Comparison of slowness vs. velocity perturbations in Bayesian seismic inversion

Bernd Trabi

Masterarbeit

Montanuniversität Leoben
Lehrstuhl für Angewandte Geophysik
Betreuer: Univ.-Prof. Dipl.-Geophys. Dr.rer.nat. Florian Bleibinhaus
Leoben, Mai, 2018

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ort, Datum Unterschrift

Abstract

A common problem in seismic tomography is to assess and quantify data uncertainties. The Bayesian approach to inverse problem by means of Markov Chain Monte Carlo (McMC) method samples relevant parts of the model space and provides an quantitative overview of the uncertainty of all model parameters. This method is very computing power intense and one important issue is to optimize the efficiency of the method. In this study, we investigate the difference between velocity-based and slowness-based McMC in refraction tomography. Whereas velocity in surface wave phase velocity inversions typically varies no more than by a factor of two, variations in refraction tomography can amount to a factor of ten, and the difference between slowness and velocity perturbations becomes more releveant. Because slowness is proportional to travel time, model perturbations need no arbitrary scaling relations. In our experiments, the associated perturbations are more uniform and show better mixing properties compared to velocity based McMC. We also investigate multivariate perturbations based on a projection of a single perturbation through the resolution matrix. Our tests show that these lead to higher acceptance ratios and/or greater step length.

Zusammenfassung

In seismischer Tomographie ist es üblicherweise sehr schwierig die Datenunsicherheiten abzuschätzen und zu bewerten. Dieser Bayesschen Ansatz zur inversen Theorie durch Markov Ketten Monte Carlo (McMC) beprobt relevante Bereiche des Modelraumes und liefert damit einen quantitativen Überblick über die Datenunsicherheiten aller Modelparameter. Diese Methode ist sehr rechenintensiv und ein wichtiger Aspekt ist es die Effizienz dieser Methode zu steigern. In dieser Studie wird der Unterschied zwischen geschwindigkeitsbasierter und langsamkeitsbasierter McMC in refraktionsseismischer Tomographie untersucht. Während die Geschwindigkeit bei Oberflächenwelleninversionen nicht mehr als um den Faktor Zwei variiert, können die Variationen in refraktionsseismischen Tomographien einen Faktor von Zehn ausmachen und der Unterschied zwischen langsamkeits- und geschwindigkeitsbasierten Perturbationen ist dadurch relevanter. Da die Langsamkeit proportional zur Laufzeit ist, benötigen die Modellperturbationen keine willkürliche Skalierungen. In unseren Experimenten sind die Perturbationen gleichmäßiger und zeigen bessere Mischungseigenschaften im Vergleich zur geschwindigkeitsbasierten McMC. Wir untersuchen auch multivariate Perturbationen, basierend auf der Auflösungsmatrix. Unsere Versuche zeigen, dass dadurch größere Akzeptanzraten und/oder größere Schrittweiten zustande kommen.

Contents

List of Figures

List of Tables

List of mathematical Symbols

M	number of model parameters
N	number of data
K	number of models in the chain
$\mathbf{m}^k$	M-dimensional vector of the model parameter values in the k^{th} position of the chain
$\boldsymbol{s}^k$	M-dimensional vector of slowness values in the k^{th} position of the chain
$\mathbf{v}^k$	M-dimensional vector of the velocity values in the k^{th} position of the chain
$\boldsymbol{\sigma}^m$	M-dimensional vector of the estimated standard deviation
$\boldsymbol{\sigma}^d$	N-dimensional vector of the estimated standard deviation of the observed data
$\mathbf{R}$	$M \times M$-dimensional Resolution matrix
$\mathbf{G}$	$M \times N$-dimensional data kernel
λ	empirical damping factor
$L(\mathbf{m})$	Likelihood function
$E(\mathbf{m})$	Cost function
p	probability distribution
n	normally-distributed random number
u	uniform-distributed random number
$\mathbf{g}(\mathbf{R})$	M-dimensional perturbation compensations
$\boldsymbol{\rho}(k)$	M-dimensional vector of the autocorrelation coefficient as a function of k
$\boldsymbol{\alpha}$	M-dimensional vector of acceptance rate
ESS	Effective Sample Size

Chapter 1

Introduction

One commonly used technique is the deterministic linearized approach to seismic travel time tomography. This approach provides a single model solution with inadequate uncertainties. It is very difficult to assess the quality and to quantify the uncertainties in this solution. Because the seismic inverse problem is non-linear a variety of models exist and this non-uniqueness is not captured by the linearized approach.
In contrast to this method the probabilistic approach is fully non-linear. It samples the whole model space and provides a quantitative overview of the uncertainty of the model parameters. When we have a large number of models which all explain the data more or less equally well we can answer some different interesting questions. Like, what is the chance to have a low velocity zone at a certain level, or in what depth it is likely to have a high velocity layer. These are questions that are important, for example, in exploration. For each model parameter we can plot a histogram and asses the distributions. One problem in Markov chains is convergence issue: When has a random walk visited enough points in the model space, so that the probability density is sufficiently sampled? This is a very broad topic which can not be answered with this thesis, but there are some tools to determine, or compare, the convergence speed of Markov Chains. The Bayesian approach to inverse problem is a very computing power and time intense method, and how long to run a Markov chain depends on the type of problem and is difficult to determine.
In this thesis the efficiency of slowness based Marcov chains will be compared to velocity based Markov chains. To evaluate the efficiency we made test runs with a well known synthetic model and also with real refraction seismic data from the Salzach valley. While the synthetic model is a known model, at the real data test model we can still compare the performance of the Markov chains. The model parameters for inversion have to be carefully selected and

this is a quite challenging iterative process. While for the known synthetic test model it is quite easy to parametrize the model, it is more difficult for the Salzach test model. But at least there are some studies with reflection-and-refraction travel time tomography (Bleibinhaus et al., 2010) and more detailed images derived by acoustic full waveform inversion (Bleibinhaus & Hilberg, 2012) from the Salzach valley to evaluate the result.

Chapter 2

Inverse Theory

The inverse theory is a method of estimating model parameters from data. The fundamentals of that topic can be read in several books like Menke (1989), and Tarantola (2005), Shearer (2009) or Aster et al. (2013) and there is many more literature. In this chapter I will give only a very rough overview for a better understanding of the following chapters.

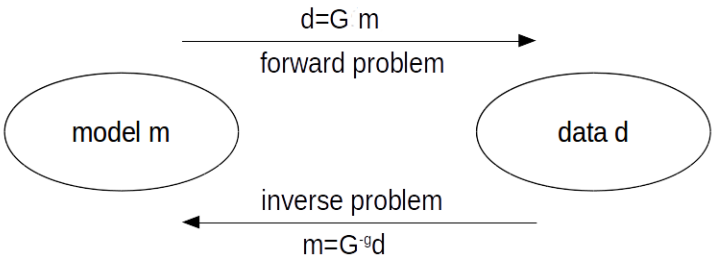


Figure 2.1: *Schematic representation of the inverse problem*

2.1 Inverse Problem

Overall, the inverse problem is to find a model $\boldsymbol{m}$ that is consistent with the observed data $\boldsymbol{d}$ (Figure 2.1). $\boldsymbol{G}$ is a $N \times M$ Jacobi matrix ($\frac{\partial d}{\partial m}$), where N is the number of data and M is the number of model parameter. $\boldsymbol{G}^{-g}$ is the generalized inverse, because $\boldsymbol{G}^{-1}$ can not generally be computed. The forward problem describes the opposite direction and the forward operator $\boldsymbol{G}$ connects the data and model parameters through a physical theory (eg.

calculating the travel time with an existing model). Most geophysical problems are non-linear to some degree and they get solved through a sequence of small linear inverse steps. With the damped least squares inversion it is possible to approximate a non-linear problem with small iterative linear steps that converge to a final model. This single solution does not reflect the uncertainties of the model parameters and its non-uniqueness. Because of that non-uniqueness a variety of models can describe the observed data within the given measurement accuracy.

2.2 Deterministic Methods

Some fundamentals of the deterministic inversion will be discussed in this chapter. It is relevant for this thesis because the deterministic solution will be used as starting model for the probabilistic inversion and is also necessary for calculating the resolution matrix. As starting model for the Markov Chain any model can be chosen, but the deterministic solution has the advantage that it shortens the burn-in phase. One can assume that the solution of the deterministic inversion is close to the equilibrium distribution (Fontanini, 2016).

2.2.1 Damped Least-Squares Solution

Most geophysical problems are mixed-determined, this means some model parameters are over determined while others are under determined. This problem demands a reasonable balance between simplicity of the solution and data fit. This can be achieved with the damped least squares solution:

$$\boldsymbol{G}^{-g} = [\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I}]^{-1}\boldsymbol{G}^T \tag{2.1}$$

where $\boldsymbol{I}$ is the identity matrix and λ is an empirical damping factor, that weights the relative importance of errors and solution norm (Gubbins, 2004). λ can be determined empirically with an trade-off test. A big damping factor generates a simple model, which is not very detailed. Small damping values generate complex models that overestimate the given limited data. With the linearisation assumptions where made that do not reflect the reality and this could lead to an inaccurate result.

2.2.2 Resolution Matrix

The resolution matrix $\boldsymbol{R}$ stems from the deterministic solution and describes the relation between the true model and the damped calculated model. It

will be used for the multivariate updating scheme. It is a square $M \times M$ symmetric matrix, where M is the number of parameters. The resolution matrix is defined as

$$\boldsymbol{R} = \boldsymbol{G}^{-g}\boldsymbol{G} \tag{2.2}$$

where $\boldsymbol{G}^{-g}$ is the general inverse and $\boldsymbol{G}$ the data kernel. An example is given in figure 2.2. The off-diagonal elements not equal zero show the dependency with the diagonal elements. Without damping the resolution matrix would be the unit matrix. The resolution matrix for the damped least square solution is calculated with

$$\boldsymbol{R} = (\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I})^{-1}\boldsymbol{G}^T\boldsymbol{G} \tag{2.3}$$

The damping number λ will be taken from the damping test of the velocity based deterministic inversion.

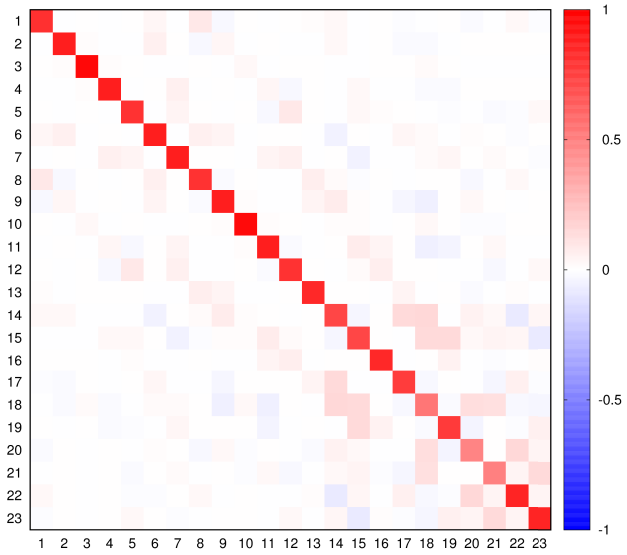


Figure 2.2: *A graphical representation of an exemplary resolution matrix*

2.3 Probabilistic Methods

Probabilistic methods sample the model space using random perturbations. These methods are totally non-linear and provide a quantitative overview of the uncertainties of all model parameters. The simplest method is a exhaustive search which explores the whole model space. It randomly samples

and evaluates the models. This method is very computing power intense. Another strategy is to sample just the important parts of the model space with the Bayesian approach.

2.3.1 Bayesian Inference

In the Bayesian approach

$$p(\boldsymbol{m}|\boldsymbol{d}^{obs}) = \frac{p(\boldsymbol{d}^{obs}|\boldsymbol{m})p(\boldsymbol{m})}{p(\boldsymbol{d}^{obs})} \tag{2.4}$$

$p(\boldsymbol{m}|\boldsymbol{d}^{obs})$ is the probability density we desire or the posteriori, the probability that the model is correct given a data set $\boldsymbol{d}$,
$p(\boldsymbol{d}^{obs}|\boldsymbol{m})$ is called the likelihood function that measures the level of fit between measurements and the prediction made using the model $\boldsymbol{m}$
$p(\boldsymbol{d}^{obs})$ is the probability that the data is observed and
$p(\boldsymbol{m})$, the priori is any kind of information on the model $\boldsymbol{m}$ that we can include in our inversion process, that are independent from our measurements: for example previous studies, physical geological knowledge or limiting of the velocity field.

2.3.1.1 The likelihood function

The likelihood function (Equation 2.5) is a function which quantifies the ability of a model to fit the observed data.

$$L(\boldsymbol{m}) = p(\boldsymbol{m}|\boldsymbol{d}^{obs}) = \frac{1}{\prod_{i=1}^{N}(\sigma_i^d\sqrt{2\pi})}\exp[-E(\boldsymbol{m})] \tag{2.5}$$

The Cost function $E(\boldsymbol{m})$ (Equation 2.6), is a weighted L_2 misfit of the observed data d_i^{obs} and the calculated data d_i^{pre} where σ_i^d is the estimated data uncertainty.

$$E(\boldsymbol{m}) = \frac{1}{2}\sum_{i=1}^{N}\left(\frac{d_i^{obs} - d_i^{pre}}{\sigma_i^d}\right)^2 = \frac{1}{2}\sum_{i=1}^{N}\left(\frac{d_i^{obs} - \boldsymbol{G}\boldsymbol{m}}{\sigma_i^d}\right)^2 \tag{2.6}$$

2.3.2 Markov Chain Monte Carlo

A Markov chain is a stochastic process that produces a sequence of variables or models where each model is just dependent of the previous one. Every new sample in the chain is created through a small random perturbation of the previous one. The core of the Markov chain is the Metropolis-Hastings algorithm.

2.3.2.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm was develeoped by Metropolis & Ulam (1949), Metropolis et al. (1953) and Hastings (1970). It is used to obtain a sequence of random samples from a probability distribution for which direct sampling is difficult. The trial model gets compared to the current model via its ratio. The schematic application of the Metropolis-Hastings algorithm in the code is used as follows:

$\gamma = \frac{L(\boldsymbol{m}_{trial})}{L(\boldsymbol{m}_{curr})}$

if $\gamma \geq 1$, accept proposed $\boldsymbol{m}_{trial}$
if $\gamma < 1$, a random number $u \in U[0, 1]$ gets generated
if $\gamma > u$, accept $\boldsymbol{m}_{trial}$
else reject $\boldsymbol{m}_{trial}$

The efficiency of the algorithm depends strongly on the step size of the model perturbation. If the model perturbation is small the distance of one model to the other one is also small, and the model is very likely to get accepted. With a too big perturbation the acceptance rate is low and many models get rejected.

2.4 Perturbation

In velocity domain the perturbation has to be scaled to a proper size, because small perturbations at shallow model parameters, where the velocity is usually small, lead to relative big changes in travel time and hence to big differences of the likelihood. The new proposed model is very unlikely to get accepted. At deeper model parameters perturbations of high velocity model parameters lead to small changes in likelihood and therefore the proposed model is more likely to get accepted. Furthermore deeper model parameters have typically longer offsets, and the offset dependent weighting also leads to a higher acceptance. In slowness domain the change in slowness is direct

proportional to the travel time and hence the change of the likelihood, which is also a function of the weighted travel time residuals. This means that the perturbation size in slowness domain does not need to be scaled. Therefore the estimated standard standard deviation of the model parameters σ_i^m is proportional to the change of travel time:

$$\sigma_i^{slowness} \propto \Delta t \tag{2.7}$$

The slowness based model parameters get perturbed with an Gaussian density distribution and the standard deviation should be set to achieve an appropriate acceptance ratio between 20 - 30% (Gelman et al., 1996). Roberts et al. (1997) suggest in their paper an acceptance rate of 23.4% for "optimal efficiency".

2.5 Compensation of the perturbations

In Metropolis-Hastings algorithm, normally one parameter gets perturbed and only one model parameter of the model vector changes:

$$\boldsymbol{m}^{trial} = \boldsymbol{m} + n\sigma_i^m \boldsymbol{e}_i \text{ for } n \in N[0,1] \tag{2.8}$$

where $\boldsymbol{m}$ is the current model, $\boldsymbol{m}^{trial}$ the trial model and σ_i^m is the standard deviation of the i^{th} model parameter. The variable n is normally-distributed random number with unity standard deviation. For slowness based perturbation σ_i^m gets exchanged with σ^m, because the standard deviation can be assumed to be the same at every model parameter, because the change of the model parameter is direct proportional to the change of travel time and therefore proportional to the change of likelihood.
In the multivariate updating scheme developed of Fontanini (2016) the applied perturbation gets compensated:

$$\boldsymbol{m}^{trial} = \boldsymbol{m} + n\sigma_i^m \boldsymbol{e}_i - an\sigma_i^m \boldsymbol{g}(R_{ij}) \tag{2.9}$$

where $\boldsymbol{g}(R_{ij})$ gets computed by using some functional of the resolution matrix and a is a scaling factor of the applied compensation. These compensations were applied to the other model parameters in the opposite direction. It allows bigger perturbation size while the change of likelihood is still kept small and the acceptance rate is therefore higher. Fontanini (2016) introduced four different functional:

$$\text{Functional 1:} \quad \boldsymbol{g}(R_{ij}) = \sum_{j \neq i}^{M} R_{ij} \boldsymbol{e}_i \tag{2.10}$$

$$\text{Functional 2:} \quad \boldsymbol{g}(R_{ij}) = \sum_{j \neq i}^{M} R_{ij} R_{ii} \boldsymbol{e}_i \tag{2.11}$$

$$\text{Functional 3:} \quad \boldsymbol{g}(R_{ij}) = \sum_{j \neq i}^{M} \frac{R_{ij}}{R_{ii}} \boldsymbol{e}_i \tag{2.12}$$

$$\text{Functional 4:} \quad \boldsymbol{g}(R_{ij}) = \sum_{j \neq i}^{M} \frac{R_{ij}}{\sum_{j \neq i}^{n} R_{ij}} \boldsymbol{e}_i \tag{2.13}$$

Tests showed that functional 3 leads to the best results (Tauchner, 2016).

2.6 Comparison of Markov Chains

One important question is how long to run a Markov chain in order to obtain observations from a stationary distribution. Was the runtime long enough to have a sufficient number of models? Which Markov chain converges faster? This are very difficult questions to answer and that is a very broad topic, but we can do several things to investigate this issue. In this thesis just some of the most important aspects will be discussed.

2.6.1 Acceptance rate

As rule of thumb the acceptance rate of an efficient Markov chain should be between 20 and 30% (Gelman et al., 1996). New generated models with very small perturbations have a similar good likelihood and a high chance to get accepted. With small perturbation the Markov chain explores very slow the model space, because a lot of chain members were needed to generate uncorrelated models. On the other hand a high step length and therefore a small acceptance rate produces fast uncorrelated models. The algorithm wastes a lot of calculation time for models that get rejected. In Both cases

the Markov chain is likely to get stuck in local minima. Consequently the acceptance rate should be maintained in the suggested range. Also a higher acceptance rate is desirable, because with a higher acceptance rate the perturbation size and step size can be further increased.
To make those two different perturbation methods comparable the perturbation was set to a specific value to achieve nearly the same acceptance rate. The Markov chain with the bigger step size is therefore more efficient, when comparing the two chains with the same acceptance rate together. The step size from slowness and velocity based Markov chains where both calculated in slowness domain. Hence the slowness values were used to determine the distance from one model to the other.

2.6.2 Acceptance rate at each model parameter

The acceptance rate for each model parameter is also very important, because some Markov chains with an overall good acceptance rate seems to perform well, but there can still be some model parameters that lie outside of the recommended range of the acceptance rate. This can lead to a situation where the forward solver wastes a lot of calculation time for perturbations that get rejected anyway. At the same time the Markov chains needs a lot of steps for other model parameter to get uncorrelated parameter. A look at each model parameter is done to determine how many perturbations of a certain model parameter get accepted. In the formula

$$\alpha_j = \frac{K_j}{K_j^{trial}} \tag{2.14}$$

the acceptance rate α_j is the acceptance rate of the model after perturbing the j^{th} model parameter. K_j is the number of accepted model and K_j^{trial} is the total number of trial models after perturbing the j^{th} model parameter. In a good Markov chain the acceptance rate of all model parameter should be the same size.

2.6.3 Distance between two models

The distance between two models is defined as the L_2-norm:

$$||\Delta s|| = \sqrt{\sum_{j=1}^{M} \Delta s_j^2} \tag{2.15}$$

$||\Delta s||$ is the distance between two models expressed in slowness. s_j are the slowness parameters. Basically it measures the distance of one model to the other model one. A Markov chain with a bigger step size samples the model space faster and is therefore desirable. Thus, for reason of comparability the distances in velocity domain were also calculated in slowness.

2.6.4 Simple Graphical Methods

In this thesis some simple graphical methods were used to assess the convergence of a Markov chain. They provide a quick overview of the convergence and are very simple to implement.

2.6.4.1 Trace plots

To compare the efficiency of the Markov chain trace plots of single parameters are very useful. It allows us to see if certain parameters have sufficient state changes and shows the values the parameters took during runtime of the chain. It also allows us to see if a parameter wanders around of its mean value, which indicates that the chain has converged. In the synthetic test it will be expected to bounce around the known real value of the model parameters. Trace plots give a good and quick qualitative overview of the performance of a Markov chain. Even Markov chains with an theoretical good acceptance rates between 20-30% can have bad mixing properties which can be quick identified by looking at trace plot. One parameter can have too big proposals that gets often rejected while at the same time another parameter can have narrow proposals that nearly always get accepted. These time series plots give a quick overview about the mixing properties.

2.6.4.2 Autocorrelation function

The autocorrelation function standardizes the values of the autocovariance and is a given by

$$\rho_j(k) = \frac{\sum_{l=1}^{K-k} (m_j^l - \bar{m}_j)(m_j^{l+k} - \bar{m}_j)}{\sum_{l=1}^{K} (m_j^l - \bar{m}_j)^2} \tag{2.16}$$

where K is the number models and $\bar{m}_j$ the overall mean of the j^{th} model parameter. The equation is approximated for reasonably large K. The correlation coefficient $\rho_j(k)$ shows the correlation of a variable j at position l

and $l + k$ in the chain, where k is the lag between both variables. High autocorrelation within chains indicate slow mixing and therefore slow convergence. The autocorrelation shows how correlated a parameter with itself at a lag k is. In the example of an autocorrelation plot (Figure 2.3) it shows that the first uncorrelated lag is approximately at 15. The green lines represent the confidence interval of 95% and autocorrelation values beyond this interval were considered to be significant. For later analysis just the first uncorrelated lag will be from interest.

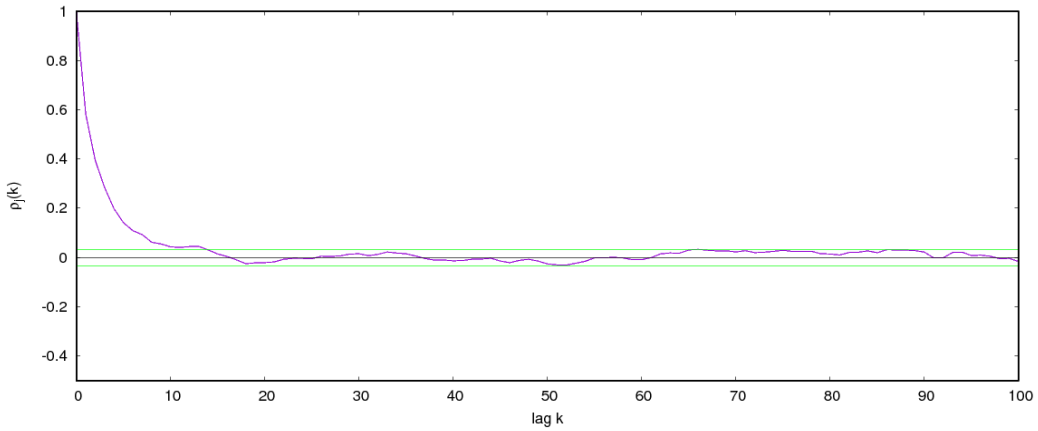


Figure 2.3: *The autocorrelation plot of a single parameter.*

With the autocorrelation function the Effective Sample Size (ESS) can be calculated. This heuristic method was proposed by Radford Neal in Kass et al. (1998). The ESS is usually defined as

$$ESS_j = \frac{K}{1 + 2\sum_{k=1}^{\infty} \rho_j(k)} \tag{2.17}$$

where K is the number of samples in the chain. Practically the summation to infinity will be truncated, when the autocorrelation $\rho(k)$ is below the confidence interval. The effective sample size measures the approximate number of independent samples in a group of partially dependent ones. It is a standard sample quality measure based on asymptotic variance (Brooks et al., 2011).

2.6.4.3 Cumulative Mean

Also a good visual indication for the convergence of a Markov chain are cumulative mean plots for single parameters. For every parameter the cu-

mulative mean will be computed as the mean of all samples values up to and including that given iteration (Smith, 2001). A a value of a chain which has not reached its stationary distribution could may change after a long run time, when the Markov chain enters or leaves a local minimum. By the fact that the cumulative mean gets divided by an ever rising number, the mean will always stabilize.

Chapter 3

Model Parametrization

For this work the simulr16 code by Bleibinhaus (2003) which was modified by Fontanini (2016) to run Markov Chains algorithms was used. The code was reprogrammed to perturb in slowness domain.

3.1 Inverse grid

A model in the simulr16 framework is parametrized by a set of irregularly distributed velocity nodes, which where set by the user. The black crosses in figure 4.2a represent the model parameters for the inversion grid.

3.2 Forward grid

For the forward modelling a finite-differences-eikonal solver of Vidale (1990) with modifications of Hole & Zelt (1995) is used, because it is fast and calculates the first arrivals with sufficient accuracy. The eikonal solver needs a fine rectangular sub grid and this is calculated in two steps. First a coarse regular grid gets interpolated from the irregularly distributed nodes with a nearest neighbour interpolation. In the second step a fine sub grid gets interpolated with a bilinear interpolation.

3.3 Slowness vs. velocity and the interpolation problem

If the Bayesian inversion is done in slowness domain it would be consistent to interpolate the fine sub grid for the eikonal solver also in slowness domain. The eikonal solver would also not require to convert the fine velocity grid in

a slowness grid to calculate the time grid. In this work the slowness perturbations get compared with velocity perturbations and a linear interpolation of slowness and velocity values at the same grid would lead to different results, because slowness is by definition the reciprocal of velocity. Therefore a linear interpolation of slowness is equivalent with a harmonic interpolation of velocity which would lead to smaller interpolation values. As result the velocity values of the model parameter would be shifted slightly to a higher value. These model parameter with the higher values and lower interpolated values would lead to a different result for the travel time calculation and hence a different likelihood. This reciprocal error prohibits a comparison of these two grids with each other.
To this issue it was decided to interpolate in velocity domain linear and in slowness domain harmonic, because velocity is more common in geophysics and it was easier to compare the results during the modification of the code to previous results.

Chapter 4

The Synthetic Test

4.1 Test model

The test model is a known synthetic seismic model (Figure 4.1) of Fontanini (2016) and it has a 3-layered structure. The total lenght of the model is 120m with a maximum depth of 36m. The acquisition geometry is 12 sources and 23 receivers evenly distributed on the surface with a 5m spacing . The synthetic travel times have been computed with the FAST algorithm (Zelt & Barton, 1998) and a gaussian random noise has been added to the data using a standard deviation of 5% of the noiseless travel time (Fontanini, 2016). The model parametrization from Fontanini (2016) was adapted, but got modified.

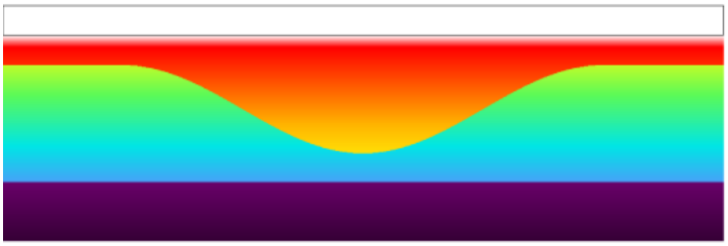
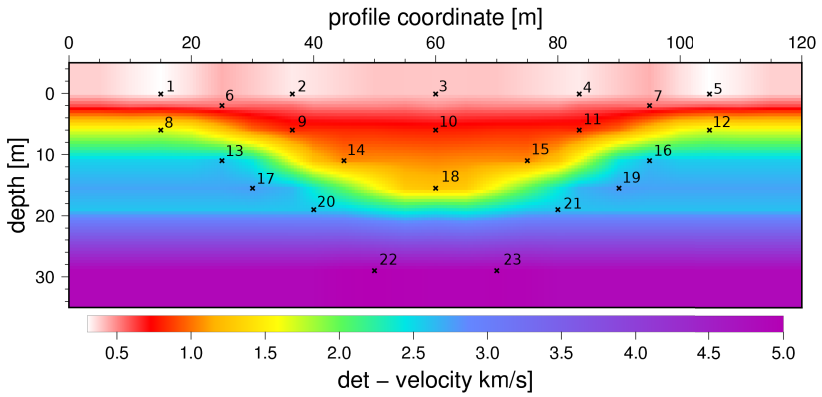


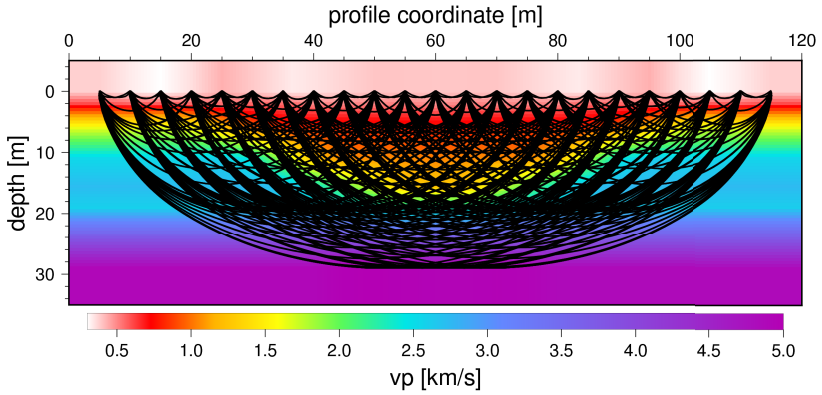
Figure 4.1: *The synthetic test model (*Fontanini*, 2016).*

Two lowest model parameter got removed and the model parameters above got slightly shifted downwards. The previous parametrization made the rays deflect just above the very lowest model parameters which allowed them to accept arbitrary values. Figure 4.2 shows the deterministic solution and the ray coverage from the synthetic test model. The model parameter get referred by numbering from left to right and top to bottom. Model parameter 1 lies on the very top at the left side and 23 is the lowest right mode parameter.

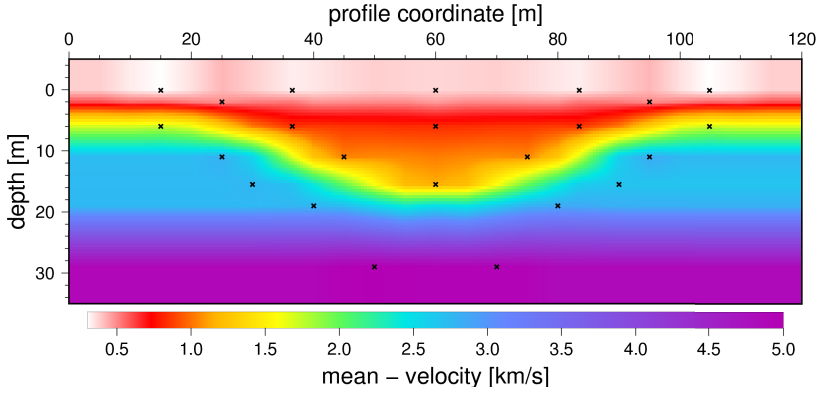
The estimated standard deviation of the data to calculate the likelihood was set 0.5 ms for σ^d_{min} and 5.0 ms for σ^d_{max} which refers to the minimum and maximum offset. Values for offsets in between were linear interpolated. Travel times values from large offsets are more uncertain than from small offsets and they are less constrained, so they are weighted by this data standard deviation. The start model of the Markov chain is the deterministic solution (Figure 4.2c). The test runs with compensations were set to the same settings and use the functional 3 (Equation 2.12). This functional shows the best performance.



(a) *Deterministic Solution*



(b) *Ray coverage*



(c) *The mean model of the probabilistic inversion*

Figure 4.2: *The deterministic solution (a) of the synthetic test model of* Fontanini *(2016) with the model parametrization, its ray coverage (b) and the mean model(c).*

4.1.1 Prior and perturbation scaling

In velocity domain Fontanini (2016) pointed out, that the magnitude of perturbation has a fundamental influence on the performance of Metropolis-Hasting based Markov chain Monte Carlo algorithms. To ensure good mixing properties it has to be carefully scaled. The main advantage in slowness domain is that there is no need for an empirical perturbation scaling. Just a scaling of the global perturbation size has to be done to achieve a reasonable acceptance rate. The applied priors were summarized in table 4.1:

Prior	Slowness domain	Velocity domain
$(m_j)_{min}$	0.1 s/km	0.1-1.0 km/s
$(m_j)_{max}$	3.33 s/km	1.0-6.0 km/s
σ_j^m	0.14 s/km	0.055-0.305 km/s

Table 4.1: *The prior information.*

Velocity domain

For velocity based perturbations a low-informative prior was used which limits the p-wave velocity to a reasonable depth-depended range (Table 4.1). The minimum and maximum velocity at the top and at the bottom of the model is set by the user and the limits in between get linear interpolated. Here the model parameters get perturbed with an depth dependent Gaussian density distribution. Depth dependent standard deviation is calculated with the formula:

$$\sigma^m(z) = c\Delta v(z) \tag{4.1}$$

$\Delta v(z)$ is the prior velocity range at each depth and c is a global perturbation constant which has to be set. A proper scaling increases the acceptance rate of shallow model parameters and decreases it at the deeper model parameters. The main disadvantage is, that a suitable velocity prior for optimal scaling assumes good prior knowledge. Nonetheless it can be estimated very roughly or the prior knowledge can come from other independent measurements. Therefore a velocity prior was set in the configuration file with $v_{min} = 0.1\,\mathrm{km/s}$ and $v_{max} = 1.0\,\mathrm{km/s}$ at the top and $v_{min} = 1.0\,\mathrm{km/s}$ and $v_{max} = 6.0\,\mathrm{km/s}$ at the bottom. The factor c (Equation 4.1) was set to 0.061. With these settings an acceptance rate of 22.95% was achieved, which is comparable to the accepatance rate of the slowness based Markov chain. All important settings were summarized in table 4.1.

Slowness domain

For slowness based Markov chains the prior is even less informative. Just the minimum possible slowness will be set by the user and for the maximum value the slowness in air gets assumed. The prior was set to an minimal slowness of 0.1 s/km which correspondents to an velocity of 10 km/s and to 3.33 s/km, the slowness in air as maximum.
The standard deviation of the perturbation size was set to 0.014 s/km to achieve an acceptance rate of about 23.04%. The perturbation size is the same at every model parameter, because the change in slowness is direct proportional to the change in travel time.

4.2 Comparing slowness vs. velocity

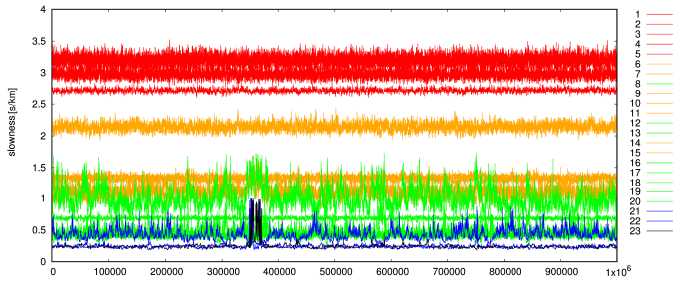
4.2.1 Step size

The average Euclidean Distance between two models is 0.088 s/km for the slowness based and 0.063 s/km for the velocity based Markov chain. This is an improvement for the step length of nearly 40% and it shows that in slowness domain the average step size is larger while both chains have the same acceptance rate. Consequently the slowness based Markov Chain explores the model space more efficient.

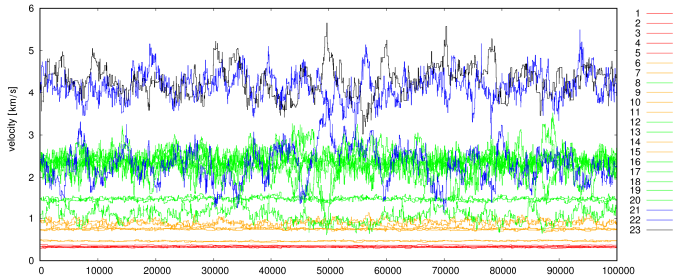
4.2.2 Trace plots

4.2.2.1 Plots of the model parameters

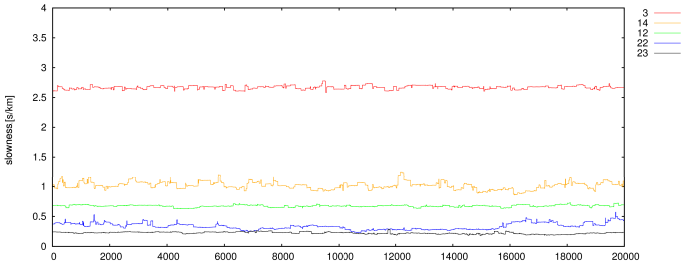
The trace plots from the slowness based Markov chain (Figure 4.3a) show the slowness of all 23 model parameters over 1 million iterations. The perturbation pattern is uniform, every parameter gets more or less equal frequent perturbed and the magnitude of perturbation size does not change with deeper model parameters. In comparison to the velocity based perturbations (Figure 4.3b) there is a very big variation at deeper model parameters, while the variation and frequency of change from shallow model parameters is very low.



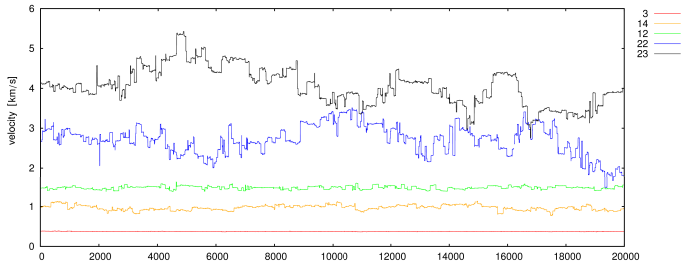
(a) *Slowness domain*



(b) *Velocity domain*



(c) *Slowness domain*

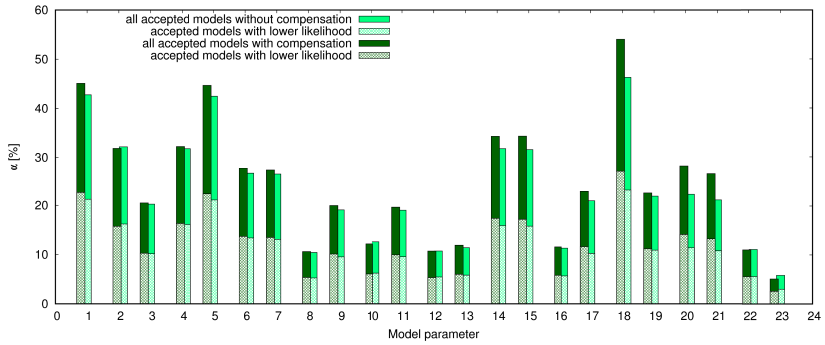


(d) *Velocity domain*

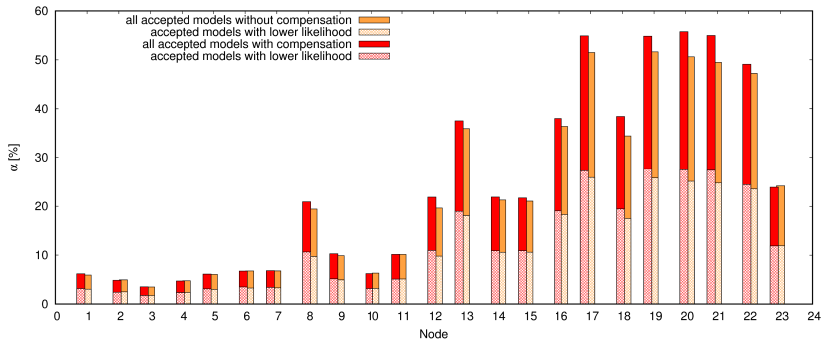
Figure 4.3: *Figure (a) and (b) show the trace plots of all model parameter and figure (c) and (d) show the first 20,000 iterations of some chosen model parameter at certain depths.*

It can be better seen in figure 4.3c and figure 4.3d, where some model parameters at certain depths were highlighted for the first 20,000 iterations. Figure

4.4 reflects the frequency of change and figure 4.5 reflects the magnitude of variation over 1 million iterations.
The acceptance rates of certain model parameters show very different results in slowness and velocity domain (Figure 4.4). In an optimal Markov chain all acceptance rates should approximately have the same value. The model parameters with the worst acceptance rates were the weakest members in the chain. The parameters with too low or too high acceptance rates are not sufficient sampled, and the applied perturbation size is not appropriately scaled. Slowness domain (Figure 4.4a) in comparison to velocity domain (Figure 4.4b) shows that acceptance rates of all model parameters are more uniform. In velocity domain, the well constrained shallow model parameters get very seldom accepted, because the perturbation size at these model parameters is probably too big. This reflects the problem of empirical perturbation scaling where the size was not ideally adjusted. It would be a matter of improved perturbation scaling which has to be applied to every model parameter as accurate as possible to perform optimal acceptance rates to every model parameter in velocity domain. But this would require prior knowledge of every model parameter in advance. The deeper model parameter get very often perturbed in velocity domain which has to do with a relative too small perturbation size. Judging from this the Markov chain has bad mixing properties. On the one hand a very low acceptance rate with relative too big perturbations and on the other hand we have relative small perturbations with high acceptance rates, even though the overall acceptance rate appears to be in the optimal suggested range. A more detailed look at figure 4.4a shows that model parameter 1, 5 and 18 get more often perturbed and are probably the least constraint model parameters in the model. Indeed model parameter 1 and 5 lie at the very top right and left corner of the model, and model parameter 18 just at the top of the synclinal structure, where there are only a few rays. In slowness domain it can be seen that poorly constrained model parameters, like that at the margin of the model or in the low velocity area in the synclinal structure get more often perturbed. Which can lead to the question if the perturbation at this model parameters could be increased further. Perturbations of model parameter 22 and 23 seem get seldom accepted even one may think they are very poorly constrained, because just rays from far offset hit this region. The perturbation size at this model parameter seems to be relatively too big. The perturbation size is not optimally scaled even in slowness domain.



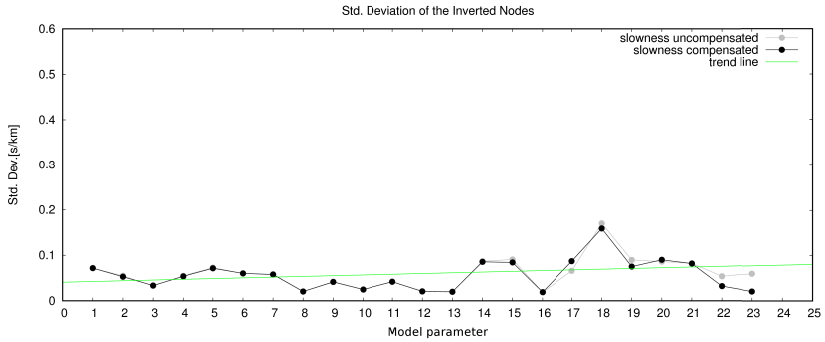
(a) *Slowness domain*



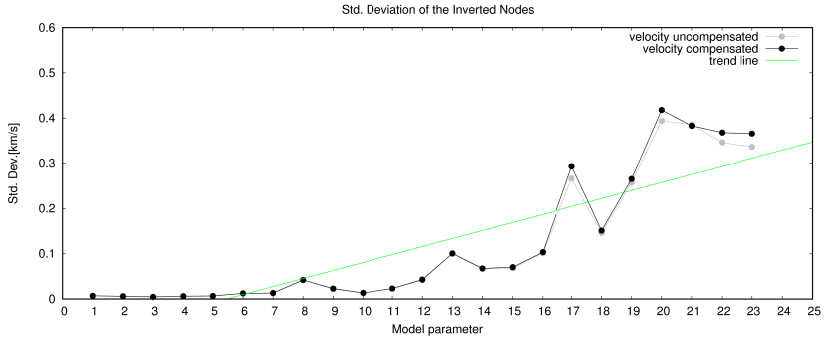
(b) *Velocity domain*

Figure 4.4: *The acceptance rate α for each model parameter*

Not only the frequency of change increases with depth in velocity domain, also the standard deviation from the parameters (Figure 4.5). While the standard deviation in velocity domain is proportional to the velocity value of the model parameter (Figure 4.5b) the standard deviation of the slowness perturbation (Figure 4.5a) does not increase with depth, bearing in mind that the slowness values from the parameters decrease with depth. The poorly constraint model parameter 18 within the synclinal structure has the highest standard deviation.



(a) *Slowness domain*

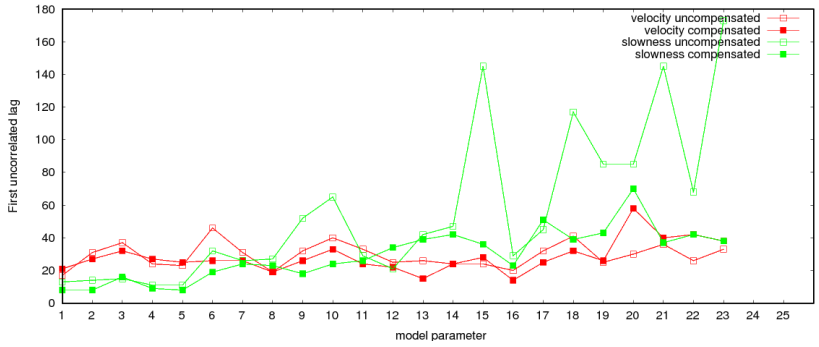


(b) *Velocity domain*

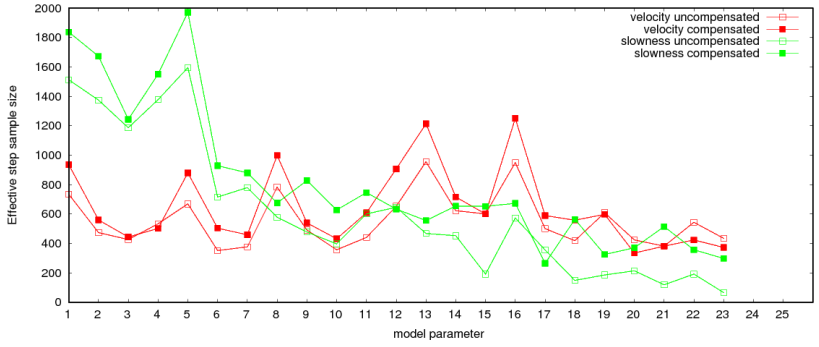
Figure 4.5: *The standard deviation of the model parameters in slowness (a) and velocity (b) domain*

4.2.2.2 Autocorrelation

Figure 4.6a compares the first uncorrelated lag of slowness and velocity based Markov chains for all model parameters. A smaller lag means that a certain parameter is earlier uncorrelated and is therefore desirable. Especially at the shallow model parameters in slowness domain the model parameter are much more uncorrelated. At deeper model parameters velocity based Markov chains get slightly better. While the difference for uncompensated chains at deeper model parameter is quite big, the difference for the compensated chains is quite insignificant. It has to be considered that in velocity domain, most of the accepted models stem from the deeper parts of the model and even with much more samples the effective sample size is about the same order like in slowness domain (Figure 4.4).



(a) *First uncorrelated lag*

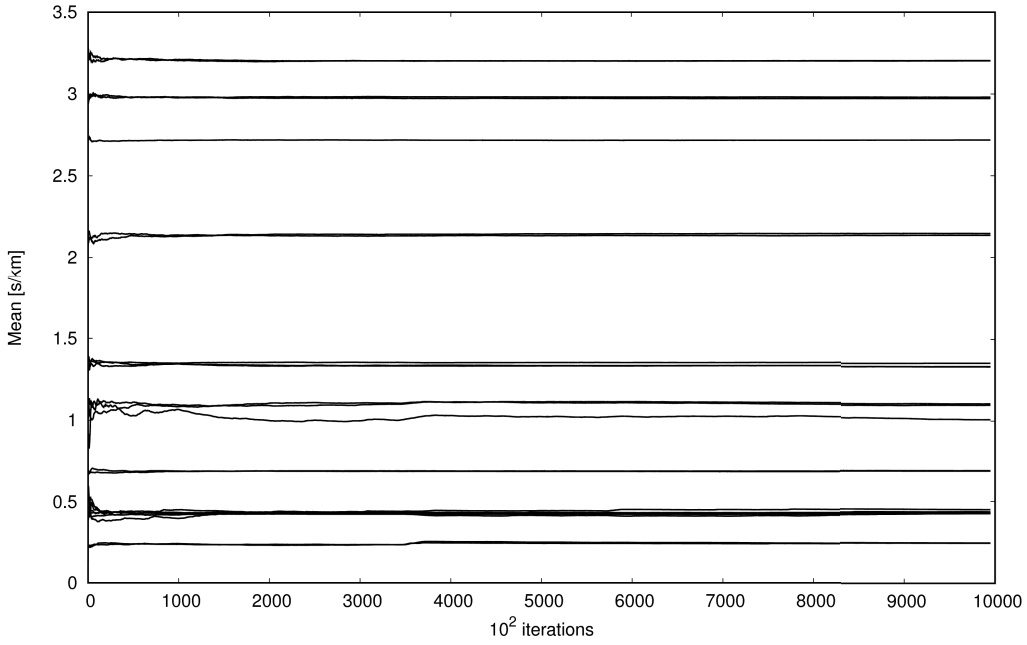


(b) *Effective sample size*

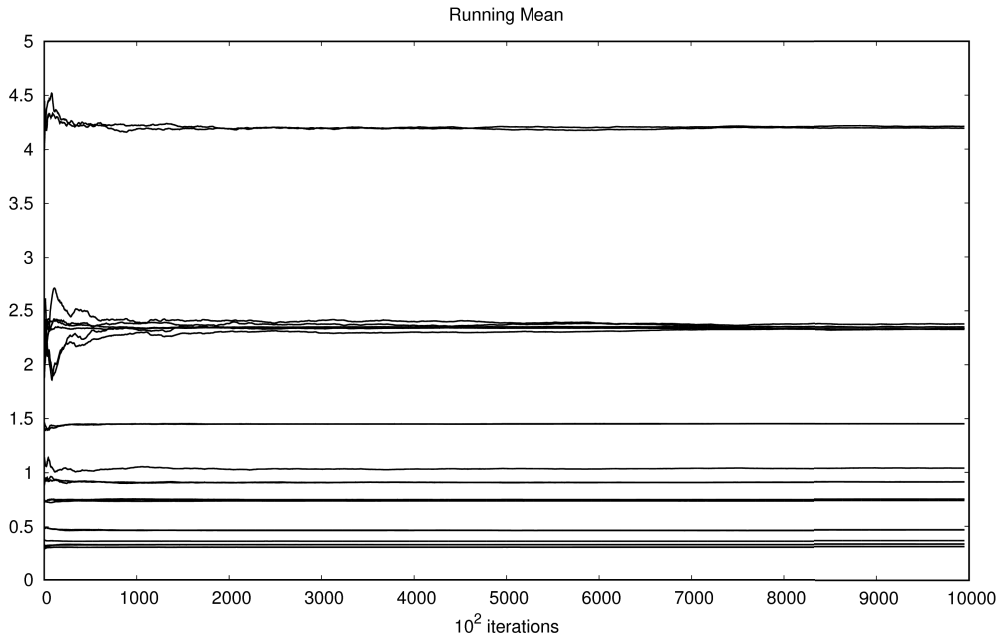
Figure 4.6: *Comparison of the first uncorrelated lag (a) and the effective sample size (b) for each model parameter in velocity and slowness domain*

4.2.2.3 Cumulative mean

The cumulative mean plots provide an indication if the Markov chain has already converged to a stationary distribution. All model model parameters were plotted over 1 million iterations (Figure 4.7 and 4.8). Both figures show a thinned chain with a finning of 100. The slowness based chain (Figure 4.7a) seem to get saturated faster than the velocity based (Figure 4.7b) which seems to stabilize after 500,000 iterations especially at model parameters with intermediate velocities.

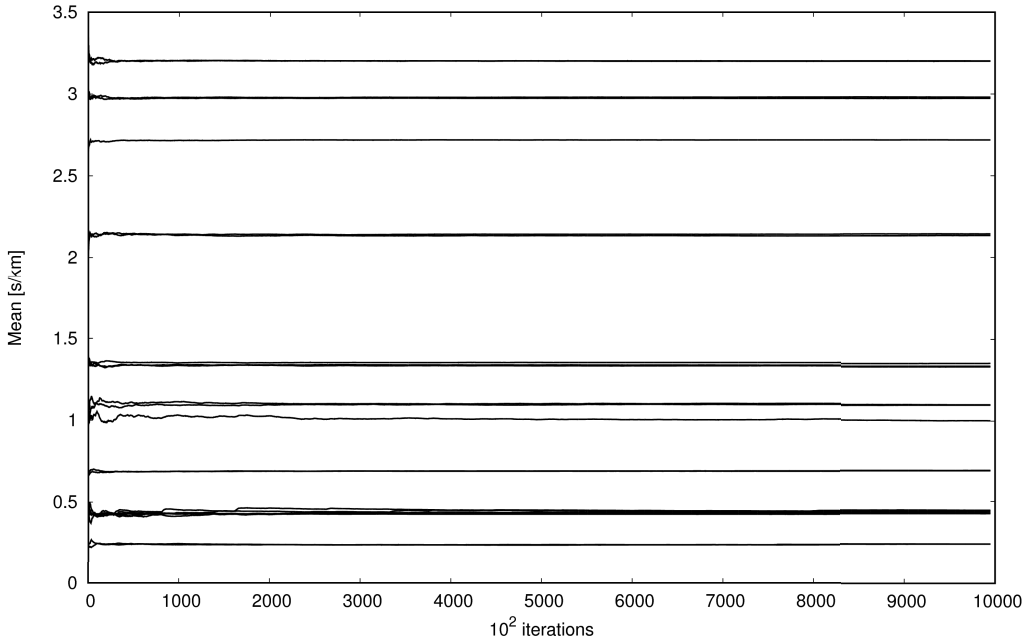


(a) *Slowness domain without compensation*

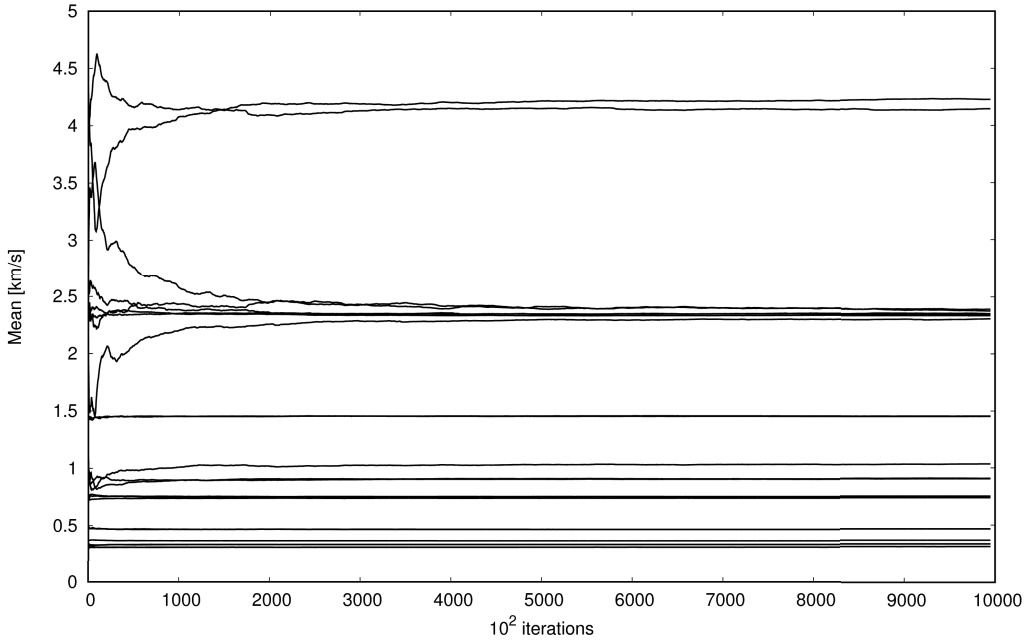


(b) *Velocity domain without compensation*

Figure 4.7: *Comparison of the cumulative mean plots*



(a) *Slowness domain with compensation*

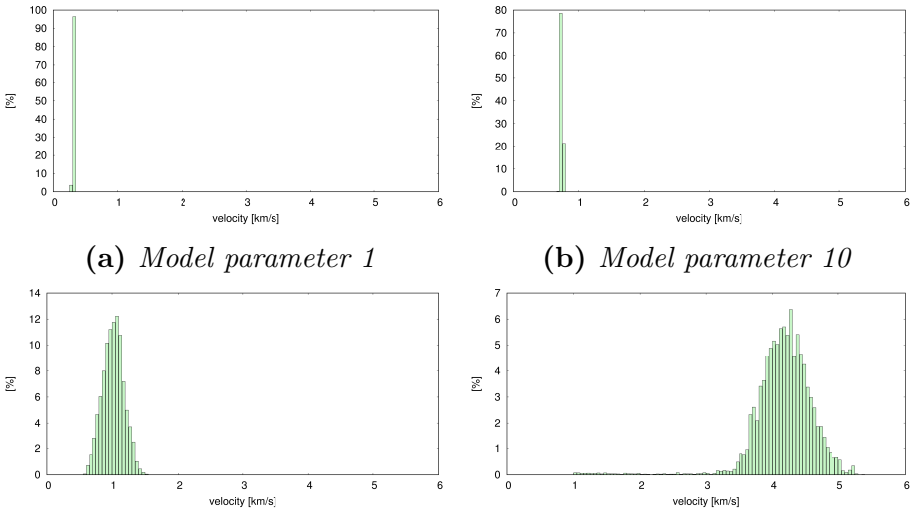


(b) *Velocity domain with compensation*

Figure 4.8: *Comparison of the cumulative mean plots*

4.2.3 Probability distributions

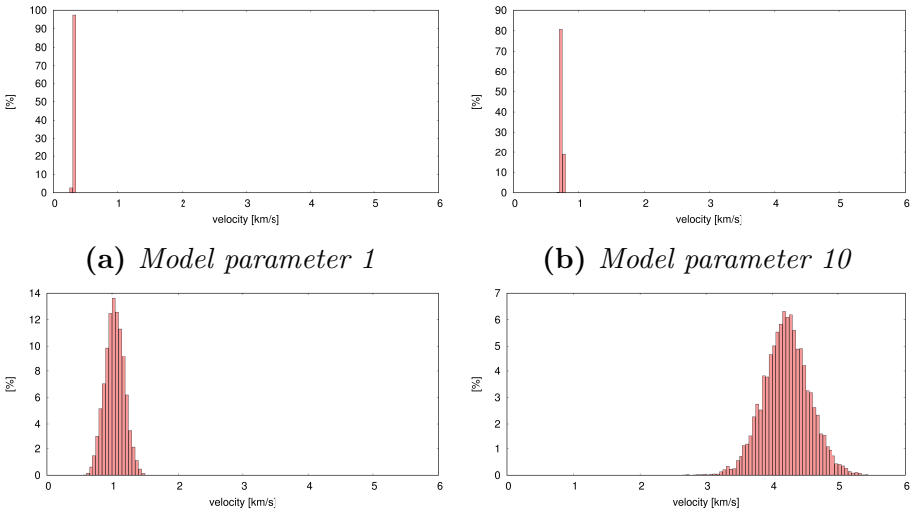
To make the histograms of the model parameter comparable they were all plotted in velocity domain (Figure 4.9 to 4.12). The bin size is 0.05 km/s. Figure 4.2a shows where the model parameter are located. Model parameter 1 at the top margin of the model and model parameter 18 in the low velocity zone within the synclinal structure should be poorly constrained. Model parameter 22 is expected to be also poorly constrained, because of its far offsets. The histogram of model parameter 22 is very broad in velocity and slowness domain, but it is slightly broader in slowness domain and there are more accepted values between 1 km/s and 3 km/s in slowness domain (Figure 4.9d and 4.11d). The applied perturbation in slowness domain seems to be slightly relative bigger compared to velocity (Figure 4.10d and 4.12d). The low acceptance rate at this model parameter (Figure 4.4a) confirms that the applied perturbation size at this model parameter is relatively too large. If we compare slowness (Figure 4.9c and 4.11c) and velocity (Figure 4.10c and 4.12c) the poorly constrained model parameter 18 has a slightly broader distribution and a higher acceptance rate in slowness domain and this suggests that the model parameter seems to have a slightly too small perturbation size in slowness domain.



(a) *Model parameter 1* **(b)** *Model parameter 10*

(c) *Model parameter 18* **(d)** *Model parameter 22*

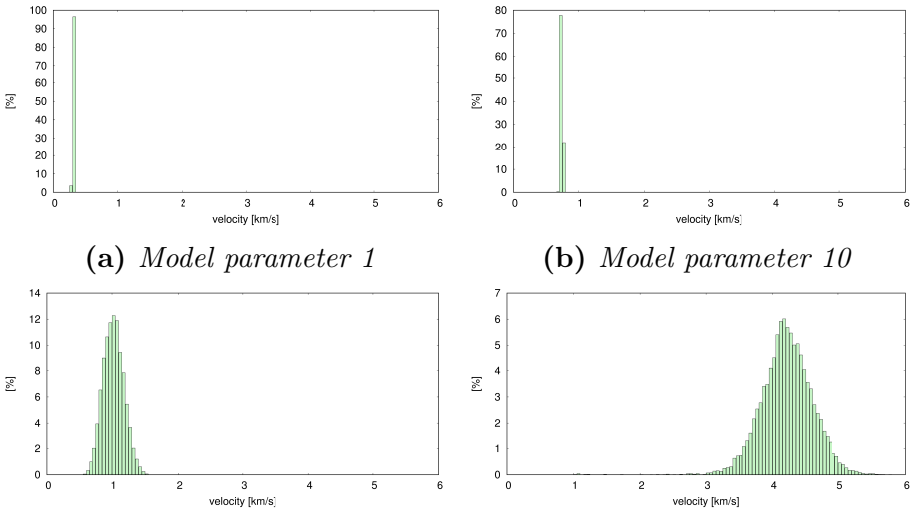
Figure 4.9: *Histograms of 4 different model parameter from the slowness based uncompensated Markov chain*



(a) *Model parameter 1* **(b)** *Model parameter 10*

(c) *Model parameter 18* **(d)** *Model parameter 22*

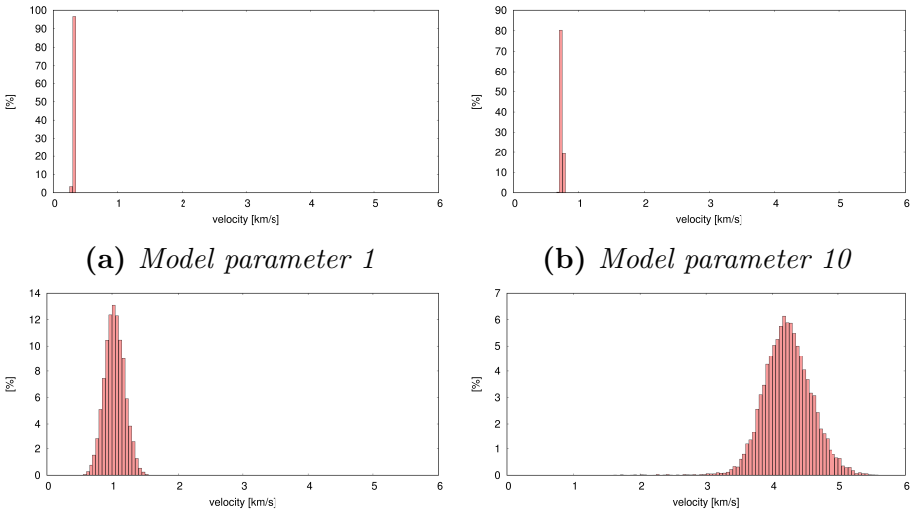
Figure 4.10: *Histograms of 4 different model parameter from the velocity based uncompensated Markov chain*



(a) *Model parameter 1* **(b)** *Model parameter 10*

(c) *Model parameter 18* **(d)** *Model parameter 22*

Figure 4.11: *Histograms of 4 different model parameter from the slowness based compensated Markov chain*



(a) *Model parameter 1* **(b)** *Model parameter 10*

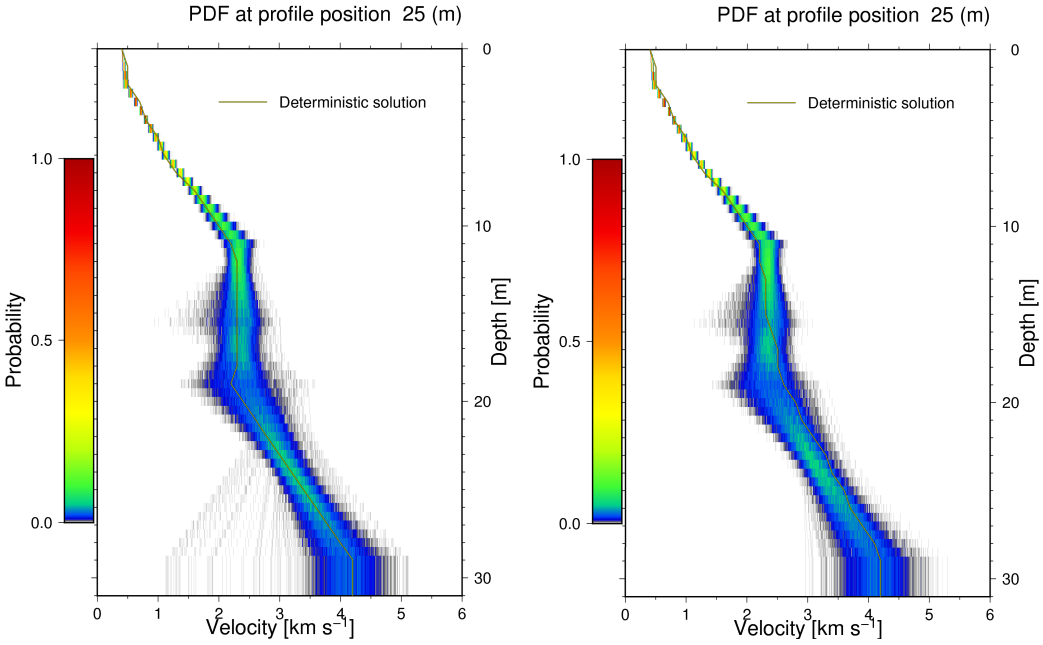
(c) *Model parameter 18* **(d)** *Model parameter 22*

Figure 4.12: *Histograms of 4 different model parameter from the velocity based compensated Markov chain*

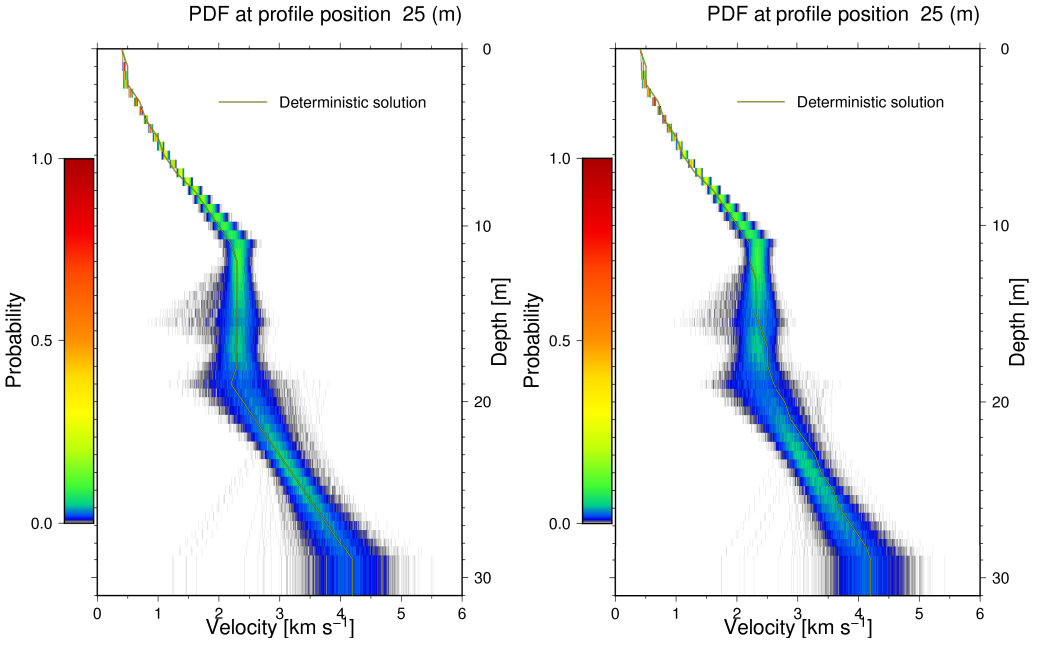
4.2.4 Spatial interpolation of probability distributions

When we normalize the binned occurrences with the total number of occurrences, we call them probability density functions (PDF), although they are, indeed, probability distributions but the shape is the same as that of a PDF. For a visualization of spatial trends, it makes sense to interpolate the PDF between different parameters (using the same linear rule as the interpolation of the values of the parameters). The PDF is referred as the solution in Bayesian inversions. It shows all possible values the parameters can take. Figure 4.13 shows the probability density function at profile position 25 m. Slowness PDF was converted to velocity to make it comparable with the velocity inversion. All figures show that shallow model parameter have a more narrow distribution than deeper parameters. Shallow parameters with a small offset have a low σ_i^d and are therefore weighted heavier in the likelihood function (Equation 2.5). The probability density functions for all Markov chains are more widely for deeper parameter.
One may notice the the narrow distribution at a depth of 17m between two broader distributions and may think the values at this level are better constrained, but this can be explained by neighbouring model parameters. Model parameters next to each other are typically correlated, if one value gets smaller, the other value is getting bigger.
In all 4 probability density function plots there is just an marginal difference.

Slowness based Markov Chains seem to tend to explorer a bigger area in the model space. If the probability density functions of slowness inversion gets overlaid with the velocity inversion it can be seen that the slowness domain is slightly broader. At the deeper model parameters there are more outliers in slowness domain (Figure 4.13a). This outliers are maybe linked to the local minima approximately between iteration 350,000 and 370,000 (Figure 4.3a). This highlights the strong non-linearity of inverse problems. Also with the velocity based Markov chain it can happen to get into such local minima. Because of the fact that this happened in slowness domain the question occurs, if the standard deviation of the deep model parameter should be smaller.



(a) *Slowness domain without compensation* **(b)** *Velocity domain without compensation*



(c) *Slowness domain with compensation* **(d)** *Velocity domain with compensation*

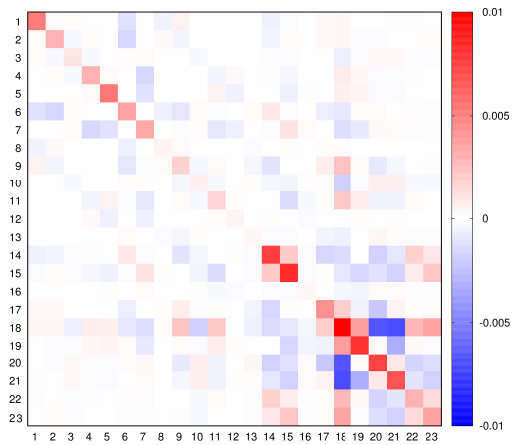
Figure 4.13: *The probability density function for profile position* 25 m

4.2.5 Covariance matrix

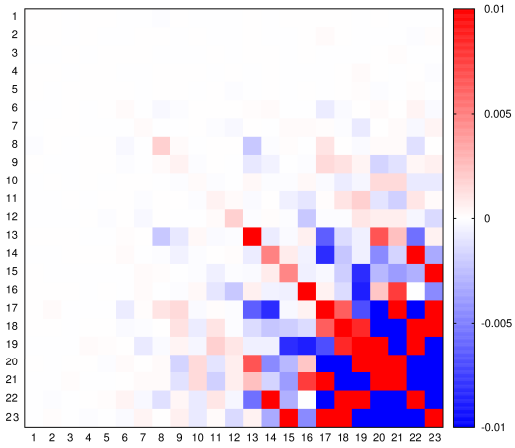
The diagonal elements visualize the variance of each model parameter and the off-diagonal elements show the covariances to other model parameters. The covariance is easily explained, for example if a model parameter has a slightly lower value the neighbouring model parameter is more likely to get an higher value to maintain the travel time. The covariance matrix can be misleading, because it depends from the magnitude of the values. For velocity values the covariance increases with depth while in slowness domain the opposite is generally the case. Therefore the covariance matrix gets normalized. For the normalized covariance the values of the standard deviations of the model parameters get divided by their mean values. We derive relative covariances:

$$cov^{rel}(m_i, m_j) = \frac{\sigma_i^m \sigma_j^m}{\bar{m}_i \bar{m}_j} \tag{4.2}$$

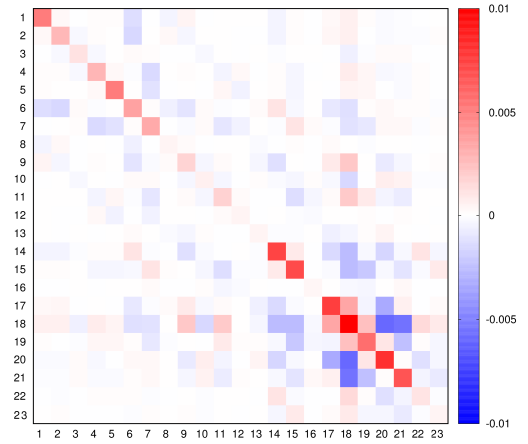
By dividing the diagonal elements which refer to the variance of the model parameter with the squared mean value we also obtain a relative variance. The absolute variance and covariance in slowness domain seem just to slightly increase with depth, while in velocity domain the increase is very strong and correlates with the magnitude of values of the model parameters (Figure 4.14). The increase of the relative covariance with depth is larger in slowness domain compared to velocity (Figure 4.15). This confirms that the relative perturbation size in slowness domain is slightly bigger than in velocity domain and this leads to the lower acceptance rate (Figure 4.4) of deep or far offset model parameters. The poorly constrained model parameter 18 within the synclinal structure has also prominently high covariances and variances in all matrices, just in figure 4.14b and 4.14d it is concealed by the neighbouring high values. In all matrices a slightly decrease of covariance for compensated runs can be seen when comparing with uncompensated.



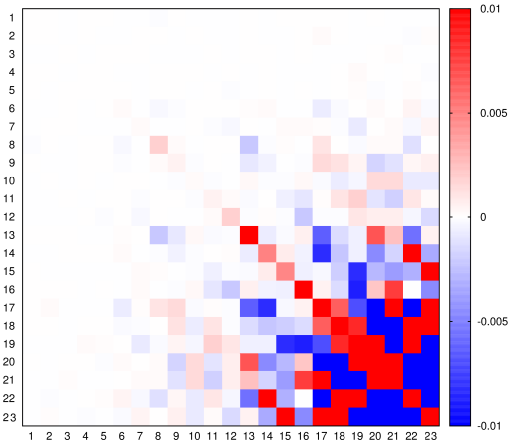
(a) *Slowness domain without compensation*



(b) *Velocity domain without compensation*

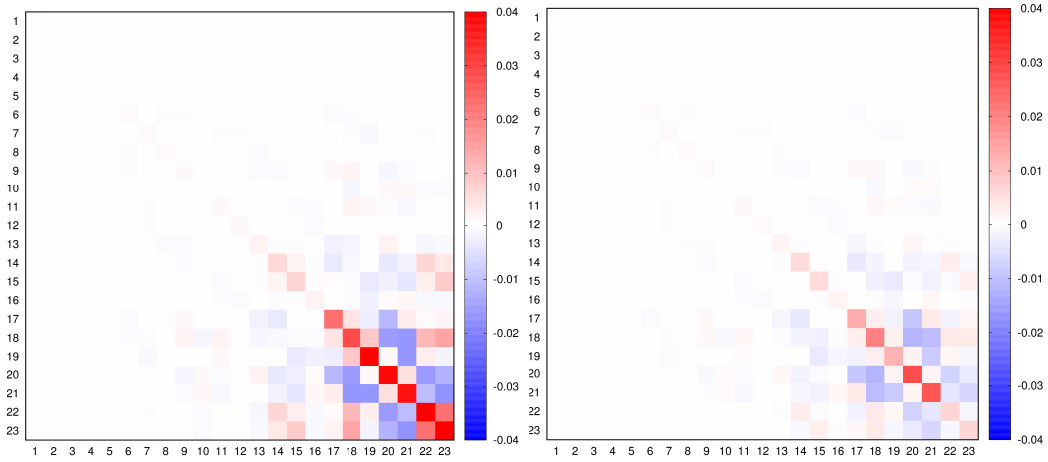


(c) *Slowness domain with compensation*

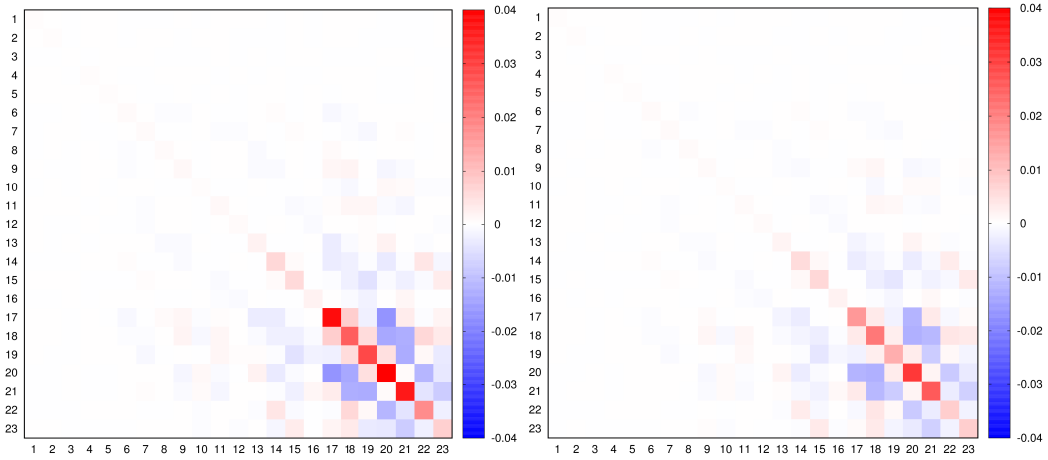


(d) *Velocity domain with compensation*

Figure 4.14: *Model covariance matrix*



(a) *Slowness domain without compensation* **(b)** *Velocity domain without compensation*



(c) *Slowness domain with compensation* **(d)** *Velocity domain with compensation*

Figure 4.15: *Normalized model covariance matrix*

4.2.6 Conclusion

In our synthetic test the slowness based Markov Chain shows a much better performance in comparison to velocity based Markov chains. The step size is increased by approximately 40% at the same acceptance rate. The mixing of the model parameters is much more uniform in terms of frequency and perturbation size. The main advantage is that less prior knowledge is needed and no arbitrary perturbation scaling has to be done. The cumulative mean in velocity domain stabilizes much later than the cumulative mean in slowness

domain. The autocorrelation plots also suggest that slowness based Markov chains reach faster the stationary distribution. For shallow model parameters the effective sample size in slowness domain is much bigger, and needs half of the time to get uncorrelated values. In deeper parts of the model the difference is not so significant and the performance gets slightly better for velocity domain. Maybe the standard deviation for the model parameters in slowness domain should not be constant and it would be better to have a smaller perturbation size at deeper model parameters.

4.3 Comparing compensated vs. uncompensated

Comparing compensated slowness and velocity based Markov chains lead to similar results like uncompensated chains. Therefore in this chapter the performance improvements in terms of acceptance rate and step size will be examined.

4.3.1 Acceptance rates and step size

For the slowness based compensated Markov chain the acceptance rate gets increased by 1.52% and the step size is more than 6% larger. In velocity based Markov chains the increase of acceptance rate (1.38%) and steps size (11%) is quite similar.

Uncompensated McMC	Slowness based	Velocity based
Acceptance rate	23.04 %	22.95 %
Avg. L_2-Distance	0.0879 km s	0.0632 km s
Compensated McMC	**Slowness based**	**Velocity based**
Acceptance rate	24.56 %	24.33 %
Avg. L_2-Distance	0.0936 km s	0.0702 km s

Table 4.2: *Performance comparison between slowness and velocity based Markov chains after 1 million iterations.*

Compensated and uncompensated Markov chains have qualitative similar results. Overall acceptance rate increases, but mostly at model parameters with still high acceptance rates (Figure 4.4). For poorly constrained model parameter there is even a decrease of acceptance rate, as a consequence of the fact that compensations at well constrained model parameter are more likely to get rejected. For example the model parameter 23 with the lowest

acceptance rate in slowness domain the acceptance rate decreases further (Figure 4.4). The same result can be observed for model parameters 2, 4, 5, 6, 7 and 10 in velocity domain.

4.4 Discussion and Conclusion

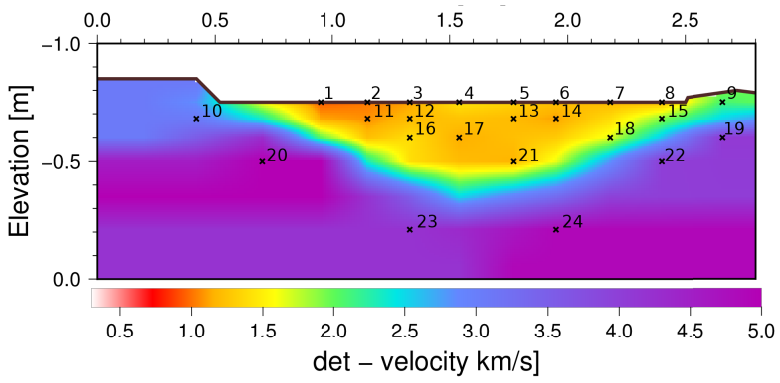
The slowness based Markov chain shows a far better performance compared to the velocity based. With a similar acceptance rate the step size is much higher. The biggest advantage is, that the model parameters get much more uniform perturbed. There is no need for perturbation scaling. While in velocity domain the frequency of perturbations increases with depth, because of an improper perturbation scaling, in slowness domain bad constraint model parameters seem to get perturbed more often. There is also slight gain in performance due to the compensations but the numbers are not very encouraging. This is consistent with the findings of Fontanini (2016). Perhaps, this inefficiency is intrinsic to the method. Another possibility is that the compensation functions might need further adjustments. For example, they might perform better when weighted by the ray length.

Chapter 5

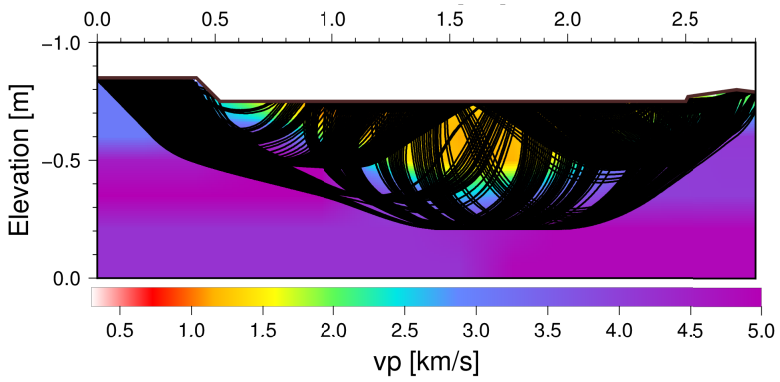
The Salzach test model

5.1 The test model

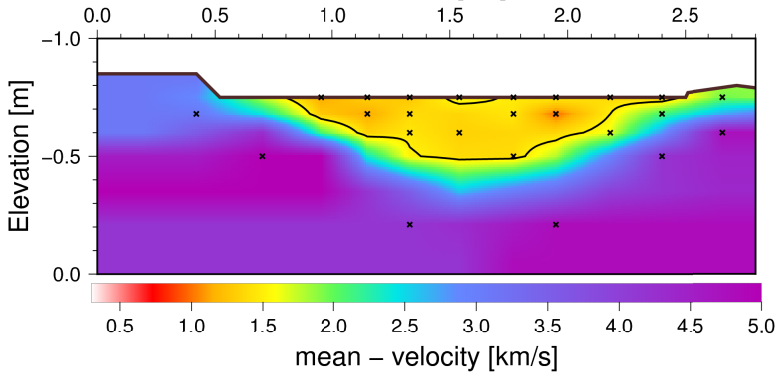
The real data came from a seismic acquisition across the Salzach valley to the west of Zell am See. It is a 3000-m-long seismic line which runs at each end a few hunded meters on bedrock. 10 Hz vertical-component geophones were spaced at 10 m and eight explosive shots were spaced at an average of 400 m (Bleibinhaus & Hilberg, 2012). Figure 5.1 shows the deterministic solution and its model parametrization. The model shows an almost symmetrical concave valley with mostly unconsolidated sedimentary infill. At the northern side of the profile there is a region were the seismic line is interrupted for 300 m because of the highway and the railway (Bleibinhaus & Hilberg, 2012). Because of the lack of receivers in this region there is an area of low ray coverage just below the subsurface. The deterministic solution is used as start model again, to shorten or even skip the burn-in phase. Functional 3 was used for the compensated test run with a compensation factor of 0.2. The applied prior is summarized in table 5.1.



(a) *Deterministic Solution*



(b) *Ray coverage*



(c) *Mean model*

Figure 5.1: *The deterministic solution of the real data test model (a) with the model parametrization, its ray coverage (b) and the mean model (c) with a 1.5 km/s contour line.*

5.1.1 Data uncertainty

To asses the data uncertainty the seismic traces were examined to estimate the picking uncertainty. The picking uncertainties were guessed, by qualitatively evaluating the seismic traces and how accurate it is possible to pick the first arrivals. This method is quite arbitrary and subjective, but it is quite easy to identify traces where the first arrivals are very unsure, due to noise or low frequencies. The range where the first arrival pick possibly lies gets estimated. At most of the traces, especially at short offsets the first arrivals are easy to identify and the data uncertainty was estimated to a low value. Some first arrivals on the other hand were difficult to identify, because of their low frequencies and the noise that can occur, especially at far offsets. These values were estimated to a high data uncertainty. For each shot the user could set 4 uncertainty coordinate points where the x-coordinate refers to the offset and the y-coordinate to the picking uncertainty. The first and the last point refers to the minimum and maximum picking uncertainty for each shot and for the points in between the values get linear interpolated (Figure 5.2). Overall data was very good and accurate and the estimated picking uncertainty was mostly far below 10 ms. Just shot 3 with its low frequencies has a significant higher picking uncertainty. The outlier in figure 5.2 refers to shot 3, where the first arrivals were very difficult to determine.

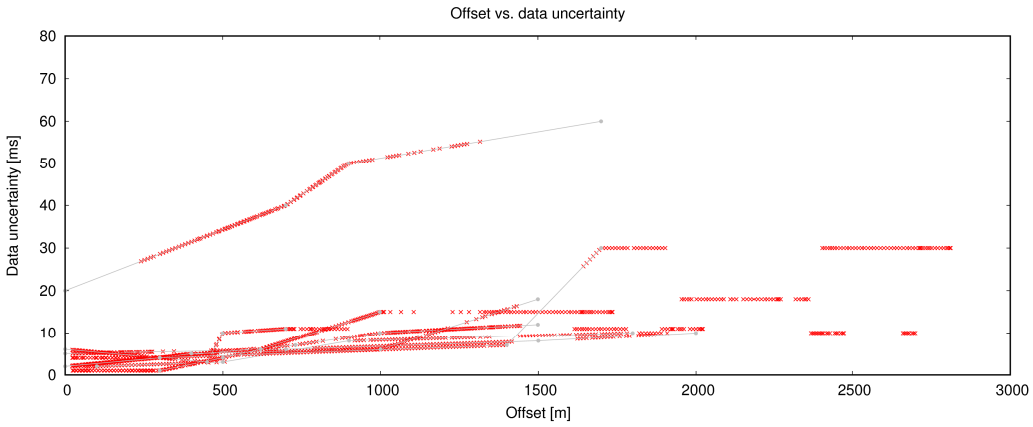


Figure 5.2: *The picking uncertainty for the Salzach model*

5.1.2 Prior and perturbation scaling in slowness domain

Slowness domain

The standard deviation of the parameter perturbation was set to 0.025 s/km. As prior a minimum slowness of 0.1 s/km and as maximum slowness the speed of sound in air was set.

Velocity domain

The depth dependent perturbation scaling was not applied. As shown in the deterministic solution (Figure 5.1) there are very shallow high velocity model parameters. The fact that the bedrock reaches the surface is not only seen in the deterministic solution also in real during the measurement as written in previous studies (Bleibinhaus & Hilberg, 2012). A scaled perturbation size would lead to very high acceptance rates with very low velocity variations at the shallow high velocity layers. The perturbation size of the model parameters where scaled with the outcome of the deterministic solution. The Gaussian density distribution with a standard deviation set by the user gets multiplied by the model parameter velocity which was calculated in the deterministic solution. When scaling the perturbation size relative to the deterministic solution one major assumption has to be taken into account and this is a very strong prior which is not applied in the slowness based inversion. All setting were summarized in table 5.1.

Prior	Slowness domain	Velocity domain
$(m_j)_{min}$	0.1 s/km	0.3 km/s
$(m_j)_{max}$	3.33 s/km	10.0 km/s
σ_j^m	0.025 s/km	$\boldsymbol{v}^{det} * 0.037$

Table 5.1: *The prior information.*

5.2 Comparison slowness vs. velocity

5.2.1 Step size

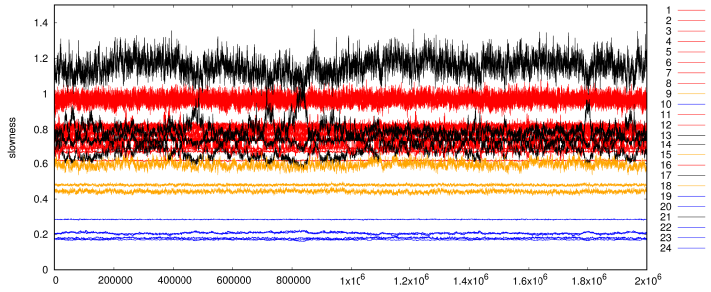
The average Euclidean Distance of one model to the next proposed and accepted model in every iteration 0.0112 s/km for the slowness based and 0.0111 s/km for the velocity based Markov chain. This is just a marginal difference of the step length of about 1.8%, but it has to be considered that

the size of the velocity perturbations were scaled with the parameter values of the deterministic inversion. No scaling has to be done for slowness domain.

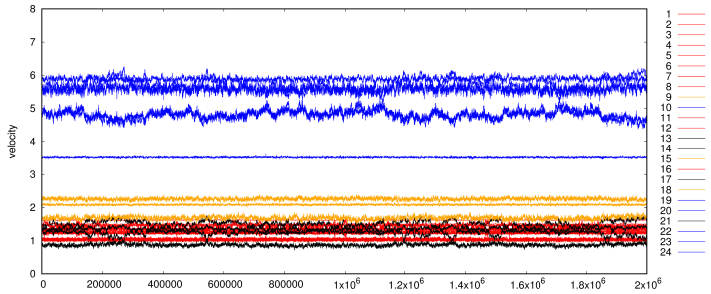
5.2.2 Trace plots

5.2.2.1 Plots of the model parameters

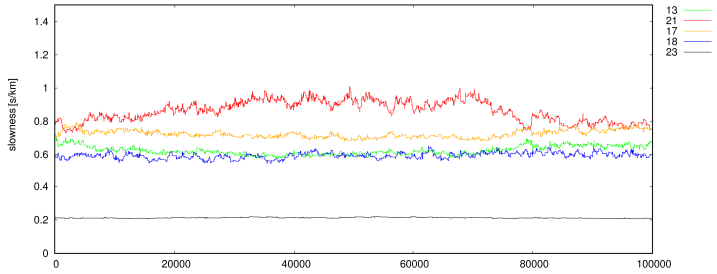
The trace plots (Figure 5.3) show the slowness and velocity of all 24 model parameters plotted over 2 million iterations. The perturbation pattern in slowness domain (Figure 5.3a) seems to be less uniform in terms of perturbation frequency and size. While the model parameters within the bedrock seem to be very well constrained other model parameters within the valley filling seem to wander up and down. This trend can also be seen in the synthetic model where the perturbations of the lowest parameters get rarely accepted. In figure 5.3a model parameters 13, 14 17 and 21 were highlighted in black color. This model parameters are located near to each other in the valley filling and seem to be correlated. In terms of model parametrization it should be considered to make a less dense node grid in this area. The trace plots for the velocity inversion (Figure 5.3b) seem to show a more uniform perturbation pattern than in slowness. Some arbitrary chosen parameters zoomed in for the first 100,000 iterations (Figure 5.3c and 5.3d) also highlight the more uniform perturbation pattern of the velocity based Markov chain. The velocity perturbation is perfectly scaled and the mixing properties for slowness domain seems to be less efficient. Figure 5.4 shows how many percent of the models got accepted by perturbing a certain model parameter. The results in slowness and velocity domain seem to be quite similar. If we compare both results it shows that model parameters 10, 19, 20, 22, 23 and 24 which lie within the bedrock have a much lower acceptance rate in slowness domain. Again this suggests not to apply the same perturbation size at every model parameter and scale it with their summed ray length.



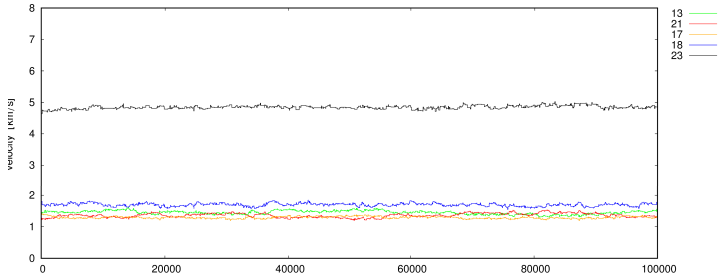
(a) *Slowness domain*



(b) *Velocity domain*

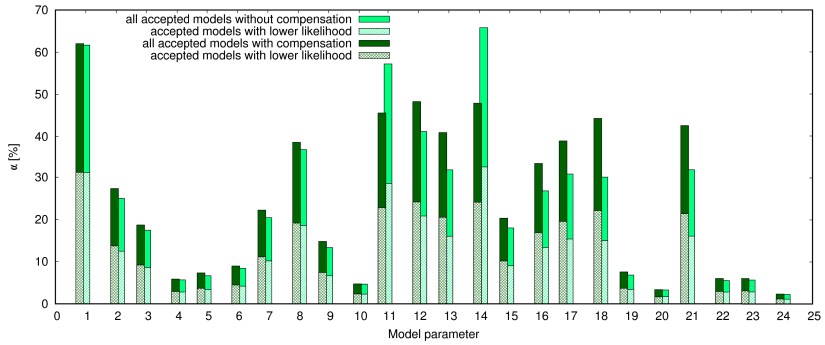


(c) *Slowness domain*

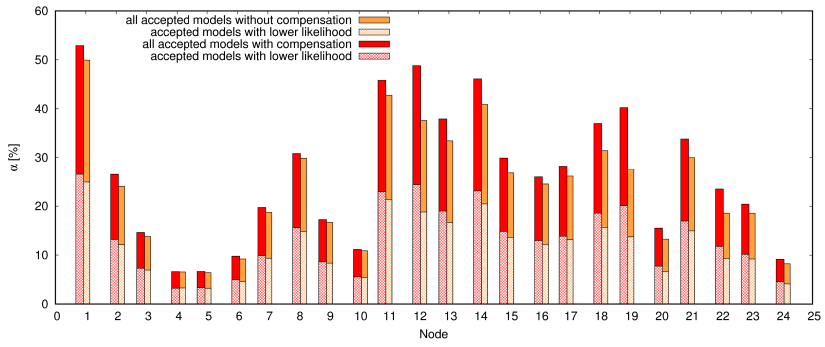


(d) *Velocity domain*

Figure 5.3: *Figure (a) and (b) show the trace plots of all model parameter and figure (c) and (d) show the first 100,000 iterations of some arbitrary model parameters at certain depths*



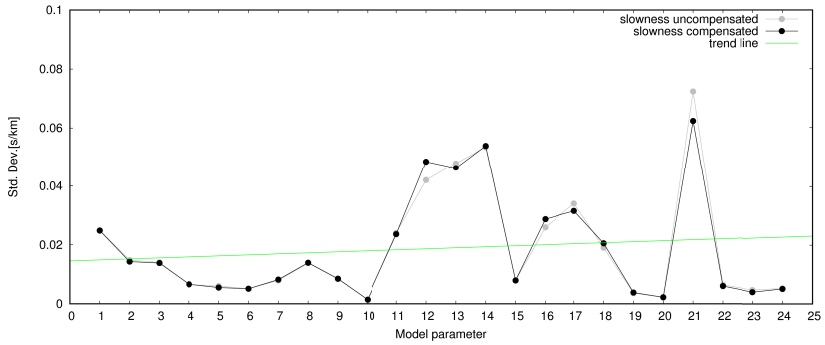
(a) *Slowness domain*



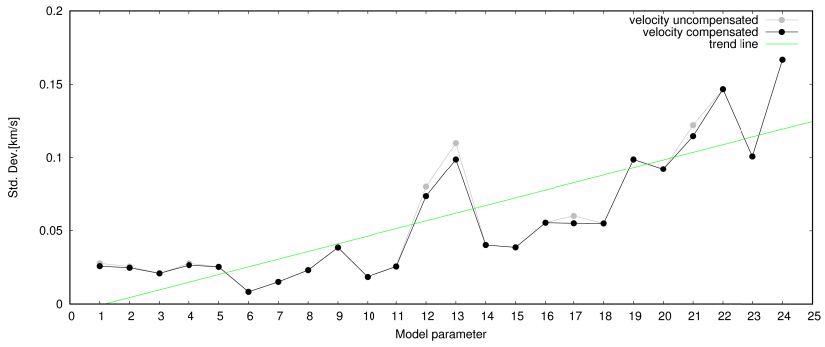
(b) *Velocity domain*

Figure 5.4: *The acceptance rate α for each model parameter*

The standard deviation for velocity (Figure 5.5b) is mainly increasing with depth. In slowness domain (Figure 5.5a) outliers with higher standard deviation can be seen better, while the standard deviation does not increase with depth. Also the model parameters which lie within the bedrock have the lowest standard deviations. Again this suggests that the perturbation size should be scaled to a smaller value.



(a) *Slowness domain*

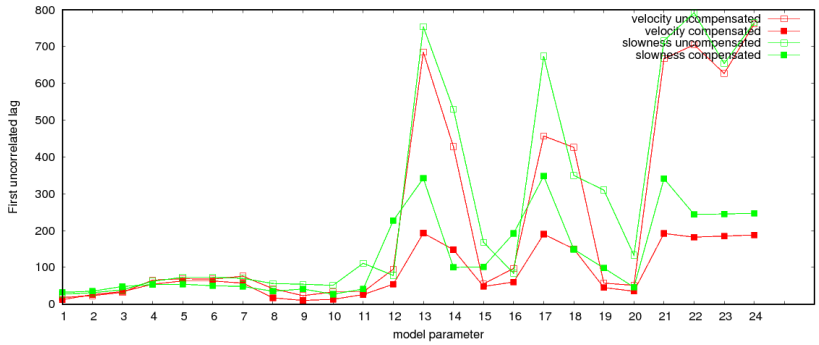


(b) *Velocity domain*

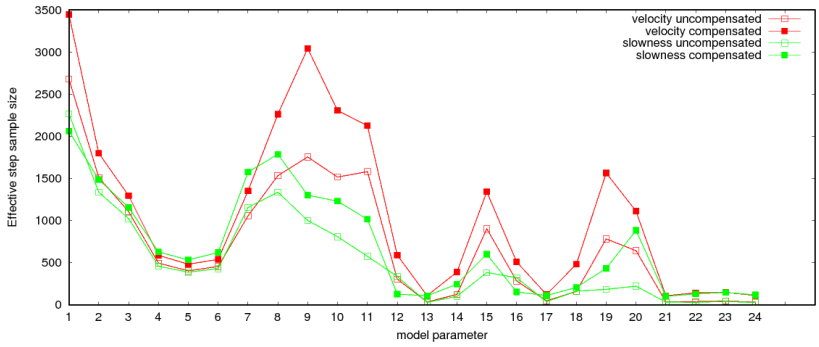
Figure 5.5: *The standard deviation of the model parameters in slowness (a) and velocity (b) domain*

5.2.2.2 Autocorrelation

Figure 5.6a compares the first uncorrelated lag of slowness and velocity based compensated and uncompensated Markov chains for all model parameters. For all chains the the autocorrelation decreases strongly when performing a compensated Markov Chain, especially for the slowness based chain, when we take a look at model parameter 21 to 24. Overall the velocity based Markov chain is performing slightly better, in particular at the deeper model parameters. The Effective sample size is much bigger for the velocity based Markov Chains, just at some few parameters the slowness based compensated Markov Chain performs better.



(a) *First uncorrelated lag*

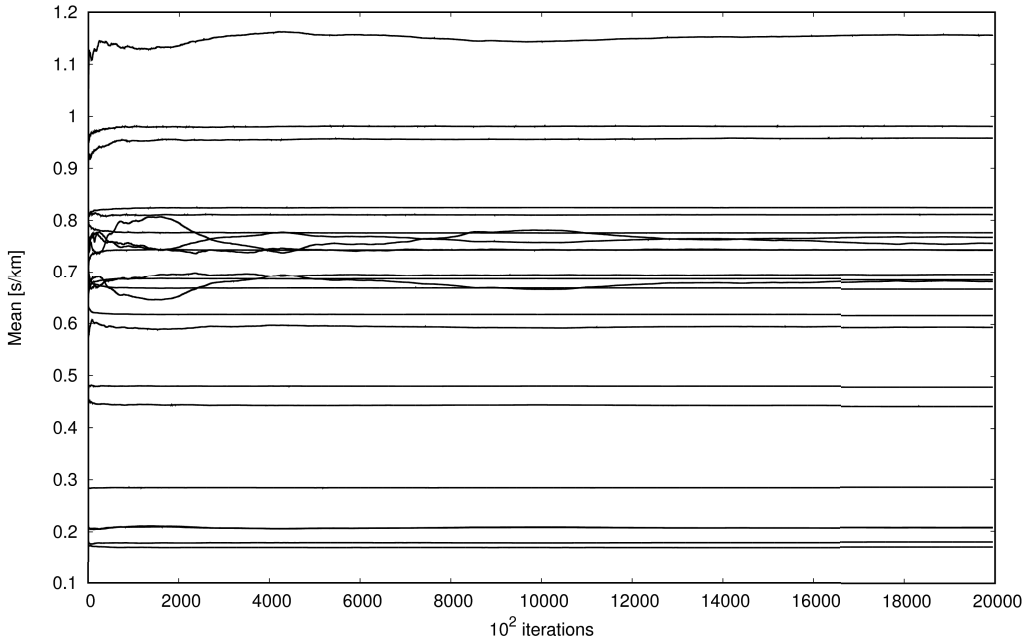


(b) *Effective sample size*

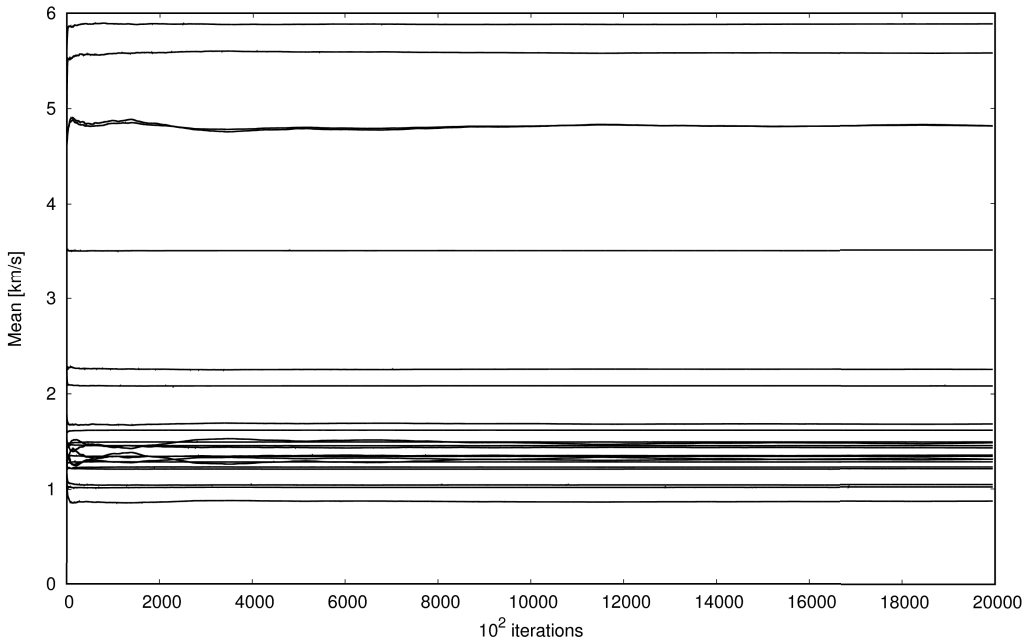
Figure 5.6: *This plots compare the slowness and velocity*

5.2.3 Cumulative mean

Normally the cumulative mean should converge quite fast, just because of the fact that it gets divided by a bigger number. Two model parameters in the slowness parametrization without compensation (Figure 5.7a) seem to change their slowness values after a very long run. To change the mean value after so many iterations the change in slowness must be significant. The mean value seem to stabilize very late, after about 1.6 million iterations. One of these parameters is model parameter 21 which has a very broad distribution (Figures 5.9c - 5.12c).

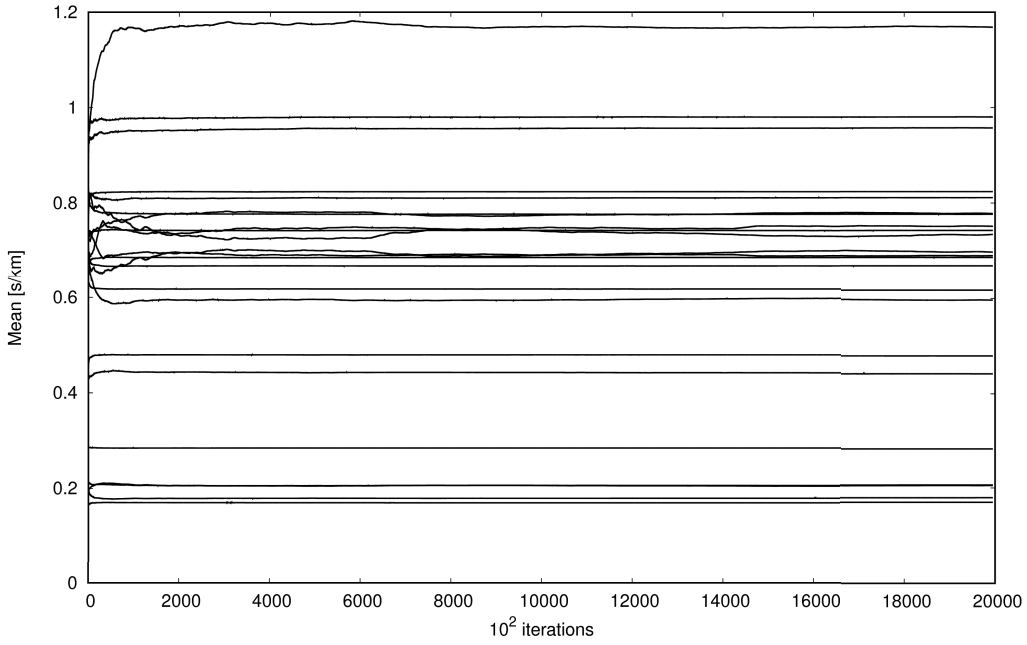


(a) *Slowness domain without compensation*

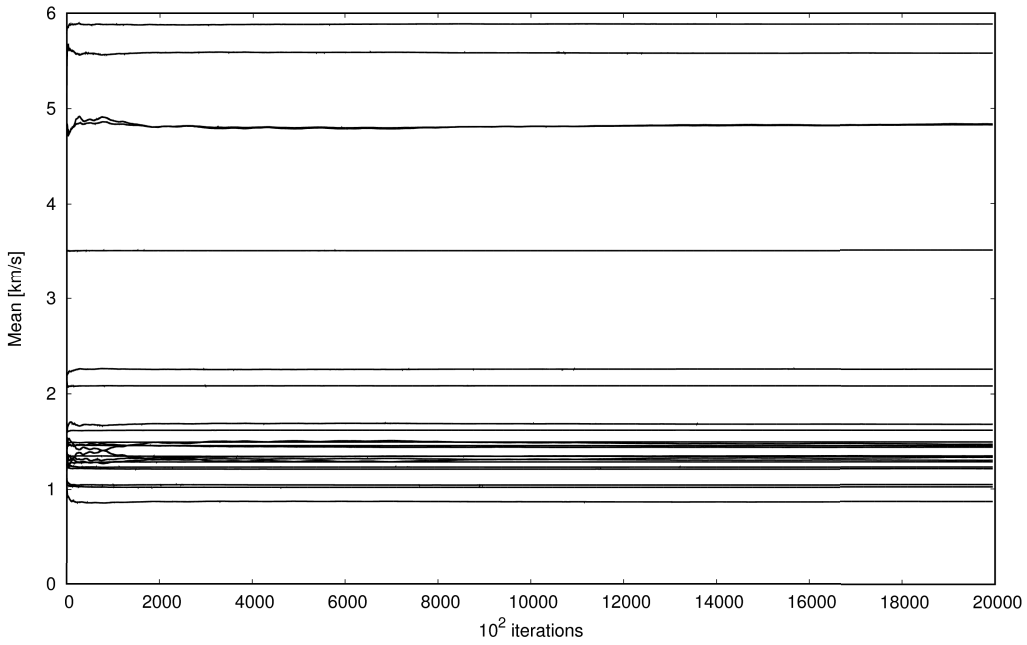


(b) *Velocity domain without compensation*

Figure 5.7: *Comparison of the cumulative mean plots*



(a) *Slowness domain with compensation*

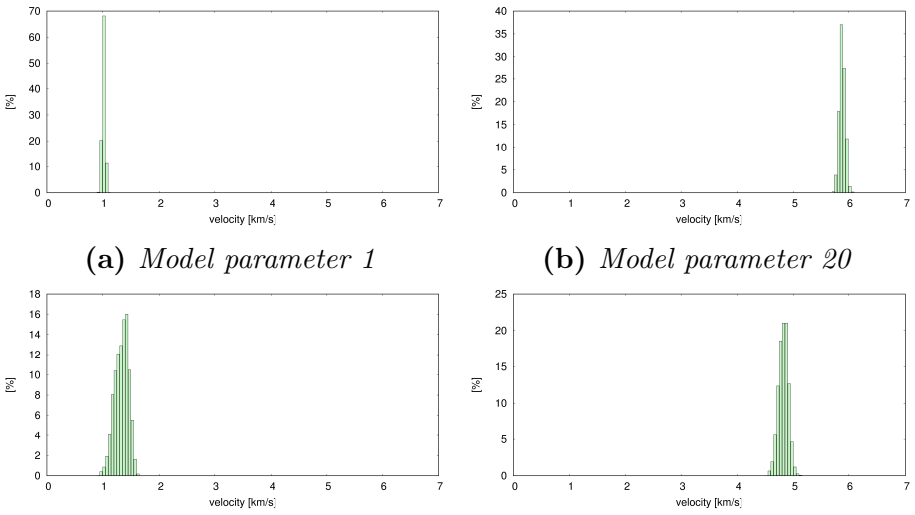


(b) *Velocity domain with compensation*

Figure 5.8: *Comparison of the cumulative mean plots*

5.2.4 Probability distributions

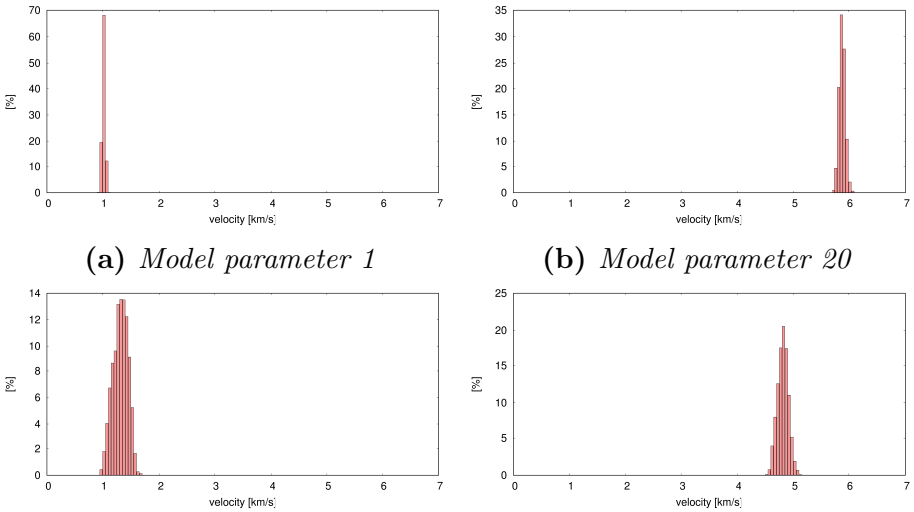
Model parameter 21 at the lowest part of the valley filling seems to have the broadest distribution in all 4 Markov chains, like it was expected (Figure 5.9c - 5.12c). Model parameter 1 has a very narrow distribution (Figure 5.9a - 5.12a), but perturbations get very often accepted. The perturbation size is considered to be relatively too small. The model parameters 20 and 23 lie within the bedrock and have a broader distribution, but perturbations at these parameters get very often rejected (Figure 5.4). In slowness domain the histograms are slightly narrower (Figure 5.9b, 5.9d, 5.11b and 5.11d) compared to velocity (Figure 5.10b, 5.10d, 5.12b and 5.12d). The small standard deviation in slowness domain (Figure 5.5a) also shows that a smaller perturbation should be applied. Perturbations in this case seem to be relatively too big. The same issue occurs at all the other model parameters within the bedrock.



(a) *Model parameter 1* **(b)** *Model parameter 20*

(c) *Model parameter 21* **(d)** *Model parameter 23*

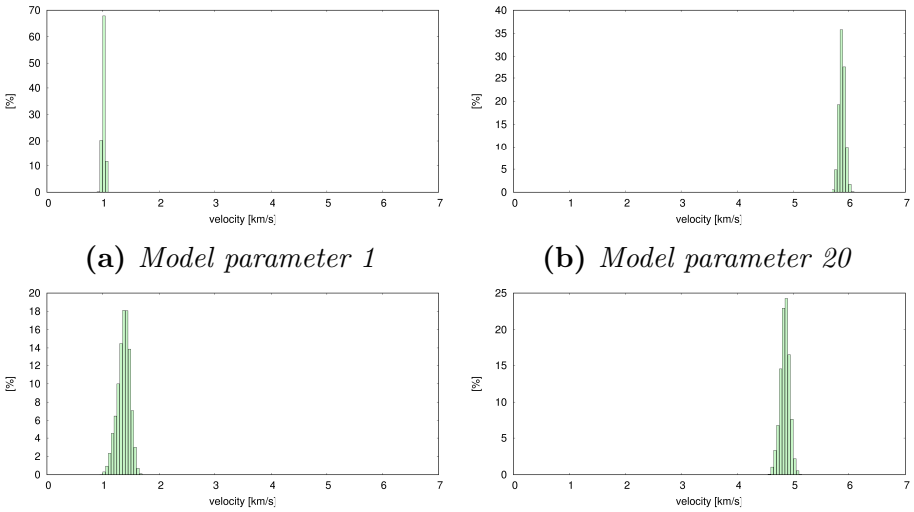
Figure 5.9: *Histograms of 4 different model parameter from the slowness based uncompensated Markov chain*



(a) *Model parameter 1* **(b)** *Model parameter 20*

(c) *Model parameter 21* **(d)** *Model parameter 23*

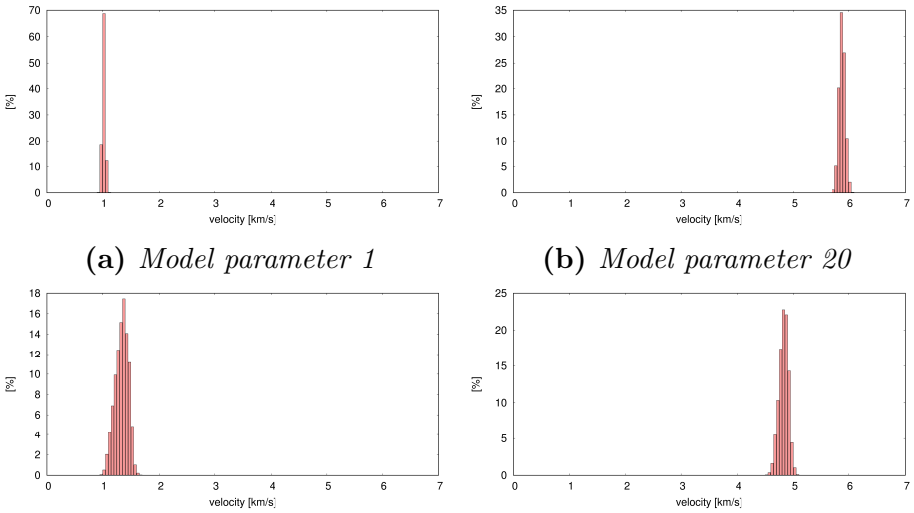
Figure 5.10: *Histograms of 4 different model parameter from the velocity based uncompensated Markov chain*



(a) *Model parameter 1* **(b)** *Model parameter 20*

(c) *Model parameter 21* **(d)** *Model parameter 23*

Figure 5.11: *Histograms of 4 different model parameter from the slowness based compensated Markov chain*



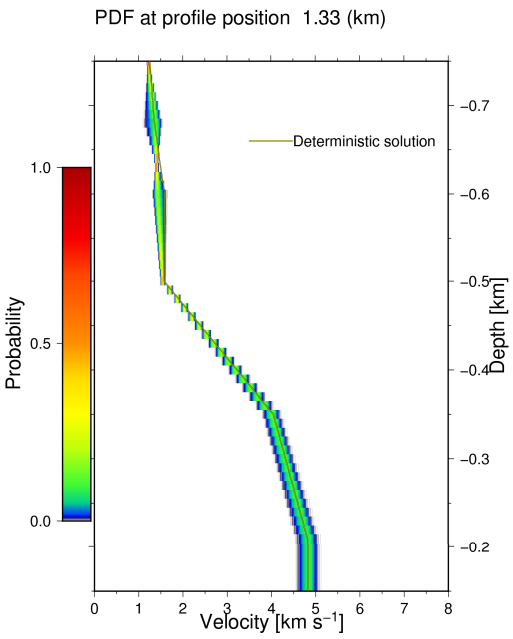
(a) *Model parameter 1* **(b)** *Model parameter 20*

(c) *Model parameter 21* **(d)** *Model parameter 23*

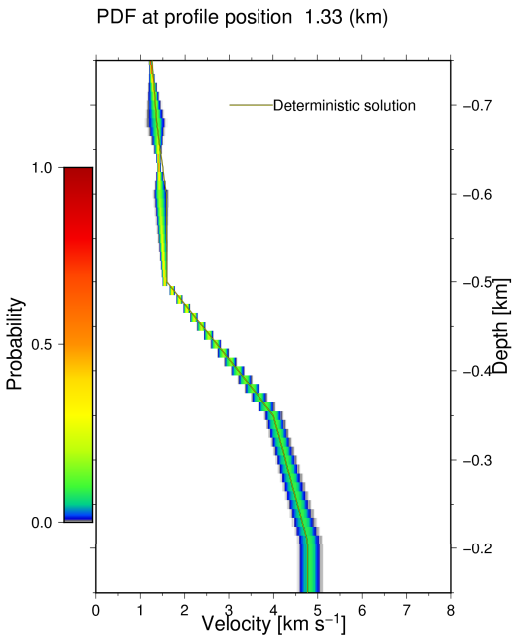
Figure 5.12: *Histograms of 4 different model parameter from the velocity based compensated Markov chain*

5.2.5 Spatial interpolation of probability distributions

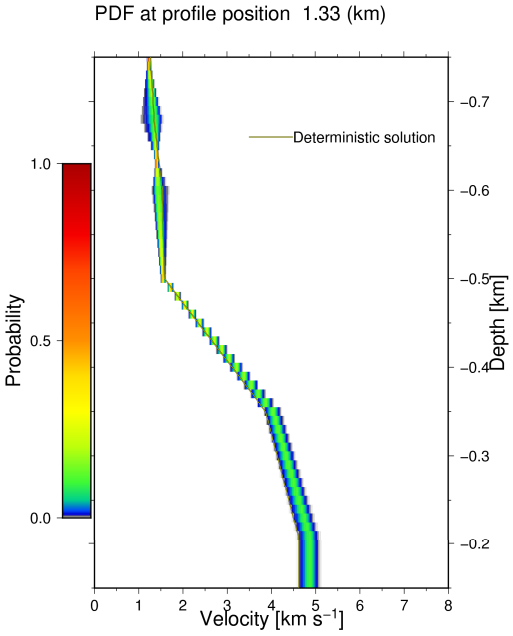
Figure 5.13 shows the probability density function at profile position 1.33 km. The slowness based probability function was converted to velocity to make it comparable with the velocity inversion. There is just a marginal difference in these plots. Slowness based Markov Chains seem to tend to explore a quite bigger area in the model space. The probability density function is slightly broader. Overall the probability density functions are very narrow with the given data uncertainty. At a depth of -0.6 km and -0.67 km there is a relative broad probability density, which reflects the fact that the PDF goes through the model parameter 12 and 16.



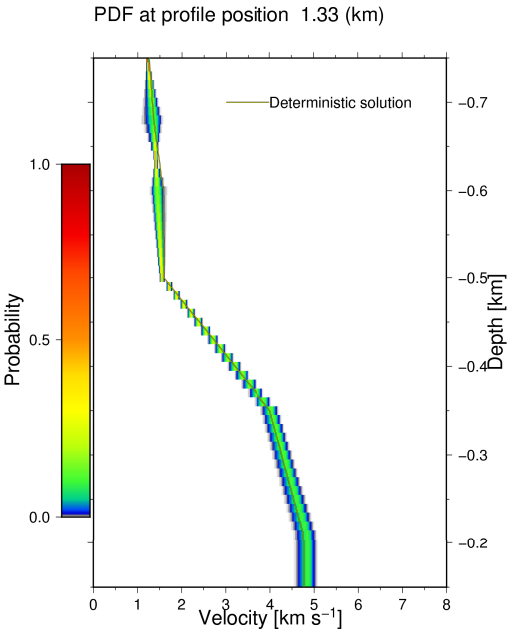
(a) *Slowness domain without compensation*



(b) *Velocity domain without compensation*



(c) *Slowness domain with compensation*

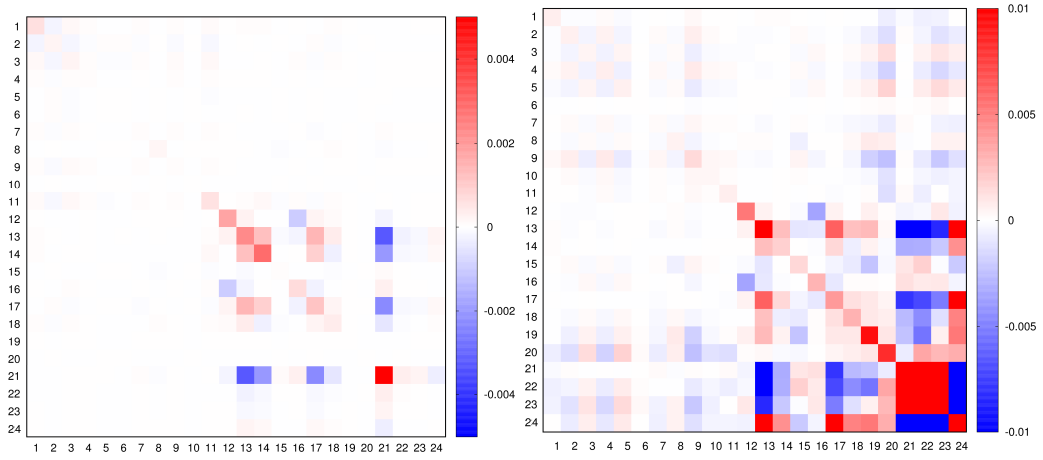


(d) *Velocity domain with compensation*

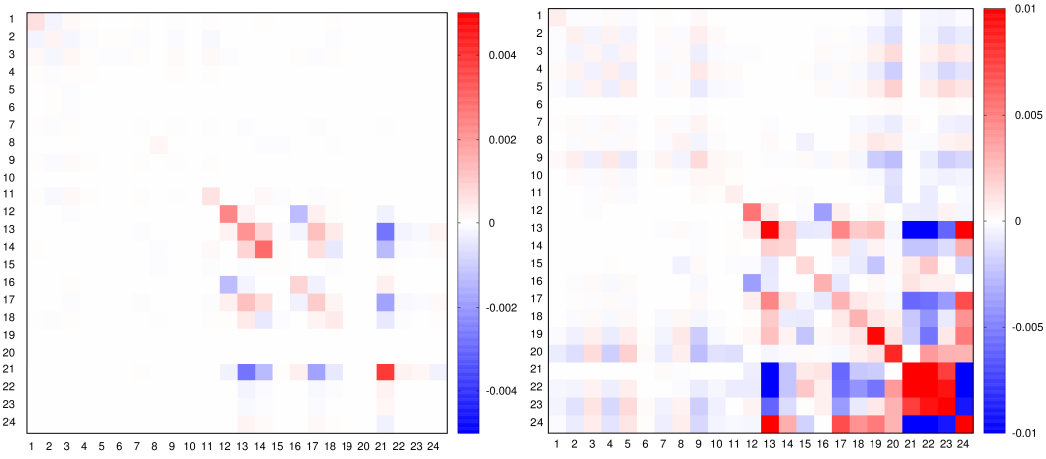
Figure 5.13: *The probability density function for profile position* 1.33 km

5.2.6 Covariance matrix

Model parameter 21 shows a very strong covariance with the model parameters 13, 14 and 17 (Figure 5.14a) at the slowness based uncompensated Markov chain. This strong covariance is slightly decreased in the compensated chain (Figure 5.14c). In velocity domain the covariance and increases with depth and velocity (Figure 5.14b an 5.14d).



(a) *Slowness domain without compensation* **(b)** *Velocity domain without compensation*

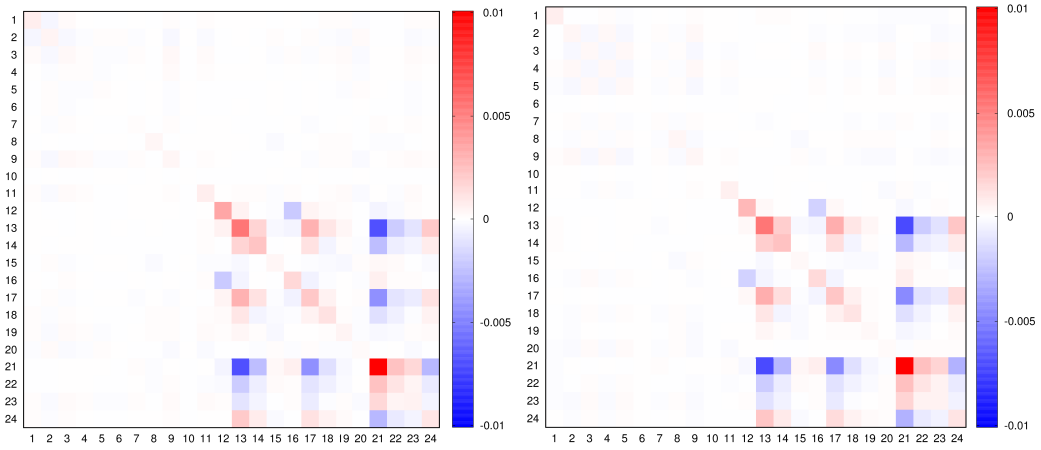


(c) *Slowness domain with compensation* **(d)** *Velocity domain with compensation*

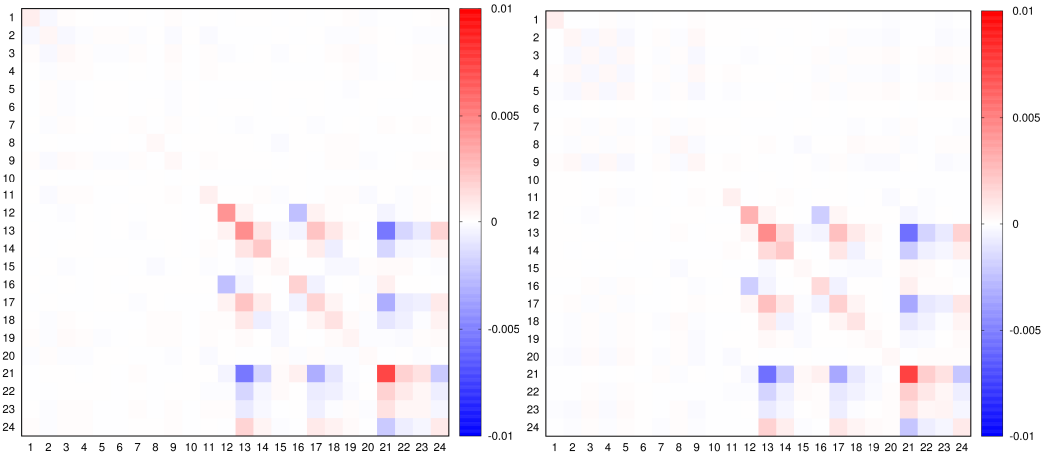
Figure 5.14: *The model covariance matrix*

The relative covariance (Figure 5.15) in all four matrices are quite similar, the only small difference is the slightly smaller covariance in both compensated

chains.



(a) *Slowness domain without compensation* **(b)** *Velocity domain without compensation*



(c) *Slowness domain with compensation* **(d)** *Velocity domain with compensation*

Figure 5.15: *The normalized model covariance matrix*

5.3 Comparing compensated vs. uncompensated Markov Chains

As seen in the previous section, the qualitative convergence assessment tools show that compensated Markov chains perform better than uncompensated. The model parameters are less correlated, step size gets increased and the

chance that a proposed model gets accepted is higher. The step size is slowness domain gets increased by approximately 2% and the acceptance rate is 1.73% higher at the same time. In velocity domain the difference is even bigger. Nearly 9% higher step size and a 3.02% higher acceptance rate. Like in the result of the synthetic test model, the greatest improvement of acceptance rate occurs at the poorly constrained model parameters where the acceptance rate already was very high. Model parameters with low acceptance rate stay at the same value, which is not desirable for a good Markov chain.

Uncompensated McMC	Slowness based	Velocity based
Acceptance rate	23.21 %	23.59 %
Avg. L_2-Distance	0.0112 km s	0.0111 km s
Compensated McMC	**Slowness based**	**Velocity based**
Acceptance rate	24.94 %	26.61 %
Avg. L_2-Distance	0.0115 km s	0.0121 km s

Table 5.2: *Performance comparison between slowness and velocity based Markov chains after* $2 * 10^6$ *iterations.*

5.4 Summed ray length

The results show that the perturbation size is not correctly scaled in slowness domain, especially at the model parameters 4, 5, 10, 20, 22, 23 and 24 with the lowest acceptance rates. The idea came up to divide the perturbation with the summed ray length of the model parameters. The summed ray length can be derived through the $\boldsymbol{G}$-matrix, by summation of the columns. It clearly shows that the model parameters with the lowest acceptance rates (Figure 5.4a) correlate with the highest values of the summed ray lengths (Figure 5.16). The ray lengths in this plot were normed by the largest value (Model parameter 10). In this master thesis there was not the time to make another chapter with some test runs and comparisons to verify this outcome, but it is quite obvious that this perturbation scaling will lead to a better result.

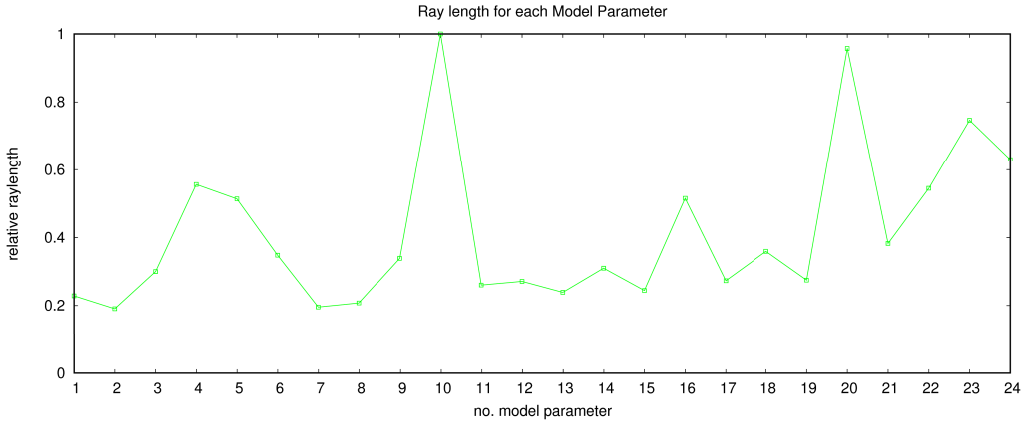
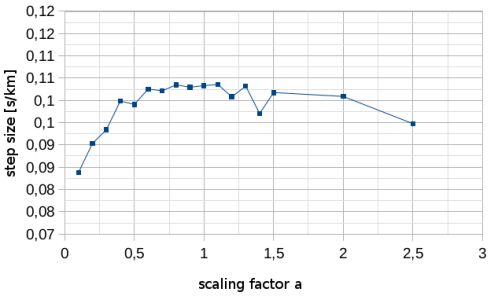


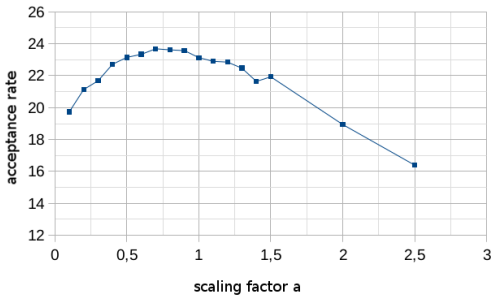
Figure 5.16: *The relative ray lengths at each model parameter*

5.5 Discussion and Conclusion

In this real data test the velocity based Markov chains show a slightly better performance. The functional for the compensation behaves different for slowness and velocity compensations and in this work I did not test out which compensation size gives the best acceptance rate and step size. Before I made the test run I decided to use the same functional with the same scaled compensation term instead of benchmarking the best performing scaling factor for both chains. The resolution matrix for slowness is slightly different to the resolution matrix in velocity domain and both were used for the compensation, so a different result can be expected. Figure 5.17 compares short test runs with different compensation scaling. It shows that the acceptance rate (Figure 5.17b) and step size (Figure 5.17a) have the maximum size at a compensation factor of about 0.8. This is just an example plot for another Markov chain with functional 1 for the compensation.



(a)



(b)

Figure 5.17: *In figure (a) the scaling factor for the compensation term is plotted against the step size. Figure (b) shows the scaling factor against the acceptance rate.*

Another in my opinion more likely reason for the worse performance of the slowness based Markov chain is the applied prior. The perturbation size is not correctly scaled, especially for the model parameters within the bedrock. When we take a look at the model parameters with a very low acceptance rate it shows that model parameter 10, 19, 20, 22, 23 and 24 lie within the much faster bedrock. Rays tend to run around low velocity zones like the valley filling and towards high velocity zones. Figure 5.1b shows the more dense ray concentration at the bedrock compared to the valley filling. These model parameter are much more constrained and change of one of these model parameter lead to a high change in likelihood because travel times at nearly all receivers got influenced. The same issue was noticed in the synthetic model, at model parameters 22 and 23 (Figure 4.2a) where we have a similar situation. The acceptance rates at these model parameters where still the lowest (4.4a). In the synthetic model this effect was clearly notable but not that significant. Compared to the Salzach test model the synthetic test model has a much more uniform ray coverage.
These leads to the question, if the perturbation is really proper scaled. The

slowness is proportional to to travel time, but the acceptance rate depends on the change of likelihood of the model. The likelihood depends on change of all travel times and if there are more rays affected by one model parameter the change of likelihood is bigger. So the proportionality can be better expressed by the equation:

$$\sigma_i^{slowness} \propto \frac{\Delta t}{l_i} \tag{5.1}$$

where the standard deviation $\sigma_i^{slowness}$ of the i^{th}model parameter in slowness domain is proportional to the change of travel time Δt divided by the summed ray length l_i at the i^{th} model parameter. The summed ray length can be obtained through the $\boldsymbol{G}$-matrix, by summation of the columns. Figure 5.16 shows the summed ray lengths at each model parameter. It clearly shows that model parameter 10, 20, 22, 23 and 24 with the lowest acceptance rates have the highest values. Those five parameters have the largest difference in acceptance rate when comparing velocity and slowness based Markov chains. Also model parameter 4, 5 and 6, which have low acceptance rates and high ray lengths, would benefit from this perturbation scaling.
The compensation seems to have a positive effect for the slowness based Markov chains. Figure 5.6a shows that the autocorrelation gets slightly decreased in velocity domain, but leads to a huge difference in slowness domain. The model parameters are connected to each other through the compensation term. This fact prevents the poorly constrained parameters against large changes, because compensations of well constrained model parameters in the apposite direction are less likely to get accepted. The model parameter pretend to have more uniform mixing properties, but not because of a proper perturbation scaling, which would be desirable for a good Markov chain.

Chapter 6

Discussion and conclusions

One of the biggest advantages of a slowness based Markov chain is that there is no need to scale the perturbation size, because the applied perturbation is direct proportional to the change of travel time. In velocity domain the perturbation needs to be scaled somehow, which requires either prior knowledge or assumptions.
In the synthetic test model with its uniform ray coverage the slowness based Markov chain showed a much better mixing performance. In velocity domain there are model parameters, where the applied perturbation is either too small or too big, which in both cases lead to bad mixing performance. For velocity domain the acceptance rate and step size of each model parameter is mainly controlled by the applied perturbation size, which has to be carefully set by the user. For slowness this issue can be omitted.
In the Salzach test model, where velocity of the model parameter does not increase just with depth, the perturbation size had to be scaled relative to the values of the deterministic solution. Prior knowledge should preferably come from an independent source and not from the data itself. The velocity based model performed slightly better. It turned out that in slowness based Markov chains it is easier to qualitatively assess bad constrained model parameters. For example in a non uniform inversion grid like in simulr16 the grid can be modified for a more appropriate model parametrization.
The Salzach model showed that the perturbation scaling of slowness has still more room for improvement. It has been shown that the model parameter with the lowest acceptance rate correlate with the reciprocal of the summed ray lengths.
The resolution matrix based compensated Markov chain shows that the proposed models have a higher chance of getting accepted. Also the step size from one model to the next accepted model is bigger. The qualitative analysis from the plots show that the effective sample size is also increasing because

the model parameters of the proposed models show a much lower autocorrelation. The third functional proposed by Fontanini (2016) with the scaling of the perturbation term (Tauchner, 2016) showed the best performance, in terms of overall acceptance rates and step size, but it turned out that this functional is not a good choice for a poorly constrained model parameter. A Markov chain is just as good as its weakest member, so it has to be considered to use another functional.

Chapter 7

Outlooks

During the work on this master thesis issues in connection with the perturbation scaling and compensation aspects came up, but also some ideas which are particularly interesting for further investigation.

7.1 Improvement of perturbation scaling

The perturbation scaling aspect has room for further investigation and improvements. In both test models there is a reason to assume that the perturbation should be better scaled. The low acceptance rates at model parameters with high summed ray lengths leads to the conclusion that these two aspects are strongly related. It is possible to derive the ray lengths through the $\boldsymbol{G}$-matrix of the deterministic inversion and to use that result to scale the perturbation size for the probabilistic inversion. The perturbation size will be more appropriate for each parameter and the acceptance rates will be much more uniform. This scaling factor can be used as prior derived through the deterministic solution. During the probabilistic inversion the ray paths will slightly change, so it would be conceivable to recalculate and update the perturbation size every i^{th} iteration. By recalculating the values of the ray lengths during the run of the Markov chain the perturbation sizes would also get independent from the values of the deterministic solution, which is then just used as a starting point.

7.2 Improvement of the compensation term

7.2.1 Improvement of the functional

It turned out that functional 3 is not the best functional for poorly constrained model parameters. Functional 2 should perform better when perturbing a poorly constrained model parameter, because the compensation size at the well constrained model parameters is smaller and therefore bigger perturbations are more likely to get accepted.
The increase of acceptance rate in compensated chains mainly occurred at model parameters which already had high acceptance rates. Either another functional or also a scaling of the compensations with the summed ray lengths could be useful, because the compensation term has the same proportionality like the perturbation itself to the change of likelihood. Again this scaling aspect very likely leads to further improvement of the performance of the multivariate updating scheme and should also be considered for further investigation.

7.2.2 Covariance matrix as compensation term

Instead of using the resolution matrix for the compensation term, the idea came up to use the covariance matrix. The covariance of the model parameters can either be extracted out of the deterministic solution of simulr16 or from the result of another similar Markov chain.

Bibliography

Aster, R., Borchers, B. & Thurber, C. (2013). Parameter Estimation and Inverse Problems 2nd Ed. Academic Press.

Bleibinhaus, F. (2003). Entwicklung einer simultanen refraktions- und reflexionsseismischen 3D-Laufzeittomographie mit Anwendung auf tiefenseismische TRANSALP-Weitwinkeldaten aus den Ostalpen. PhD thesis, Ludwig-Maximilians-Universität München.

Bleibinhaus, F. & Hilberg, S. (2012). Shape and structure of the Salzach Valley, Austria, from seismic traveltime tomography and full waveform inversion. Geophysical Journal International, **189(3)**: 1701–1716.

Bleibinhaus, F., Hilberg, S. & Stiller, M. (2010). First results from a Seismic Survey in the Upper Salzach Valley, Austria. Austrian Journal of Earth sciences, **103(2)**: 28–32.

Brooks, S., Gelman, A., Jones, G. & Meng, X. (2011). Handbook of Markov Chain Monte Carlo. CRC Press.

Fontanini, F. (2016). Optimization strategies for Markov chain Monte Carlo Inversion of seismic tomographic data. PhD thesis, Friedrich Schiller Universität Jena.

Gelman, A., Roberts, G. & Gilks, W. (1996). Efficient Metropolis jumping rules. Bayesian Statistics, **5**: 599–607.

Gubbins, D. (2004). Time Series Analysis and Inverse Theory for Geophysicists. Cambridge University Press.

Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika, **57**: 97–109.

Hole, J. & Zelt, B. (1995). 3-D finite-difference reflection travel times. Geophys. J. Int., **121(2)**: 427–434.

Kass, R., Carlin, B., Gelman, A. & Neal, R. (1998). Markov Chain Monte Carlo in Practice: A Roundtable Discussion. The American Statistician, **52**: 93–100.

Menke, W. (1989). GeophysicalData Analysis: Discrete Inverse Theory. Academic Press, San Diego, Calif.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. J.Chem.Phys, **21**: 1087–1092.

Metropolis, N. & Ulam, S. (1949). The Monte Carlo method. J. Amer. Stat. Assoc., **44**: 335–341.

Roberts, G., Gelman, A. & Gilks, W. (1997). Weak convergence and optimal scaling of random walk Metrolopis algorithms. Ann. Appl. Probab., **7(1)**: 110–120.

Shearer, P. M. (2009). Introduction to Seismology. Cambridge University Press.

Smith, B. (2001). Bayesian output analysis program (NOA) (Version 1.0.0). IA: University of Iowa, College of Public Health.

Tarantola, A. (2005). Inverse problem Theory and Methods for Model Parameter Estimation. SIAM.

Tauchner, C. (2016). Effizienz stochastischer Methoden zur Bestimmung von Modellunsicherheiten. Master's thesis, Montanuniversität Leoben.

Vidale, J. (1990). Finite-difference calculation of traveltimes in three dimensions. Geophysics, **55(5)**: 521–526.

Zelt, C. A. & Barton, P. J. (1998). Three-dimensional seismic refraction tomography: A comparison of two methods applied to data from the Faeroe Basin. Journal of Geophysical Research: Solid Earth, **103**: 7187–7210.