := \frac{1}{6} \sum_{j \in \{g,o,c\}} \left(\text{Corr}\left(\frac{\partial h_{R_0}}{\partial x_j}, \widehat{\frac{\partial h_{R_0}}{\partial x_j}} \right) + \text{Corr}\left(\frac{\partial h_G}{\partial x_j}, \widehat{\frac{\partial h_G}{\partial x_j}} \right) \right) \tag{4.28}$$

and mean MSE

$$\text{MSE}\left(\boldsymbol{\nabla h}, \widehat{\boldsymbol{\nabla h}}\right) := \frac{1}{6} \sum_{j \in \{g,o,c\}} \left(\text{MSE}\left(\frac{\partial h_{R_0}}{\partial x_j}, \widehat{\frac{\partial h_{R_0}}{\partial x_j}} \right) + \text{MSE}\left(\frac{\partial h_G}{\partial x_j}, \widehat{\frac{\partial h_G}{\partial x_j}} \right) \right) \tag{4.29}$$

for all the covariates j. The correlation, MSE and computation time are reported in Table 4.4. Again, the computation time is the average of 50 computation times. One should keep in mind when reading Table 4.4, that the correlation and MSE are calculated using the numerical approximations as $\boldsymbol{\nabla h}$.

According to Table 4.4, $\text{MARS}_{\boldsymbol{\nabla}}$ is the best at approximating the gradient of the forward model in terms of correlation, MSE and computation time. In Chapter 5, this approximation is used by MALA. The gradient of the two MARS models $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{MARS}}$ and $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$, had the second and third highest correlation after $\text{MARS}_{\boldsymbol{\nabla}}$. These two models also had the second and third lowest MSE. As opposed to in the 2D example, $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{MARS}}$ is slightly better than $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$ in terms of correlation and MSE. The computation time for the gradient of these models is very high. Using binary search in the algorithm for extracting the partial derivatives reduces the computation time, however, the binary search is most efficient for non-interaction terms. Therefore, the computation time depended on the number of interactions in the MARS model. For a MARS model without around 40 non-interaction terms, the computation time was approximately 10% of the computation time of $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{MARS}}$. However, reducing the degree of the MARS model worsened the MSE and correlation. Using more data points in the NPKR model leads to a higher correlation and a lower MSE for the gradient of the NPKR model, though at the cost of increasing the computation time. Using the K-means approach to approximate $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$ was faster than calculating $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$, but the K-means approach for approximating $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$ was slower than computing $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$.

The computation time of $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$ was close to the computation time of $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{MARS}}$ and lower than the computation time for the gradient of the larger MARS model. The K-means approach with 10 clusters and 5 neighbours is not particularly fast. It was however, a bit better at predicting the gradient of the forward model compared to $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$ and $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$.

$\widehat{\boldsymbol{\nabla}\boldsymbol{h}}$	correlation	MSE $[10^{-5}]$	computation time [sec]
$\text{MARS}_{\boldsymbol{\nabla}}$	0.984	0.2	0.32288
$\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{MARS}}$	0.879	1.7	20.59836
$\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{MARS}_{\text{OF}}}$	0.872	1.9	33.72662
$\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{NPKR}_{1000}}$	0.621	6.2	22.40592
$\boldsymbol{\nabla}\widehat{\boldsymbol{h}_{\text{NPKR}_{1000}}}$ (K-means)	0.679	6.1	43.58878
$\boldsymbol{\nabla}\widehat{\boldsymbol{h}}_{\text{NPKR}_{4000}}$	0.702	4.7	62.19348
$\boldsymbol{\nabla}\widehat{\boldsymbol{h}_{\text{NPKR}_{4000}}}$ (K-means)	0.844	3.7	54.28604

Table 4.4: Correlation, MSE and computation time for approximations $\boldsymbol{\nabla}\widehat{\boldsymbol{h}}$.

4.4.2 Adjusting for the uncertainty added to the forward model

To adjust for the additional error caused by replacing $\boldsymbol{h}(\boldsymbol{x})$ by $\widehat{\boldsymbol{h}}_{\text{MARS}}(\boldsymbol{x})$, the relationship between the variable of interest and the data can be modified to

$$\boldsymbol{y}_s = \widehat{\boldsymbol{h}}_{\text{MARS}}(\boldsymbol{x}) + \widetilde{\boldsymbol{\omega}}, \quad \widetilde{\boldsymbol{\omega}} \sim \mathcal{N}\left(\boldsymbol{0}, \widetilde{\boldsymbol{\Omega}}_0\right), \tag{4.30}$$

where $\widetilde{\boldsymbol{\Omega}}_0$ is the covariance matrix $\boldsymbol{\Omega}_0$ in equation (3.19) with empirical variance and covariance are added to it. A homoscedastic independence of the covariates is assumed. Plots of $|\boldsymbol{h}(\boldsymbol{x}) - \widehat{\boldsymbol{h}}_{\text{MARS}}(\boldsymbol{x})|$ against the covariates used to investigate dependency between the absolute error and the covariates, indicated that this is not an unreasonable assumption. The idea is that an average sample variance in the approximation partially compensates for using $\widehat{\boldsymbol{h}}_{\text{MARS}}$. The adjusted covariance matrix is, where the subscript MARS is omitted for simplicity,

$$\widetilde{\boldsymbol{\Omega}}_0 = \boldsymbol{\Omega}_0 + \begin{bmatrix} \widehat{\text{Var}}\left[\widehat{h_{R_0}}\right] & \widehat{\text{Cov}}\left[\widehat{h_{R_0}}, \widehat{h_G}\right] \\ \widehat{\text{Cov}}\left[\widehat{h_{R_0}}, \widehat{h_G}\right] & \widehat{\text{Var}}\left[\widehat{h_G}\right], \end{bmatrix} \tag{4.31}$$

with

$$\widehat{\text{Var}}\left[\widehat{h_{R_0}}\right] = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{h_{R_{0_i}}} - h_{R_{0_i}}\right)^2, \tag{4.32}$$

$$\widehat{\text{Var}}\left[\widehat{h_G}\right] = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{h_{G_i}} - h_{G_i}\right)^2, \tag{4.33}$$

and

$$\widehat{\text{Cov}}\left[\widehat{h_{R_0}}, \widehat{h_G}\right] = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{h_{R_{0_i}}} - h_{R_{0_i}}\right)\left(\widehat{h_{G_i}} - h_{G_i}\right). \tag{4.34}$$

Here $\widehat{h_{R_{0i}}}$ is the prediction for test data point i and $h_{h_{R_{0i}}}$ denotes the response at test data point i. The error-adjusted covariance matrix for the seismic AVO log-likelihood is thus

$$\widetilde{\mathbf{\Omega}} = \begin{bmatrix} \left[\widetilde{\mathbf{\Omega}}_0\right] & 0 & \dots & 0 & 0 \\ 0 & \left[\widetilde{\mathbf{\Omega}}_0\right] & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \left[\widetilde{\mathbf{\Omega}}_0\right] & 0 \\ 0 & 0 & \dots & 0 & \left[\widetilde{\mathbf{\Omega}}_0\right] \end{bmatrix}. \tag{4.35}$$

5

MCMC results from the Alvheim field

In this chapter, MCMC results from the Alvheim field are presented and discussed. The results are divided into three sections and discussed at the end of each section. In the first section, MCMC results from using the exact forward model are compared to MCMC results from using the approximation to the forward model. The approximate likelihood function, where $\boldsymbol{h}$ is replaced by $\widehat{\boldsymbol{h}}_{\text{MARS}}$ from Chapter 4, is tested with using both $\boldsymbol{\Omega}$, from equation (3.22), and $\widetilde{\boldsymbol{\Omega}}$, presented in equation (4.35), as covariance matrix. In Section 5.2, results from the four MH algorithms with the four proposal distributions are compared by looking at trace plots, autocorrelation plots and ESS per time. The forward model is replaced by $\widehat{\boldsymbol{h}}_{\text{MARS}}$ for this part. Due to the computation time of the exact forward model and the extensive comparison, these MCMC computations are only performed on a smaller area in the Alvheim field. Finally, the surrogate, $\widehat{\boldsymbol{h}}_{\text{MARS}}$, is used to perform approximate MCMC over the entire Alvheim field. MCMC results from using the MH algorithm with proposal distribution q_3 are compared to the results of Spremić et al. (2024) in Section 5.3.

In Section 5.1 and Section 5.3, MCMC samples are compared by mean gas, oil and brine saturations and mean clay content. The samples are also compared by their uncertainties, which are represented as the difference between the 90th and the 10th quantiles. The procedures for calculating these quantities are described in this paragraph. To obtain the mean saturations and mean clay content, logistic transformation, described in Chapter 3, of the sample mean is used. At every location l, let the sample mean be denoted by $\boldsymbol{x}^l_{\text{mean}} = [\text{mean}(\boldsymbol{x}^l_g), \text{mean}(\boldsymbol{x}^l_o), \text{mean}(\boldsymbol{x}^l_c)]$ where $\text{mean}(\boldsymbol{x}^l_j)$ is the mean of the $\widetilde{m}$ samples after the burn-in of covariate j. The mean saturations of gas, oil and brine at location l are found by evaluating $\boldsymbol{x}^l_{\text{mean}}$ in the equation (3.4), equation (3.5) and equation (3.6), from Chapter 3, respectively. Similarly, the mean clay content is found by evaluating $\boldsymbol{x}^l_{\text{mean}}$ in equation (3.7). To find the uncertainty at location l, the $\widetilde{m}$ samples after burn-in are sorted and transformed using the logistic transformations. Approximately 10% of transformed and sorted samples have lower values than the 10th quantile and approximately 10% of transformed and sorted samples have a higher value than the 90th quantile. The difference between these two saturations serves as the uncertainty. The uncertainty is measured this way as the sample variance for variables between 0 and 1 is not very informative (Spremić et al., 2024).

5.1 Comparison of the surrogate and the exact forward model

In Chapter 4, an approximation to the slow exact forward function was created. This surrogate was approximately 32 times faster. It also had a low MSE and a high correlation. However, as it is an approximation, the MCMC samples are sampled from a posterior which is an approximation to the

posterior π in equation (3.8). For simplicity, the approximated posterior, where $\boldsymbol{h}$ is replaced by $\widehat{\boldsymbol{h}}_{\text{MARS}}$, is denoted $\widehat{\pi}$. To check how similar $\widehat{\pi}$ is to π, MCMC samples are sampled using both $\boldsymbol{h}$ and $\widehat{\boldsymbol{h}}_{\text{MARS}}$ as forward models. In addition, MCMC samples with $\widehat{\boldsymbol{h}}_{\text{MARS}}$ as forward model and $\widetilde{\boldsymbol{\Omega}}$, from equation (4.35), as the covariance matrix of the likelihood model is tested to see if the samples from this posterior are more similar to the samples of π than the samples of $\widehat{\pi}$. The posterior with forward function $\widehat{\boldsymbol{h}}_{\text{MARS}}$ and covariance matrix $\widetilde{\boldsymbol{\Omega}}$ in the likelihood is denoted by $\widetilde{\pi}$. The approximation $\widehat{\boldsymbol{h}}_{\text{MARS}}$ and the covariance matrix $\widetilde{\boldsymbol{\Omega}}$ are described in Chapter 4. The covariance matrix $\boldsymbol{\Omega}$ is described in Chapter 3.

The MH algorithm with proposal distribution q_2 is used to sample the posteriors π, $\widehat{\pi}$ and $\widetilde{\pi}$. The tuning parameter s is tuned such that the acceptance rate is approximately 23.4%. The value $s = 0.027$ was used, which gave the acceptance rate 23.5% when using the exact forward model, 23.4% when using $\widehat{\boldsymbol{h}}_{\text{MARS}}$ as forward model and 23.5% when using $\widehat{\boldsymbol{h}}_{\text{MARS}}$ as forward model and $\widetilde{\boldsymbol{\Omega}}$ as covariance matrix of the likelihood. The initial sample was the mean of the prior distribution. The number of iterations was $m = 1,000,000$ where every 10th sample is saved. The first half of the saved MCMC samples are discarded as burn-in, such that the remaining $50,000$ samples was 10 time iterations apart and presumably from the stationary distributions of the Markov chains.

Due to computation time, this was only done for a smaller area located at inlines $\times$ crosslines $= [928, 1160] \times [4788, 5020]$, where every other inline and crossline is omitted, such that the location of the measurements are approximately 100 meters apart. This means that the smaller area is discretized by $30 \times 30 = 900$ grid points. Hence $\boldsymbol{x}_g, \boldsymbol{x}_o, \boldsymbol{x}_c \in \mathbb{R}^{900}$. Figure 5.1 shows the smaller area at the Alvheim field where the MCMC was performed. There is a well in the area. Particularly near the well, the samples will be influenced by the likelihood model for the well-log data presented in Chapter 3, such that it is most interesting to investigate the similarities and dissimilarities further from the well.

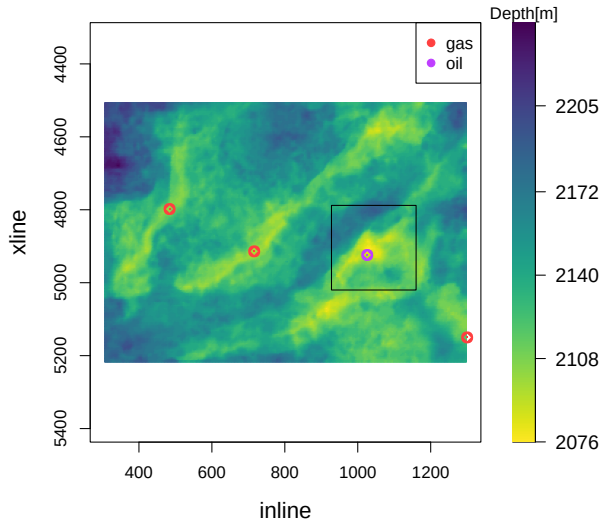
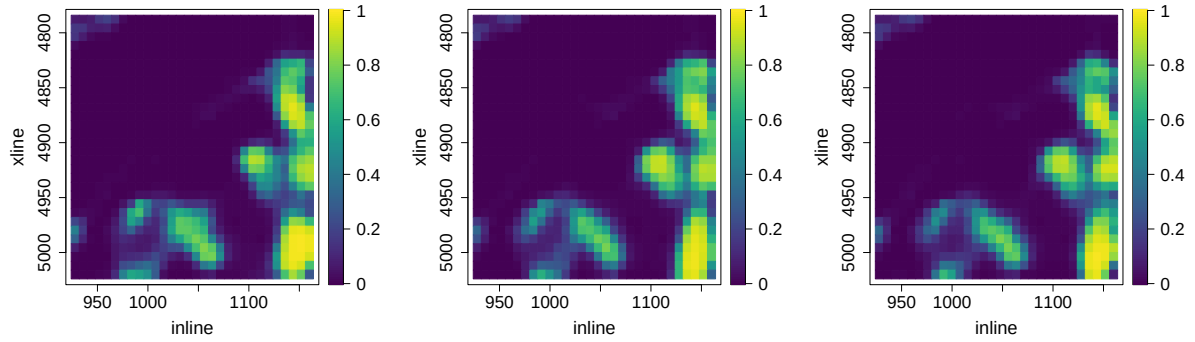


Figure 5.1: The depth of the top-reservoir of the Alvheim field. Four wells are marked by coloured circles, where the colour indicates the majority type of hydrocarbons found at that well. The wells are located at (inline,crossline)-locations $(484, 4798)$, $(716, 4914)$, $(1026, 4924)$ and $(1300, 5150)$. The black square is a smaller area where MCMC with the forward function is performed.

Figure 5.2 shows the mean gas saturations and the corresponding uncertainties from the three sets of

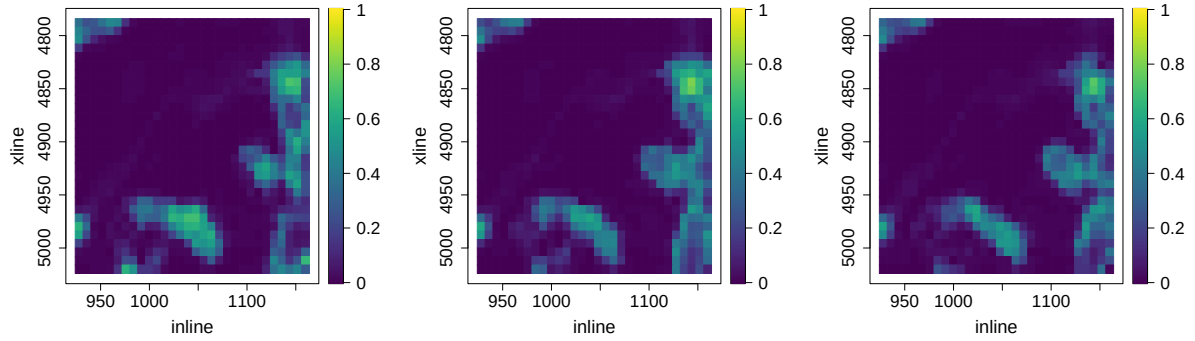
posterior samples. The mean gas saturations and the corresponding uncertainties are very similar. A small difference between the mean gas saturations is that the light green area at approximately inline = 1050 and crossline = 4955 is wider in Figure 5.2a than in Figure 5.2b and Figure 5.2c. In this same area, the uncertainty is also larger in Figure 5.2d compared to Figure 5.2e and Figure 5.2f. Slightly to the left of that area, at approximately inline = 1000 and crossline = 4950, the uncertainties in Figure 5.2d and Figure 5.2e are more similar to each other, while the uncertainty in Figure 5.2f at this area is lower. There is another small difference between the gas saturations in the yellow area at approximately inline = 1100 and crossline = 4920. Figure 5.2b and Figure 5.2c are quite similar in this area, while this area is slightly less intense in Figure 5.2a. The uncertainties show the same tendency in this area, scarcely less intense for the MCMC results using the exact forward model compared to the MCMC results using the surrogate. There are also some differences in the uncertainties in the bottom right corner, where again the MCMC samples using the surrogate are similar to each other and slightly different to the MCMC samples using the exact forward model.



(a) Mean gas saturation from MCMC samples from π.

(b) Mean gas saturation from MCMC samples from $\widehat{\pi}$.

(c) Mean gas saturation from MCMC samples from $\widetilde{\pi}$.



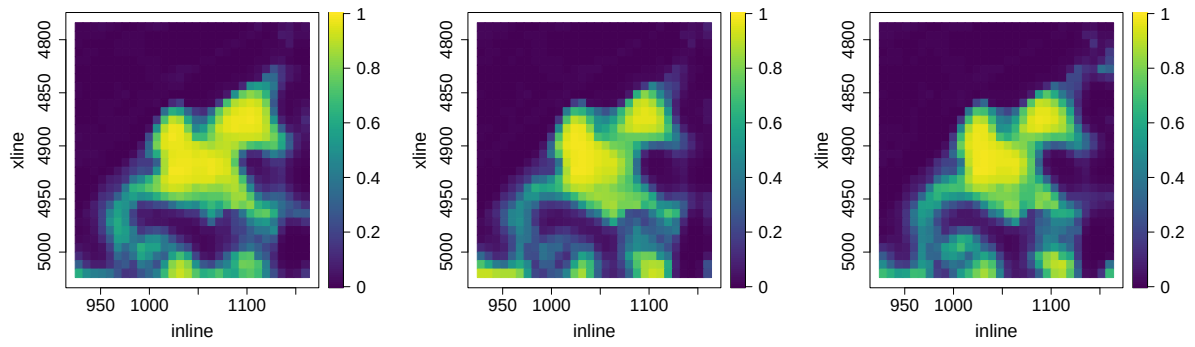
(d) Uncertainty of gas saturation from MCMC samples from π.

(e) Uncertainty of gas saturation from MCMC samples from $\widehat{\pi}$.

(f) Uncertainty of gas saturation from MCMC samples from $\widetilde{\pi}$.

Figure 5.2: Mean gas saturations and uncertainties from MCMC samples from three posteriors. The posterior in the Alvheim field from equation (3.8) is π. The posterior where $\widehat{\boldsymbol{h}}_{\text{MARS}}$ is surrogate is $\widehat{\pi}$. In the last posterior, $\widetilde{\pi}$, both the surrogate $\widehat{\boldsymbol{h}}_{\text{MARS}}$ and the error correcting covariance matrix $\widetilde{\boldsymbol{\Omega}}$ are used. The MH algorithm with proposal distribution q_2 with $s = 0.027$ was used in all three cases. Uncertainty is expressed as the difference between the 90th and 10th quantiles.

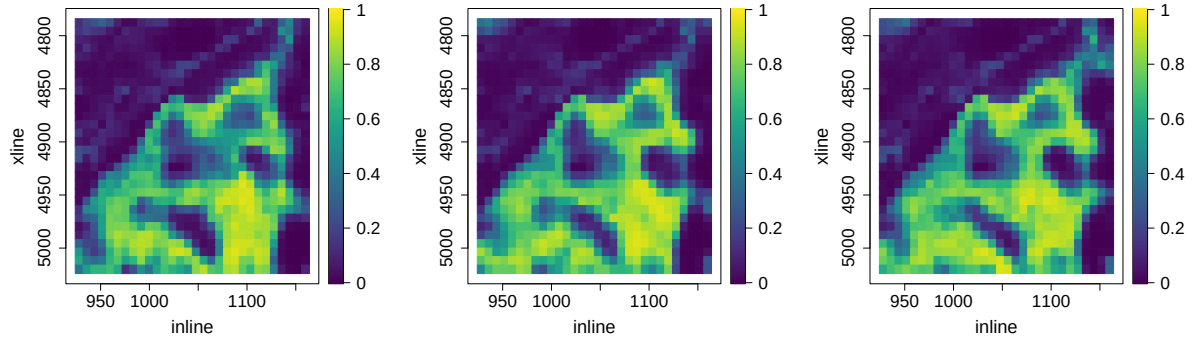
Also, the mean oil saturations and the oil saturation uncertainties in Figure 5.3 are very similar in all three cases. There is a difference in the bottom left corner, where samples from π show lower oil saturation than the samples from $\widehat{\pi}$ and $\widetilde{\pi}$. Using the error-correcting covariance matrix improved the approximation to π because the oil saturation in this area in Figure 5.3c shows lower oil saturation compared to the oil saturation in Figure 5.3b. However, in the bottom right corner, Figure 5.3c shows a small green area which is not present in neither Figure 5.3b nor in Figure 5.3a. The MCMC samples from $\widetilde{\pi}$ also have some slightly visible green areas in the top right corner, which is not found in the other two cases. More visible are the differences in the uncertainty in this area. The uncertainties in Figure 5.3d and in Figure 5.3e are more similar compared to the uncertainty in Figure 5.3f, which has a higher uncertainty. An area where the uncertainties in the samples from $\widehat{\pi}$ and $\widetilde{\pi}$ are similar to each other and different to the uncertainty in the samples from π is at approximately inline = 1060 and crossline = 4900. The uncertainty of the samples from π is lower. Another example of where the samples from $\widehat{\pi}$ and $\widetilde{\pi}$ are more similar to each other compared to the samples from π, is in the oil saturation at approximately inline = 1060 and crossline = 5020. In this area, the oil saturation in Figure 5.3b and in Figure 5.3c show a mean saturation of oil of approximately zero, while in Figure 5.3a, the mean saturation of oil is approximately 0.6 in the same area.



(a) Mean oil saturation from MCMC samples from π.

(b) Mean oil saturation from MCMC samples from $\widehat{\pi}$.

(c) Mean oil saturation from MCMC samples from $\widetilde{\pi}$.



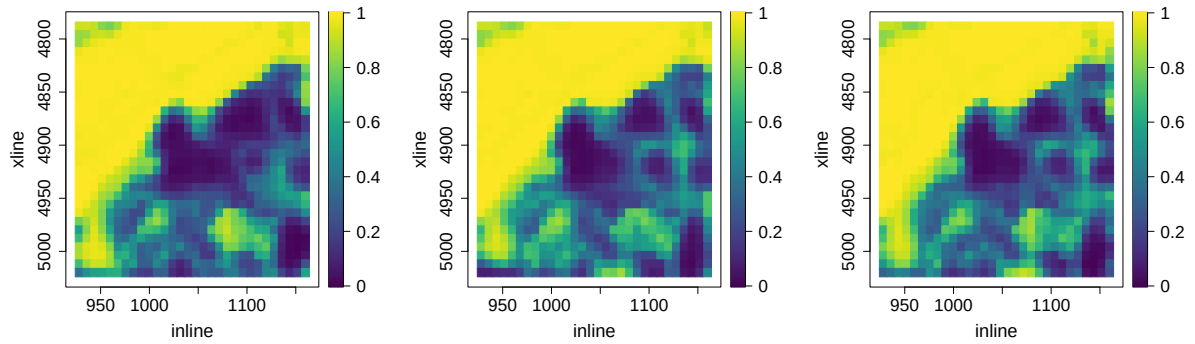
(d) Uncertainty of oil saturation from MCMC samples from π.

(e) Uncertainty of oil saturation from MCMC samples from $\widehat{\pi}$.

(f) Uncertainty of oil saturation from MCMC samples from $\widetilde{\pi}$.

Figure 5.3: Mean oil saturations and uncertainties from MCMC samples from three posteriors. The posterior in the Alvheim field from equation (3.8) is π. The posterior where $\widehat{\boldsymbol{h}}_{\text{MARS}}$ is surrogate is $\widehat{\pi}$. In the last posterior, $\widetilde{\pi}$, both the surrogate $\widehat{\boldsymbol{h}}_{\text{MARS}}$ and the error correcting covariance matrix $\widetilde{\boldsymbol{\Omega}}$ are used. The MH algorithm with proposal distribution q_2 with $s = 0.027$ was used in all three cases. Uncertainty is expressed as the difference between the 90th and 10th quantiles.

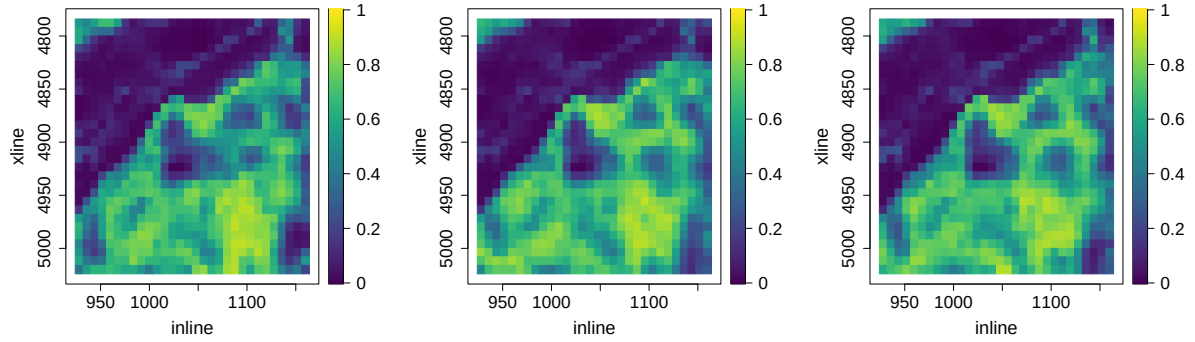
The mean saturation of brine and corresponding uncertainty in Figure 5.4 are also similar in the three cases. A difference is found in approximately inline = 1060 and crossline = 5020. The brine saturations in Figure 5.4b and in Figure 5.4c show high values, whereas the brine saturation in Figure 5.4a show a lower brine saturation in this area. This is the same area where there was a difference in the oil saturation. The uncertainties in the brine saturation in the three sets of samples are the most different in the bottom left and right corners. In these areas, the similarities between Figure 5.4c and Figure 5.4d are stronger compared to the similarities between Figure 5.4e and Figure 5.4d.



(a) Mean brine saturation from MCMC samples from π.

(b) Mean brine saturation from MCMC samples from $\widehat{\pi}$.

(c) Mean brine saturation from MCMC samples from $\widetilde{\pi}$.



(d) Uncertainty of brine saturation from MCMC samples from π.

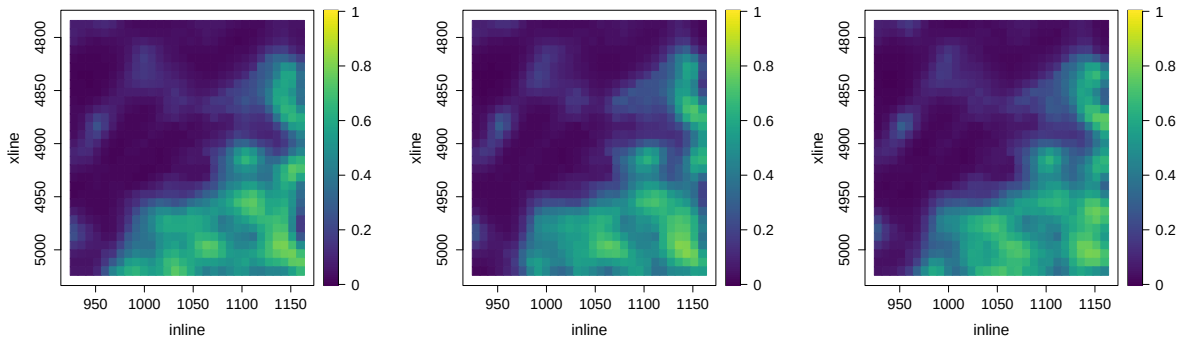
(e) Uncertainty of brine saturation from MCMC samples from $\widehat{\pi}$.

(f) Uncertainty of brine saturation from MCMC samples from $\widetilde{\pi}$.

Figure 5.4: Mean brine saturations and uncertainties from MCMC samples from three posteriors. The posterior in the Alvheim field from equation (3.8) is π. The posterior where $\widehat{\boldsymbol{h}}_{\text{MARS}}$ is surrogate is $\widehat{\pi}$. In the last posterior, $\widetilde{\pi}$, both the surrogate $\widehat{\boldsymbol{h}}_{\text{MARS}}$ and the error correcting covariance matrix $\widetilde{\boldsymbol{\Omega}}$ are used. The MH algorithm with proposal distribution q_2 with $s = 0.027$ was used in all three cases. Uncertainty is expressed as the difference between the 90th and 10th quantiles.

The mean content of clay shown in Figure 5.5a, Figure 5.5b and in Figure 5.5c are very similar to each other. There are some slight differences to the right in the areas at approximately crossline = 4955, the mean clay content in Figure 5.5a is much lower compared to the mean clay contents in Figure 5.5b and Figure 5.5c. In this area, the samples from $\widehat{\pi}$ look slightly more like the samples from π, compared to how similar samples from $\widehat{\pi}$ and π are. The uncertainty in Figure 5.5d, Figure 5.5e and in Figure 5.5f are very similar. In general, the uncertainty might be a bit larger in Figure 5.5d for example at inline

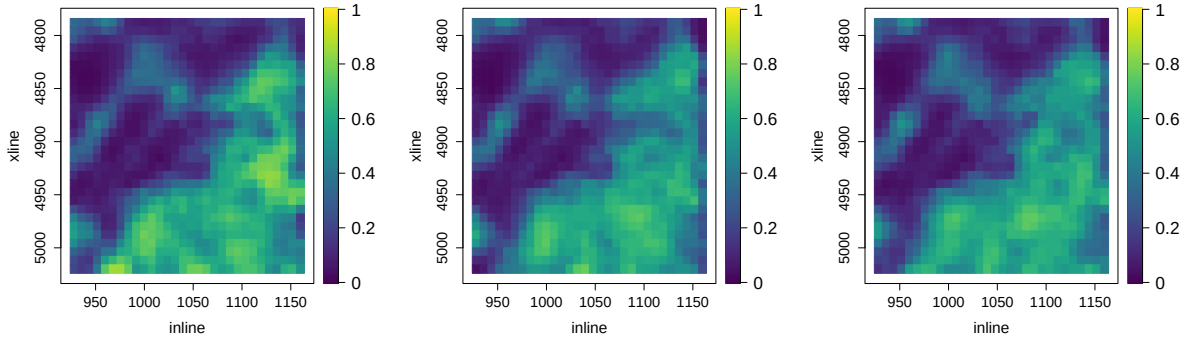
$= 900 - 1000$ at the highest crosslines, and in the area at approximately inline $= 1130$ and crossline $= 4930$.



(a) Mean clay content from MCMC samples from π.

(b) Mean clay content from MCMC samples from $\widehat{\pi}$.

(c) Mean clay content from MCMC samples from $\widetilde{\pi}$.



(d) Uncertainty of clay content from MCMC samples from π.

(e) Uncertainty of clay content from MCMC samples from $\widehat{\pi}$.

(f) Uncertainty of clay content from MCMC samples from $\widetilde{\pi}$.

Figure 5.5: Mean clay content and uncertainties from MCMC samples from three posteriors. The posterior in the Alvheim field from equation (3.8) is π. The posterior where $\widehat{\boldsymbol{h}}_{\text{MARS}}$ is surrogate is $\widehat{\pi}$. In the last posterior, $\widetilde{\pi}$, both the surrogate $\widehat{\boldsymbol{h}}_{\text{MARS}}$ and the error correcting covariance matrix $\widetilde{\boldsymbol{\Omega}}$ are used. The MH algorithm with proposal distribution q_2 with $s = 0.027$ was used in all three cases. Uncertainty is expressed as the difference between the 90th and 10th quantiles.

Figure 5.2, Figure 5.3, Figure 5.4 and Figure 5.5 show overall great similarity in the results from the MCMC samples from π, $\widehat{\pi}$ and $\widetilde{\pi}$ on the smaller area in the Alvheim field. This indicates that substituting $\boldsymbol{h}$ with $\widehat{\boldsymbol{h}}_{\text{MARS}}$ gives a good approximation to the posterior π, both when using the covariance matrix $\boldsymbol{\Omega}$ and when using $\widetilde{\boldsymbol{\Omega}}$ as covariance matrix in the likelihood model. MARS models were also considered as good surrogates for the time-consuming forward models in Chen et al. (2014) and Chen et al. (2013). A MARS model does not assume a functional form of the response, which could be the reason it applied to all three problems.

The use of the adjusted correlation function $\widetilde{\boldsymbol{\Omega}}$ showed minor improvements in some areas, while in other areas, using $\boldsymbol{\Omega}$ gave the means and uncertainties closest to the mean and uncertainties of the MCMC samples of π. The empirical variance and covariance added to the covariance matrix to adjust for the error between $\widehat{\boldsymbol{h}}$ and $\boldsymbol{h}$ were very small, such that $\widetilde{\boldsymbol{\Omega}}$ and $\boldsymbol{\Omega}$ are quite similar. That the covariance matrices are very similar could be the reason for the similar result.

Most of the differences between the exact and approximate samples are found at the edges of the area. As mentioned, there is a well in that area, such that samples close to the well are affected by the likelihood model for the well-log data in addition to the likelihood model for the seismic AVO data. However, the mean and uncertainty of the saturations are in general very similar, indicating that $\widehat{\pi}$ and $\widetilde{\pi}$ are similar to π. In MCMC, there is a randomness when proposing and accepting samples, such that the exact same samples might not be considered for the three different Markov chains. This could lead to some differences in the means and uncertainties, however, this difference vanishes with an increasing number of MCMC iterations if π, $\widehat{\pi}$ and $\widetilde{\pi}$ were the same distributions.

5.2 Comparing the proposals distributions

Next, the performance of the MH algorithm with the four different proposal distributions, q_1, q_2, q_3 and q_4, from Chapter 2 are compared. The tuning parameter is, as in Chapter 2, denoted by s. To refresh, there was the standard random walk

$$q_1(\boldsymbol{x}^p|\boldsymbol{x}_t) = \varphi(\boldsymbol{x}_t, s^2\boldsymbol{I}), \tag{5.1}$$

a random walk with the covariance matrix from the prior

$$q_2(\boldsymbol{x}^p|\boldsymbol{x}_t) = \varphi(\boldsymbol{x}_t, s^2\boldsymbol{\Sigma}), \tag{5.2}$$

the pCN proposal distribution with the covariance matrix from the prior

$$q_3(\boldsymbol{x}^p|\boldsymbol{x}_t) = \varphi\left(\boldsymbol{\mu} + \sqrt{1-s^2}(\boldsymbol{x}_t - \boldsymbol{\mu}), s^2\boldsymbol{\Sigma}\right), \tag{5.3}$$

and the MALA

$$q_4(\boldsymbol{x}^p|\boldsymbol{x}_t) = \varphi\left(\boldsymbol{x}_t + \frac{s^2}{2}\nabla \log\left(\pi(\boldsymbol{x}_t)\right), s^2 I\right). \tag{5.4}$$

The approximation $\text{MARS}_{\boldsymbol{\nabla}}$ from Chapter 4 is used instead of the gradient of the log posterior in the MALA to reduce computation time.

The MH algorithms are used to draw samples from $\widehat{\pi}$, from the previous section, on the small area. That is, $\widehat{\boldsymbol{h}}_{\text{MARS}}$ is used as the forward model and the covariance matrix of the likelihood model is $\boldsymbol{\Omega}$. The objective is to compare how efficiently the four MH algorithms explore the domain of $\widehat{\pi}$. The ESS per computation time is measured to assess which proposal gives the most information about $\widehat{\pi}$ per time. The reported ESS is the mean ESS of all $3N$ stochastic processes $x_g^1, x_o^1, x_c^1, x_g^2, x_o^2, x_c^2, ..., x_g^N, x_o^N, x_c^N$. The tuning parameter s and acceptance rates, which are connected to the efficiencies of the algorithms are reported. The four MH algorithms were tuned such that the acceptance rates were approximately 23.4% when using the proposal distributions q_1, q_2 and q_3 and approximately 57.4% when using the proposal distribution q_4 as discussed in Section 2.2.3. As the acceptance rates and ESS should be computed on MCMC samples after burn-in, the Markov chain starts at the mean of the posterior samples whose logistic transformation is shown in Figure 5.2a, Figure 5.3a and Figure 5.5a. The MH algorithms are run for $1,000,000$ iterations where every 100th sample is saved. This gives $10,000$ MCMC samples from $\widehat{\pi}$, 100 time iterations apart. Due to computation time and memory limitations, this is also done for the small area described in the previous section.

The acceptance rate, computation time, ESS and ESS per computation time for the four MH algorithms are reported in Table 5.1. These results show that the MH algorithm with the proposal distribution q_3 was the most efficient. The $10,000$ MCMC samples this algorithm produced give approximately the same

proposal	s	acceptance rate [%]	computation time [sec]	ESS [#]	ESS / time [#/sec]
q_1	0.006	25.5	13546.24	11.01340	0.000813023
q_2	0.027	23.3	16547.33	93.88295	0.005673601
q_3	0.045	22.8	11408.03	212.64163	0.018639645
q_4	0.038	58.9	44418.04	39.72408	0.000894323

Table 5.1: Comparison of the efficiency of four MH algorithms. The acceptance rate, computation time and ESS are calculated on $10,000$ MCMC samples which are 100 time iterations apart.

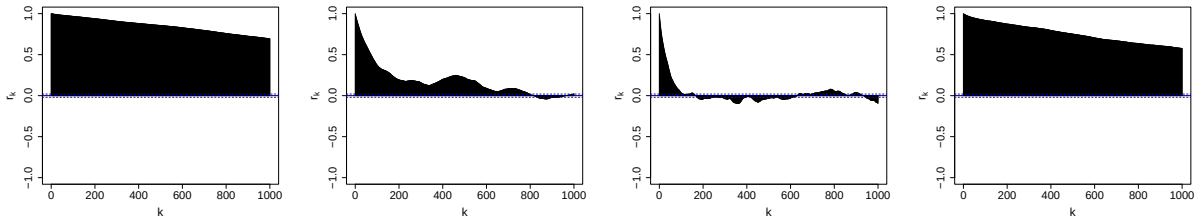
information as 212 i.i.d samples. The algorithm produced approximately 0.0186 independent samples per second. This is much higher than the ESS per computation time for the MH algorithm with the other proposals distributions. The second most efficient algorithm was the MH algorithm with the proposal distribution q_2. The difference in the ESS between the most efficient and second most efficient algorithms is large. The ESS of the MH algorithm with proposal distribution q_3 is twice as high as the ESS for the MH algorithm with q_2 as proposal distribution. The MH algorithm with proposal distribution q_3 was also about 1 hour and 25 minutes faster than the second most efficient algorithm. The slowest algorithm was the MALA, which used more than 12 hours. For comparison, the fastest algorithm used 3 hours and 10 minutes. The MH algorithm with proposal distribution q_1 was the second fastest algorithm. The MALA used more than three times as much time as q_1, but because the samples were less correlated, the MH algorithm with q_4 was slightly more efficient than the MH algorithm with q_1 in this case.

Figure 5.6 show autocorrelation plots of every 100th value of $\{x_{g,t}^l\}_{t=0}^{1000000}$, $\{x_{o,t}^l\}_{t=0}^{1000000}$ and $\{x_{c,t}^l\}_{t=0}^{1000000}$ at location $l = 435$ using the four MH algorithms. These autocorrelation plots agree with the results in Table 5.1. The autocorrelation plots in 5.6 show that the MH algorithm with q_3 as proposal distribution produces the least correlated samples among the four algorithms because the curve decreases the fastest. The autocorrelation plots for this algorithm, shown in Figure 5.6c, Figure 5.6g and Figure 5.6k, show that samples approximately $15,000$ time iterations apart, as there are 100 time iterations between each lag, are not significantly correlated to each other. Compared to the autocorrelation plots of the least efficient algorithm, the MH algorithm with q_1 as proposal distribution, the autocorrelation plots of the MH with autocorrelation q_3 decrease much faster. The autocorrelation plots for the MH algorithm with proposal distribution q_2 also decrease fast compared to the autocorrelation plots of the MH algorithms with proposal distributions q_1 and q_4. The autocorrelation in Figure 5.6a and in Figure 5.6d seem to decrease at the same rate. However, the autocorrelation plot in Figure 5.6h and in Figure 5.6l from the MALA decrease noticeably faster than the autocorrelation plots in Figure 5.6e and in Figure 5.6i, which belong to the MH algorithm with q_1 as proposal distribution.

The trace plots in Figure 5.7 also correspond with the results in Table 5.1. The trace plots show that the MH algorithm with q_3 as proposal distribution mixes best because the path moves around a centred value many times. The MH algorithm with proposal distribution q_2 also seems to mix well according to the trace plots, however, there is slightly less coverage compared to the most efficient algorithm. The trace plots indicate that the MALA mixes much better than the standard random walk, though the MALA does not mix as well as the MH algorithms with q_2 and q_3 as proposal distributions. The trace plots indicate that the MH algorithm with q_1 as proposal distribution has not yet converged because there seems to be a trend indicating that the algorithm was still exploring the posterior. When comparing the trace plots for the standard random walk to the other algorithms, it is clear that this algorithm has not

yet converged. For example, the trace plot in Figure 5.7a has not yet reached values close to -10, which is an area visited by all three other algorithms.

In this location, the autocorrelation plots and trace plots agreed with the results in Table 5.1. It must be mentioned that mixing, and hence trace plots and autocorrelation plots can vary at different locations. However, the ESS in the table is the mean ESS of every location, which tells us that the MH algorithm with q_3 as proposal distribution had the best overall mixing.

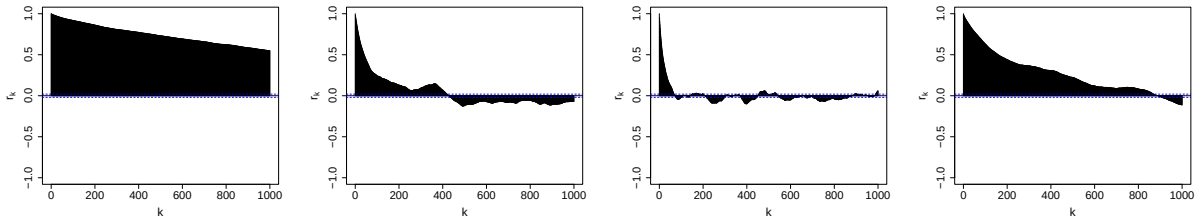


(a) Autocorrelation plot from MCMC computations with q_1 as proposal distribution.

(b) Autocorrelation plot from MCMC computations with q_2 as proposal distribution.

(c) Autocorrelation plot from MCMC computations with q_3 as proposal distribution.

(d) Autocorrelation plot from MCMC computations with q_4 as proposal distribution.

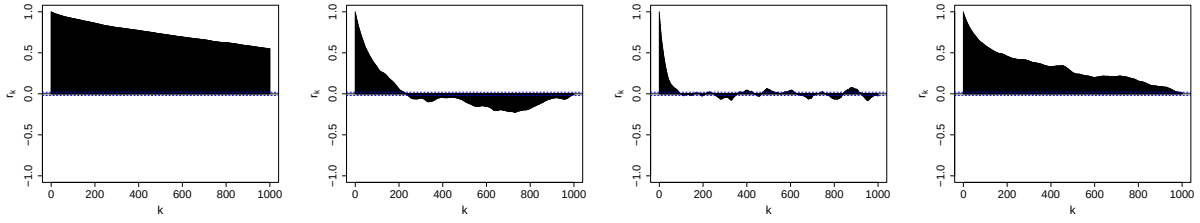


(e) Autocorrelation plot from MCMC computations with q_1 as proposal distribution.

(f) Autocorrelation plot from MCMC computations with q_2 as proposal distribution.

(g) Autocorrelation plot from MCMC computations with q_3 as proposal distribution.

(h) Autocorrelation plot from MCMC computations with q_4 as proposal distribution.



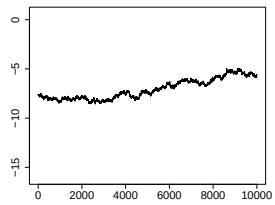
(i) Autocorrelation plot from MCMC computations with q_1 as proposal distribution.

(j) Autocorrelation plot from MCMC computations with q_2 as proposal distribution.

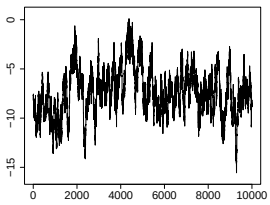
(k) Autocorrelation plot from MCMC computations with q_3 as proposal distribution.

(l) Autocorrelation plot from MCMC computations with q_4 as proposal distribution.

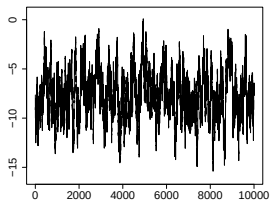
Figure 5.6: Autocorrelation plot up to lag $k = 1000$ of every 100th value of $\{x_{g,t}^l\}_{t=0}^{1000000}$ (first row), $\{x_{o,t}^l\}_{t=0}^{1000000}$ (second row) and $\{x_{c,t}^l\}_{t=0}^{1000000}$ (third row) at location $l = 435$ using the four MH algorithms. The MH algorithm with proposal distribution q_1, q_2, q_3 and q_4 was used to get the results in the first, second, third and fourth columns respectively.



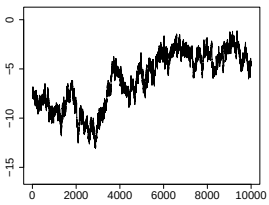
(a) Trace plot from MCMC computations with q_1 as proposal distribution.



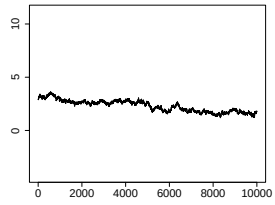
(b) Trace plot from MCMC computations with q_2 as proposal distribution.



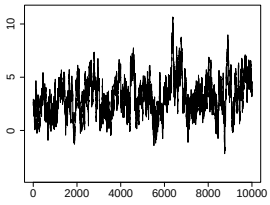
(c) Trace plot from MCMC computations with q_3 as proposal distribution.



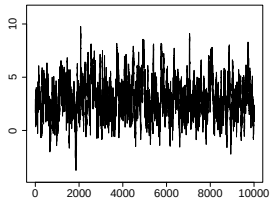
(d) Trace plot from MCMC computations with q_4 as proposal distribution.



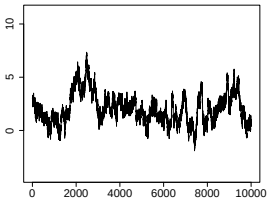
(e) Trace plot from MCMC computations with q_1 as proposal distribution.



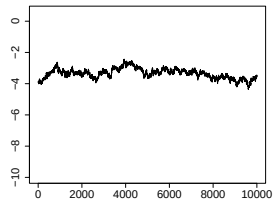
(f) Trace plot from MCMC computations with q_2 as proposal distribution.



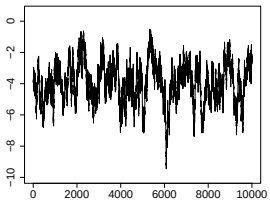
(g) Trace plot from MCMC computations with q_3 as proposal distribution.



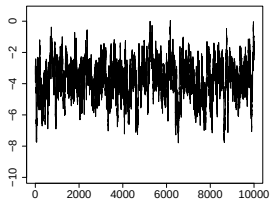
(h) Trace plot from MCMC computations with q_4 as proposal distribution.



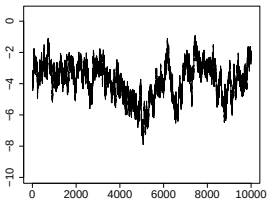
(i) Trace plot from MCMC computations with q_1 as proposal distribution.



(j) Trace plot from MCMC computations with q_2 as proposal distribution.



(k) Trace plot from MCMC computations with q_3 as proposal distribution.



(l) Trace plot from MCMC computations with q_4 as proposal distribution.

Figure 5.7: Trace plots of every 100th value of $\{x^l_{g,t}\}_{t=0}^{1000000}$ (first row), $\{x^l_{o,t}\}_{t=0}^{1000000}$ (second row) and $\{x^l_{c,t}\}_{t=0}^{1000000}$ (third row) at location $l = 435$ using the four MH algorithms. The MH algorithm with proposal distribution q_1, q_2, q_3 and q_4 was used to get the results in the first, second, third and fourth columns respectively.

Among the four MH algorithms tested for seismic AVO inversion in the small area in the Alvheim field, using MH with the proposal distribution q_3 was the most efficient sampling scheme in terms of ESS per computation time. Between the four algorithms, the MH algorithm with proposal distribution q_3 actually had both the highest ESS and the lowest computation time. Thus the MH algorithm with q_3 was clearly the most efficient sampling scheme in the Alvheim case. However, this does not necessarily mean that it is more efficient than the other algorithms in other situations. MCMC methods applicable for a variety of inverse problems are generally hard to find, and an appropriate MCMC scheme is problem-dependent (Eidsvik & Tjelmeland, 2006; Khoshkholgh et al., 2021). On the other hand, the MH algorithm with q_3 as proposal distribution has been efficient for other problems than the Alvheim case as well. Using the

proposal distribution q_3 in the MH algorithm was much more efficient compared to using the proposal distribution q_1 in Rudolf and Sprungk (2016), especially as the dimension of the inversion problem increased. The high dimension in the Alvheim case is a probable reason for the MH algorithm with q_3 to be the most efficient sampler because as opposed to the MH algorithm with q_1 as proposal distribution, its efficiency does not depend on the dimension.

As the two most efficient algorithms were the MH algorithm with the proposal distribution q_3 and the MH algorithm with the proposal distribution q_2, this suggests that using correlation in the proposal distribution might lead to more efficient sampling. Intuitively this makes sense because if there is a covariance pattern between the random variables, it is unlikely to draw a sample which has a covariance structure similar to the covariance pattern of the posterior when all N locations are sampled independently. This reasoning would suggest that this issue increases with the dimension N. In addition, if the covariance pattern of the prior resembles the covariance pattern of the posterior in the Alvheim case, then using this information in the proposal distribution leads to a more efficient sampler (Khoshkholgh et al., 2021).

Using the MH algorithm with the proposal distribution q_3 gives faster MCMC sampling compared to using the MH algorithm with the other three proposal distributions. It is reasonable that using the proposal distribution q_4 gives the slowest MCMC sampling because it is neither symmetric, as q_1 and q_2, nor reversible with respect to the prior, as q_3, and gradients need to be evaluated. Moreover, it is reasonable that q_2 is the second slowest proposal because it proposes samples with correlation, as q_3, and needs to evaluate both the prior and likelihood, as q_1. Using the proposal q_1 one does not need to propose samples with correlation, as is necessary for q_3. However, the prior needs to be evaluated when using q_1. This is not the case when using q_3, where only the likelihood function needs to be evaluated. The computation time in Table 5.1 thus show that proposing a sample with correlation is faster than evaluating the proposal distribution.

For this example, the MALA mixes considerably better than the MH algorithm with q_1 as proposal distribution. This indicates that using information about the posterior is important as described in (Roberts & Rosenthal, 1998; Khoshkholgh et al., 2021). The computation time used by MALA was more than three times the computation time used by the MH algorithm with proposal distribution q_1. However, as the samples were less correlated, the MALA was able to produce marginally more information about $\widehat{\pi}$ per second.

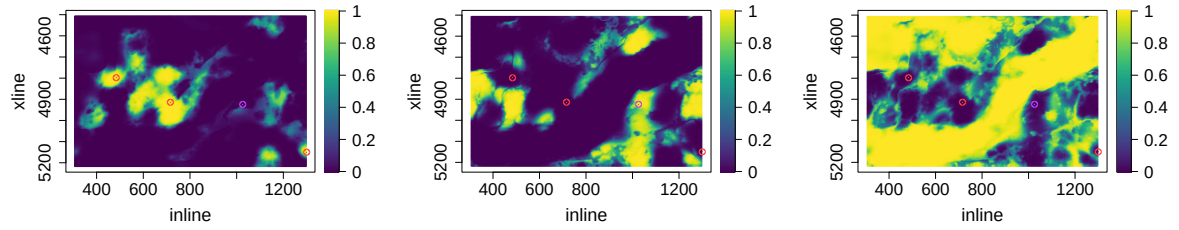
5.3 The MCMC samples from the approximate posterior in the Alvheim case

In this section, MCMC samples from $\widehat{\pi}$ over the entire Alvheim field are compared to the results of Spremić et al. (2024). The two results are compared by mean saturation of gas, oil and brine, mean clay content, uncertainties and ternary plots from one deep and one shallow location in the Alvheim field. How the saturation mean and uncertainty are computed is described at the beginning of this chapter.

The MH algorithm with q_3 as proposal distribution is used to draw $500,000$ MCMC samples after burn-in where every 50th sample is saved. That gives $10,000$ MCMC samples 50 time iterations apart, presumably from the posterior $\widehat{\pi}$. The tuning parameter was $s = 0.0088$ and the acceptance rate was 25.4%. The Markov chain is started in two locations to check that the Markov chain converges to the same values. The MCMC results are regarded before the comparison.

The MCMC results are shown in Figure 5.8 and Figure 5.9. Figure 5.8 shows the mean and uncertainty

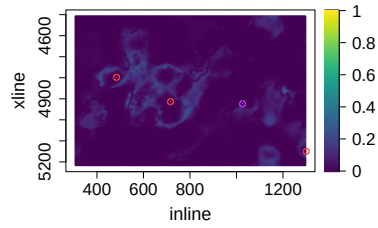
of gas, oil and brine saturations, while the mean clay content and corresponding uncertainty are shown in Figure 5.9. The wells are marked by coloured circles. As in Chapter 3, red circles indicate wells where mainly gas was found and the violet circle indicates the well where the primarily oil was found. Figure 5.8 shows that near the wells where mainly gas was found, the mean gas saturation is high and the mean oil saturation is low. Likewise, the mean gas saturation is low and the mean oil saturation is high near the well where most oil was retrieved. The uncertainty is generally low near the four wells. The mean gas and oil saturations are generally low at the deeper areas, such as in the bottom left corner, the top left corner, in the top middle and between the gas and oil wells in the area that stretches from approximately (inline,crossline) = (600,5200) to (1300,4500). The uncertainty of the saturations is generally low, however higher for the oil saturations than the gas saturations. The uncertainties are low in the middle of areas which show high oil or gas saturations, and higher where the saturations transition from high to low values, such that the uncertainties look like circles enclosing the areas with high gas or oil saturations.



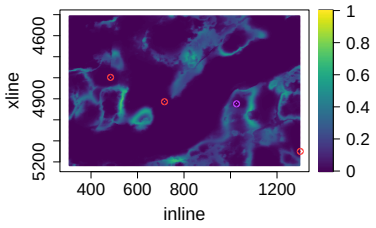
(a) Mean saturation of gas.

(b) Mean saturation of oil.

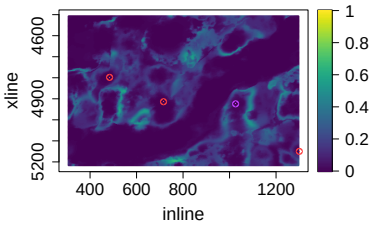
(c) Mean saturation of brine.



(d) Uncertainty of gas saturation.

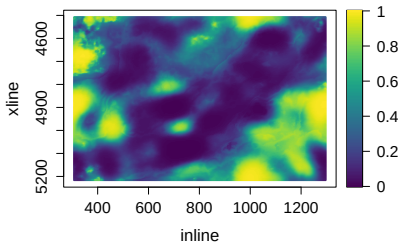


(e) Uncertainty of oil saturation.

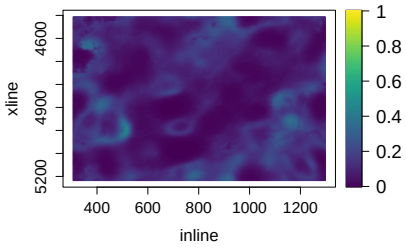


(f) Uncertainty of gas saturation.

Figure 5.8: Mean and uncertainty of gas, oil and brine saturations from MCMC samples from $\widehat{\pi}$. The uncertainty is the difference in the 90th and 10th quantile.



(a) Mean clay content.



(b) Uncertainty of clay content.

Figure 5.9: Mean and uncertainty of clay content from MCMC samples from $\widehat{\pi}$. The uncertainty is the difference in the 90th and 10th quantile.

The results of Spremić et al. (2024) are shown in Figure 5.10 and Figure 5.11. The ensemble based method described in (Spremić et al., 2024) is used to sample 100 samples approximately from the posterior π in equation (3.8). Mean gas and oil saturations and the uncertainties of these saturations are shown in Figure 5.10. The clay content and uncertainty of the clay content are shown in Figure 5.11. The three wells marked by red circles and the well marked by the violet circle in Figure 5.8 are marked by red squares and a green triangle in Figure 5.10, respectively.

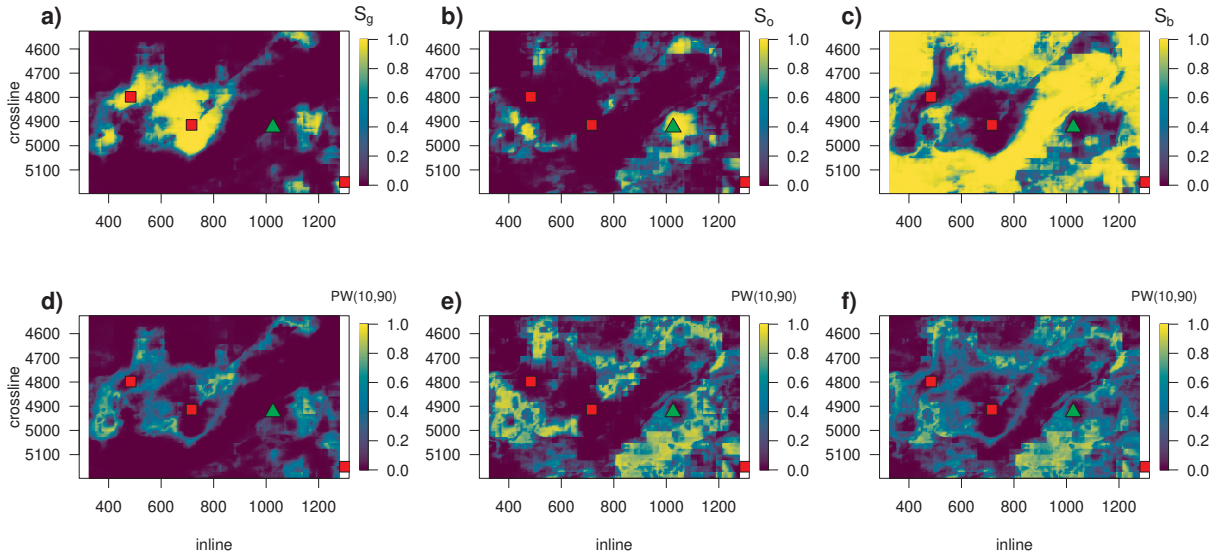


Figure 5.10: Mean and uncertainty of gas, oil and brine saturations from the ensemble based method described in Spremić et al. (2024). The uncertainty is the difference in the 90th and 10th quantile. a) mean gas saturation, b) mean oil saturation, c) mean brine saturation, d) uncertainty of gas saturation, e) uncertainty of oil saturation and f) uncertainty of brine saturation.

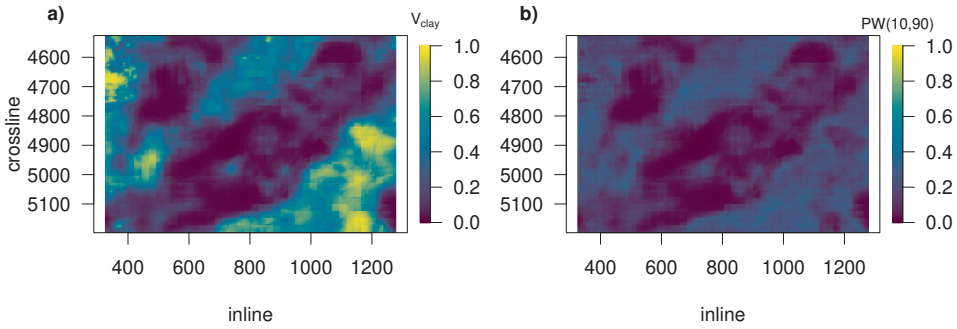


Figure 5.11: Mean and uncertainty of clay content from the ensemble based method described in Spremić et al. (2024). The uncertainty is the difference in the 90th and 10th quantile. a) mean clay content and b) uncertainty of clay content.

There are several similarities between the two results. The results in Figure 5.10 show high mean saturation of gas and oil near the wells where mostly gas and oil were found respectively. There is also low uncertainty close to the four wells. In general, the areas of high mean gas saturation look similar to each other, but the mean saturation of gas from the MCMC samples has more defined shapes compared to the mean gas saturation in Figure 5.10, which is more square-shaped and blurred around the edges of

high mean saturation areas. Another similarity is the low mean saturation of gas and oil between the gas and oil wells in the bottom left area, the top left area and the area that stretches from approximately (inline,crossline) = (600,5200) to (1300,4500). The uncertainties of the clay content in Figure 5.11 and Figure 5.9 also look very similar.

The mean oil saturations in the two results show some dissimilarities. For example, in Figure 5.10 below the well marked by the green triangle, there are modest tendencies toward high oil saturations, whereas in the same area in Figure 5.8b, mean oil saturation values are close to one. There is also an area to the right of the well marked by the violet circle in Figure 5.8b which shows high mean oil saturation. This area shows very low mean oil saturation in Figure 5.10. Another example of areas where the MCMC results show higher oil saturation compared to the results of Spremić et al. (2024) are two areas below the well located the furthest to the left at approximately inline = 500 and crossline = 4800. Conversely above this well at approximately crossline $4600 - 4700$, there is an area where the mean oil saturation from the MCMC results is very low and the mean oil saturation in Figure 5.10 is high.

Another significant difference is the difference in the uncertainties for the mean gas and oil saturations. In Figure 5.10 the uncertainty is high for the mean oil saturations in the areas described in the previous paragraph. That is the area below the well marked by the green triangle, the area below the leftmost well and the area over the leftmost well. In addition, the uncertainty of the mean oil saturation is high above the well between the leftmost well and the well marked by the green triangle. The uncertainty of the gas saturations in Figure 5.10 are high in the same areas as the mean gas saturation is high, except near the wells. The uncertainty of the mean gas saturation from the MCMC results in Figure 5.9b are generally considerably lower.

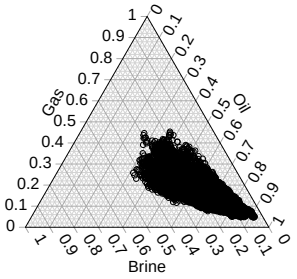
There is also a difference in the mean clay content, which shows higher values for the MCMC results compared to the results of Spremić et al. (2024). The mean clay content is high in the same areas, however, much higher in Figure 5.9 compared to the mean clay content in Figure 5.11.

Ternary plots at two locations in the Alvheim field for the two sets of samples from the two approximation methods are shown in Figure 5.12. Figure 5.12a and 5.12b show the $10,000$ values of MCMC samples from $\widehat{\pi}$ at locations $(800, 4800)$ and $(436, 5432)$ respectively. Ternary plots for the 100 approximate posterior samples from the ensemble based approach at location $(800, 4800)$ and $(436, 5432)$ are shown in Figure 5.12c and 5.12d, respectively. The depth at the location $(800, 4800)$ is 2112 meters and the depth at the location $(436, 5432)$ is 2190 meters.

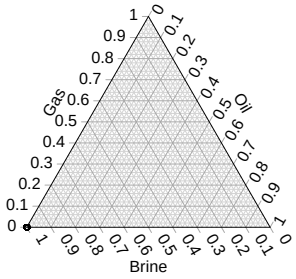
The ternary plots in Figure 5.12b and Figure 5.12d show that the obtained samples in the two methods give the same composition at the deep location. In the shallow location, however, there are some differences. The MCMC results show less uncertainty in the oil and gas saturations. The oil saturations were also higher in Figure 5.12a compared to Figure 5.12c. This agrees with the higher mean oil saturations in the MCMC results shown in Figure 5.8b compared to the mean oil saturations in Figure 5.10. The uncertainty of the brine saturations in Figure 5.12a and Figure 5.12c is about the same. The ternary plots are only from two locations, such that general conclusions cannot be drawn. However, they indicate that the MCMC samples had more oil and lower uncertainty in the shallow area compared to the results of Spremić et al. (2024), which agrees with results in Figure 5.8 and Figure 5.10.

Even though there are dissimilarities between the results from the two different methods for approximating the posterior π, the results show the same tendencies. For most of the areas where the MCMC result shows a high mean oil saturation and the ensemble-based approach shows a lower mean oil saturation, the ensemble-based approach also has a higher uncertainty of the mean oil saturation. That means that the two results are compatible with each other. There is an exception around inline 1250 and crossline

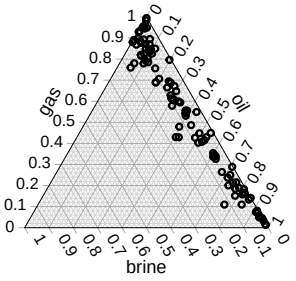
4900 where the MCMC results show a high mean oil saturation and the ensemble-based approximation to the posterior shows a low mean oil saturation and low uncertainty. A possible reason could be that this area is located at the edge of the area such that there is no AVO data to the right of this area in the conditioning in the ensemble-based approximation.



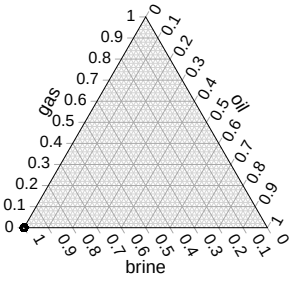
(a) Ternary plots for the MCMC samples from $\widehat{\pi}$ at location (800, 4800) in the Alvheim field.



(b) Ternary plots for the MCMC samples from $\widehat{\pi}$ at location (436, 5432) in the Alvheim field.



(c) Ternary plots for the approximate posterior samples from Spremić et al. (2024) at location $(800, 4800)$ in the Alvheim field.



(d) Ternary plots for the approximate posterior samples from Spremić et al. (2024) at location $(436, 5432)$ in the Alvheim field.

Figure 5.12: Ternary plots for samples from the two approximations to the posterior π in equation (3.8) at two locations at the Alvheim field. The depth in location (inline,crossline)= $(800, 4800)$ is 2112 meters while the depth at (inline,crossline)= $(436, 5432)$ is 2190 meters.

In the approach described in Spremić et al. (2024), the prior samples from the total area are divided into smaller patches when conditioning on the AVO data, to prevent false correlations between the AVO data which can occur when using such methods (Spremić et al., 2024). Two patch sizes were compared to each other, and when using larger patches, there was more oil in the results, which resembles the MCMC results more.

Spremić et al. (2024) added an iterative loop of the approximate posterior samples to improve the posterior approximation. The results indicated a possible increase in the integrated oil saturation, which resembles the tendencies in the results from MCMC.

Another explanation could be that $\widehat{\pi}$ has less uncertainty than π. Figure 5.5 showed that the approximate posterior samples had slightly lower uncertainty of the clay content. However, the uncertainty of the clay content is one of the similarities between the results from the MCMC and the method described in Spremić et al. (2024). The uncertainties of the gas and oil saturations in Figure 5.2 and in Figure 5.3 did not indicate that using the surrogate leads to a lower uncertainty. If it is the case that $\widehat{\pi}$ has less uncertainty than π it is not clear from the results in Section 5.1.

6

Closing Remarks

In this thesis, the forward function of the likelihood model in the Alvheim case was approximated by a MARS model to speed up the MCMC algorithm. The approximation was about 32 times faster. The model had a high correlation and low MSE. The goodness of fit was also confirmed by a comparison between MCMC samples obtained using both the exact forward model and the surrogate. Using the MARS model, the MH algorithms with the proposal distributions q_1, q_2, q_3 and q_4 were compared on a smaller part of the Alvheim field. A comparison of ESS per computation time showed that the MH algorithm with q_3 as proposal distribution was the most efficient algorithm in the Alvheim case. This algorithm was therefore used with the surrogate to perform MCMC on the entire Alvheim field. The MCMC results had similarities to the results in Spremić et al. (2024), however, the MCMC results showed more oil and less uncertainty for the gas and oil saturations.

For the experiments in this thesis, computation time was a limitation in the sense that the proposal distributions were not fine tuned for the particular problem at the Alvheim field. Instead, the results from previous work (Roberts et al., 1997; Roberts & Rosenthal, 1998; Cotter et al., 2013) were used as guidelines to tune the proposal distributions. Spending more time investigating the relationship between the acceptance rate and the ESS for the specific case at the Alvheim field, could lead to higher efficiency. However, for the random walk in the high-dimensional limit case in Roberts et al. (1997), the efficiency decreased slowly as the acceptance rate moved away from 23.4%, indicating that an acceptance rate of exactly 23.4% was not important. This suggests that one might not gain very much from fine-tuning the acceptance rate.

Computation time was also a challenge when using the exact forward model $\boldsymbol{h}$. To further investigate how similar $\widehat{\pi}$ is to π, it would be interesting to continue to compare the MCMC samples from $\widehat{\pi}$ to MCMC samples from π for other small areas. Dividing the area of $44,144$ grid points into areas of 900 grid points gives approximately 50 smaller areas, such that performing MCMC on all these areas is tedious, but some of the smaller areas could be investigated. It could be interesting to check some of the areas which are further from the wells, as there is approximately no influence by the likelihood model for the well-log data here. However, if the purpose is to increase the oil recovery near existing wells, this might not be attractive for oil entrepreneurs.

Using $\widehat{\boldsymbol{h}}_{\text{MARS}}$ as a surrogate opens for drawing samples from $\widehat{\pi}$ at the total area of the Alvheim field. The information from these samples could be used in several ways. First, one could use the mean saturations to narrow down a smaller area, which looks interesting for some reason, and perform MCMC sampling with the exact forward model there. Using an approximation to locate interesting areas could however be done using any approximation to the posterior. An advantage of using $\widehat{\boldsymbol{h}}_{\text{MARS}}$ in this way is that results

from the smaller area in Chapter 5 showed that MCMC samples from $\widehat{\pi}$ are very similar to the MCMC samples from π. On the other hand, other approximations to the posterior, such as the ensemble based method in Spremić et al. (2024), are faster than using MH with q_3 to approximate the posterior, which used around 20 hours to compute $500,000$ MCMC samples where every 50th sample is saved. Memory storage was also a limitation for the experiments performed in this thesis. Whereas 100 posterior samples were sampled in Spremić et al. (2024), $10,000$ MCMC samples were saved when performing MCMC on the Alvheim field in this thesis.

Another way to use the MARS approximation is to use the samples from $\widehat{\pi}$ to create an informed independent proposal. Khoshkholgh et al. (2021) found that using an independent proposal distribution created on the information of the posterior, increased the efficiency compared to a dependent standard random walk proposal distribution. An independent proposal could also run in parallel to speed up the computations.

An attempt was made to sample from π and $\widehat{\pi}$ simultaneously. The MH algorithm with proposal distribution q_3 was used to sample $500,000$ MCMC samples from $\widehat{\pi}$. The proposed sample was at each time iteration either accepted or rejected by the acceptance rate where $\widehat{\boldsymbol{h}}_{\text{MARS}}$ served as the forward model, as usual. In addition, at each 1000th time iteration, the proposed sample was either accepted or rejected with acceptance probability calculated with the exact forward model instead of the approximation to the forward model. The proposal distribution, q_3, at 1000th time iterations apart served as independent proposals changing as the Markov chain moved around the approximate posteriors' support. The acceptance rate was unfortunately zero for the exact MCMC sampling. An issue with combining dependent and independent proposals is that they have different optimal tuning. An independent proposal is more efficient if the acceptance rate is closer to one (Givens & Hoeting, 2013). For the MH algorithm with proposal distribution q_3 tuned such that the acceptance rate is approximately 23.4% the proposed samples might be too extreme/far away from the posterior for the independent proposal to accept them. This approach could therefore maybe work better with MALA which has a higher optimal acceptance rate.

Since the MARS model is non-parametric, it can in principle model functions of arbitrary shape. This suggests that MARS can approximate other forward models in other problems as well. The forward model in the Alvheim example takes four continuous input parameters, hence the covariate space is $\mathbb{R}^4$. If there are more covariates, one probably need more terms in the model, which would increase the computation time a bit. However, the input space of the MARS model which served as a surrogate in Chen et al. (2014) had dimension 15, and the MARS model still reduced the computation in that problem as well. The conclusion that MARS is a good surrogate model for any forward model in any problem can not be made. However, as the model is non-parametric, the approximation could work for similar problems with complicated and computationally inefficient forward models in a Bayesian setting with similar input spaces.

References

Ammon, C. J., Velasco, A. A., Lay, T., & Wallace, T. C. (2021). Chapter 12 - body waves and ray theory – travel times. In C. J. Ammon, A. A. Velasco, T. Lay & T. C. Wallace (Eds.), *Foundations of modern global seismology (second edition)* (Second Edition, pp. 339–362). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-815679-7.00020-3

Auestad, K. (2023). *Using markov chain monte carlo sampling to assess the oil and gas distribution at the alvheim field* (Project report, Norwegian University of Science and Technology).

Chen, M., Sun, Y., Fu, P., Carrigan, C. R., Lu, Z., Tong, C. H., & Buscheck, T. A. (2013). Surrogate-based optimization of hydraulic fracturing in pre-existing fracture networks. *Computers & Geosciences*, *58*, 69–79. https://doi.org/https://doi.org/10.1016/j.cageo.2013.05.006

Chen, M., Tompson, A. F., Mellors, R. J., Ramirez, A. L., Dyer, K. M., Yang, X., & Wagoner, J. L. (2014). An efficient bayesian inversion of a geothermal prospect using a multivariate adaptive regression spline method. *Applied Energy*, *136*, 619–627. https://doi.org/https://doi.org/10.1016/j.apenergy.2014.09.063

Christen, J. A., & Fox, C. (2010). A general purpose sampling algorithm for continuous distributions (the t-walk). *Bayesian Analysis*, *5*(2), 263–282. http://ba.stat.cmu.edu/journal/2010/vol05/issue02/christen.pdf

Cotter, S. L., Roberts, G. O., Stuart, A. M., & White, D. (2013). Mcmc methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, *28*(3). https://doi.org/10.1214/13-sts421

Efron, B., & Hastie, T. (2016). *Computer age statistical inference algorithms, evidence and data science*. Cambridge University Press.

Eidsvik, J., & Tjelmeland, H. (2006). On directional metropolis–hastings algorithms. *Statistics and Computing*, *16*, 93–106. https://doi.org/10.1007/s11222-006-5536-2

Friedman, J. H., & Roosen, C. B. (1995). An introduction to multivariate adaptive regression splines. *Statistical Methods in Medical Research*, *4*, 197–217. https://api.semanticscholar.org/CorpusID:6633418

Givens, G. H., & Hoeting, J. A. (2013). *Computational statistics* (2nd ed.). John Wiley & Sons Inc.

Gray, R. M. (2006). Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, *2*(3), 155–239. https://doi.org/10.1561/0100000006

Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100–108. Retrieved January 14, 2024, from http://www.jstor.org/stable/2346830

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning. data mining, inference, and prediction* (2nd ed.). Springer.

Holm-Jensen, T., & Hansen, T. (2019). Linear waveform tomography inversion using machine learning algorithms. *Mathematical Geosciences*, *52*. https://doi.org/10.1007/s11004-019-09815-7

IEA. (2022). World energy outlook 2022, iea. https://www.iea.org/reports/world-energy-outlook-2022

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning* (2nd ed.). Springer New York, NY.

Kamatani, K. (2020). Random walk metropolis algorithm in high dimension with non-gaussian target distributions. *Stochastic Processes and their Applications*, *130*(1), 297–327. https://doi.org/https://doi.org/10.1016/j.spa.2019.03.002

Kass, R. E., Carlin, B. P., Gelman, A., & Neal, R. M. (1998). Markov chain monte carlo in practice: A roundtable discussion. *The American Statistician*, *52*(2), 93–100.

Khoshkholgh, S., Zunino, A., & Mosegaard, K. (2021). Informed proposal monte carlo. *Geophysical Journaal International*, *226*, 1239–1248.

Malinverno, A., & Briggs, V. A. (2004). Expanded uncertainty qualifications in inverse problems: Hierarchical bayes and empirical bayes. *Geophysics*, *69*(4), 1005–1016.

Mosegaard, K., & Tarantola, A. (1995). Monte carlo sampling of solutions to inverse problems. *Journal of Geophysical Research*, *1001*, 12431–12448. https://doi.org/10.1029/94JB03097

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization*. Springer New York, NY.

Pinksy, M. A., & Karlin, S. (2011). *An introduction to stochastic modeling* (4th ed.). Elsevier Inc.

Pinski, F. J., Simpson, G., Stuart, A. M., & Weber, H. (2015). Algorithms for kullback–leibler approximation of probability measures in infinite dimensions. *SIAM Journal on Scientific Computing*, *37*(6), A2733–A2757. https://doi.org/10.1137/14098171X

Racine, J., & Li, Q. (2004). Nonparametric estimation of regression functions with both categorical and continuous data. *Journal of Econometrics*, *119*(1), 99–130. https://EconPapers.repec.org/RePEc:eee:econom:v:119:y:2004:i:1:p:99-130

Rimstad, K., Avseth, P., & Omre, H. (2012a). Hierarchical bayesian lithology/fluid prediction: A north sea case study. *Geophysics*, *77*(2), B69–B85.

Rimstad, K., Avseth, P., & Omre, H. (2012b). Hierarchical bayesian lithology/fluid prediction: A north sea case study. *GEOPHYSICS*, *77*(2), B69–B85. https://doi.org/10.1190/geo2011-0202.1

Robert, C., & Casella, G. (2011). A short history of markov chain monte carlo: Subjective recollections from incomplete data. *Statistical Science*, *26*(1), 102–115.

Robert, C. P., & Casella, G. (2004). *Monte carlo statistical methods* (2nd ed.). Springer.

Roberts, G. O., Gelman, A., & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk metropolis algorithm. *The Annals of Applied Probability*, *7*(1), 110–120.

Roberts, G. O., & Rosenthal, J. S. (2001). Optimal scaling for various metropolis-hastings algorithms. *Statistical Science*, *16*(4), 351–367.

Roberts, G. O., & Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *60*(1), 255–268. https://doi.org/10.1111/1467-9868.00123

Rudolf, D., & Sprungk, B. (2016). On a generalization of the preconditioned crank–nicolson metropolis algorithm. *Foundations of Computational Mathematics*, *18*(2), 309–343. https://doi.org/10.1007/s10208-016-9340-x

Spremić, M., Eidsvik, J., & Avseth, P. (2024). Bayesian rock-physics inversion using a localized ensemble-based approach — with an application to the alvheim field. *GEOPHYSICS*, *89*(2), R95–R108. https://doi.org/10.1190/geo2022-0764.1

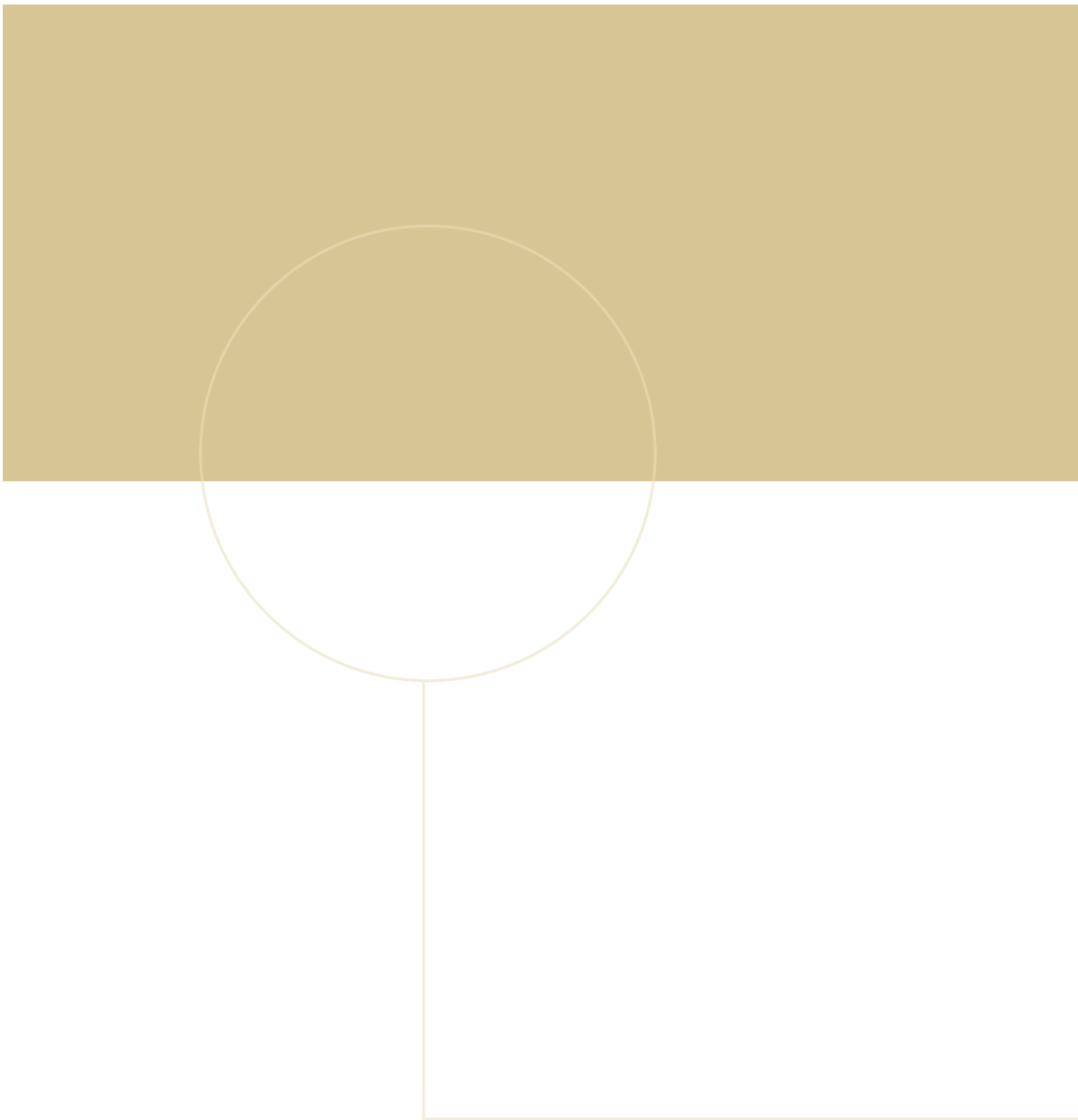
Ulrych, T. J., Sacchi, M. D., & Woodbury, A. (2001). A bayes tour of inversion: A tutorial. *Geophysics*, *66*(1), 55–69.

van Ravenzwaaij, D., Cassey, P., & Brown, S. D. (2018). A simple introduction to markov chain monte–carlo sampling. *Psychon Bull Rev*, *25*, 143–154.

Venables, B., & Ripley, B. (2002, January). Modern applied statistics with s. https://doi.org/10.1007/b97626

Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2012). *Probability and statistic for engineers and scientists* (9th ed.). Pearson Education, Inc.

Zeng, L., Shi, L., Zhang, D., & Wu, L. (2012). A sparse grid based bayesian method for contaminant source identification. *Advances in Water Resources*, *37*, 1–9. https://doi.org/https://doi.org/10.1016/j.advwatres.2011.09.011



NTNU

Norwegian University of
Science and Technology