Time-Lapse Seismic Imaging and Uncertainty Quantification

by

© Maria Kotsi

A thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Department of Earth Sciences

Memorial University

March 2020

St. John's, Newfoundland and Labrador, Canada

Abstract

Time–lapse (4D) seismic monitoring is to date the most commonly used technique for estimating changes of a reservoir under production. Full–Waveform Inversion (FWI) is a high resolution technique that delivers Earth models by iteratively trying to match synthetic prestack seismic data with the observed data. Over the past decade the application of FWI on 4D data has been extensively studied, with a variety of strategies being currently available. However, 4D FWI still has challenges unsolved. In addition, the standard outcome of a 4D FWI scheme is a single image, without any measurement of the associated uncertainty. These issues beg the following questions: (1) Can we go beyond the current FWI limitations and deliver more accurate 4D imaging?, and (2) How well do we know what we think we know? In this thesis, I take steps to answer both questions. I first compare the performances of three common 4D FWI approaches in the presence of model uncertainties. These results provide a preliminary understanding of the underlying uncertainty, but also highlight some of the limitations of pixel by pixel uncertainty quantification. I then introduce a hybrid inversion technique that I call Dual–Domain Waveform Inversion (DDWI), whose objective function joins traditional FWI with Image Domain Wavefield Tomography (IDWT). The new objective function combines diving wave information in the data–domain FWI term with reflected wave information in the image–domain IDWT term, resulting in more accurate 4D model reconstructions. Working with 4D data provides an ideal situation for testing and developing new algorithms. Since there are repeated surveys at the same location, not only is the surrounding geology well–known and the results of interest are localized in small regions, but also they allow for better error analysis. Uncertainty quantification is very valuable for building knowledge but is not commonly done due to the computational challenge of exploring the range of all possible models that could fit the data. I exploit the structure of the 4D problem and propose the use of a focused modeling technique for a fast Metropolis–Hastings inversion. The proposed framework calculates time–lapse uncertainty quantification in a targeted way that is computationally feasible. Having the ground truth 4D probability distributions, I propose a local 4D Hamiltonian Monte Carlo (HMC) — a more advanced uncertainty quantification technique — that can handle higher dimensionalities while offering faster convergence.

Thesis Supervisor: Alison E. Malcolm
Title: Associate Professor

To my grandparents, Efterpi and Panayiotis...

Acknowledgements

Time flies incredibly fast, and honestly, I do not know where the past five years went. It feels like yesterday when I first arrived in St. John's in that super cold night of late December. I am incredibly grateful for having the opportunity to complete my Ph.D. under the mentorship and guidance of Dr. Alison Malcolm and everything that this experience taught me, scientifically and personally. I feel that Maria in 2019 has almost nothing to do with Maria in 2015. Completing your doctorate is a significant accomplishment that comes, however, with many doubts and struggle at times. I am incredibly thankful to Alison Malcolm for her constant encouragement, guidance, and support at every level. She is the best role model a student could look up to.

I would also like to thank all my collaborators and coauthors, Gregory Ely, Jonathan Edgar, and Sjoerd de la Ridder, for their tremendous support, advice, and fruitful discussions on both uncertainty quantification and Full–Waveform Inversion. In addition, I would like to thank the other two members of my Ph.D. committee, Dr. Colin Farquharson and Dr. Charles Hurich, for their suggestions that helped improve the content of this thesis. Special thanks to all the friends and fellow researchers at MUN that I met along the way in Newfoundland. Your friendship, support, and continuous waves of laughter created a special bonding and an almost second family that gave me warmth every time I was homesick.

Last but not least, I would like to thank my big Greek family who raised me with all their hearts! Your constant love and support brought me to where I am today, and I will forever owe you for that. Big thank you to Craig for his generous support, care, and love that motivated me over the past few years.

Statement of contribution

This research project was first initiated by Alison Malcolm and was then designed by Alison Malcolm and Maria Kotsi. The identification of specific questions to be examined was refined by Maria Kotsi during the course of the study with the guidance of Alison Malcolm. For all the Chapters listed below, research and numerical calculations, as well as drafting and maintaining each manuscript were performed by Maria Kotsi.

Chapter 2 was presented by Maria Kotsi at the Geological Association of Canada (GAC) Newfoundland and Labrador Section Annual Meeting (2017), 13th International Conference on Mathematical and Numerical Aspects of Wave Propagation (WAVES 2017) and at the 87th SEG Annual International Meeting (2017). All Expanded Abstracts have been co–authored by Alison Malcolm, who provided research and editorial guidance.

Chapter 3 is published in Geophysics and co–authored by Jonathan Edgar, Alison Malcolm and Sjoerd de Ridder. This research took place during a summer internship at the Geoscience Research Centre of TOTAL E&P UK in Aberdeen, and Jonathan Edgar provided research supervision and helped shaping the ideas. Alison Malcolm and Sjoerd de Ridder provided editorial guidance and helped with the organization of the paper. The paper was reviewed by Jeffrey Shragge, Partha Routh, Musa Maharramov, and two anonymous reviewers. In addition, this work was presented by Maria Kotsi at the Geological Association of Canada (GAC) Newfoundland and Labrador Section Annual Meeting (2018).

Chapter 4 is currently under revision in Geophysical Journal International and co–authored by Alison Malcolm and Gregory Ely. The version appeared in this thesis, is an extended version of the manuscript since the Editor (Ludovic Métivier) has requested some sections to be removed. Alison Malcolm and Gregory Ely helped shaping ideas and organize the paper. In addition, parts of this paper have been previously presented by Maria Kotsi at SIAM Conference on Imaging Science (2018), 88th SEG Annual International Meeting (2018), SIAM Conference on Mathematical & Computational Issues in the Geosciences (2019), and 89th SEG Annual International Meeting (2019). Both the manuscript and Expanded Abstracts have been written by Maria Kotsi.

Chapter 5 is expected to be submitted in Geophysics under the section of Geophysical Software and Algorithms, and co–authored by Alison Malcolm and Gregory Ely. Alison Malcolm and Gregory Ely provided editorial assistance.

Table of contents

List of tables

List of figures

Chapter 1

Introduction

Seismic material properties (e.g. P–wave velocity) describe how waves travel through the Earth's subsurface. Over the cycle of a reservoir's life, hydrocarbon production processes, such as fluid extraction and injection, may introduce changes to pore fluids, pore pressure, and effective stress in the reservoir. Consequently, seismic properties of the reservoir will change too [91] [164]. For instance, Gassmann's equations [43] are the most widely used ones to infer seismic velocity changes from different pore fluid saturations in a reservoir. Changes in pore pressure lead to changes in the effective stress field [150] of a reservoir and its surrounding rocks [154]. As a consequence, the stiffness of the rock matrix is modified, resulting in opening and closing cracks, which then affects the seismic velocity. In some cases, a steep decline in the pore pressure can even lead to reservoir compaction and overburden subsidence [45]. This causes changes in seismic velocity and layer thickness, which thenceforth alter the seismic travel times. All of these actions, indicate that there is a complicated relationship between the seismic properties and the physical processes that take place within a reservoir. Time–lapse seismology aims to estimate these changes and relate them to physical changes within a reservoir. Geophysical monitoring of a reservoir allows for estimation of the extraction efficiency and determination of remaining reserves.

When looking at time–lapse data, the differences that are typically used are those of amplitude and time–shifts. For example, amplitude changes can be caused by new structures in the target area or reflectivity differences at interfaces, while time–shifts can be caused by physically shifted geologic boundaries or velocity perturbations along the wave path. Understanding which of these two mechanisms control the changes in the observed data is important to better estimate and interpret the time-lapse changes. Traditional time–lapse analyses are conducted on full- or partial- stack data [8] [72] [73] [153]. In cases of simple and thin reservoirs (i.e. reservoirs with only two fluids) direct interpretation of amplitude changes in the full–stack data will be adequate, whereas in cases of complicated reservoirs more attributes will be necessary for an accurate analysis [39]. For example, time–shifts can sometimes constrain interpretation of amplitude changes. Time–shifts are usually measured at an interface of interest by summing all changes of traveltimes in the layers above; the derivative of those time–shifts — the so–called time strain — can indicate relative velocity changes [17]. In addition, Amplitude Variation with Offset (AVO) is commonly used in order to separate fluid saturation from pressure effects [149].

However, most of these approaches do not look for the actual property changes (i.e. P–wave velocity) but rather indicators of them. In addition, they rely on simple model assumptions and high frequency approximations, which are far from realistic. Ideally, one would like to use an approach that is based on the physics of the wave equation, has no assumptions, and delivers higher resolution images (compared to traveltime tomography and migration) such as Full Waveform Inversion (FWI). FWI aims to deliver directly an Earth model (e.g. P–wave velocity) of the subsurface by iteratively trying to match synthetic seismograms that were generated solving wave equations with observed data. Even though FWI was conceptualized in the 1980s [71] [145], it is only recently that it has become a part of time–lapse seismological analysis. In contrast with the traditional methods discussed above, FWI uses prestack data, which means that the risk of potentially losing useful information is eliminated. 4D FWI has seen already numerous applications both on synthetic [173] [84] [6] [173] and real cases [118] [57] [172] [85]. Amongst the most popular 4D FWI workflows are Parallel FWI [110], Sequential FWI [6], Double Difference FWI [167] [182], and Joint FWI [84] [173] (in the context of this thesis, we will refer to the Joint FWI scheme from [173] which is called Alternating FWI). Figure 1.1 (a)-(d) provide a flowchart of how each methodology works.

1.1 Challenges and Contributions

Despite its numerous advantages, 4D FWI has its challenges too. First and foremost, FWI is sensitive to initial model inaccuracies and prone to cycle–skipping. If, for example, the synthetic data generated on a starting model are more than half a wavelength apart from the observed data, FWI will get trapped into a local minima which will lead to erroneous estimates of the Earth's property changes [161]. Figure 1.2 (a) shows an erroneous model recovery for a horizontal layered model, when the velocities in the starting model have been perturbed by 5%. One way to mitigate the cycle–skipping issue would be to consider the low frequency component of the data; which in practise would require very large offsets. Typically 4D seismic monitoring involves looking for changes in relatively small regions, where a large offset survey would be expensive. A small offset survey, however, cannot record the diving rays necessary for FWI to successfully recover velocity changes at depth as it can be seen

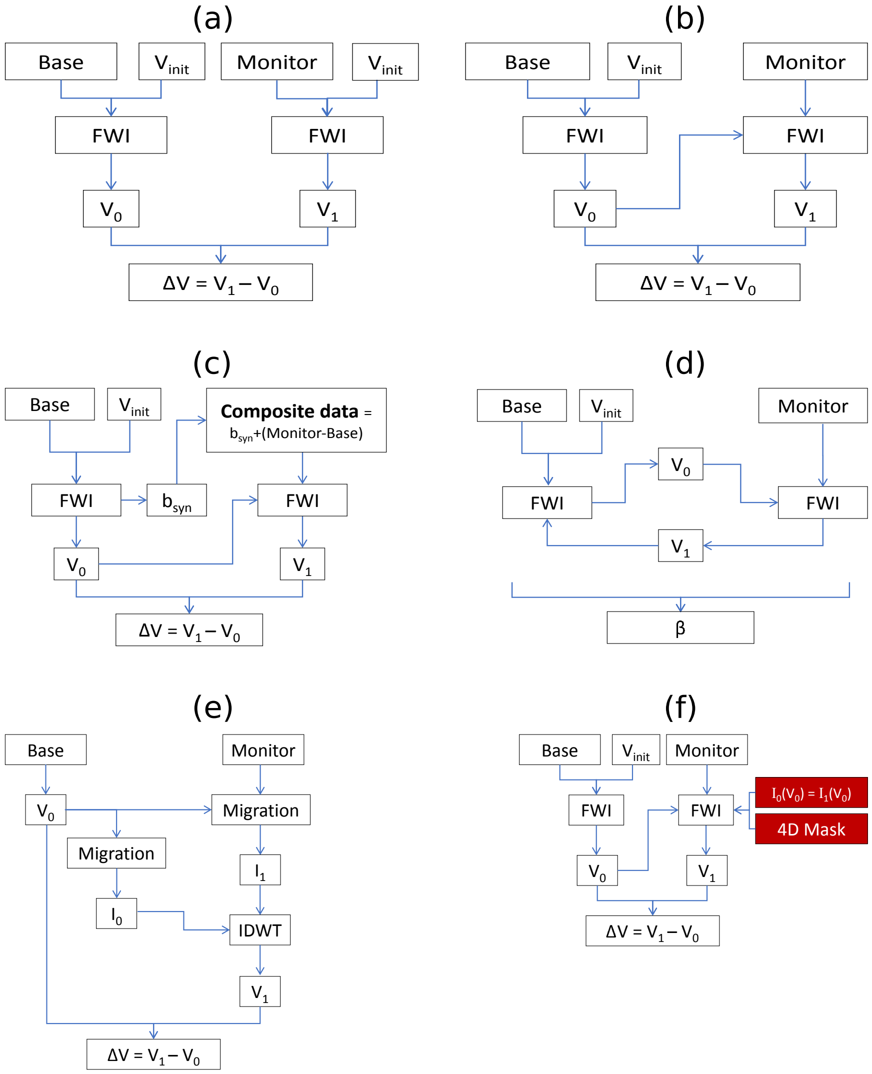

Figure 1.1: Flowcharts of all gradient–based 4D Full–Waveform Inversion approaches used in this thesis: (a) Parallel FWI (PFWI), (b) Sequential FWI (SFWI), (c) Double–Difference FWI (DDFWI), (d) Alternating FWI (AFWI), (e) Image Domain Wavefield Tomography (IDWT), and (f) Dual–Domain FWI (DDWI). The abbreviations noted in the above flowcharts are: **Base**: observed baseline data, **Monitor**: observed monitor data, $\mathbf{V}_{init}$: initial velocity model, $\mathbf{V_0}$: inverted baseline velocity model, $\mathbf{V_1}$: inverted monitor velocity model, $\mathbf{\Delta V}$: time–lapse velocity change, $\mathbf{b_{syn}}$: synthetic data generated on V_0, $\mathbf{I_0}$: baseline migrated image, and $\mathbf{I_1}$: monitor migrated image.

in Figure 1.2 (b).

In addition, it is essential that the baseline and monitoring surveys are repeatable. This includes but is not limited to the positions of sources/receivers, type of source, environmental conditions and noise in the data. Since the actual 4D changes are relatively small, any data differences generated by non–repeatable conditions, can either saturate the actual 4D changes or be misinterpreted as a property change too. Some authors have suggested using a model regularization which basically smooths the result [178] [132], however regularization risks delivering unrealistic results [134]. While FWI could potentially fail to solve the kinematics of the wave equation properly (due to the issues mentioned above), image domain methods, such as the one in Figure 1.1 (f), can overcome that while also retrieving the low-wavenumber components of the velocity model. Ideally one would want a 4D inversion strategy that can handle complex 4D changes while also being immune to most of FWI's challenges. In **Chapter 3**, we tackle this by proposing a Dual–Domain Wavefield Inversion (DDWI) scheme (Figure 1.1 (f)). Our new objective function combines data and image domain terms, while the velocity model updates are constrained using both terms simultaneously.

A major challenge of gradient–based FWI, as also in all geophysical inverse problems, is the non–uniqueness of the solution [7]. Since FWI is a highly nonlinear problem [26], any locally derived approximations of the objective function will lead to an underestimation of the underlying uncertainty and overconfidence in the best–fit model. 4D FWI depends on a mathematical model of the underlying physics, and any mathematical model carries uncertainty for various reasons. This begs the question: how well do we know what we think we know? In **Chapter 2**, we investigate the performances of Parallel FWI, Double Difference FWI and Alternating FWI (Figure 1.1 (a)(c)(d)) in the presence of model uncertainties. To do so, we generate different

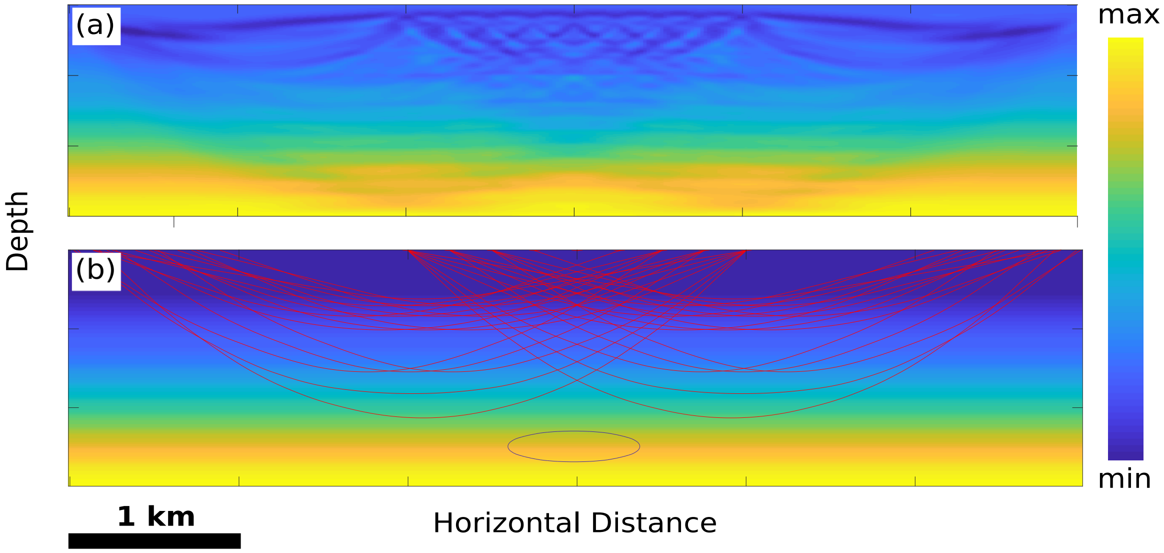

Figure 1.2: (a) The degraded FWI result when there is a 5% amplitude error in the smooth starting model. (b) When the target of interest is deep (e.g. deeper than 1 km), and the available acquisition does not have large offsets, diving waves (the red lines in this figure), that mainly FWI relies on, do not penetrate the target of interest.

model noise realizations (one could think of it as adding heterogeneity to the models) and computed different statistical estimators. While this analysis provides valuable information for the effectiveness of each method, it is somewhat adhoc in terms of its statistical robustness. In addition, one could argue that this uncertainty analysis is mainly local, due to the local optimization solution of the objective function. If the objective function is smooth and has a single minimum, approaches like this one, could potentially deliver a meaningful measure of the global uncertainty. However, in cases where the objective function is abrupt and highly multimodal, finding a global minimum can be extremely onerous, and hence one can only rely on local measures of uncertainty. This work builds the preceeding steps and motivation for the following chapters by highlighting the need of a rigorous uncertainty quantification framework.

Uncertainty quantification (UQ) is valuable for building knowledge but is not commonly done due to the computational challenge of exploring the range of all possible

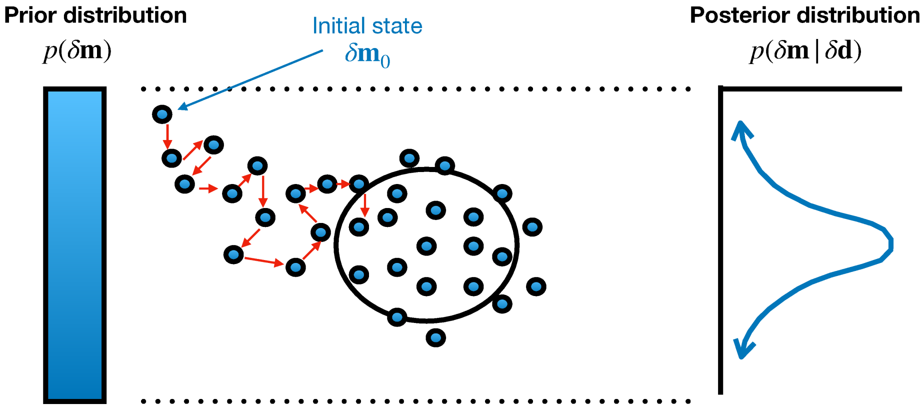

Figure 1.3: Simple illustration of the mechanics of 4D Metropolis–Hastings algorithm.

models that could equally fit the data. Ideally, one would want a computationally–feasible framework that drops the linearity assumption that all gradient–based FWI algorithms rely on, and also provides a robust estimate of the associated uncertainty. Both non–linearity and uncertainty quantification can be handled elegantly by Markov Chain Monte Carlo (MCMC). MCMC is a family of algorithms that aims to sample a probability distribution that cannot be estimated analytically or sampled directly [95] [131]. MCMC originates in the pioneering work of Nicholas Metropolis [93] where he combined the concept of the Monte Carlo sampling [94] with the ergodic properties of a Markov Chain (i.e. Markovian property and reversibility) to simulate distribution of states for molecules. Typically, these algorithms start by an assumption of the probability distribution (the so–called prior distribution), and after finite time of sampling the probability space they reach equilibrium and yield the so–called posterior distribution, which is the probability distribution of interest (Figure 1.3). MCMC methods are inherently stable because they do not rely on the objective function being smooth nor require any matrix inversion [131]. In addition, they appraise the solution by providing estimates of the associated uncertainty. In the context of time–lapse seismic imaging, UQ describes the distribution of 4D velocity changes given observed data.

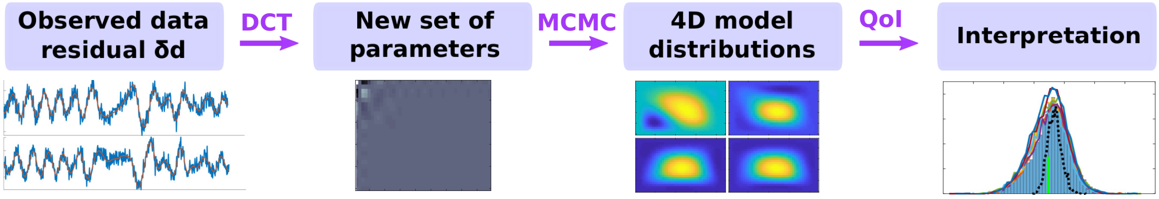

Figure 1.4: Illustration of the proposed 4D Metropolis–Hastings FWI from the observed data residual δd to distributions of quantities of interest.

Conveying an interpretation of the uncertainty in a velocity model can be challenging and a UQ framework, such as our proposed one in Figure 1.4, is likely the most useful when asking a specific question, such as how well known is the 4D velocity change in terms of its magnitude or its extent.

There are a variety of UQ approaches that one could choose from depending on the nature of the problem; evaluation of the forward solver, number of Degrees of Freedom (DoF), and dimensionality of the parameter space [143]. Sampling–based UQ methods, such as random–walk and Adaptive Metropolis–Hastings, tend to be slow in space exploration; however, they do not make assumptions about the distributions of the underlying uncertainty. This means that these distributions can be any type, including non–Gaussian and multimodal. This characteristic is particularly useful in seismic imaging, where the structure of model distributions is unknown; therefore, vigorous and straightforward algorithms such as Metropolis–Hastings are optimal to ground–truth these distributions. In **Chapter 4**, we choose Metropolis–Hastings (both random–walk and adaptive) to perform a 4D Full–Wavefield Bayesian Inversion. The main difference between random–walk and adaptive Metropolis–Hastings resides in how they generate the proposal distribution and hence how they sample the probability space. At each iteration of a random–walk Metropolis–Hastings algorithm a new sample is drawn from a Gaussian proposal distribution centred at the previous sample with a covariance C. This covariance C typically stays fixed for all iterations of the algorithm. The adaptive Metropolis–Hastings, on the other hand, updates this covariance C at each iteration using the history of the models already sampled in the chain. By doing so, the probability space can be sampled more efficiently allowing as such for a faster convergence compared to the random–walk Metropolis–Hastings. Metropolis–Hastings algorithms typically require far fewer degrees of freedom, 10s to 100s at most [51] [37], than the thousands used to describe even a simple velocity model. To provide a meaningful estimate of the velocity model distribution using Metropolis–Hastings we commonly assume that either the underlying distributions are Gaussian or parameter space needs to be reduced. Assuming that the velocity model distributions are Gaussian, basically means that we linearize the wave equation, which is not always a good approximation for our problem. To allow for nonlinearity in the model parameters and non-Gaussian distributions of models, the number of DoF needs to be reduced considerably. Since the nature of the time–lapse change is quite localized, simulations of wavefield propagation through the entire Earth model would be wasteful. In this thesis, we exploit this characteristic and use an exact acoustic local solver developed by [169]. By doing so, not only do we reduce the number of DoF, but we can also have very fast wavefield computations, an essential component for the hundreds of thousands realizations that are needed in order to allow the Metropolis–Hastings algorithm to converge. While the employment of the local domain significantly reduces the number of model parameters, it is still not enough. Additional model parameterizations need to be adopted (such as the one in **Chapter 4**), that not only respect the physics of the inverse problem, but also accurately describe the 4D velocity model with far fewer parameters.

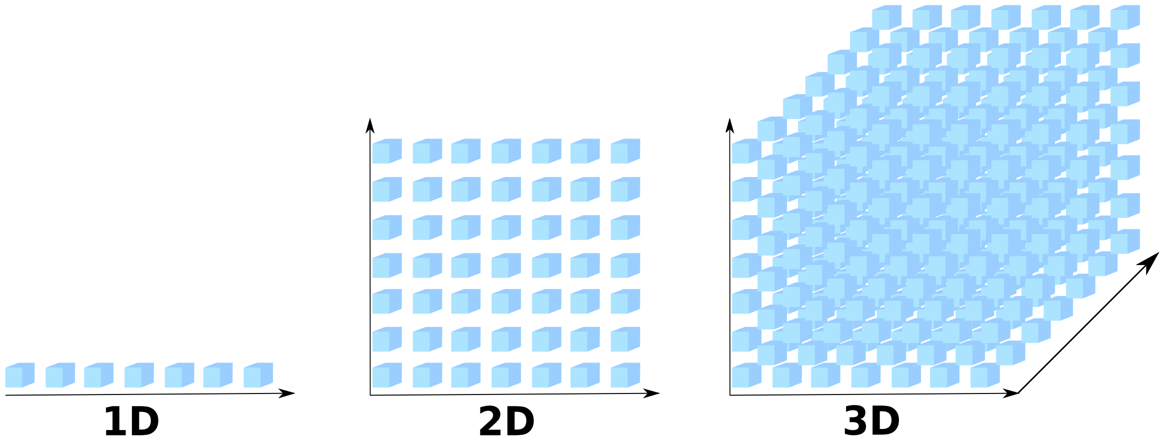

Figure 1.5: The curse of dimensionality: the number of configurations that need to be assessed grows exponentially as the underlying space dimensionality increases [12]. For example, if the number of data points that are needed to evaluate a function in a 1-dimensional space are 7, when we move to a 2-dimensional space the number of these data points increase to 7^2, and in a 3-dimensional space to 7^3 respectively.

However, not all reduced parameterization approaches allow for high fidelity models. In addition, Metropolis–Hastings suffers from the so–called "curse of dimensionality". In simple words, this means that as the number of dimensions grows, the features representing these dimensions grow exponentially. Figure 1.5 aims to provide a schematic understanding of dimensionality. If seven equally spaced features (cubes in Figure 1.5) are needed to describe a function in 1–dimension, with the addition of another dimension, we need $7^2 = 49$ (to keep the same space characteristic and fill the whole space). In n-dimensions, the number of these features will be 7^n. The effect of the increasing dimensionality eventually requires smaller step sizes in the Metropolis–Hastings proposals, which leads to far slower space exploration and subsequent correlated samples. Thus, obviously, the road ahead would be having a time–lapse nonlinear Full–Wavefield methodology that has no restrictions on the number of variables describing the 4D model and can provide uncertainty information. Hybrid Monte Carlo (as it was first introduced by [32] in a quantum chromodynamics study) or Hamiltonian Monte Carlo (HMC) as it is now most commonly known has seen a widespread application in numerous research fields such as neural networks [99] and molecular simulations [33] to name a few. However, it is not until recent years that it has been applied in geophysical inverse problems [133] [44] [27]. HMC allows for faster convergence by doing long–distance moves in the parameter space while also maintaining high acceptance rates. Hamiltonian Monte Carlo (HMC) explores a target distribution by incorporating information about its differential geometry into the search–proposal process. While adjoint state methods [109] typically allow for fast gradient calculations (when compared to direct finite differencing for example), they could still be a big computational burden given large models and thousands of iterations. To address this challenge, we utilize local domains (since 4D changes are localized) that considerably reduce the computational cost of a wavefield simulation, and generate a robust 4D HMC framework. A summary of strengths and shortfalls of our developed frameworks can be found in Table 1.1.

Although in this thesis we explore sampling–based UQ methods, there are non–sampling techniques available too. For example, one can find alternative methods that are based on approximations of the likelihood function such as the Polynomial Chaos Expansion (PCE) method [29] [35], where the algorithm tries to approximate the distributions of the uncertain parameters using a set of polynomials, resulting in computational cost reduction. However, PCE is commonly limited to low dimensional spaces. Another approach would be using Data Assimilation approaches or Ensemble Kalman Filters [38], which try to characterize a dynamic system over time. Such approaches solve inverse problems with a large number of DoFs while also providing an estimate of uncertainty through posterior convariance. Thus far, they have been applied to weather forecasting, oceanography and climatology, as well as geophysical imaging [98] [127] [74] [152]. In addition, computation and storage of the Hessian matrix, potentially through the help of machine learning algorithms or using local

	Gradient–based FWI	MH FWI	HMC FWI
Number of DoF:	$10^3 - 10^6$	$1 - 10^2$	$10^3 - 10^6$
Number of iterations:	≤ 50	$10^4 - 10^6$	$10^3 - 10^5$
Computational Expense:	Moderate	Low	High
Robust to local minima:	No	Yes	Yes
Independent of Initial model:	No	Yes	Yes
Uncertainty quantification:	No	Yes	Yes

Table 1.1: Performance and trade–offs comparison of the three solvers that are used in this thesis: Gradient–based, Metropolis–Hastings, and Hamiltonian Monte Carlo 4D Full–Waveform Inversion.

domains as in this thesis, can also be used as an uncertainty indicator when the solution of the minimization problem is close to the global minimum.

1.2 Thesis Outline

Here, we will briefly discuss the content of each chapter of this thesis. The order of the chapters reflects the chronological order that each research project took place.

To get a first understanding of how different gradient–based 4D FWI methods work and extract simple measures of uncertainty, in **Chapter 2** we compare Parallel, Double Difference, and Alternating FWI on two simple numerical examples; one of two horizontal reflectors and a five layer model. Although the considered models are trivial, we are aiming at broad understanding of the underlying uncertainty. In particular, we are interested in understanding how errors in the model are translated into errors in the final image, and thus in the interpretation of those images. To introduce the model errors, we add correlated Gaussian noise to either velocity or slowness models. Visualizing and interpreting the conveying uncertainty could be challenging, and hence we look at distributions of different spatial characteristics, such as the area of the 4D change for example. Even though we were limited to test only limited number of realizations (compared to what is typically done in UQ methods), we notice that for the particular numerical examples all three methods perform reasonably well. In addition, we found that when adding noise to the velocity, rather than to the squared slowness, we get significantly poorer recovered images. This is likely because the wave equation depends linearly on the squared slowness but non linearly on the velocity, resulting in a deterioration in the recovered models in the latter case.

In **Chapter 3**, we address some of the limitations of conventional 4D FWI, such as the need of large offsets, and accurate starting models, by proposing a Dual–Domain Waveform Inversion (DDWI) technique. The new hybrid objective function nicely combines information from the diving waves in the data-domain term with information from the reflected waves in the image-domain term. Our framework can handle complicated 4D changes without resorting to manual interventions in either data or image domain. At each iteration, we update the velocity model using constraints from both terms simultaneously. An essential aspect of this new hybrid methodology is that the image-domain constraint is not relaxed during the inversion, meaning that the inversion cannot deviate from the geologic prior. This characteristic is beneficial in cases where the starting model is not in the vicinity of the true model. We test our algorithm on a variety of synthetic models of increasing complexity and find that it outperforms traditional FWI or image domain inversions.

In **Chapter 4** we develop a computationally feasible framework that enables time-lapse uncertainty quantification in a targeted way (Figure 1.4). We achieve fast wavefield simulations by exploiting the structure of the 4D problem and use a local solver to perform an inversion focussed on a specific sub-domain of interest within a larger model. We add random noise to our data and we want to answer UQ questions related to time–lapse change, such as the magnitude of the change or its spatial extent. While the added measurement noise is Gaussian, we do not assume a specific type of distribution for the posterior. We derive a 4D Bayesian scheme that only depends on time–lapse changes both in terms of models and data. One key assumption of our derivation is that the posterior calculation is independent of the background model; we address this assumption both analytically and numerically. To reduce the number of variables describing the 4D velocity model, we use image compression techniques, where the model is expressed with a set of coefficients. We apply our framework to and justify its success with both single and multiple DoF examples.

Metropolis Hastings (MH) is typically used to ground truth probability distributions, which is particularly useful in seismic imaging where the distributions of most structures of interest are not well known. However, MH tends to be less efficient and slow to converge. Building upon our work on 4D–MH, in **Chapter 5**, we focus on more advanced uncertainty quantification methods, such as Hamiltonian Monte Carlo (HMC), which can handle higher dimensions while providing faster convergence. Extending our framework to a method that can handle high dimensional models allows us to more accurately model the subsurface. HMC requires the gradient calculation of the likelihood function, which is computationally feasible when it is computed in a local domain using adjoint state methods. We first provide a proof of principle that HMC can be used for a probabilistic 4D inversion by demonstrating the algorithm on low dimensional examples where one can easily visualize aspects of the algorithm, such as trajectories in phase space. We then focus on a significantly higher dimensional problem by inverting for the full local domain.

In **Chapter 6** we summarize our innovations and discuss potential improvements and future directions.

Chapter 2

A statistical comparison of three 4D Full-Waveform Inversion schemes

2.1 Summary

Multiple seismic data sets are often recorded to monitor changes in Earth properties. To image these changes, several different 4D Full Waveform Inversion (FWI) schemes have been successfully applied over the past decade. We compare three different 4D FWI schemes on two simple numerical examples to quantify how each method performs. To do this, we create correlated Gaussian noise realizations and add them to our models to determine how errors in the models are translated to errors in the final images. We computed spatial characteristics of the recovered models and compare the performance of the different 4D FWI schemes. Our results indicate that while there are minor differences between the different proposed methods all perform reasonably

well for this type of noise in these simple models. The methods that specifically target 4D changes do result in fewer artifacts outside the region of true change, but all methods recover the true change with similar accuracy.

2.2 Introduction

During hydrocarbon production changes occur in the reservoir geometry and pore fluid properties. Geophysical monitoring of these changes allows for the estimation of the extraction efficiency and determination of the remaining reserves [103]. Time-lapse (4D) seismic is the most commonly used technique for geophysical monitoring. In 4D seismic, differences between multiple surveys at the same site reveal changes in the reservoir. The first survey acquired is called the baseline survey and all subsequent surveys are called monitor surveys [79]. Results from studies using Full Waveform Inversion (FWI) to recover 4D changes have been encouraging thus far [6]. Like all inversion methods, the objective of FWI is to deliver a velocity model of the subsurface by iteratively matching predicted and observed seismic data [145] [161]. FWI can be extended to the time-lapse case successfully, however artifacts may arise due to the non linearity of the inverse problem and the non-repeatability of the surveys. To overcome this challenge different FWI approaches have been developed and used [167] [182] [171] [173] [84]. Parallel, Double Difference, and Alternating FWI are used in this study and they will be explained in more detail in the following section. Since 4D monitoring involves looking for small changes in localized regions, understanding the uncertainty in the measurement of those changes is key. For this study we use two different numerical examples: one of two horizontal reflectors and a five layer model. We then try to understand the uncertainty in the recovered changes by comparing the performances of the three methods. To do this, we introduce coherent noise and observe how errors in the models propagate to the final image.

Although we consider simple models here, we are aiming at a more comprehensive understanding of uncertainty. Characterizing the uncertainty pixel-by-pixel in a large model is not computationally feasible and it is not clear that doing so would help in the interpretation due to the volume of information generated. We focus instead on the idea of characterizing the uncertainty of key elements of the image. As a first step towards this goal, we estimate some parameters of the image which we refer to as spatial characteristics, and we explore how are recovered by different FWI methods.

2.3 Theory

The most commonly used objective function in FWI is a least squares measure:

$$J(m) = \sum_{x_s} ||F(m, x_s) - d(x_s)||_2^2, \tag{2.1}$$

where m is the model, usually $\frac{1}{c^2}$ where c is the velocity, d is the observed data and F is the forward modeling operator. The extension of FWI to the 4D case can be straight forward or complicated in a number of ways. In this section, we describe the methods compared in this paper.

The first method we explore is Parallel FWI. This method is the most straightforward 4D extension of FWI. In this scheme, we perform FWI runs independently on both baseline,

$$J(m_0) = \sum_{x_s} ||F(m_0, x_s) - d_0(x_s)||_2^2, \tag{2.2}$$

and monitor,

$$J(m_1) = \sum_{x_s} ||F(m_1, x_s) - d_1(x_s)||_2^2. \tag{2.3}$$

The difference between the two recovered models illuminates the areas of the time-lapse changes. Even though PFWI is easy to apply, it is also very naive considering the complexity of the problem. In realistic scenarios, there is usually noise in the data from different origins such as equipment and environment. For a successful 4D analysis, the surveys need to be almost perfectly repeatable. In a standard marine environment for example, non–repeatability and noise can arise from acquisition geometry mismatch, seasonal water velocity and salinity changes, overburden changes, and source wavelet inconsistencies. Therefore, noise from any of these sources is usually different between the two sets of data. When we are solving for the successive linearized problem, we can end up in different local minima, in which case subtracting the two results leaves artefacts that are not related to the true 4D change. In order to overcome those challenges different schemes have been developed. One of those schemes, even though it is not studied in this chapter, is Sequential FWI. In Sequential FWI (SFWI), we first perform FWI runs on the baseline model given a smooth initial model m_{init} to recover a baseline velocity model. We then use the recovered m_0 as an initial model for the monitor model inversion. The final 4D signal is computed by subtracting the two models. If the recovered baseline velocity model is good, SFWI provides fewer artifacts in the time-lapse recovery than PFWI. However, if the recovered baseline model is not good, in the sense of converging to the wrong local minima, then this wrong local minima remains in the recovered monitor model. This will lead to FWI updating the model in areas other than the 4D change.

The second method we compare is Double Difference FWI (DDFWI). Double Difference FWI is another FWI extension that is often utilized for its robustness [31] [171] [175] [167], particularly to different surveys falling into different local minima. The concept was first introduced to locate earthquakes' hypocenters [162] and in recent decades has been extensively used in 4D imaging. The method is based on the sequential approach with an extra step between the two inversions. Similar to SFWI, we first invert for the baseline model given a smooth initial model m_{init}. We then take the inverted baseline model $\hat{m_0}$ and generate synthetic data, which we call b_{syn}. Then, we add b_{syn} to the data difference to create the so–called composite data d_{comp}

$$d_{comp} = b_{syn} + (d_1 - d_0), \tag{2.4}$$

where d_1 and d_0 are the monitor and baseline observed data respectively. Then, we perform FWI runs for the monitor model, starting from the baseline inverted $\hat{m_0}$, but we use d_{comp} rather than d_1. The model to be minimized can be expressed as

$$J(m_1) = \sum_{x_s} ||F(m_1, x_s) - d_{comp}(x_s)||_2^2, \tag{2.5}$$

$$= \sum_{x_s} ||F(m_1, x_s) - (b_{syn}(x_s) + (d_1(x_s) - d_0(x_s)))||_2^2, \tag{2.6}$$

$$= \sum_{x_s} ||(F(m_1, x_s) - b_{syn}(x_s)) + (d_1(x_s) - d_0(x_s))||_2^2, \tag{2.7}$$

$$= \sum_{x_s} ||(F(m_1, x_s) - F(m_0; x_s)) + (d_1(x_s) - d_0(x_s))||_2^2. \tag{2.8}$$

The two differences in equation (2.8) is the reason why this method is called Double Difference FWI. In this way, DDFWI is more robust with respect to the starting model since the only signal in the composite data that is not explained by the starting model is the 4D signal. However, due to the data subtraction (equation 2.8) there is a strong dependence on the acquisition geometry of the surveys; a scenario that is not optimal in real cases.

The third method we study is Alternating FWI (AFWI). This method attempts to both mitigate the reliance on perfectly repeatable surveys and give us a measure of uncertainty [173]. To do this, a set of weights, β, are calculated from the differences in how the baseline and monitor models converge. More precisely, we first perform FWI runs on the baseline data for n number of iterations giving an update. We then use this update to perform n FWI iterations on the monitor data until we get a new update. We keep alternating — hence the name of the method — between the two datasets for multiple times. Once we have all these model updates, we can calculate β by

$$\beta = \sum_i (1 - sgn[(m_{i-1} - m_i)(m_{i+1} - m_i)])|m_{i+1} - m_i|. \tag{2.9}$$

This set of weights β can be thought of as a confidence map of changes that highlights areas that have the highest probability of change. This is possible because β is derived from the convergence curves of the model parameters (equation 2.9). In areas with no time-lapse changes β will be small, whereas at places where there are time-lapse changes, β will be big. β is then used as a regularization parameter to constrain the final joint inversion for the change in the material properties. The models to be minimized in this case are

$$\begin{aligned} J(m_0, m_1) = &\sum_{x_s} ||F(m_0, x_s) - d_0(x_s)||_2^2 + \sum_{x_s} ||F(m_1, x_s) - d_1(x_s)||_2^2 + \\ &\frac{1}{2}\left\|\frac{m_0 - m_1}{\beta}\right\|^2, \end{aligned} \tag{2.10}$$

where m_0 and d_0 are the baseline model and data, and m_1 and d_1 are the monitor model and data respectively. In the next section, we describe the two numerical examples and the spatial characteristics calculation for the respective examples. We then compare the recovered spatial characteristics for the three methods on two simple models.

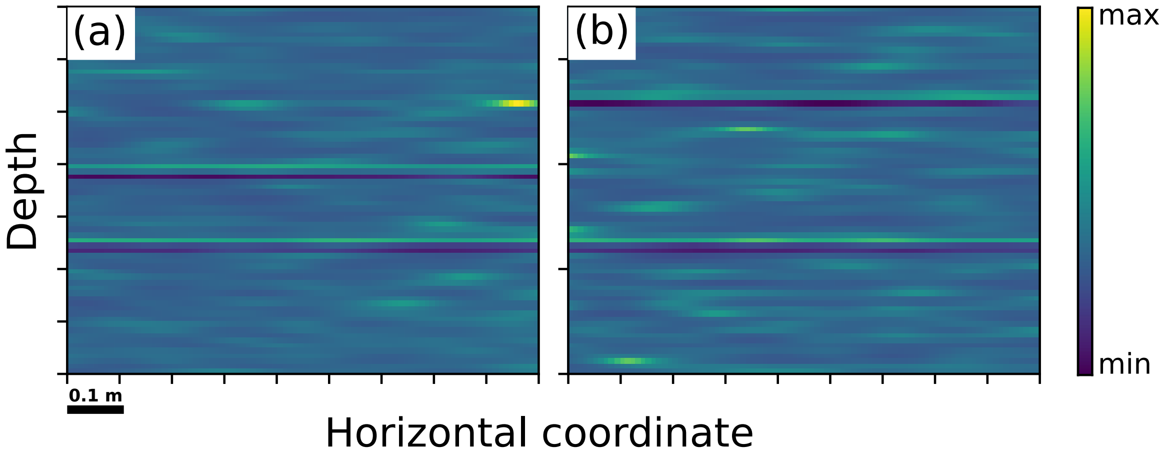

Figure 2.1: Horizontal reflectors example: the true noisy baseline model (a) and the true noisy monitor model (b) for one of the noise realizations.

2.4 Numerical Examples

2.4.1 2D Horizontal Reflectors

For the first example we use a simple model with a homogeneous background and two horizontal reflectors. All of our calculations are done with the PySit package [56]. To introduce a change between baseline and monitor models we shift the position of the top reflector [65]. We generate one hundred different realizations of Gaussian distributed random correlated noise, with a correlation length that is bigger in the x-direction than in the z-direction. We then scale these random distributions so that they have the same average velocity as the true model and so that we are adding a zero-mean random field to the squared slowness. Specifically, we take the normalized perturbations ($normPert$) and scale them via:

$$velRan = \frac{velAve}{\sqrt{1 + pertAmpl * normPert}}, \tag{2.11}$$

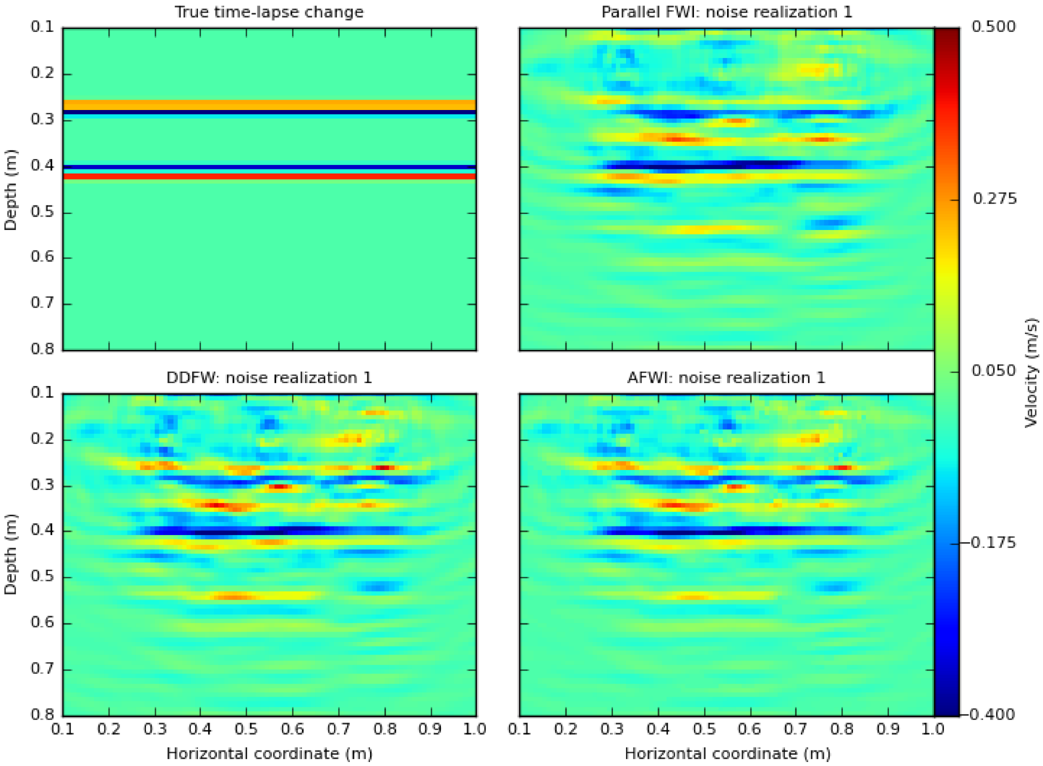

Figure 2.2: Horizontal reflectors example: True time-lapse change together with the recovered changes from all three FWI schemes time-lapse changes, for the same model realization.

where $velRan$ is the final random noise velocity model, $velAve$ is the average velocity used, and $pertAmpl$ is the amplitude of the perturbations.

We then add these random perturbations to the true baseline and monitor velocity models (Figure 2.1) and generate noisy data sets (note that we use different noise realizations for the baseline and monitor data sets). For each of the 100 noise realizations we apply each of the three FWI methods described above to recover the time-lapse changes (Figure 2.2). We then apply an image stacking procedure as follows. For each horizontal position, we compute the distance Δ_z between the recovered reflectors by taking the maximum value of the image in a window (Figure 2.3). We then average

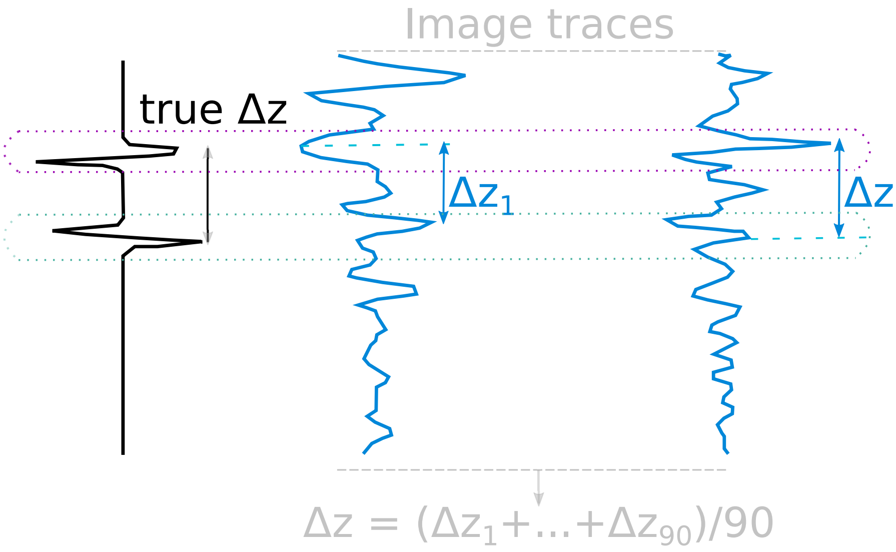

Figure 2.3: Simple schematic illustration showing how we compute the distance Δ_z between the recovered reflectors.

the recovered distances over the entire image to obtain an average Δ_z for a particular image. Figure 2.4 shows the histograms of all the recovered Δ_z for each of the FWI schemes together with their calculated standard deviations.

The recovered Δ_z from all three histograms are approximately normally distributed with most of the results being concentrated around the $\Delta_z = 16$, which is the true value. In order to measure the spread of the values of Δ_z, we compute their standard deviationa via

$$\Sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Delta z_i - \overline{\Delta z})^2}, \tag{2.12}$$

where $N = 100$ is the number of realizations, Δz_i represents each value in the retrieved histograms, and $\overline{\Delta z}$ is the mean value of each histogram. Even though the deviations in Figure 2.4 are sometimes large, none of them are larger than 1 from the true Δ_z. It is important to mention that defining an accurate picking window is important. In this particular case a larger window allows more artifacts to be included in the calculation of Δ_z leading to less reliable results in terms of higher standard deviations.

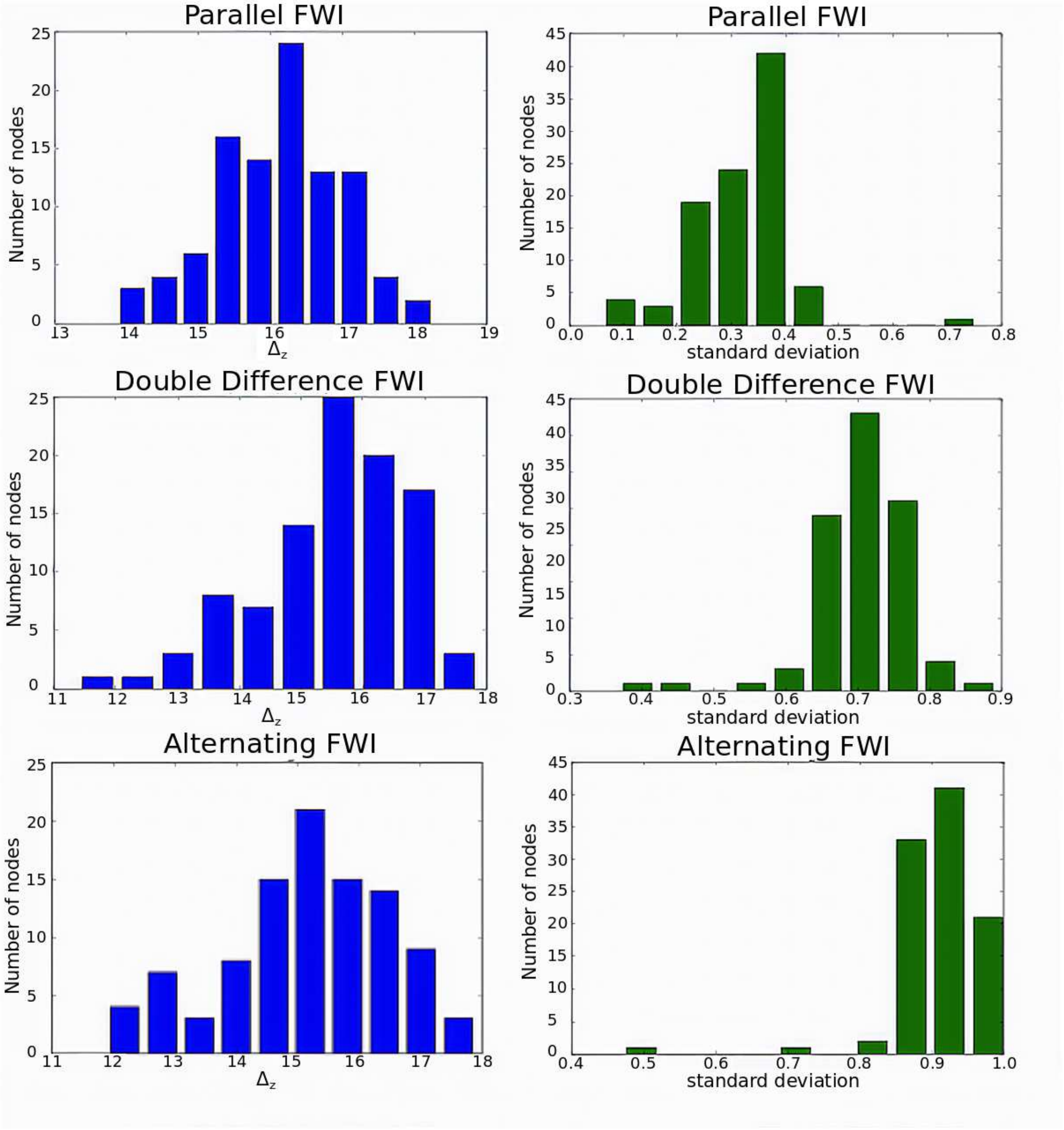

Figure 2.4: Left: Histograms of the Δ_z of the recovered time-lapse changes from the three FWI schemes. Right: the corresponding standard deviations.

2.4.2 Layered Model

We use a five layer model to incrementally increase the complexity of the model. The thickness of the middle layer is increased from baseline to monitor introducing a velocity perturbation of magnitude 0.5 km/s and a change in thickness of half the layer thickness. The initial model for the inversion was created by applying a Gaussian smoothing filtering with a σ of 10 to the true baseline model (Figure 2.5). To see how random noise affects this model, we consider two cases. In case 1 we calculate the perturbed velocity model as in equation 2.11 by applying zero-mean perturbations around the slowness squared. In case 2 we add the normalized correlated noise directly to the velocity instead of the slowness.

We create fifty noise realizations and add them to the true models. Once again, we apply all three 4D FWI approaches to estimate the time-lapse changes (Figure 2.6). Due to the different nature of the time-lapse change in this experiment, the spatial characteristic we are interested in is the *area* of the recovered changes. To compare the different methods, we compute the area of the changed region. To do this, we define a target velocity change, we use ±30% of the true change, and compute the number of pixels within a depth dz, that are within our velocity range, giving us an estimate of the area of the recovered change. Note that this calculation does not consider any lateral-discontinuities that might be present in the recovery of the layer.

The true change has an area of 6.3 m^2. For comparison we performed the 4D FWI schemes on the noise free case. The resulting area for parallel FWI is 3.76 m^2, for DDFWI 4.16 m^2, and for AFWI 3.7 m^2. To be able to compare noisy and noise free cases instead of plotting the histograms of areas, we plot the deviations from the areas in the noise free case, in other words we normalize the results of each method by the recovered area in the noise free case for that method. We calculate the area

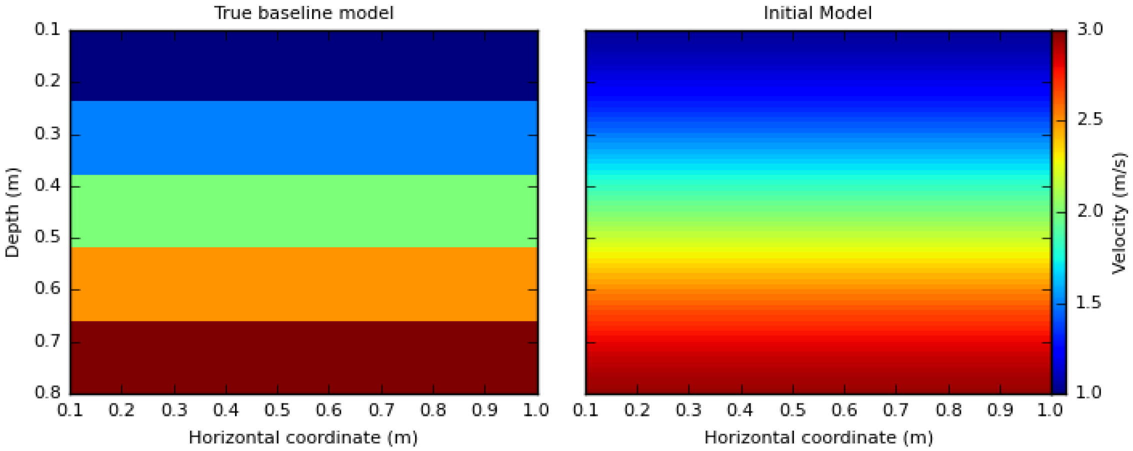

Figure 2.5: The true five layer baseline model on the left and the initial guess used for the inversion on the right.

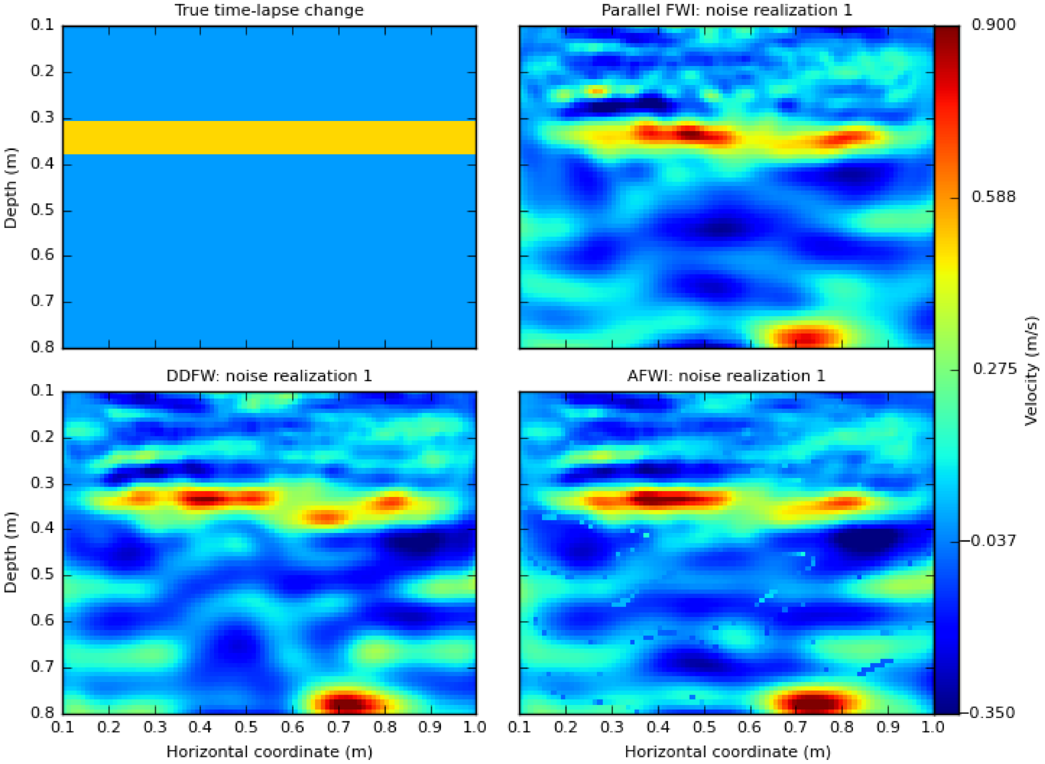

Figure 2.6: Five layer example: True time-lapse change together with the recovered for all three FWI schemes time-lapse changes for the same model realization in type 1 noise.

both within the depth range dz and throughout the entire model. The area within dz gives us a measure of how well each method recovers the true change and the calculated area within the entire domain gives us a measure of how many artifacts are introduced into the recovered change image. Figure 2.7 shows the histograms of the area deviations from the case 1 noise in the depth dz on the left and throughout the total depth on the right. In the depth dz all three methods perform well with the peaks of the histograms being near 1 as expected. Parallel FWI has the flattest distribution indicating that it is the method with the least precision in recovering the final model. Of course extremes are also present in the other two cases, but they are fewer and the distributions are thus a bit sharper around the optimal recovered area. When we are looking at the whole depth, we are letting all the potential artifacts be included in our calculation. An overestimation is therefore expected. DDFWI and AFWI perform better compared to the parallel FWI, meaning they are more effective at suppressing random noise and artifacts.

For the case 2 noise we do not scale the normalized correlated noise and we are thus adding higher amplitude noise, which translates into a higher noise level in our final images. Therefore, we extend the velocity range we consider to be a correct recovery by a factor of 2. Figure 2.8 shows the resulting histograms of the area deviations. All of the histograms are broader indicating that the accuracy of all of the methods is diminished. This is to be expected as adding noise to the velocity instead of the squared slowness introduces a more complicated error because the wave equation is linear in squared slowness but non-linear in velocity. Both AFWI and DDFWI clearly outperform parallel FWI in this case, as evidenced by their significantly narrower distributions with fewer outliers particularly when comparing areas within the region of interest.

2.5 Conclusions

In this chapter we compared three 4D FWI approaches to evaluate their relative performance in a statistical way. We used two simple numerical examples, one with two horizontal reflectors in a homogeneous background and a five layer model. We added different realizations of correlated Gaussian random noise to our models, and we calculated spatial characteristics in the recovered images. In the two horizontal reflector example we calculated the distance between the two recovered reflectors in the time-lapse change. The histograms are approximately normally distributed and the standard deviations show similarly good performance for all three FWI methods. In the five layer model we calculated the area of the time-lapse change in the final image for two different types of noise. We found that when adding noise to the velocity, rather than to the squared slowness, we get significantly poorer recovered images. This is likely because the wave equation depends linearly on the squared slowness but non–linearly on the velocity, resulting in a deterioration in the recovered models in the latter case. This observation could be linked to the smaller phase variation when considering slowness perturbations rather than velocity perturbations as shown by the homogenization theory (e.g. [24]). The occured deterioration of results primarily flattens the associated histograms and is particularly noticeable for the parallel FWI case. Our results also indicate that both AFWI and DDFWI are successful at attenuating artifacts outside of the region of true change. Last but not least, when comparing different methodologies for FWI it is important to mention how their ability to fit the data is affected by the acquisition geometry. Repeatability of seismic surveys is a key factor for a successful time–lapse analysis. Data differences caused by non–repeatability sometimes can be stronger than the actual time–lapse changes, leading to a potential break down for these methods. Some of the repeatability issues, such as source wavelets discrepancies and survey geometry, can be mitigated with pre–processing. However, changes such as overburden changes, are much harder to be accounted for. In general, both Parallel and Double–Difference FWI are more sensitive to non–repeatability compared to Alternating FWI. This is because AFWI tries to highlight time–lapse model changes rather time–lapse signal changes.

2.6 Acknowledgments

This work is supported by Chevron and with grants from the Natural Sciences and Engineering Research Council of Canada Industrial Research Chair Program and the Research and Development Corporation of Newfoundland and Labrador and by the Hibernia Management and Development Corporation. We would like to thank Oleg V. Poliannikov from MIT for sharing his code for the generation of the Gaussian perturbed velocity models.

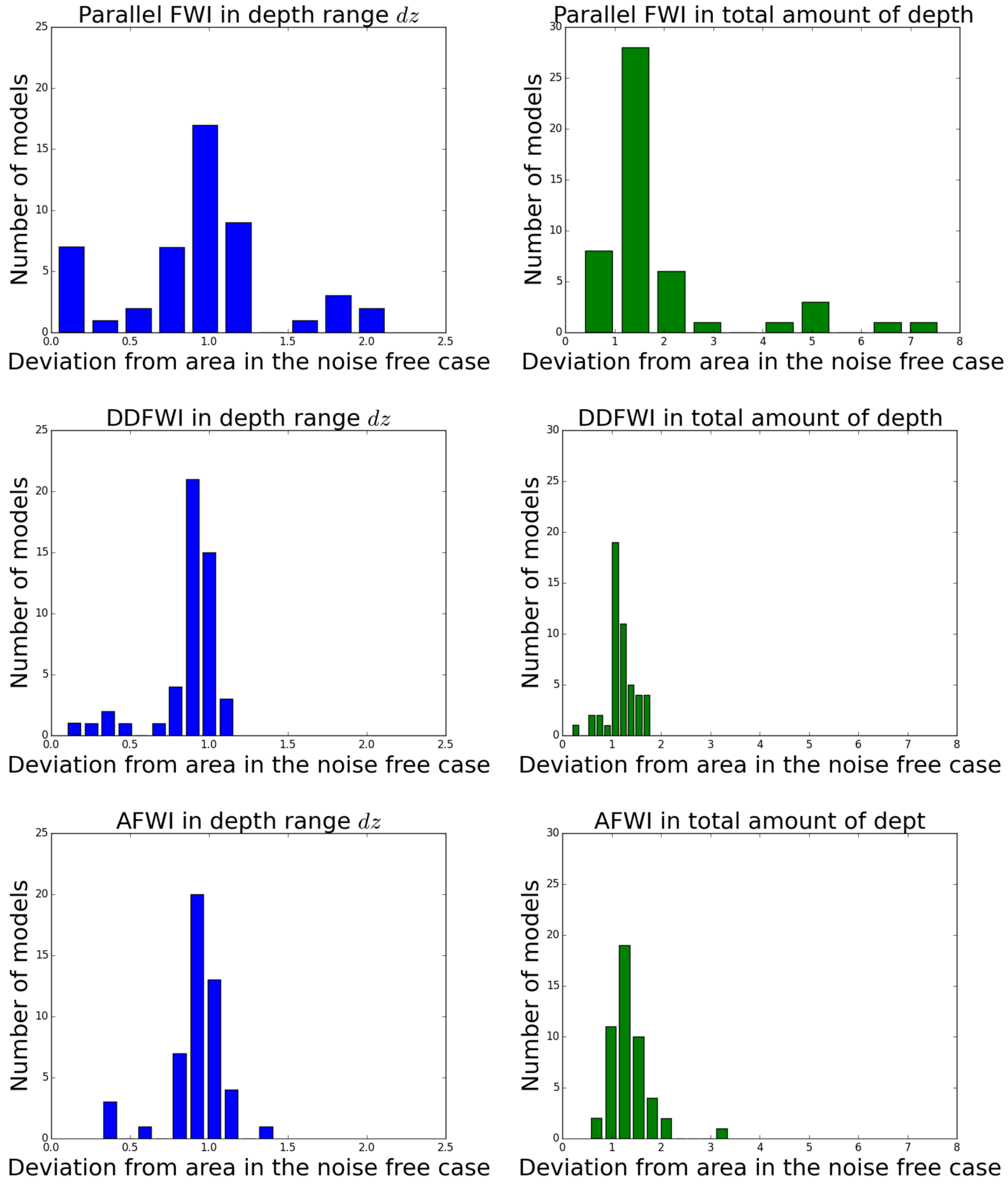

Figure 2.7: Case 1 noise results from all three FWI schemes. Left: Histograms of the area deviations in depth dz. Right: Histograms of area deviation in the total depth.

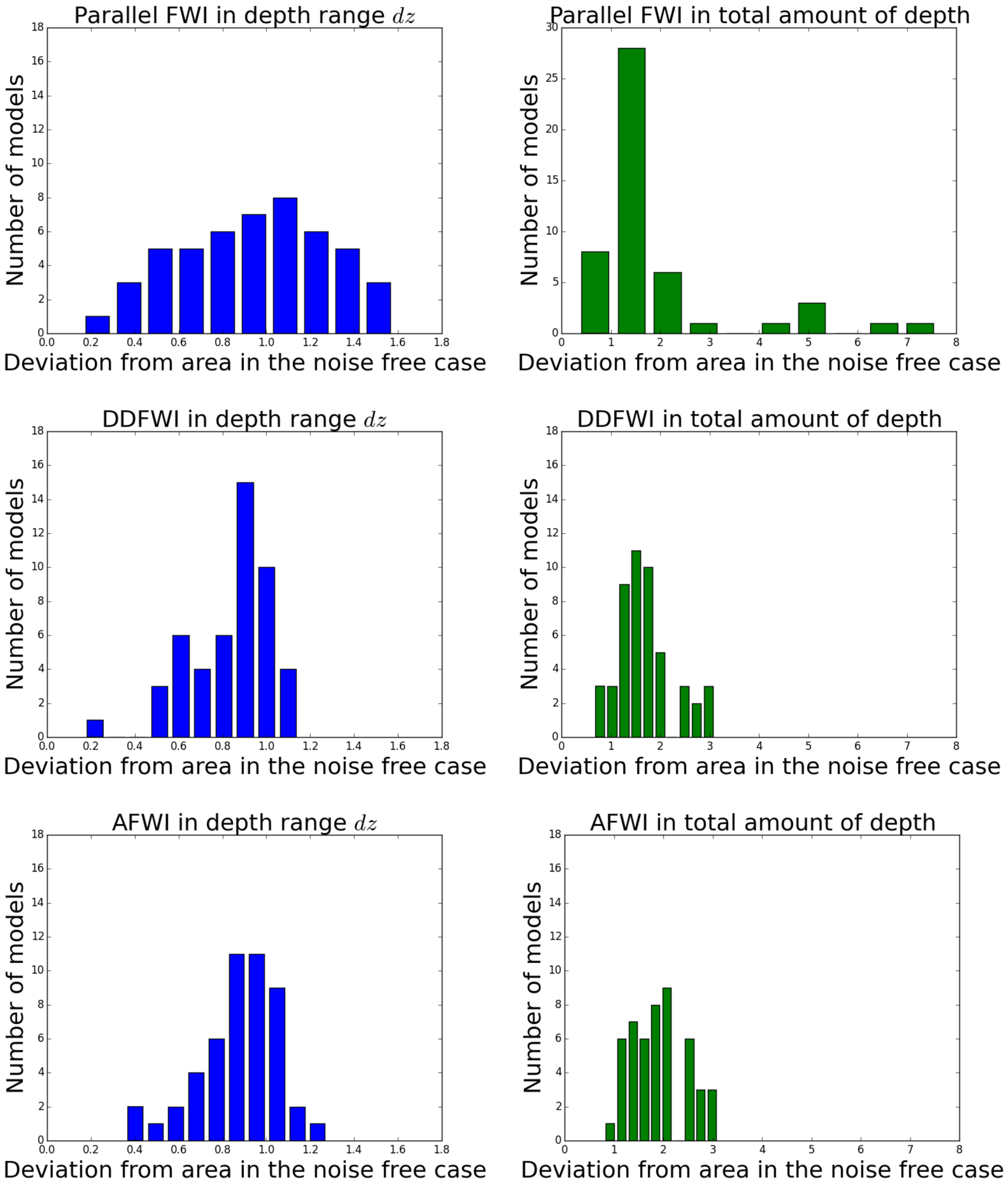

Figure 2.8: Case 2 noise results from all three methods. Left: Histograms of the area deviations in depth dz. Right: Histograms of area deviation in the total amount of depth.

Chapter 3

Combining reflection and transmission information in time-lapse velocity inversion: A new hybrid approach

3.1 Summary

Full-Waveform Inversion (FWI) uses the information of the full wavefield to deliver high resolution images of the subsurface. Conventional time-lapse FWI uses primarily the transmitted component (diving waves) of the wavefield to reconstruct the low wavenumber component of the velocity model. This requires large offset surveys and low frequency data. When the target of interest is deep, diving waves cannot reach the target and FWI will be dominated by the reflected component of the wavefield. Consequently, the retrieved model resembles a least-squares migration instead of a

velocity model. Image Domain methods, especially Image Domain Wavefield Tomography (IDWT), have been developed to obtain a model of time-lapse velocity changes in deeper targets using reflected waves. The method is able to recover models of deep targets. However, it also tends to obtain smeared time-lapse velocity changes. We present a form of time-lapse waveform inversion that we call Dual Domain time-lapse Waveform Inversion (DDWI), whose objective function joins FWI and IDWT, combining information from the diving waves in the data-domain FWI term with information from the reflected waves in the image-domain IDWT term. During the non-linear inversion, the velocity model is updated using constraints from both terms simultaneously. Similar to sequential time-lapse waveform inversion we start the time-lapse inversion from a baseline model recovered with FWI. We test DDWI on a variety of synthetic models of increasing complexity and find that it can recover time-lapse velocity changes more accurately than when both methods are used independently or sequentially.

3.2 Introduction

During hydrocarbon production, changes occur in the reservoir geometry and pore fluid properties. Monitoring these changes allows us to estimate the extraction efficiency and determine remaining reserves. Geophysical monitoring of an oil field provides valuable information about these changes during production, particularly offshore where well control is sparse and drilling is expensive. The technique most commonly used in monitoring is time-lapse seismic. Time-lapse (4D) seismic is the acquisition and analysis of multiple seismic surveys over the same site over time [79]. The first survey acquired is called the baseline survey and subsequent surveys are called monitor surveys. Differences between surveys reveal changes in the reservoir and overburden. Time-lapse changes can result from amplitude changes and/or time shifts. Amplitude changes can be caused by new structures in the target area or reflectivity differences at interfaces. Time shifts are usually the result of physically shifted geological boundaries or velocity perturbations along the wave path. Understanding which of these two mechanisms control the observed time shifts is important to better estimate and interpret the time-lapse changes.

Most 4D analyses are conducted on full- or partial-stack data [8] [72] [73] [153], where a lot of information and sometimes 4D signal has been lost. To mitigate this potential loss of 4D signal in stacking, a variety of pre-stack methods have been proposed. [34] propose a ray based tomography without the 1D approximation of seismic energy propagation resulting in more accurate seismic energy modeling and thus more accurate inversion. [102] address the repeatability issue of 4D monitoring by using a cost efficient subsampled acquisition and they propose a joint recovery scheme that is based on the curvelet coefficients of the pre-stack 4D vintages. [139] assess the effect of 4D seismic noise in the amplitude and impedance domain. They conclude that the amplitude domain is more robust and can be used for quantifying reservoir fluid properties. [88] apply a frequency based local domain solver to Double Difference FWI and observe a decrease in both computational cost and number of iterations. [137] address the repeatability issue of time-lapse surveys by applying an interferometric least-squares migration on both synthetic and field data.

FWI is built upon original ideas from [145] to estimate a velocity model of the subsurface by iteratively matching modelled and recorded data; see [161] for a recent overview. FWI can be extended to the time-lapse case successfully and results from studies using Full Waveform Inversion (FWI) to recover 4D changes have been encouraging thus far. [167] successfully apply a differential waveform inversion in crosswell seismic data to monitor a gas reservoir under production. [182] utilize and compare three elastic 4D FWI schemes on OBC data. [129] apply a broadband FWI in a parallel and sequential scheme and compare their robustness on a synthetic example based on a producing field in West Africa. [173] introduce a regularized joint FWI where the regularization parameters are obtained from the convergence curves of the model parameters by fitting baseline and monitor datasets in an alternating manner. [84] introduce a simultaneous 4D FWI with a model-difference regularization and a cross-updating method and compare their performances with other 4D FWI techniques on noisy synthetic data. [85] apply the TV-regularized simultaneous inversion to the Genesis field in the Gulf of Mexico. [6] evaluate and compare three different 4D FWI schemes on noisy and noise free data and propose a target oriented imaging and regularized FWI with the use of an a priori model. [57] compare the performance of the standard parallel 4D FWI with their proposed common-model 4D FWI on the Grane field in North Sea, and they observe more accuracy in the 4D recovery. [172] apply Double Difference FWI on OBC data from the Valhall field in the North Sea. [66] provide a statistical comparison of the performance of three 4D FWI schemes on noisy models. [86] use a cascaded 4D FWI by choosing different regularization penalties and achieve better recovery compared to the traditional parallel FWI.

For a successful application of FWI low frequency data and large offsets are required [138]. Typically 4D seismic monitoring involves looking for small changes in localized regions, where a large offset survey would be expensive. A small offset survey, on the other hand, cannot record the diving rays necessary for FWI to successfully recover velocity changes at depth. Additionally, while FWI is robust in recovering amplitude changes, it is often not able to solve the kinematics properly due to cycle skipping issues and initial model inaccuracies. This is a problem when the 4D change is primarily due to time shifts, because FWI tends to interpret the time shifts as an amplitude change, and therefore introduces a false 4D signal. Image Domain methods are better at solving the kinematics of the wave equation and recovering the low wavenumber components of the velocity model. [83] propose a layer stripping approach in wave equation image difference tomography and they achieve rapid convergence, with fewer local minima, and their method is less susceptible to cycle skipping. [176] improve the accuracy and robustness of Differential Semblance Optimization (DSO)— an implementation of image domain wavefield tomography — by replacing the conventional penalty operator with one that compensates for uneven illumination by measuring the illumination effects on space-lag extended images. [174] propose IDWT specifically for the small-offset case, to recover velocity changes from time shifts. In that study, time-shifts are translated into velocity changes using dynamic warping [53]. In other words, the migrated images of the baseline and monitor data are calculated first, and then a warping function is applied to measure how much one image is shifted from the other and translate that shift into a 4D velocity change. The method is appealing in cases where there are acquisition limitations and survey non-repeatability. In IDWT, we assume that the reflectors do not shift in depth during the period of the time-lapse surveys. [174] show that the velocity changes are smeared through the reflectors at the layer boundary that the anomaly is placed on top of. Here, we use a mask that helps us recover a less smeared velocity change and speeds convergence.

In real case scenarios, the 4D signal is a complicated combination of time shifts and amplitude changes. This can result in decreased performance for both methods, depending on the nature of the complication. This situation highlights the need for a methodology that is able to handle these more complicated cases without having to separate the two mechanisms of change nor needing to run different methods sequentially for a more accurate estimation with less uncertainty. Several approaches have been proposed for velocity model building that incorporate both reflected and diving wave information. [14] propose extending the velocity model along the time-shift τ axis to make the modeling of large time shifts more efficient. This allows them to develop an FWI approach that successfully recovers a velocity model from an inaccurate starting model when both reflected and diving waves are recorded. [4] propose an updated FWI objective function based on unwrapped phase attributes of the wavefield and the misfit of the instantaneous traveltime between observed and modelled data. They recover high resolution velocity models with less dependence on the initial velocity model. [183] propose a Joint FWI based on conventional FWI and Reflection Waveform Inversion (RWI) for more accurate velocity building using both diving and reflected waves. [60] address the cycle-skipping issue of the traditional FWI with a source-receiver extended FWI for better velocity model reconstruction. All of these methods are designed to perform an accurate velocity model building.

In this study, we are interested in accurate 4D velocity change estimation and therefore we address similar issues with a different approach. Our goal is to provide a framework that can handle complicated 4D changes without resorting to manual interventions in either FWI or IDWT. We therefore propose a Dual Domain Wavefield Inversion (DDWI), where we perform FWI runs but add an image domain constraint. To calculate the joint gradient we introduce a weight that defines the amount of contribution from each domain. We use a sequential approach, where we first perform FWI runs on the baseline data to estimate a baseline model, which we then use as an input to DDWI. In this way, what we do is essentially to use IDWT to compute a mask, which is then applied “softly” during the inversion. By “softly” we mean that depending on the weights chosen on the two terms in the objective function the influence of the image and FWI constraints can be adjusted, thus adjusting how far the final result can stray from the locations of the mask.

The paper is organized as follows. We first recall the theory of IDWT and explain how we calculate the image domain mask. We then recall the conventional FWI and introduce the theory of DDWI in this framework. In this study, we compute the joint gradient with equal contributions from the data and image domains. We follow this with a series of examples, in which we validate the robustness of the method compared to typical 4D FWI schemes and IDWT both alone and followed by FWI. We conclude with a discussion on the current limitations and potential future extensions of this approach.

3.3 Image Domain Wavefield Tomography (IDWT)

The aim of IDWT is to use short-offset data to exploit primarily kinematic changes rather than amplitude changes. To focus on these kinematic changes, IDWT uses an objective function that compares the migrated baseline and monitor images [174]. More precisely, we find the model by solving the following minimization problem

$$\hat{m} = \underset{m}{\operatorname{argmin}} \left\{ \sum_{x_s} ||I_1(m; x, z, x_s) - I_0(m_0; x, z, x_s)||_2^2 \right\}, \tag{3.1}$$

where I_1 is the migrated image of the monitor data with the monitor velocity model, I_0 is the migrated image of the baseline data with the baseline velocity model m_0, x_s is the shot position, and m is the model (either velocity or squared slowness). By trying to minimize equation 3.1 two problems arise: cycle skipping when the reflectors are shifted by more than half of a wavelength resulting in an insufficient minimization of the objective function and physical movement of reflectors when amplitude changes are also present in the migrated images. To mitigate these issues, in IDWT we assume that one image is a warped version of the other and make use of a warping function

[53] to align the baseline and monitor images. We consider only vertical shifts since horizontal shifts are usually less significant. With these assumptions, the monitor image can be expressed as a vertically warped version of the baseline image. We thus define the warping function w [53] as the solution to the following minimization problem

$$\hat{w}(m|m_0; x, z) = \underset{w}{\operatorname{argmin}} \left\{ \sum_{x_s} ||I_1(m; x, z, x_s) - I_0(m_0; x, z + w(x, z), x_s)||_2^2 \right\}. \quad (3.2)$$

The model $\hat{m}$ from equation 3.1 can then alternatively be obtained as

$$\hat{m} = \underset{m}{\operatorname{argmin}} ||\hat{w}(m|m_0; x, z)||_2^2, \quad (3.3)$$

which replaces equation 3.1 as our objective function.

Choosing the right norm for a minimization problem is important. Full wavefield problems are commonly formulated as an L_2 norm. Other norms have also been considered in the literature such as the L_1 norm [146] [28] [20], the Huber criterion [50], and the hybrid L_1/L_2 [22]. Here, we choose the L_2 norm for the minimization of the warping function in equation 3.3. The least squares norm is the correct norm for optimization problems in which the true data residuals (those which would remain even if the true model was known - i.e. the data error) form a Gaussian distribution. According to the Central Limit Theorem, aggregation of independently sampled errors will be Gaussian irrespective of the underlying distributions being sampled from. Migration aggregates the noisy data over shots and offsets, and therefore the resulting error distribution is quite likely to be Gaussian. There is no obvious compelling reason to expect it to be any other distribution.

To calculate the gradient G of the IDWT objective function in equation 3.3 we use the adjoint state method (e.g. [109]) as was first done for the IDWT case in [174]. We summarize Yang's derivations here. The gradient can be expressed as an inner product of wavefields

$$G_{\mathrm{IDWT}}(x,z) = -\sum_{x_s} \underbrace{\int_{t=0}^{T} (\frac{\partial^2 \lambda_s(x,z,t,x_s)}{\partial t^2} u_s(x,z,t,x_s) + \frac{\partial^2 \lambda_r(x,z,t,x_s)}{\partial t^2} u_r(x,z,t,x_s))dt}_{G}, \tag{3.4}$$

where G is the single source gradient, $u_s(x,z,t,x_s)$ and $u_r(x,z,t,x_s)$ are the modeled source and backpropagated receiver wavefields, and $\lambda_s(x,z,t,x_s)$ and $\lambda_r(x,z,t,x_s)$ are the adjoint wavefields. The adjoint wavefields satisfy the wave equation

$$m\frac{\partial^2 \lambda(x,z,t)}{\partial t^2} - \Delta\lambda(x,z,t) = d, \tag{3.5}$$

where d are the adjoint sources

$$d_s(x,z,t,x_s) = \alpha(x,z,x_s)u_r(x,z,t,x_s), \tag{3.6}$$

and

$$d_r(x,z,t,x_s) = \alpha(x,z,x_s)u_s(x,z,t,x_s), \tag{3.7}$$

where

$$\alpha(x,z,x_s) = \frac{w(x,z,x_s)\frac{\partial I_0(x,z+w(x,z,x_s),x_s)}{\partial z}}{\frac{\partial I_0(x,z+w(x,z,x_s),x_s)}{\partial z}^2 - \frac{\partial^2 I_0(x,z+w(x,z,x_s),x_s)}{\partial z^2}(I_1(x,z,x_s) - I_0(x,z+w(x,z,x_s),x_s))}. \tag{3.8}$$

Here we take this one step further and use the recovered w to construct a mask to isolate the 4D change. We multiply the 4D gradient by a mask derived from image shifts, thus restricting model updates to regions in which the two images are misaligned. Following the ideas of [122] and [170] about the relationship between time shifts and velocity changes, we relate the warping function to a fractional depth strain $\phi^z(x, z_j)$ as

$$w(x, z_j) = \Delta z \sum_{i=1}^{N_Z} \phi^z(x, z_j), \tag{3.9}$$

where z_j is the discretized depth, Δz is the depth discretization, and $\phi^z(x, z_j)$ is the unitless fractional depth strain. Inserting equation 3.9 into equation 3.2 we compute the fractional depth strain by solving

$$\hat{\phi}^z(m|m_0; x, z) = \underset{\phi^z}{\operatorname{argmin}} ||I_1(m; x, z) - I_0(m; x, z + \Delta z \sum_{i=1}^{N_Z} \phi^z(x, z_j))||_2^2. \tag{3.10}$$

Assuming that the waves propagate purely vertically, the fractional depth strain at some depth in the subsurface is related to the fractional velocity strain (i.e. fractional velocity change) in the overburden. Thus, equation 3.10 is a proxy for velocity change, and therefore we use it to derive a mask for target-oriented time lapse inversion (Figure 3.1). At the 1$^{\text{st}}$ iteration the mask ψ is computed via

$$\psi_1(x_k, z_i) = \frac{1}{4N^2} \sum_{l=-N}^{N} \sum_{j=-N}^{N} \left|\phi^z(x_{k+l}, z_{i+j})\right|, \tag{3.11}$$

where N is the total number of discretized nodes in each direction. For the following n iterations, the mask is updated as

$$\psi_n(x_k, z_i) = \frac{1}{4N^2} \sum_{l=-N}^{N} \sum_{j=-N}^{N} \left|\frac{m_{1,n}(x_{k+l}, z_{i+j}) - m_0(x_{k+l}, z_{i+j})}{m_0(x_{k+l}, z_{i+j})}\right|, \tag{3.12}$$

where m_0 is the baseline velocity model and $m_{1,n}$ is the monitor velocity model updated at iteration n.

Having estimated the warping function, we then use a moving average smoothing via convolution with a boxcar function to smooth the absolute values in equations 3.11 and 3.12 and construct the final mask. In 1D this is expressed as

$$\theta_n = \sum_{m=-M}^{M} |\phi|_{n-m} b_m, \tag{3.13}$$

where θ_n is the value of the mask at sample n, $|\phi|_{n-m}$ is the absolute value of the model parameters at sample $n-m$, b_m is the value of the smoothing function at sample m, and $2M+1$ is the length of the smoothing function.

To find a local minimum in equation 3.3 we use a non linear conjugate gradient method [113] and we iteratively invert for the velocity $c = \frac{1}{\sqrt{m}}$. At each iteration the model is updated via

$$m_{n+1} = m_n + \alpha_n s_n, \tag{3.14}$$

where α is the step length and s is the search direction. To compute the search direction s, we use the Polak-Ribiere formula [111]. For the first iteration, the search direction is equal to the steepest descent direction, $s_1 = -G_1$. For the following n iterations this direction is updated via

$$s_n = -G_n + \beta_n s_{n-1}, \tag{3.15}$$

where

$$\beta_n = \frac{G_n^T(G_n - G_{n-1})}{G_{n-1}^T G_{n-1}}, \tag{3.16}$$

where G_n is the gradient at the n^{th} iteration. See [108], [161], and [114] for the step length computation.

Thus far we have followed the same workflow that [174] proposed with the addition of the 4D mask. Their workflow is based on pre-stack data, and therefore the vertical shifts are computed individually for each of the partial images (from each shot). The partial vertical shifts are then added to evaluate the objective function. Here we propose a post-stack extension of this workflow, in which we first form the full image and then compute the vertical shifts. We modify their workflow to:

1. given the baseline model m_0, we create the baseline image I_0,
2. for each shot x_s migrate the monitor data with the model m_0 and get the monitor image $I_1(x, z, x_s)$,
3. sum over shots and get the full migration image $I_1(x, z)$,
4. use image warping to compute the vertical shifts $w(x, z)$ between the images in 1 and 3,
5. evaluate the cost function (equation 3.3),
6. compute the adjoint source $\alpha(x, z)$,
7. using the post-stack adjoint source $\alpha(x, z)$ compute the adjoint wavefields λ_s, λ_r and the partial gradient $G(x, z, x_s)$ for each shot x_s,
8. sum over the partial gradients to get $G_{IDWT}(x, z)$,
9. calculate the search direction and step length to update the model,
10. remigrate the model with the new update and go to step 2.

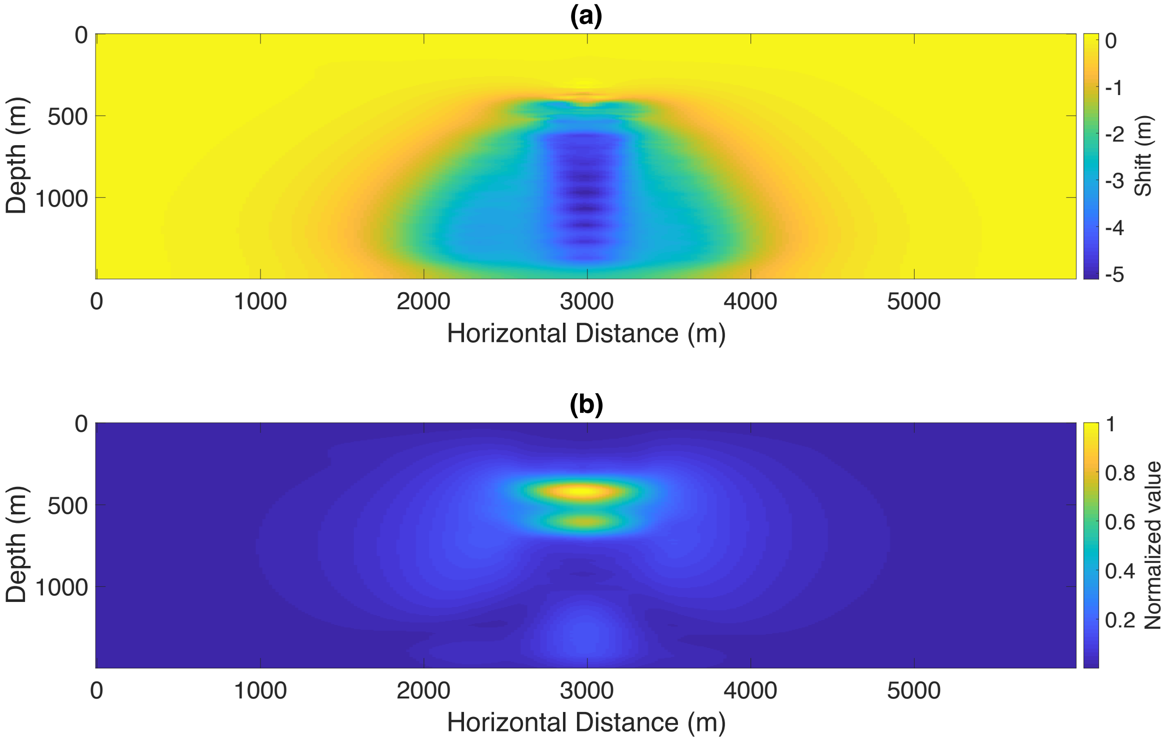

Figure 3.1: (a)The warping function from Example 1 calculated for one shot at $x_s =$ 3000 m. (b) The mask computed from the warping function. The mask is updated at each iteration, similar to the warping function.

3.4 Sequential Full Waveform Inversion (SFWI)

The most commonly used objective function in FWI is a least squares measure, and the recovered model is found by solving the following minimization problem:

$$\hat{m} = \underset{m}{\operatorname{argmin}} \left\{ \sum_{x_s} ||F(m|m_{init}; x_s) - d(x_s)||_2^2 \right\}, \tag{3.17}$$

where m is the model, in this case we invert for velocity $c = \frac{1}{\sqrt{m}}$, d is the observed data, F is the forward modeling operator, and x_s is the shot position. Usually a smooth initial model (m_{init}) is used to create the predicted data, and the model is updated at every iteration. To calculate the gradient of the objective function the adjoint state method is used [109]. Therefore, the gradient is simply a cross-correlation between the incident and the adjoint wavefields,

$$G_{\mathrm{FWI}}(x, z) = -\sum_{x_s} \int_{t=0}^{T} \left[\frac{\partial^2 \lambda_0(x, z, t, x_s)}{\partial t^2} u_0(x, z, t, x_s) \right] dt, \tag{3.18}$$

where the adjoint wavefield is the residual wavefield backpropagated from the receivers. Similar to lDWT, we update the model using a non linear conjugate gradient method [113].

We use the Sequential FWI (SFWI) 4D extension of FWI. In this approach, we first perform FWI on the baseline model

$$\hat{m_0} = \underset{m_0}{\operatorname{argmin}} \left\{ \sum_{x_s} ||F(m_0|m_{init}; x_s) - d_0(x_s)||_2^2 \right\}, \tag{3.19}$$

given a smooth initial model m_{init} to recover a baseline velocity model. We then use the recovered m_0 as an initial model for the monitor model inversion,

$$\hat{m_1} = \underset{m_1}{\operatorname{argmin}} \left\{ \sum_{x_s} ||F(m_1|m_0; x_s) - d_1(x_s)||_2^2 \right\}. \tag{3.20}$$

The final 4D signal is computed by subtracting the two updates

$$dV = \frac{1}{\sqrt{\hat{m_1}}} - \frac{1}{\sqrt{\hat{m_0}}}. \tag{3.21}$$

3.5 Dual Domain Waveform Inversion (DDWI)

In DDWI we combine diving wave information from FWI and reflection wave information from the migrated images obtained with IDWT via a joint objective function. As mentioned in the Introduction of this Chapter, several methods have been proposed in the literature for incorporation of both reflection and diving wave information into the inverse problem [14] [183] [60].

Our joint method is based on sequential 4D FWI. Therefore, we first perform FWI runs on the baseline model to get the starting model for DDWI (Figure 3.2). Our modified misfit function has two components and the model to be recovered is

$$\begin{aligned} \hat{m_1}(x, z) = (\mu) \underset{m_1}{\operatorname{argmin}} \left\{ \sum_{x_s} ||F(m_1|m_0; x_s) - d_1(x_s)||_2^2 \right\} \\ + (1 - \mu) \underset{m_1}{\operatorname{argmin}} \left\{ \sum_{x_s} ||\hat{w}(m_1|m_0; x, z, x_s)||_2^2 \right\}, \end{aligned} \tag{3.22}$$

for the pre-stack version, while for the post-stack version, equation 3.22 becomes

$$\hat{m_1}(x,z) = (\mu)\underset{m_1}{\operatorname{argmin}}\left\{\sum_{x_s} ||F(m_1|m_0;x_s) - d_1(x_s)||_2^2\right\} \tag{3.23}$$

$$+ (1-\mu)\underset{m_1}{\operatorname{argmin}}\left\{||\hat{w}(m_1|m_0;x,z)||_2^2\right\},$$

where m_0 is the recovered baseline model, m_1 is the monitor model, d_1 is the monitor observed data, F is the forward modeling operator, w is the warping function (equation 3.2), and μ denotes the weighting parameter between the two components of the misfit function. The choice of the weight reflects how much contribution one wants from the data and image domain in the gradient and misfit functions. Compared to sequential FWI, our method is less sensitive to acquisition limitations and to cycle skipping issues because of the image domain constraint. Compared to IDWT, DDWI provides better reconstructed 4D changes without a significant increase in the already high computational cost of IDWT.

To calculate the gradient we use the adjoint state method [109]. This gradient can be expressed as a sum of cross correlated wavefields

$$G_{\text{DDWI}}(x,z) = (\mu)\tilde{G}_{\text{FWI}}(x,z) + (1-\mu)\tilde{G}_{\text{IDWT}}(x,z) \tag{3.24}$$

where $\tilde{G}_{\text{FWI}}$ and $\tilde{G}_{\text{IDWT}}$ are the normalized gradients from FWI (equation 3.18) and IDWT (equation 3.4) computed via

$$\tilde{G}_{\text{FWI}}(x,z) = \frac{N_X N_Z \psi(x,z) G_{\text{FWI}}(x,z)}{\sqrt{\sum_{j=1}^{N_X}\sum_{i=1}^{N_Z}\psi^2(x_j,z_i)G_{\text{FWI}}^2(x_j,z_i)}}, \tag{3.25}$$

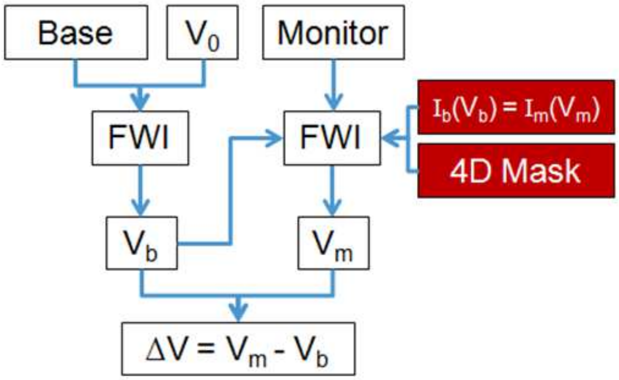

Figure 3.2: Flowchart for Dual Domain Waveform Inversion (DDWI).

and

$$\tilde{G}_{\mathrm{IDWT}}(x,z) = \frac{N_X N_Z \psi(x,z) G_{\mathrm{IDWT}}(x,z)}{\sqrt{\sum_{j=1}^{N_X} \sum_{i=1}^{N_Z} \psi^2(x_j,z_i) G^2_{\mathrm{IDWT}}(x_j,z_i)}}, \tag{3.26}$$

where ψ is the mask (equations 3.11 and 3.12), and N_X and N_Z are the total number of discretized nodes in the x and z directions respectively. In this particular case, we use equal weights for both gradients and therefore set $\mu = 0.5$. Similar to both FWI and IDWT, we use a non linear conjugate gradient method to update the model.

3.6 Numerical Examples

In this section, we compare the performance of the Dual Domain Waveform Inversion (DDWI) with IDWT, Sequential FWI, Double-Difference FWI [171], and IDWT followed by FWI [174] on different synthetic models. From simple to more complicated models, we demonstrate the improvements the method delivers. We use a finite difference method to solve the wave equation and we generate constant density acoustic data. Reverse Time Migration (RTM) [11] [92] is used to create the baseline and monitor migration images. We use the same time-domain solver to generate the data and solve the inverse problem. For the boundary conditions we use absorbing boundary conditions on all boundaries.

3.6.1 Example 1: Smooth velocity perturbation spanning a layer

We begin with a layered model in which we introduce 200 m/s or 10% smooth velocity perturbation. The anomaly is shallow and spans a layer (Figure 3.3), resulting in a combination of time shifts and amplitude changes. We discretize the model using 300 nodes in both directions, with node spacing 20 m in the x-direction and 5 m in the z-direction. We use the same 15 Hz Ricker wavelet for both baseline and monitor surveys. We use a total of 9 shots to generate the synthetic seismograms, which we record with 148 receivers placed at a depth of 5 m and an interval of 40 m. We first invert for the baseline model, and then use the update to perform three 4D inversions: SFWI, IDWT, and DDWI.

Figure 3.4 shows the 4D changes reconstructed from all three inversions. Because there are both phase and amplitude changes IDWT struggles somewhat to recover the change, the anomaly is smeared vertically, with a break at the location of the spanned layer. SFWI performs better than IDWT providing a more accurate anomaly reconstruction. Despite providing a better result than IDWT alone, there is still a lot of energy below the anomaly and it almost looks as though FWI is taking information from the reflections instead of the diving waves. DDWI produces better results than the other two methods. The 4D change is fully reconstructed in shape without visible artifacts.

One of the main points of our new objective function, is that it aligns the two migrated images. Figure 3.5 illustrates this by comparing the baseline and monitor migrated images at the beginning and the end of the DDWI scheme. We choose the shot location of $x_s = 3000$ m and $z_s = 5$ m for the display. In panels (a) and (b) we see that initially there is a big misalignment between the two images, especially in the reflectors below the anomaly. The difference between the two is shown in panel (c). On the contrary, after 25 iterations of the DDWI scheme, the migrated images are aligned, with the only difference being where the anomaly is located.

In any inversion scheme, it is important to have a quality control both in terms of model fit and data fit (Figure 3.6). In panel (a) we display 1D velocity profiles extracted from the true and recovered monitor velocity models at $x = 3000$ m, which goes through the top of the anomaly. The DDWI line is closer to the true, especially around the anomaly location, compared to the SFWI and IDWT lines. Panel (b) shows the shot gather at $x_s = 3000$ m, $z_s = 5$ m using the DDWI recovered velocity model. We show a single trace in panel (c) and focus on the shallow part where the 4D change is (panel d). We see that the data fit is good for all of the methods but DDWI clearly has the best match.

3.6.2 Example 2: Deeper smooth velocity perturbation

To complicate the model, we now move the velocity perturbation to deeper layers, so that the anomaly is deeper than the recorded diving waves penetrate (Figure 3.7). We do this to show that in our hybrid approach, we are able to recover an accurate model without first determining which parts of the data illuminate the regions of interest. The velocity perturbation is the same as in the previous example (200 m/s or 10% velocity change), and for consistency we keep the same acquisition setup and simulation grids.

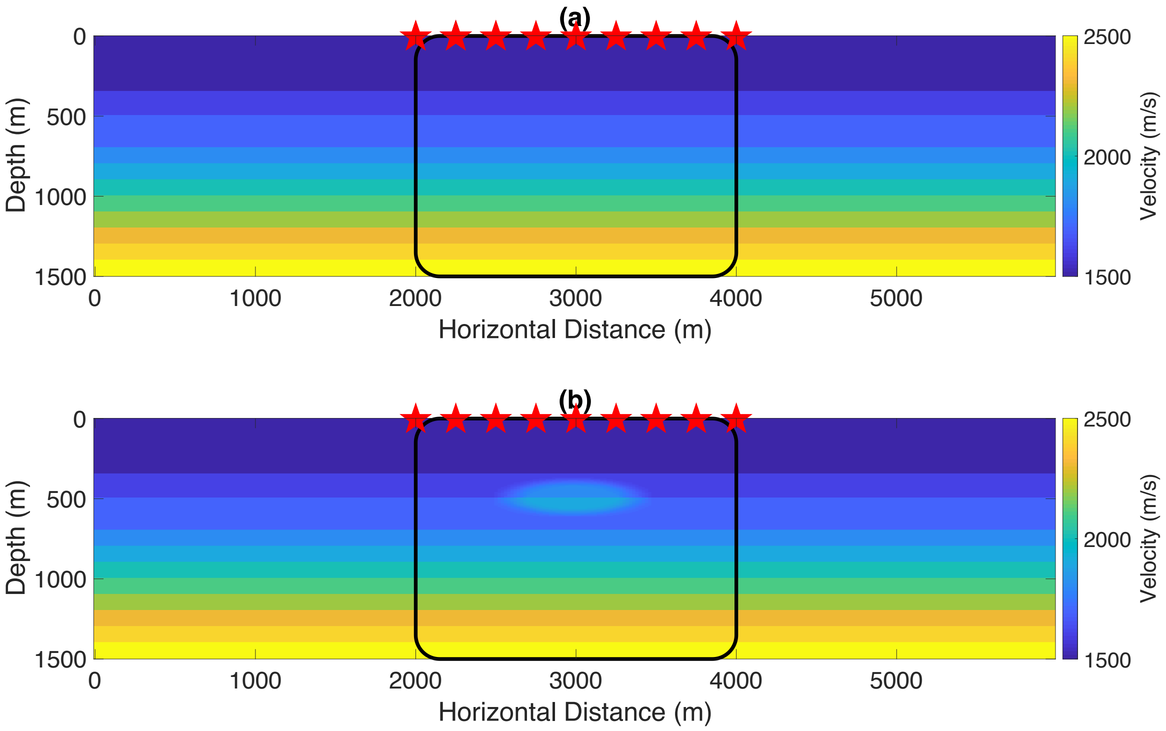

Figure 3.3: (a)The true baseline model. The red stars represent the positions of the sources. (b)The true monitor model. In both figures the black box shows the zoomed in area where the time-lapse changes are.

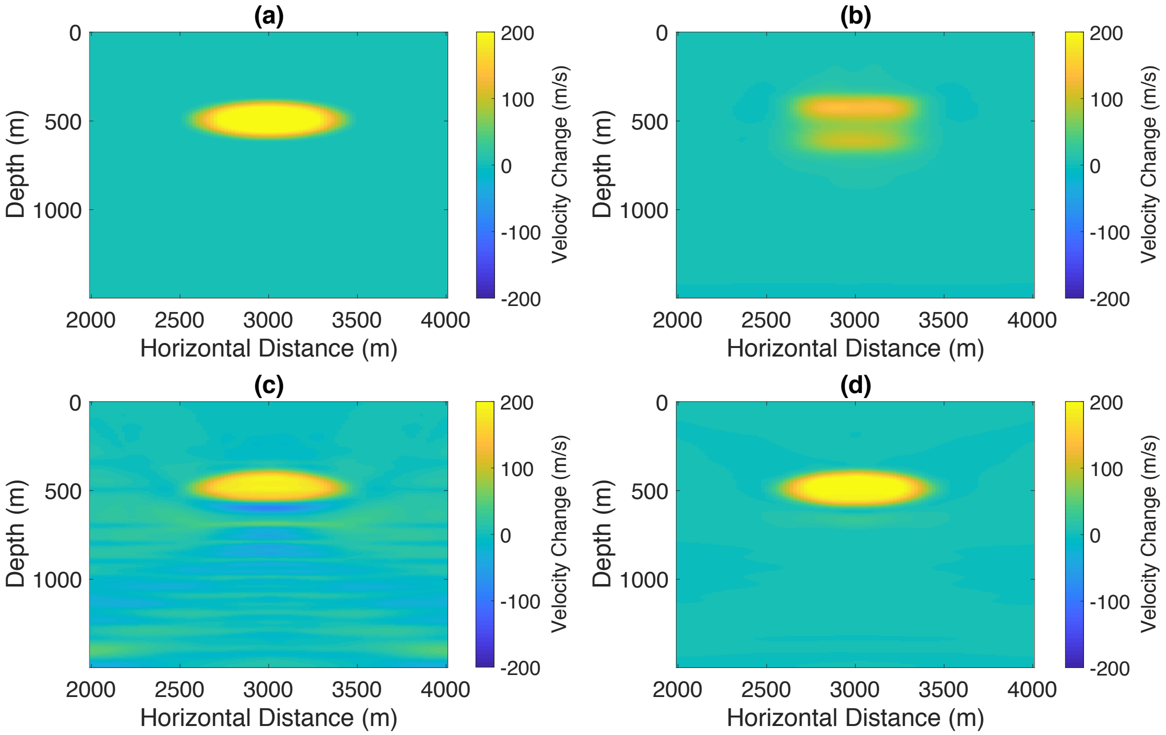

Figure 3.4: (a) True time-lapse velocity change. (b) Time-lapse velocity change recovered from IDWT. (c) Time-lapse velocity change recovered from SFWI. (d) Time-lapse velocity change recovered from DDWI.

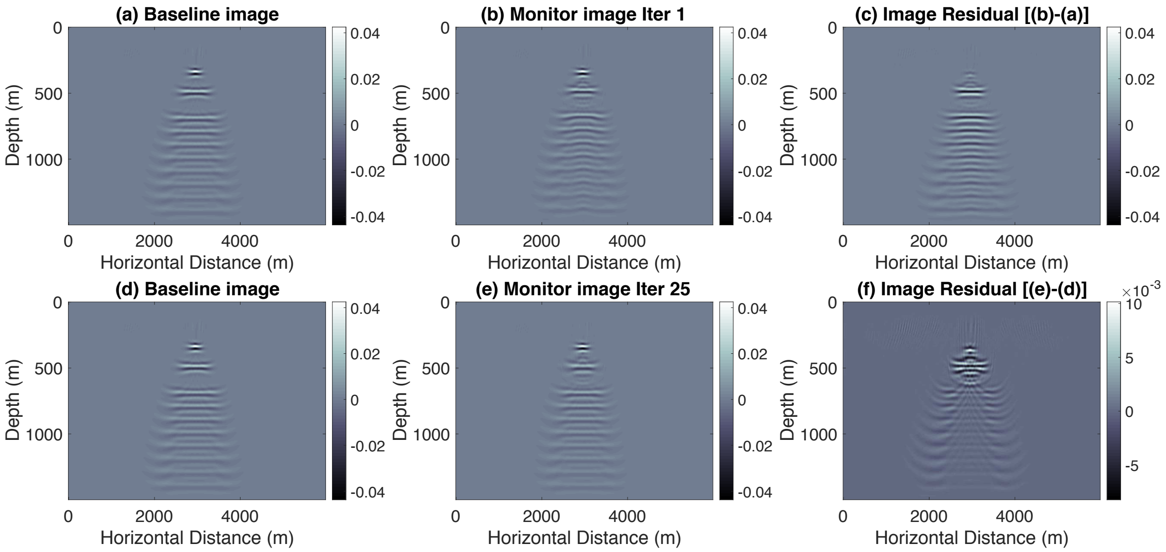

Figure 3.5: For the shot located at $x_s = 3000$ m in Example 1: (a) The migrated baseline image, (b) the migrated monitor image at iteration 1, (c) the difference between the monitor image **b** and the baseline image **a** (d) The migrated baseline image same as in panel **a**, (e) the migrated monitor image after 25 iterations, (f) the difference between the monitor image **e** and the baseline image. Note the difference in scale between (c) and (f).

In this example we also compare the performance of DDWI with Double Difference FWI [167] [171] [180] and IDWT followed by FWI [174]. For all inversions we perform 25 iterations. We use this number of iterations because the reduction of the cost function in all schemes reach a plateau after approximately 20 iterations. Figure 3.8 shows the results obtained from all inversion schemes. IDWT performs very well. The resolved anomaly is localized to the area of the true change and appears bounded by the reflectors above and below it. As expected, SFWI delivers a poorly constrained anomaly, completely failing to update the lower part of the anomaly. Double Difference FWI is unable to constrain the anomaly fully both in shape and amplitude, with a performance similar to SFWI. However, this is expected due to the nature of the anomaly; it is placed deeper than the diving waves penetrate and therefore there is little information to contribute to the FWI reconstruction. When we apply IDWT

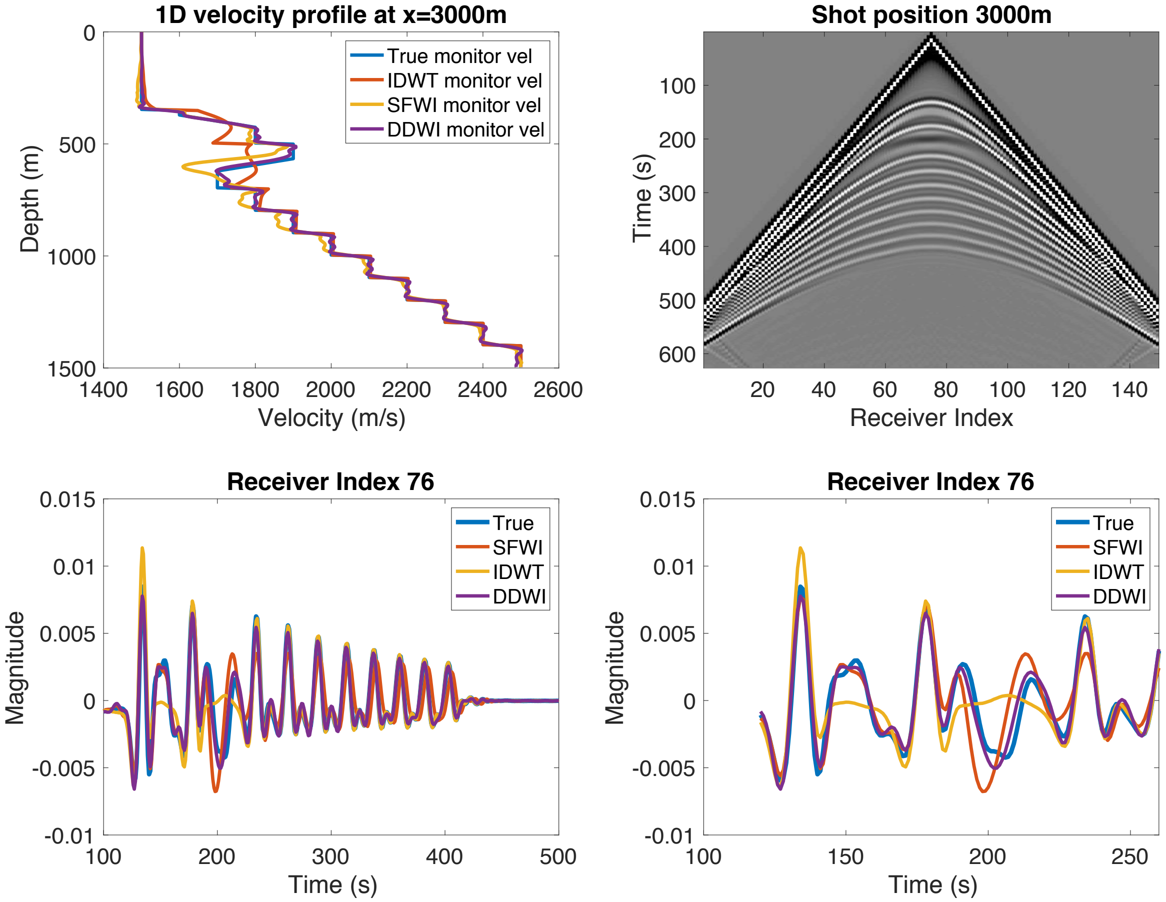

Figure 3.6: Top left: 1D velocity profile of the true and recovered monitor models in Example 1, at $x_s = 3000$ m. Top right: Shot gather at $x_s = 3000$ m, $z_s = 5$ m using the recovered DDWI velocity model. Bottom left: 1D plot of the shot gather in the top right panel; the receiver index is 76 and we plot between 100-500 seconds to exclude the direct wave. Bottom right: Zoom-in at the location of the anomaly (120-260 seconds).

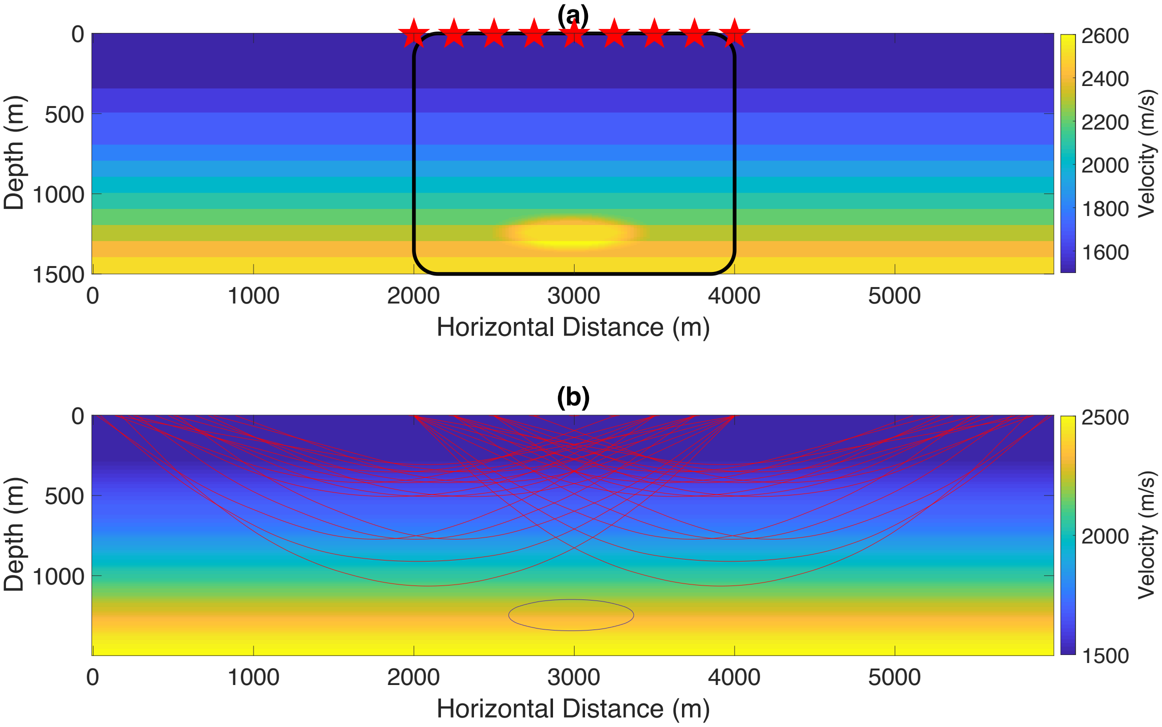

Figure 3.7: (a) The true monitor model for the deeper smooth velocity perturbation. The black rectangle shows the area where the time lapse change is, and all the subsequent figures will show only the part in the area of interest. (b) Diving wave penetration analysis.

followed by FWI the anomaly is well reconstructed both in shape and magnitude, indicating the advantages of using image domain information. However, a few layering artifacts remain from the FWI above and below the anomaly, and we see a pattern from the IDWT recovery within the anomaly. DDWI, on the other hand, delivers an excellent reconstructed anomaly with fewer artifacts than IDWT followed by FWI. This is likely due to the implicit incorporation of what amounts to a mask in the inversion. More specifically, when the objective function includes both IDWT and FWI components the IDWT component acts as a weight encouraging the updates to remain in the vicinity of the changes estimated by IDWT. In DDWI both components are present throughout the process making the mask a stronger constraint in DDWI than it is in IDWT followed by FWI.To quantitatively compare the two strategies, we computed the difference between their retrieved 4D signal and the true anomaly (Figure 3.9). We see that DDWI provides a more accurate result with an error of 10 m/s (about 5% of the true change calculated via $\frac{TrueChange-Recovered}{TrueChange} \cdot 100\%$) compared to IDWT followed by FWI with an error of approximately 60 m/s (approximately 30% of the true change). This shows that the joint scheme mitigates the individual limitations of IDWT and FWI.

3.6.3 Example 3: Inaccurate starting model

For all of the previous models, we assumed that the initial velocity model was accurate. In practice though, this assumption does not usually hold. We know that FWI, in contrast with IDWT is very sensitive to the velocity of the starting model [161] [174]. In a sequential time-lapse approach this will have an effect on the final estimate of the velocity change. We anticipate that this sensitivity to initial model will be larger with FWI than IDWT. To show this, we use the model from Example 2 with the smooth

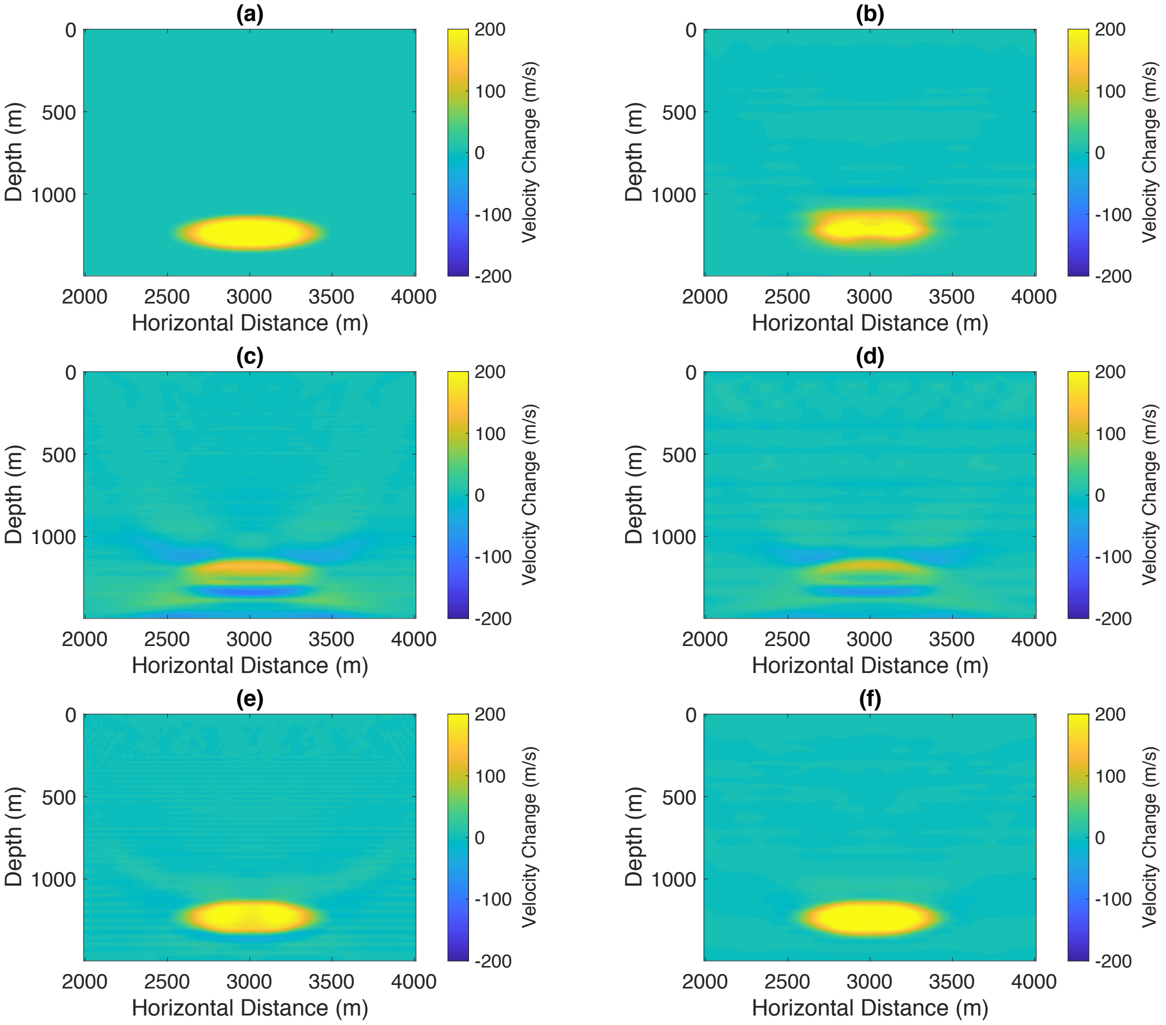

Figure 3.8: (a) True time-lapse velocity change for the deeper smooth velocity perturbation example. (b) Time-lapse velocity change recovered from IDWT. (c) Time-lapse velocity change recovered from SFWI. (d) Time-lapse velocity change recovered from Double Difference FWI. (e) Time-lapse velocity change recovered from IDWT followed by FWI. (f) Time-lapse velocity change recovered from DDWI.

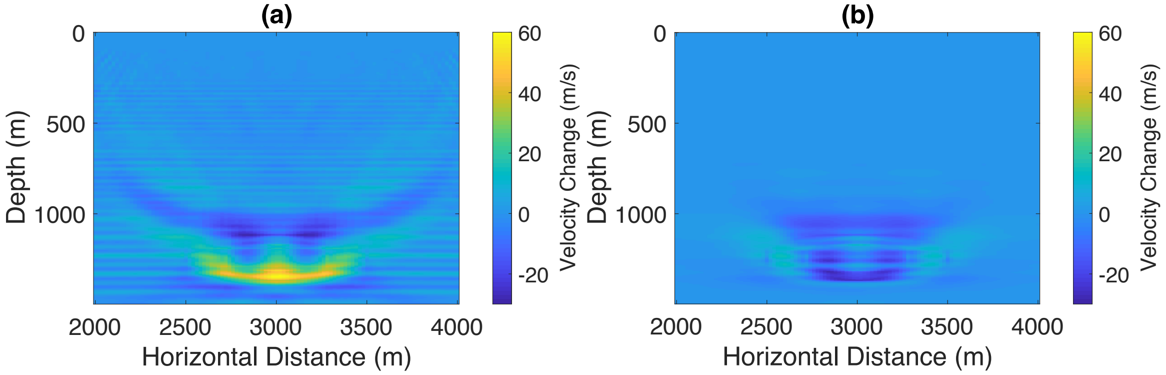

Figure 3.9: (a) Difference between the anomaly recovered with IDWT followed by FWI and the true anomaly. (b) Difference between the anomaly recovered with DDWI and the true anomaly. We see that DDWI delivers more accurate results.

velocity perturbation spanning a layer (Figure 3.3). The velocity anomaly is shallow enough for the diving waves to penetrate it. We use a smooth starting model, in which we introduce a 5% amplitude perturbation in slowness and perform the baseline model inversion (Figure 3.10). We see that the performance of FWI deteriorates with the introduction of a lot of artifacts in the shallow part, and layered artifacts in the deeper parts of the model. A recovered baseline model like this one will never be accepted in practise, and further processing and quality control steps will be needed. However, in this example we want to show, that even if the baseline model is so bad that it would never be acceptable in practise, IDWT still recovers a good 4D model, whereas the FWI fails. This is because the kinematic differences (on which IDWT relies) are less damaged by an inaccurate baseline model than are amplitude differences. As in all previous examples, we use the recovered baseline model as a starting model for the 4D inversions.

Figure 3.11 shows the results of all of the 4D inversion schemes we compare. As we expected, IDWT is almost completely unaffected by the velocity errors, yielding the same final result as the case where the starting model was accurate (Figure 3.4). Our result is in-line with the results of [174], where they tested the robustness of

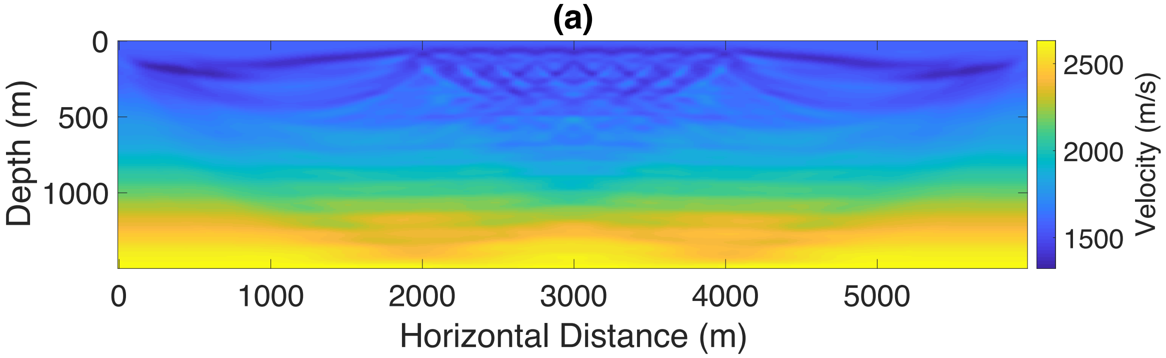

Figure 3.10: The inverted baseline model staring from a smooth initial model with 5% perturbation.

IDWT over baseline velocity errors (Figures 6 and 7 in their paper). The amplitude discrepancy we introduced in the starting model is sufficiently large to cause cycle skipping in the low frequency components of the data, and this is something we see clearly with the SFWI result. In many cases, the result of SFWI could be improved by the use of a mask (typically derived from data domain constraints). This type of mask will work as a model-space geological prior to the inverse problem without adding any sensitivity to the anomaly. Image domain constraints on the other hand, fundamentally alter the sensitivities of the inverse problem by adding constraints to the time-lapse change with a penalty sensitive to the reflecting energy in the input data. Therefore, the sensitivity to the reflecting energy is enhanced, making IDWT a powerful constraint in any time-lapse inversion scheme (hence we propose DDWI). Here, the performance of DDWI is not perfect. Artifacts in the 4D model are well suppressed when compared to SFWI, and the 4D change is in the right location. However the shape of the anomaly is not recovered, and this is something requiring further study and improvement. One of the main advantages of the DDWI and fundamental difference with IDWT followed by FWI [174], is that we never relax the image domain constraint. Even though IDWT can provide a good starting model for FWI, there are scenarios in which FWI still ends up in the wrong local minimum, as for instance here where we see a poor performance of IDWT followed by FWI. This is because FWI is walking away from the solution given by IDWT. Any FWI scheme will deviate from its starting model if there is a lower residual to be found. If the starting model is good, the inversion is typically regularized to stay close to it (i.e. zero-order Tikhonov). If the starting model is not good (as in this example), an approach such as FWI being anchored to IDWT will provide more meaningful results.

3.6.4 Example 4: Subsalt smooth velocity perturbation

To mimic a more realistic scenario we create a salt model. To do this, we take the model from Example 3 and add the salt bodies from the Pluto velocity model on top of the anomaly (Figure 3.12). The salt has a velocity of 3500 m/s, creating a high contrast with the background layers' velocity ($1600 - 1800$ m/s). In this example, we modify the acquisition setup to provide larger offsets so we can image better beneath the salt. We use a total of 11 sources and 148 receivers equally distributed over a distance of 5 km on the surface for both the baseline and monitor surveys. To invert for the baseline model, we assume that the salt geometry is known and we therefore create a smooth background that includes the salt body. The background baseline model is well recovered (Figure 3.12); as above we use this as the starting model of the 4D inversions.

Thus far we have used only the pre-stack version of IDWT and DDWI (equation 3.22). However, in this example, having such a high contrast body present in the model, leads to weaker amplitude reflectors in the migrated image. Therefore, in this example we use the post-stack version of IDWT and DDWI (equation 3.23), where instead of computing the warping function separately for each shot, we create the full migration image first. In addition, we use an automatic gain control (AGC) function

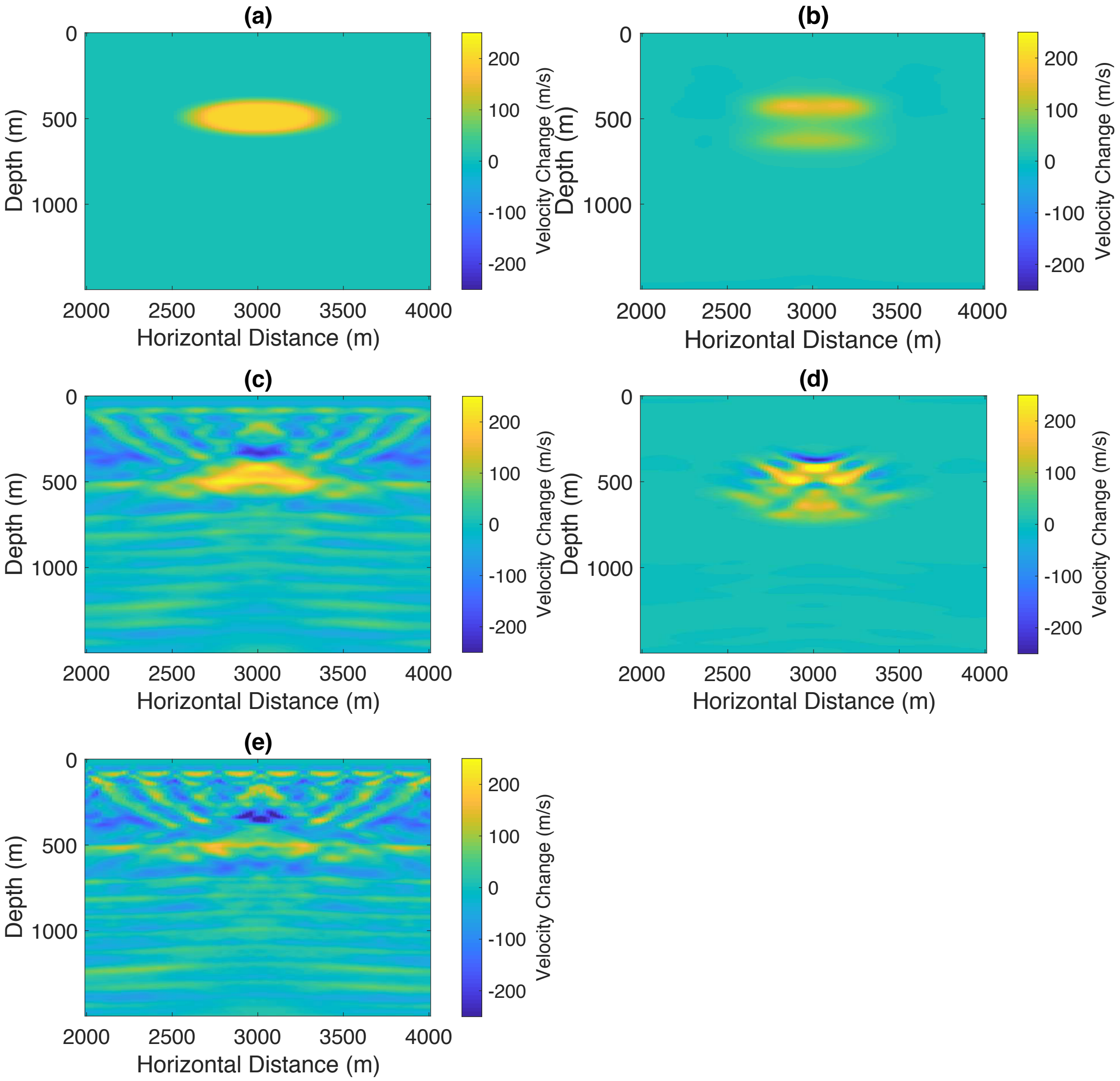

Figure 3.11: (a) The true time-lapse change for Example 3 (b) The time-lapse change recovered with IDWT. (c) The time-lapse change recovered with SFWI. (d) The time-lapse change recovered with DDWI. (e) The time-lapse change recovered from IDWT followed by FWI.

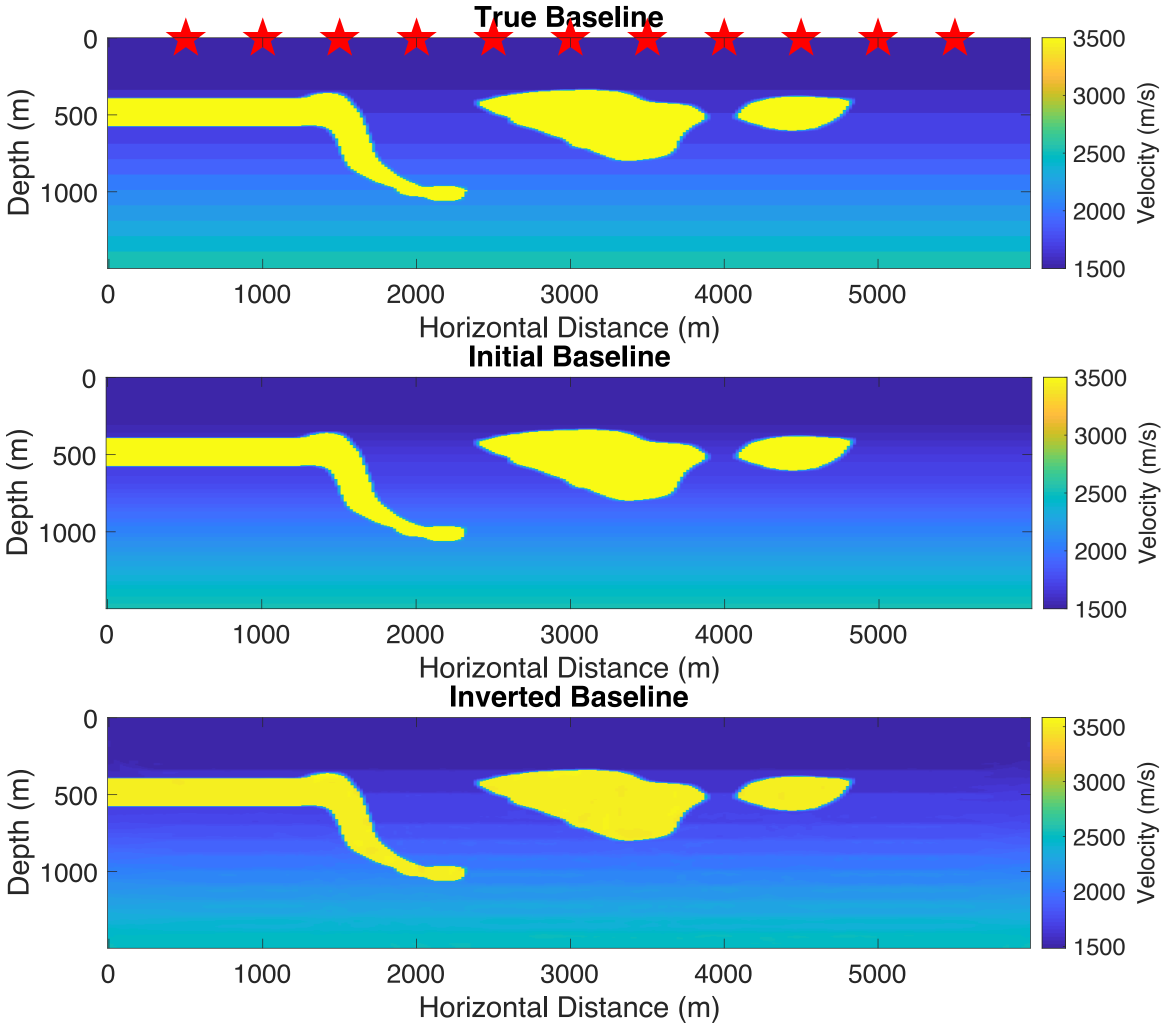

Figure 3.12: (a) The true subsalt baseline model. We use a total of 11 shots shown with the red stars. (b) The smooth initial velocity model with the known salt geometry that was used for the baseline FWI. (c) The recovered baseline model after 50 iterations.

to enhance the amplitude of the reflectors beneath the salt so that we have a more accurate estimation of the vertical shifts. Therefore, the updated misfit function from equation 3.2 is

$$\hat{w}(m|m_0;x,z) = \underset{w}{\operatorname{argmin}} ||AGC(I_1(m;x,z)) - AGC(I_0(m_0;x,z+w(x,z)))||_2^2. \quad (3.27)$$

We then use $AGC(I_1(m;x,z))$ and $AGC(I_0(m_0;x,z))$ to compute the updated adjoint source α and the gradient G_{IDWT}. To do this we simply replace the standard images with the AGC images in the system of equations A-11 from Appendix A of [174]. If we now take the gradient of equation A.1

$$\frac{\partial w}{\partial m} = 2\left(AGC(I_1(m)) - AGC(I_0(m))\right)\frac{\partial AGC(I_1(m))}{\partial m}, \quad (3.28)$$

and apply the chain rule to the last term, we get

$$\frac{\partial AGC(I_1(m))}{\partial m} = \frac{\partial AGC(I_1(m))}{\partial I_1(m)}\frac{\partial I_1(m)}{\partial m}. \quad (3.29)$$

We expect that $\frac{\partial I_1}{\partial m}$ varies on a shorter wavelength scale than $\frac{\partial AGC(I_1(m))}{\partial I_1(m)}$, when we choose an AGC window that spans several events in the image. Furthermore, we expect that the AGC value between the two images will be similar, and can be approximated by a constant factor that varies slowly across the image (and which is in any case close to constant). The constant could have been absorbed into the scaling factor for the image space constraint in the FWI objective function and we neglect the further contributions of the AGC in the gradient computation. A more detailed explanation can be found in Appendix A of this thesis.

Figure 3.13 shows the 4D recovery from the three algorithms. We see that SFWI after 25 iterations performs better than before for this anomaly without the salt, because we use larger offsets and therefore diving waves can better penetrate the anomaly. However, the shape is not perfectly recovered and there are still a few artifacts present as well as the signature of the salt bodies. When we run the SFWI for 50 iterations, the recovered anomaly is improved in shape, however artifacts are still present. The recovery from IDWT is good, but worse if it is compared with the same anomaly without the salt. This is most likely due to the big internal multiples we get from the salt body. The extra processing steps to enhance the images do not correct the internal multiple energy from the salt body, that maps to below the salt in the migrated image. More sophisticated migration algorithms capable of avoiding artifacts from internal multiples, such as Marchenko imaging [166] would benefit DDWI in this situation. [116] use a Simultaneous Joint Migration Inversion that utilizes internal multiple energy to enhance the illumination of deeper targets. Image space constraints, as are used in this study, are founded on the geological assumption that reflectors do not move (significantly) in space over time. Combining the two techniques may help in scenarios where illumination of deeper targets is poor. This remains a topic of future study in which we expect our algorithm to perform similarly to IDWT followed by FWI or other FWI algorithms when this coherent noise is present. DDWI performs better than the other two methods. It delivers a better constrained anomaly, which is located in the right place with a more accurate shape.

3.7 Discussion

Thus far, we have only compared the results of three different time-lapse inversion strategies. It is important to also consider computational cost. For one iteration of the standard FWI we need three wavefield simulations, this means that in 25 iterations

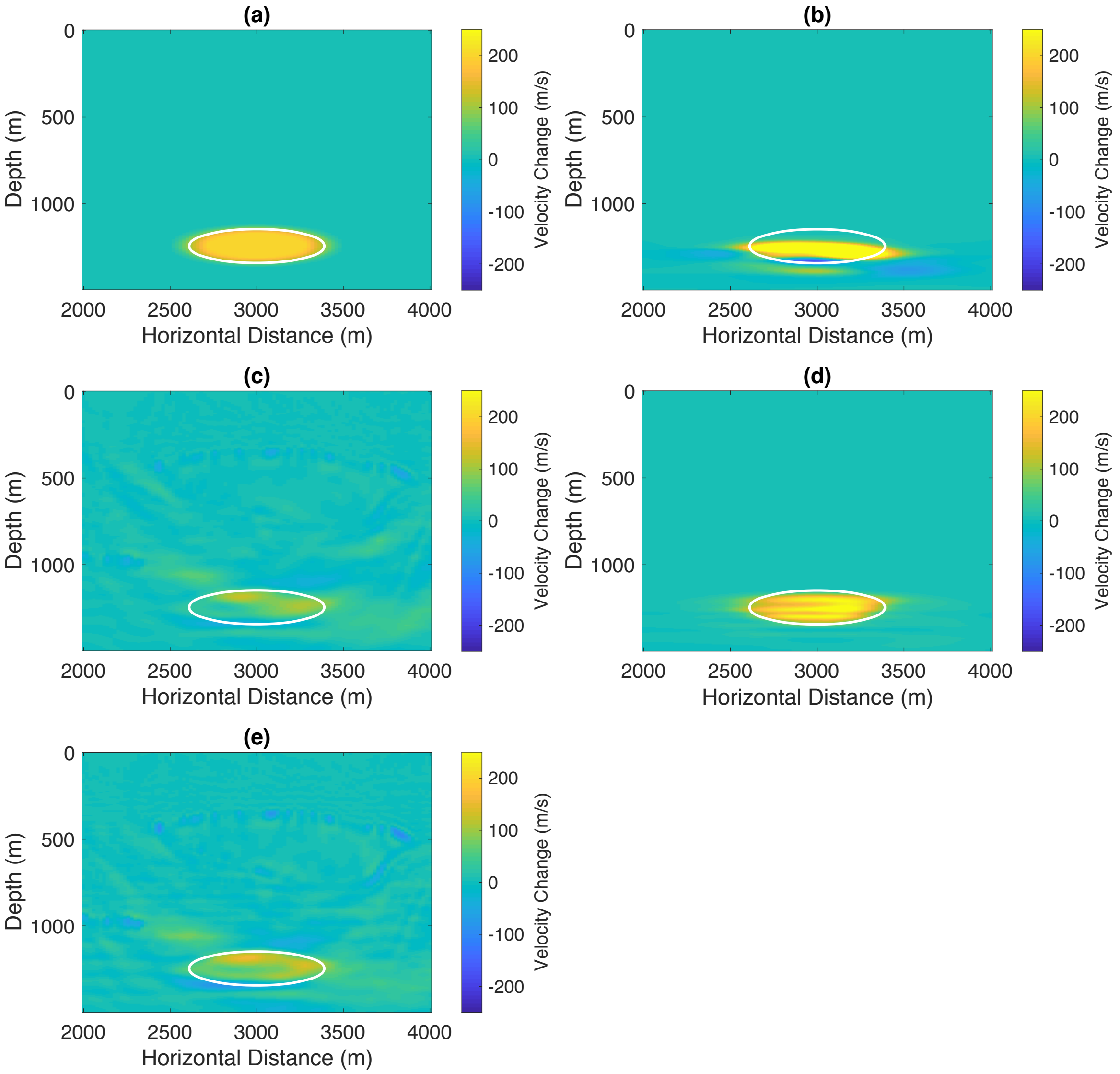

Figure 3.13: (a) The true time-lapse change for the subsalt anomaly. The white circle highlights the area of the true change. (b) The time-lapse change recovered by IDWT after 25 iterations. (c) The time-lapse change recovered from SFWI after 25 iterations. (d) The time-lapse change recovered from DDWI after 25 iterations. (e) The time-lapse change recovered from SFWI after 50 iterations.

need to perform 75 wavefield simulations. For one iteration of IDWT we need to perform six wavefield simulations, this means that in 25 iterations we need to perform a total of 150 wavefield simulations. In DDWI, we need seven wavefield simulations in one iteration, and thus 175 in 25 iterations. When a good starting model is available, performing FWI after IDWT, improves the final image over either method alone [174]. But doing the two together saves one wavefield solve per iteration. When the starting velocity model is far from the truth, FWI moves away from the correct solution and suffers from cycle skipping issues.

A fundamental advantage of our proposed methodology, is that we do not relax the image domain constraint in our objective function, and hence FWI cannot deviate from our geological prior. This is highlighted in the example of the inaccurate starting model, where IDWT followed by FWI [174] is not able to deliver a meaningful result and performs worse than DDWI. A second important contribution of DDWI, is the use of a mask that is data driven. In contrast with other available masks for localized 4D updates, we do not need to pre-construct the mask prior the inversion procedure. Instead, the mask is automatically derived from the migrated images, hence the data themselves. In this way, we avoid both the bias of a precomputed mask as well as localizing updates in areas where they may not actually occur. Additionally, using image domain constraints, that are predominantly function of kinematic changes, enhances the sensitivity of the inverse problem to reflections. This is an advantage over the data domain masks that mainly work as a geological prior to the inversion without adding any sensitivity to the anomaly. Therefore, this contribution, could be considered as an alternative way of deriving a mask for a targeted inversion.

A seismic reflection can be shifted as a result of a reflector shift or a velocity change. In IDWT we assume that the reflectors shifts are not as significant as the shifts due to velocity changes and therefore we ignore any actual structural changes. This assumption is valid in many examples, however in cases where structural shifts are present and significant, such as reservoir compaction, this assumption can lead to inaccurate results. One way to compensate for this would be to relax the image constraint in DDWI at later iterations and let FWI correct for depth shifts. This could be easily achieved by changing the weights in the calculation of the joint gradient (equation 3.24).

An important component of a successful seismic time-lapse analysis is the survey repeatability. In most cases, 4D FWI approaches require extra pre-processing steps to compensate for the dislocations of shots. [174] provide a detailed study about the performance of IDWT when source positioning errors are present. More precisely, they test both random (their Figure 7) and systematic (their Figure 8) source perturbations also with errors in the baseline velocity model, and they conclude that the method is robust and able to deliver meaningful results. Since half of our objective function is based on IDWT, and we never relax the image domain component, we expect that DDWI will be similarly robust.

Our current warping function is both amplitude and phase shift dependent. This is why we need to perform image pre-processing steps to recover the 4D changes in the salt model. Alternatively, a phase-shift only warping function may mitigate some of the challenges of working with complicated velocity models with strong impedance contrasts. Additionally, for our subsalt example we have assumed that the the geometry and location of the salt bodies were perfectly known. We know that such an assumption is far from realistic. Future work could incorporate salt inversion strategies such as the one shown by [77].

Last but not least, for all of our numerical examples we use the same solver for the generation of the synthetic datasets and the inverse problem. Future work might investigate the performance of DDWI in a non-inverse crime scenario.

3.8 Conclusions

Imaging the low-wavelength component of the velocity model in deeper regions of the Earth is a significant challenge. Full waveform inversion relies on diving waves that lack penetration depth. Image domain tomography overcomes this challenge by relying on reflected waves. In this manuscript, we proposed to combine full waveform inversion with image domain tomography to setup time-lapse seismic inversion as a single minimization problem. Having systematically compared the performance of three methods (full waveform inversion, image domain tomography, and dual domain tomography) we made several key observations. For shallow velocity anomalies in geology without complications, all three methods perform similarly. When the anomaly is deeper, the image domain tomography lacks vertical resolution, while full waveform inversion has to rely on reflected energy which fails in realistic scenarios when the time-lapse velocity is smooth. Dual domain waveform inversion (combining these two methods in one objective function) is able to combine the strengths and cancel the weaknesses of both methods. Image domain tomography seems to be quite sensitive to amplitude changes in seismic images. This is particularly challenging in cases like subsalt imaging. We show that pre-processing the seismic images, before measuring the depth-shifts, overcomes this problem. Finally, we found it advantageous to combine both methods in a single minimization problem rather than applying image domain tomography and full waveform inversion sequentially because the continued application of the image constraint keeps FWI from straying into the wrong local minima.

3.9 Acknowledgments

This work took place during a summer internship at the Geoscience Research Centre of TOTAL E&P UK in Aberdeen. We are thankful to TOTAL E&P UK for giving permission to publish this research. We acknowledge all the researchers and staff from the Geoscience Research Centre, and in particular Constantin Gerea, Mohammad Shahraeeni, Angus Lomas and Nathan Amaral, for insightful discussions. We would like to thank Jeffrey Shragge, Partha Routh, Musa Maharramov and the two anonymous reviewers for the constructive comments that helped us improve our paper. We are also grateful for support at Memorial provided by Chevron and with grants from the Natural Sciences and Engineering Research Council of Canada Industrial Research Chair Program and InnovateNL and by the Hibernia Management and Development Corporation.

Chapter 4

Uncertainty quantification in 4D seismic imaging: a full-wavefield approach

4.1 Abstract

Time-lapse seismic monitoring using Full-Wavefield methods aims to accurately and robustly image rock and fluid changes within a reservoir. These changes are typically very small and localized. Quantifying the uncertainty related to those changes is crucial for decision making, but traditional methods that use pixel by pixel quantification with large models are computationally infeasible. We exploit the structure of the 4D seismic problem for much faster wavefield computations using a numerically exact local acoustic solver. This allows us to perform a Bayesian inversion using a Metropolis Hastings algorithm to sample our posterior distribution. We assume that the measurement noise is Gaussian, however, we make no assumptions about the posterior

distribution which could be any type, including multi–modal and non–Gaussian. We compute single and multiple degrees of freedom numerical examples and address the dimensionality problem using image compression techniques.

4.2 Introduction

4.2.1 Background Information and Related Work

Time–lapse (4D) seismic monitoring is to date the most commonly used technique for geophysical monitoring of a reservoir. 4D monitoring is achieved by acquiring and analyzing multiple seismic surveys at the same site over time [79]. The first survey acquired over a field is called the baseline survey. All subsequent surveys are called the monitor surveys. Full-Waveform Inversion (FWI) delivers high resolution images of the subsurface using prestack data [145] [161]. There are several extensions of FWI to the 4D case that aim to accurately image changes that are then used for further decisions (i.e. drilling and production in the context of hydrocarbon exploration). There are a variety of 4D schemes available including: Parallel FWI [110], Sequential FWI [6], Double Difference FWI [167] [182], Joint FWI [84] [173], cascaded schemes [129] or even schemes that couple FWI with image domain constraints [64]. All of these schemes try to solve for the successive linearized problem using local optimization techniques.

As with any inverse problem, 4D FWI depends on a mathematical model of the underlying physics. Any mathematical model carries uncertainty for various reasons. If we cannot quantify and control these uncertainties, then we cannot trust the outcomes of our model. Because FWI is a non–linear problem and prone to local minima, it is important to determine how likely particular solutions are. [173] provide a preliminary framework for uncertainty quantification by computing a confidence measure of the model changes. Their idea is to perform FWI for baseline and monitor data sets in an alternating fashion and observe the behaviour of the objective function. [66] study time-lapse model uncertainties by comparing the performances of Parallel, Double Difference, and Alternating FWI. These methodologies, however, are somewhat naive in terms of statistical estimations, and are computationally intensive. For comprehensive statistical estimations one needs to compute the posterior probability of the quantities of interest, such as the average change in velocity in a given region of interest, using Bayes' theorem [10].

In a Bayesian seismic inversion we are interested in obtaining a posterior probability density function (*pdf*) of some model parameters given some observations [147]. In this framework, we turn the traditional FWI optimization problem into a sampling problem. Sampling is usually done using Monte Carlo algorithms [95]. Markov Chain Monte Carlo (MCMC) is a family of algorithms that originate in the 1950s [93] but became more widely used in a variety of subjects in the 1990s [124]. These algorithms generate random samples from a probability distribution that is otherwise difficult to sample directly [19]. When the probability space is explored fully and the Markov Chain reaches an equilibrium state, this probability distribution is equivalent to the posterior distribution. However, in order for an equilibrium state to be reached, thousands of models need to be evaluated, which in Bayesian seismic inversion terminology relates to thousands of wavefield solves. This can be a significant computational burden, particularly when most of this computational time is spent on standard Finite-Difference solutions of proposed models while a majority of these models are then rejected due to the low acceptance rate of these algorithms [3]. For an efficient algorithm, ideal acceptance rates range from 10% to 60% with optimal acceptance rates at approximately $20 - 25\%$ [126].

In seismic imaging, there are multiple sources of uncertainty at all different stages from acquiring the data to interpreting the final images [80]. Of course, there are also uncertainties that we cannot even think of; unknown unknowns. Uncertainty estimation is thus a crucial topic at all scales (i.e. exploration scale, crustal scale) of geophysical imaging. Since the early study of [63], work is well underway to understand these uncertainties and quantify them. Here, we refer to some of the most recent work. [105] provide a strategy and overview for uncertainty quantification in seismic tomography that is related to oil and gas exploration and production. [112] estimate uncertainties that are propagated in migrated images through velocity model and picking errors using a Bayesian framework. [141] propose a two-stage Markov Chain Monte Carlo method where the unacceptable velocity proposals are filtered out with an upscaling operator and the accepted ones are then used to solve the fine-grid problem. [133] propose a reversible jump Hamiltonian Monte Carlo method that combines the traditional reversible jump MCMC (RJMCMC) and Hamiltonian Monte Carlo and they achieve faster convergence than the traditional RJMCMC. [151] propose a combination of Ensemble Kalman filters and FWI that provides an uncertainty estimate of the recovered model through the posterior covariance matrix. [119] carry out a 2D Bayesian inversion using a tree based parameterization and trans-dimensional sampling in the wavelet transform domain. [37] use an Adaptive Metropolis Hastings algorithm with a fast forward solver based on the field expansion method [87] to estimate uncertainties in velocity models. [179] perform uncertainty quantification in 3D surface wave tomography using the reversible–jump MCMC and a model parameterization based on Voronoi polyhedrals. [42] describe the Hamiltonian Monte Carlo method for linear and nonlinear tomographic inverse problems. [61] perform a probabilistic FWI on ground penetrating radar data using a geostatistical subsurface parameterization. Even though this literature focuses mostly on sampling based uncertainty quantification techniques, one can find alternative methods that are based on approximations of the likelihood function such as the Polynomial Chaos Expansion (PCE) method [29] [35], where the algorithm tries to approximate the distributions of the uncertain parameters using a set of polynomials, resulting in computational cost reduction.

Even though most of the uncertainty quantification frameworks in the literature use expensive forward solvers, a few attempts to overcome this have been proposed recently. For example, [37] use a fast Helmholtz solver, called field expansion, that provides fast approximate solutions of the wave equation. This allows them to perform fast uncertainty quantification in velocity model building. Another way could be using homogenization or upscaling techniques. These techniques could be used in terms hierarchically moving from cheaper to more expensive forward solvers [158]. For instance, [141] first evaluate models in a coarser grid (less computational time) and if these models are accepted, are then evaluated upscaled to a finer grid. An encouraging alternative is using neural networks to approximate solutions of the wave equation. [128] present an early application of neural networks for geophysical inverse problems, where the algorithm estimates a 1D velocity model given a shot gather. More recently, [54] replace the forward solver and travel time picking for GPR first arrival travel time inversion with a neural network. This allows them to perform a probabilistic inversion with Monte Carlo sampling three orders of magnitude faster compared to traditional forward solvers. Last but not least, a different option could be found in exact localized wavefield techniques [155] [157] [169] [18] [177]. The fundamental idea behind these techniques is to update a model only in a subdomain of interest while still taking into account all data available. This means that no data redatuming needs to take place. Such techniques, are particularly appealing in 4D imaging where we are trying to image small changes in localized regions. Therefore, solving the forward problem in the entire domain while we are interested in a small subdomain (i.e. reservoir) is inefficient. In this work, we use the frequency based local solver developed by [169].

4.2.2 Main Contribution and Overview

The main goal of this work is to provide a computationally feasible framework that enables time-lapse uncertainty quantification. In addition, we study the 4D problem and evaluate different assumptions, such as using an incorrect background model and the presence of local minima at different frequencies, while also providing a way of addressing the dimensionality issue. To our knowledge, there is no work in the literature studying the time-lapse problem using stochastic Full-Waveform inversion. The paper is structured as follows. In Section 4.3 we briefly describe how the local acoustic solver works and explain how we setup the 4D problem in a Bayesian regime. For an efficient posterior calculation, we need an expression only in terms of model differences and data differences, without having to compute a joint distribution over the full domain baseline model. We test our problem setup using a single degree of freedom (DoF) example (Section 4.4). Having such a simple numerical example can prove beneficial in studying a range of potential issues. For example, one key assumption of our derivation is that the posterior calculation is independent of the background model. In Section 4.5, we try to address the effect of such an assumption both analytically in terms of amplitude and travel time at the boundary of the local domain, and numerically using six different inverted background models. Our results show that for a reasonable noise level, all background models result in similar distributions (measured by the recovered mean and median). This implies that a single background model assumption can be valid. However, characterizing the time-lapse uncertainty using only a single DoF is far from realistic. In Section 4.6, we examine a more complex 4D reconstruction using multiple degrees of freedom. To reduce the dimensionality of the problem, we parameterize the 4D model using image compression techniques, where the model is expressed with a set of coefficients. For a standard FWI to reach a global minimum a good starting model is needed. Using a Bayesian FWI, instead, removes such a requirement. However, it is still possible that various frequencies might converge to different local minima in the presence of high noise. In Section 4.7 we discuss this matter for both single and multiple DoF examples used in this study.

4.3 Theory

4.3.1 The Helmholtz equation

A mechanical disturbance in a medium is accompanied with a force that tries to restore the equilibrium situation in the medium [165]. The acoustic wave equation describes the evolution of the acoustic pressure u as a function of position x and time t [41] and it is expressed via a partial differential equation,

$$\nabla \cdot \left(\frac{1}{\rho(x)} \nabla u(x,t) - \frac{1}{K(x)} \frac{\partial^2 u(x,t)}{\partial t^2} \right) = -\frac{\partial^2 s(x,t)}{\partial t^2} + \nabla \cdot \left(\frac{1}{\rho(x)} f_v(x,t) \right), \quad (4.1)$$

where ρ is the density, K is the bulk modulus, $\frac{\partial^2 s(x,t)}{\partial t^2}$ is the point source of volume injection, $\nabla \cdot \left(\frac{1}{\rho(x)} f_v(x,t) \right)$ is the point source of force, and ∇ is the divergence. In this thesis the discussion is limited to the 2D case only. If we consider a monopole source and multiply by density [46], equation (4.1) becomes

$$\rho(x,z)\nabla\cdot\left(\frac{1}{\rho(x)}\nabla u(x,t)-\frac{1}{c^2(x,z)}\frac{\partial^2 u(x,t)}{\partial t^2}\right)=-\rho(x,z)\frac{\partial^2 s(t)}{\partial t^2}\delta(x-x_s), \tag{4.2}$$

where $c=(K/\rho)^{1/2}$ is the acoustic velocity, $\frac{\partial^2 s(t)}{\partial t^2}$ is the time derivative of the rate that a volume is added at a region with a delta function source $\delta(x-x_s)$ at location x_s. Also, in this research, we consider the case of constant density and therefore equation 4.2 is reduced and rewritten as

$$\nabla^2 u(x,t)-\frac{1}{c^2(x)}\frac{\partial^2 u(x,t)}{\partial t^2}=-f(x,t), \tag{4.3}$$

where $f(x,z)$ is a general expression of the source function.

Equations 4.1 - 4.3 take place in time domain. Using a Fourier transform, we can transfer the time dependent wave equation to the frequency dependent Helmholtz equation, such that

$$c^2(x)\nabla^2 U(x,\omega)+\omega^2 U(x,\omega)=-g(x,\omega). \tag{4.4}$$

Dividing all terms with $c^2(x)$, we obtain the final form of the Helmholtz equation, which is stated as

$$\nabla^2 U(x,\omega)+\frac{\omega^2}{c^2(x)}U(x,\omega)=-F(x,\omega). \tag{4.5}$$

4.3.2 Local Acoustic Solver

Following the previous subsection, the term “acoustic” in the title refers to the scalar wave equation in frequency domain (Helmholtz equation, equation 4.5), while the term “local” stands for targeted wavefield solver as opposed to a full domain solver.

Here, we briefly describe how the local solver works; for a detailed explanation please refer to [169]. The first step to apply the local solver is to split the full domain into the exterior domain Ω and the local domain Ω^s. The model in the exterior Ω together with the initial guess in the local domain Ω^s form the background model $\mathbf{m_0}$. In the background model, we compute background Green's functions. The background wavefield u_0 satisfies the Helmholtz equation

$$\nabla^2 u_0 + \mathbf{m_0}\omega^2 u_0 = f, \tag{4.6}$$

where ω is the angular frequency and $\mathbf{m_0}$ is the background model in terms of squared slowness. We assume we have a perturbation $\delta\mathbf{m}$ that only exists within the local domain Ω^s. Therefore, the model $\mathbf{m}$ consists of the background model $\mathbf{m_0}$ and the perturbation $\delta\mathbf{m}$ such as $\mathbf{m} = \mathbf{m_0} + \delta\mathbf{m}$. The total wavefield u inside Ω^s will also be present only inside Ω^s , where it is given as $u = u_0 + \delta u$. This total wavefield has also to satisfy the Helmholtz equation

$$\nabla^2 u + \mathbf{m}\omega^2 u = f. \tag{4.7}$$

Equation 4.7 is used to accurately compute the total field inside the local domain. In order to use this scheme, we need to precompute the Green's functions in the background model $\mathbf{m_0}$ in the full domain. We need to do this background computation only once, and then we can update the model and recompute the wavefield using only the local domain resulting in significant computational savings.

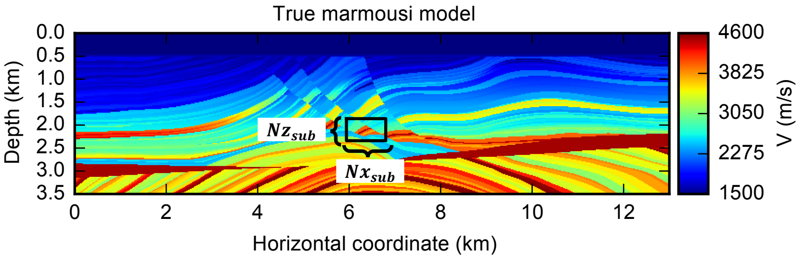

Figure 4.1: The true baseline velocity model with the black box representing the location of the local domain.

4.3.3 4D Problem Setup and Posterior Calculation

Statistical inference is a methodology which concludes properties about populations of data by evaluating evidence. There are two fundamentally different categories of statistical inference: the Frequentist statistics and the Bayesian statistics [40]. One of the main differences between the two categories is that Bayesian statistics use the laws of probability while the Frequentist statistics do not. Specifically, Bayesian statistics use prior probabilities of quantities, which they later update to posterior probabilities given some evidence. Frequentist statistics on the other hand, do not use probabilities (prior or posterior), but rather compute characteristics such as p-values or confidence intervals from the distribution of the data. For example, Chapter 2 of this thesis can be considered as Frequentist statistics. Both types of statistics are widely used in science, however Bayesian statistics are more logically rigorous and allow for deductive logic.

Bayesian statistics are build upon Bayes' theorem. Bayes' theorem describes the relationship between a hypothesis and given evidence. In seismic imaging, this is expressed in terms of velocity models ($\mathbf{m}$) and observed data ($\mathbf{d}$),

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{p(\mathbf{d})}, \tag{4.8}$$

where $p(\mathbf{m}|\mathbf{d})$ is the quantity of interest for any probabilistic inversion, or the probability that a given model, $\mathbf{m}$, is the truth given the existence of the data, $\mathbf{d}$; $p(\mathbf{m}|\mathbf{d})$ is called the posterior. In equation 4.8, $p(\mathbf{d}|\mathbf{m})$ is the likelihood function calculated by [146]

$$\begin{aligned} \mathcal{L}(\mathbf{m}) &\equiv p(\mathbf{d}|\mathbf{m}) \\ &\propto exp\left[-\frac{1}{2}(F(\mathbf{m})-\mathbf{d})^T\Sigma^{-1}(F(\mathbf{m})-\mathbf{d})\right], \end{aligned} \tag{4.9}$$

where F is the forward solver and Σ^{-1} is the inverse covariance matrix of the data noise, $p(\mathbf{m})$ is the input prior model distribution and $p(\mathbf{d})$ is considered to be a normalization constant.

Following [67], we now explain how we are going to set up our problem for the Metropolis Hastings algorithm. Let $\mathbf{m}$ be the velocity model and $\mathbf{n}$ be zero mean Gaussian noise with a covariance matrix Σ, then

$$\mathbf{d} = G(\mathbf{m}) + \mathbf{n}, \tag{4.10}$$

where G is the forward modeling solver (i.e. the local domain solver here). In the 4D case, we have $\mathbf{d_1} = G(\mathbf{m_1}) + \mathbf{n_1}$ for the baseline model, and $\mathbf{d_2} = G(\mathbf{m_2}) + \mathbf{n_2}$ for the monitor. If we let $\delta\mathbf{m} = \mathbf{m_2} - \mathbf{m_1}$ (which is equivalent to the model perturbation described in the section above) and $\delta\mathbf{d} = \mathbf{d_2} - \mathbf{d_1}$, then our goal is to find the probability of the model differences given the two datasets,

$$p(\delta\mathbf{m}|\mathbf{d_1}, \mathbf{d_2}). \tag{4.11}$$

In this expression there is a hidden variable, $\mathbf{m_1}$. In order to calculate the distribution over $\delta\mathbf{m}$, we will first need to calculate the joint distribution $p(\mathbf{m_1}, \delta\mathbf{m}|\mathbf{d_1}, \mathbf{d_2})$ and then integrate over $\mathbf{m_1}$ to get the distribution on only $\delta\mathbf{m}$. However, this would be computationally expensive because we have to sample for both $\delta\mathbf{m}$ and $\mathbf{m_1}$, where $\mathbf{m_1}$ is the baseline velocity model, which needs to be computed in the full domain. Therefore, we will need an expression that is only in terms of $\delta\mathbf{m}$. To obtain this, we rewrite the forward modeling expression in equation 4.10 as,

$$\begin{aligned}\delta\mathbf{d} &= G(\mathbf{m_2}) + \mathbf{n_2} - G(\mathbf{m_1}) - \mathbf{n_1} \\ &= G(\mathbf{m_2}) - G(\mathbf{m_1}) + (\mathbf{n_2} - \mathbf{n_1}) \\ &= G(\mathbf{m_1} + \delta\mathbf{m}) - G(\mathbf{m_1}) + (\mathbf{n_2} - \mathbf{n_1}). \qquad (4.12)\end{aligned}$$

The sum or difference of two zero mean Gaussians $(\mathbf{n_1}, \mathbf{n_2})$ is equal to a single Gaussian $(\mathbf{n_3})$ with a covariance $\Sigma_3 = \Sigma_1 + \Sigma_2$. From this, we can rewrite equation 4.12 as

$$\delta\mathbf{d} = G(\mathbf{m_1} + \delta\mathbf{m}) - G(\mathbf{m_1}) + \mathbf{n_3}. \qquad (4.13)$$

If we let $F(\mathbf{m_1}, \delta\mathbf{m}) = G(\mathbf{m_1} + \delta\mathbf{m}) - G(\mathbf{m_1})$, equation 4.13 becomes

$$\delta\mathbf{d} = F(\mathbf{m_1}, \delta\mathbf{m}) + \mathbf{n_3}. \qquad (4.14)$$

Equation 4.14 is now almost entirely in terms of $\delta\mathbf{m}$. If we now assume that $F(\mathbf{m_1}, \delta\mathbf{m})$ is independent of the initial model $\mathbf{m_1}$, then at any model update

$$F(\mathbf{m_{1a}}, \delta\mathbf{m}) = F(\mathbf{m_{1b}}, \delta\mathbf{m}), \qquad (4.15)$$

where $\mathbf{m_{1a}} \approx \mathbf{m_{1b}}$. This statement is valid for any small perturbation in $\mathbf{m_1}$. To show this numerically we consider the true Marmousi model (Figure 4.1) with a perturbation in one of the background layers outside the local domain (Figure 4.2 (a)). Because we compare model differences to data differences, which is similar to Double Difference FWI schemes, we expect that any perturbations in the background model that are not explained by the data will be cancelled out. We, therefore, compare the data residual $F(\delta\mathbf{m})$ generated for a time–lapse perturbation $\delta\mathbf{m} = 75$ m/s using both the true (Figure 4.1) and perturbed (Figure 4.2 (a)) Marmousi as the background model (Figure 4.2 (b)). The two residual wavefields are almost identical with tiny differences highlighted with the two dashed circles. Assuming that this time–lapse perturbation ($\delta\mathbf{m} = 75$ m/s) is also the true perturbation, we can further compare the residual wavefields ($F(\delta\mathbf{m})$) with the observed data residual $\delta\mathbf{d}$, by looking at the data misfit $F(\delta\mathbf{m}) - \delta\mathbf{d}$ that is used in the likelihood (Figure 4.2 (c)). When the background model is correct, this data misfit is zero. When the background model is slightly perturbed, this data misfit is very small but not exactly zero. Here, the data misfit is two orders of magnitude smaller than the data difference.

The forward model is now only dependent on the difference of models, difference of observed data, and the sum of some known covariance matrices,

$$\delta\mathbf{d} = F(\delta\mathbf{m}) + \mathbf{n_3}. \tag{4.16}$$

Bayes' theorem from equation 4.8 now becomes

$$p(\delta\mathbf{m}|\delta\mathbf{d}) = \frac{p(\delta\mathbf{d}|\delta\mathbf{m})p(\delta\mathbf{m})}{p(\delta\mathbf{d})}, \tag{4.17}$$

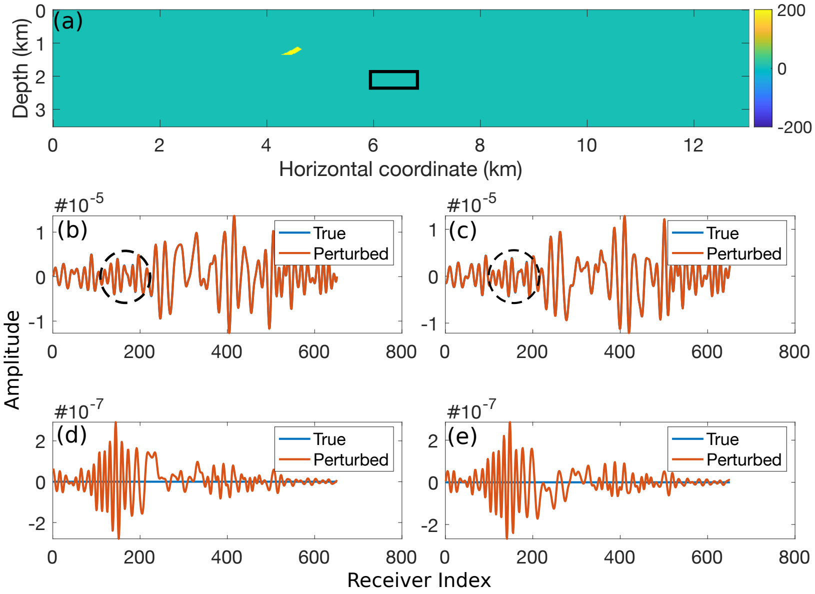

Figure 4.2: (a) Error of 200 m/s in the background model with the black box representing the location of the local domain (which contains the time-lapse perturbation). (b) & (c) Real and imaginary parts of the data residuals using the true and the perturbed Marmousi as the background model. (d) & (e) Real and imaginary parts of the data misfit $F(\delta\mathbf{m}) - \delta\mathbf{d}$ from the residual wavefields used in (b).

and the likelihood function from equation 4.9 is

$$\begin{aligned} \mathcal{L}(\delta\mathbf{m}) \equiv p(\delta\mathbf{d}|\delta\mathbf{m}) \propto & \\ & exp\left[-\frac{1}{2}(F(\delta\mathbf{m}) - \delta\mathbf{d})^T {\Sigma_3}^{-1}(F(\delta\mathbf{m}) - \delta\mathbf{d})\right]. \end{aligned} \tag{4.18}$$

For an accurate posterior calculation, a large number of samples need to be generated, and then the first half are discarded in order to reduce the impact of the starting model [19]. This is because during the so–called burn–in phase the algorithm tries to move from the initial models to areas with higher posterior density. Once in these areas, the algorithm samples the solutions corresponding to the posterior density. To sample the posterior of the single DOF example we use a Metropolis Hastings algorithm [55] [125]. An advantage of the Metropolis Hastings algorithm is that it does not require the computation of $p(\mathbf{d})$ or the calculation of the gradient of the likelihood function. The pseudocode in Algorithm 1 provides a brief explanation of the algorithm. At each iteration i, we get a new proposal $\delta\mathbf{m}_*$ by adding a zero mean perturbation to the current $\delta\mathbf{m_{i-1}}$ such that $\delta\mathbf{m}_* = \delta\mathbf{m_{i-1}} + n, n{\sim}N(0,C)$, where $N(0,C)$ is a zero–mean vector drawn from the Gaussian distribution with a covariance C. The acceptance of the new proposal is determined by the ratio of the likelihood functions of the proposal and current samples, $\alpha_i = \frac{L(\delta\mathbf{m})_*}{L(\delta\mathbf{m})_{i-1}}$. Typically, the acceptance probability is defined using the Metropolis–Hastings rule [55] via

$$\alpha = p(\delta\mathbf{m}^*|\delta\mathbf{m}_{i-1}) = min\left[1, \frac{p(\delta\mathbf{m}^*)}{p(\delta\mathbf{m}_{i-1})}\frac{p(\delta\mathbf{d}|\delta\mathbf{m}^*)}{p(\delta\mathbf{d}|\delta\mathbf{m}_{i-1})}\frac{Q(\delta\mathbf{m}_{i-1}|\delta\mathbf{m}^*)}{Q(\delta\mathbf{m}^*|\delta\mathbf{m}_{i-1})}\right], \tag{4.19}$$

where Q is called the transition kernel, which is simply a way of transitioning randomly to a point $\delta\mathbf{m}^*$ given a point $\delta\mathbf{m}_{i-1}$. Equation 4.19 simplifies to a Metropolis update [93] if the proposal distribution is symmetrical and if the prior distribution is uniform. As mentioned above, the proposal distribution is usually recentered after each step at the value that was last generated by the chain. Generally, in a random walk the proposal distribution is Gaussian, in which case it satisfies the symmetry requirement. In addition, for the numerical example of this study we define a bounded uniform prior distribution. Therefore, the acceptance probability is simplified to the ratio of likelihood functions of the proposal and current samples, such as

$$\alpha = p(\delta\mathbf{m}^*|\delta\mathbf{m}_{i-1}) = min\left[1, \frac{p(\delta\mathbf{d}|\delta\mathbf{m}^*)}{p(\delta\mathbf{d}|\delta\mathbf{m}_{i-1})}\right]. \tag{4.20}$$

If it is accepted, the proposal becomes the new current. If it is rejected the current proposed model is reused.

Algorithm 1 4D Metropolis Hastings algorithm

```
Require: $\delta\mathbf{m_0}$                                ▷ initial perturbation
Require: N                                         ▷ maximum number of iterations
Require: C                                          ▷ proposal covariance matrix
$L(\delta m_0)$                                        ▷ likelihood of initial model
for $i = 1, ..., N$ do
    $n \leftarrow Normal(0, C)$                                      ▷ proposed jump
    $\delta m_* \leftarrow \delta m_{i-1} + n$                        ▷ proposed model perturbation
    $L(\delta m_*)$                                     ▷ likelihood of the proposal
    $\alpha_i = \frac{L(\delta m_*)}{L(\delta m_{i-1})}$                               ▷ acceptance probability
    $u \leftarrow U[0, 1]$                                      ▷ uniform distribution
    if $u < \alpha_i$ then
        $\delta m_i \leftarrow \delta m_*$                                     ▷ accept proposal
    else
        $\delta m_i \leftarrow \delta m_{i-1}$                                   ▷ reject proposal
    end if
end for
```

It is well known that for fast convergence of a Metropolis Hastings procedure an effective proposal distribution is needed (in terms of size and spatial orientation). In cases where the classic Metropolis Hastings algorithm does not converge, we can use an Adaptive Metropolis Hastings algorithm that uses the history of the process to tune the proposal distribution [51]. The pseudocode in Algorithm 2 summarizes the Adaptive Metropolis Hastings algorithm. We typically run the MCMC with a fixed step size $C_i = C_0$ for a number of iterations N_c, and then start updating C_i using the covariance of the models already in the chain. The choice of N_c reflects the trust in the initial estimate of C_0; if C_0 is assumed to be equivalent to $Cov[\delta\mathbf{m_0}, ..., \delta\mathbf{m_{i-1}}]$ then N_c is set to a small number. Otherwise, N_c is set to a larger value. In this study, we use Algorithm 2 for all the runs in the multiple DoF examples and Algorithm 1 for the single DoF examples. We find that a choice of $N_c = 1000$ leads to good convergence. It is important to mention that even though we assume that the measurement noise is Gaussian, we make no assumptions about the posterior velocity model distribution. These MCMC algorithms generate a number samples $\delta\mathbf{m}_0, \delta\mathbf{m}_1, ...\delta\mathbf{m}_N$ that are drawn from the posterior distribution.

Algorithm 2 4D Adaptive Metropolis Hastings algorithm

```
Require: $\delta\mathbf{m_0}$                                  ▷ initial perturbation
Require: $C_0$                                                 ▷ initial step size
Require: N                                                     ▷ maximum number of iterations
Require: $N_c$                                                 ▷ number of iterations before the updating starts
Require: $d = length(\delta\mathbf{m_0}, ..., \delta\mathbf{m_{i-1}})$
Require: $S_d = \frac{2.4^2}{d}$                               ▷ value from [51]
Require: $\epsilon \ll 1$
    $L(\delta m_0)$
    for $i = 1, ..., N$ do
        if $i < N_c$ then
            $C_i = C_0$                                        ▷ fixed step size
        else
            $C_i = S_d(Cov[\delta\mathbf{m_0}, ..., \delta\mathbf{m_{i-1}}] + \epsilon\mathbf{I}_d)$,   ▷ tune step size based on covariance of models
        end if
        $n \leftarrow Normal(0, C_i)$                          ▷ proposed jump
        $\delta m_* \leftarrow \delta m_{i-1} + n$             ▷ proposed model perturbation
        $L(\delta m_*)$                                        ▷ get the likelihood of the proposal
        $\alpha_i = \frac{L(\delta m_*)}{L(\delta m_{i-1})}$   ▷ acceptance probability
        $u \leftarrow U[0, 1]$                                 ▷ uniform distribution
        if $u < \alpha_i$ then
            $\delta m_i \leftarrow \delta m_*$                 ▷ accept proposal
        else
            $\delta m_i \leftarrow \delta m_{i-1}$             ▷ reject proposal
        end if
    end for
```

4.4 Single Degree of Freedom Estimation

4.4.1 Numerical Example

To setup a simple numerical illustration, we use the standard Marmousi model [159] as the true baseline model and as the background model $\mathbf{m_0}$ on which we compute the full subsurface Green's functions (Figure 4.1). The black box shows the location

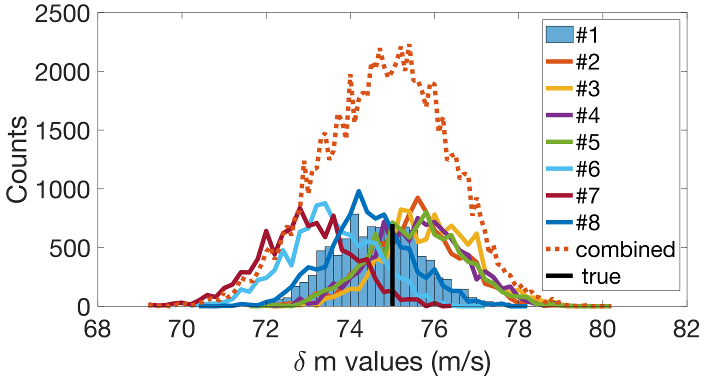

Figure 4.3: Histograms of the recovered time-lapse velocity change from eight different noise realizations.

of the local subdomain in which the likelihood function is evaluated at each iteration i. Nx_{sub} and Nz_{sub} represent the number of grid points in the subdomain in the $x-$ and $y-$ direction respectively. The number of these grid points is significantly smaller than the number of grid points in the full domain; in this example $Nx_{sub} = 44$ and $Nz_{sub} = 25$, whereas $Nx_{full} = 651$ and $Nz_{full} = 176$.

We create the monitor model by adding a perturbation of 75 m/s in one of the layers in the local domain. For the simulations, we use a single shot located in the middle of the model and 651 equally spaced receivers at the surface. The source is a Ricker wavelet with a peak frequency of 6 Hz. We evaluate the likelihood for the single frequency of 8 Hz. In the inversions we use noisy $\delta\mathbf{d}$ to which we add Gaussian noise with covariance matrix Σ_d. The signal to noise ratio is 1.9, calculated as the ℓ_2 norm of the noiseless signal over the ℓ_2 norm of the noise.

To examine the effect of random noise in the recovered posterior distributions, we generate eight different noise realizations. All realizations have a similar signal to noise ratio and are described by the same covariance matrix Σ_d. We run the nonlinear inversion for 20,000 iterations with a fixed step-size $C_i = \sigma$ for each realization and we discard the first half. The initial guess is $\delta\mathbf{m}_0 = 0$ m/s, and we invert only the magnitude of the 4D change by keeping the shape fixed (hence the single parameter estimation). Figure 4.3 shows the resulting histograms for all noise realizations with the straight black line representing the true perturbation of 75 m/s. As expected, each noise realization recovers slightly different means. However, if we take all histograms together in one (orange line in Figure 4.3), we recover a mean of 74.86 m/s which has an error of 0.18% (measured using mean absolute percentage error) from the true value.

4.4.2 Linearity of the Results

Traditional time-lapse seismic analysis includes looking at migrated data cubes [49] [81]. These methodologies assume that the monitor model is well estimated by a linear perturbation of the baseline model. In a case like this, the 4D problem is likely to be a rather linear problem. Of course, in real cases where strong changes in the physical properties of the reservoir (and/or overburden) are introduced through production, the linearity assumption might not hold anymore [136]. Here, we try to further understand the Gaussian distribution of the histograms in Figure 4.3 mathematically. The observed data retrieved at the receivers can be expressed as

$$\mathbf{d} = \mathcal{P}u \tag{4.21}$$

where u is the wavefield everywhere in the subsurface and $\mathcal{P}$ is a projection matrix that projects the wavefield to the receivers. The wavefield u can be obtained by applying a wave equation operator $\mathcal{W}$ given a source s,

$$\mathcal{W}u = s. \tag{4.22}$$

Since we work in frequency domain $\mathcal{W} = \nabla^2 + \mathbf{m}\omega^2$ is the Helmholtz operator. At time 1 (baseline) and time 2 (monitor) we will consequently have

$$\mathbf{d_1} = \mathcal{P}u_1,\ \mathcal{W}_1 u_1 = s_1, \tag{4.23}$$

and

$$\mathbf{d_2} = \mathcal{P}u_2,\ \mathcal{W}_2 u_2 = s_2, \tag{4.24}$$

generated by a small time–lapse perturbation such as the one in the numerical example above ($\delta\mathbf{m} = 75$ m/s), and we want to examine the relationship of

$$\mathbf{d_2} - \mathbf{d_1} = \mathcal{P}(u_2 - u_1). \tag{4.25}$$

Typically, in a 4D experiment we try to repeat the source as best as possible which means that we can assume $s_2 = s_1$, and hence

$$\mathcal{W}_2 u_2 - \mathcal{W}_1 u_1 = 0. \tag{4.26}$$

If we express $\mathcal{W}_2 = \mathcal{W}_1 + \delta\mathcal{W}$, and $u_2 = u_1 + \delta u$, then equation 4.20 becomes

$$\mathcal{W}_1 \delta u + \delta\mathcal{W} u_1 + \delta\mathcal{W}\delta u = 0. \tag{4.27}$$

This equation is very similar to the Born approximation. To proceed further, we will look into $\delta\mathcal{W}$ and see how it behaves. The third term $\delta\mathcal{W}\delta u$ is very small given the small model perturbation (which means that $\delta u \ll u_1, u_2$ and $\delta\mathcal{W} \ll W_1, W_2$) and it can be neglected. In the second term

$$\delta \mathcal{W} u_1 = \left[(\nabla^2 + \mathbf{m_2}\omega^2) - (\nabla^2 + \mathbf{m_1}\omega^2) \right] u_1 \tag{4.28}$$
$$= \omega^2(\mathbf{m_2} - \mathbf{m_1})u_1,$$

which can be thought of as a source term. Substituting equation 4.28 to 4.27 we find

$$\mathcal{W}_1 \delta u = -\omega^2(\delta \mathbf{m})u_1, \tag{4.29}$$

which is a wave equation, where the wavefield δu is linear with respect to the source $(-\omega^2 \delta \mathbf{m} u_1)$. Recall that our objective is to understand whether the time–lapse problem can be considered linear for small time–lapse perturbations, by studying the data difference relationship in equation 4.25. One way to show that is by looking at the right hand side ($\mathcal{P}\delta u$) of equation 4.25; if the right hand side has linear behaviour, then so does the left hand side. A projection matrix ($\mathcal{P}$) is by definition a linear operator, and δu is shown to be linear in equation 4.29. Therefore, we can assume that the data difference $\mathbf{d_2} - \mathbf{d_1}$ is also linear, meaning that the 4D problem can be treated as linear for small time–lapse perturbations.

To further validate our argument, we consider 41 different time-lapse velocity models $\delta \mathbf{m}$ ranging from -200 m/s to 200 m/s, and their respective data residuals $\delta \mathbf{d}$. The simplest way to see what relationship models versus data exhibit is by plotting them against each other. To do so, we first organize them into a matrix form. Let $\underset{j \times 41}{\boldsymbol{\mathcal{M}}}$ represent the time-lapse velocity models matrix where each column represents a vectorized model. The subscript j represents the size of the vector such that $j = nz \times nx$. Similarly, we define $\underset{i \times 41}{\boldsymbol{\mathcal{D}}}$ as the data residual matrix where each column represents the data $\delta \mathbf{d}$ for the equivalent $\delta \mathbf{m}$ in the $\boldsymbol{\mathcal{M}}$. To avoid any complications when using complex numbers, each data residual is represented as a concatenation of real and imaginary parts such that $\delta\mathbf{d} = [real(\delta\mathbf{d}); imag(\delta\mathbf{d})]$. Please note that the dimension i is different from the dimension j. Plotting $\boldsymbol{\mathcal{M}}$ against $\boldsymbol{\mathcal{D}}$ would be challenging for two reasons. First, their dimensions are too high for a classic 2D visualization. Second, even if the dimensions were not a problem, these two matrices have different sizes.

To reduce their dimensions, we use the commonly used Singular Value Decomposition (SVD). When we plot the singular values of both $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{D}}$, we see that they decay extremely fast (Figure 4.4 (a),(b)). Matrix $\boldsymbol{\mathcal{M}}$ appears to be only rank 1, which is not surprising given the single degree of freedom of the problem. Matrix $\boldsymbol{\mathcal{D}}$ on the other hand appears to be rank 2, with the second singular vector being significantly smaller than the first. We can then reduce the dimensionality of both matrices, by performing an SVD using only the 1st singular value of $\boldsymbol{\mathcal{M}}$ and the 1st and 2nd singular values of $\boldsymbol{\mathcal{D}}$. Plotting the resulting right singular vectors against each other reveals the relationship between the two matrices. However, to make this a clearer visualization, we plot the time–lapse perturbations used in $\boldsymbol{\mathcal{M}}$ against the two right singular vectors of $\boldsymbol{\mathcal{D}}$ (Figure 4.4 (c),(d)). There is clearly a relationship between them. Panel (c) shows that the relationship between the first right vectors is almost linear; the slight decay from linear represents the non-linearity of the problem. When plotting the first vector of $\boldsymbol{\mathcal{M}}$ against the second right vector of $\boldsymbol{\mathcal{D}}$ (Panel (d)) the relationship is quadratic. This shows, therefore, that for a single DoF the relationship of model and data is very close to linear, and this is the reason that the PDFs in section 4.4.1 are Gaussian. However, this changes when moving to higher DoFs as shown later.

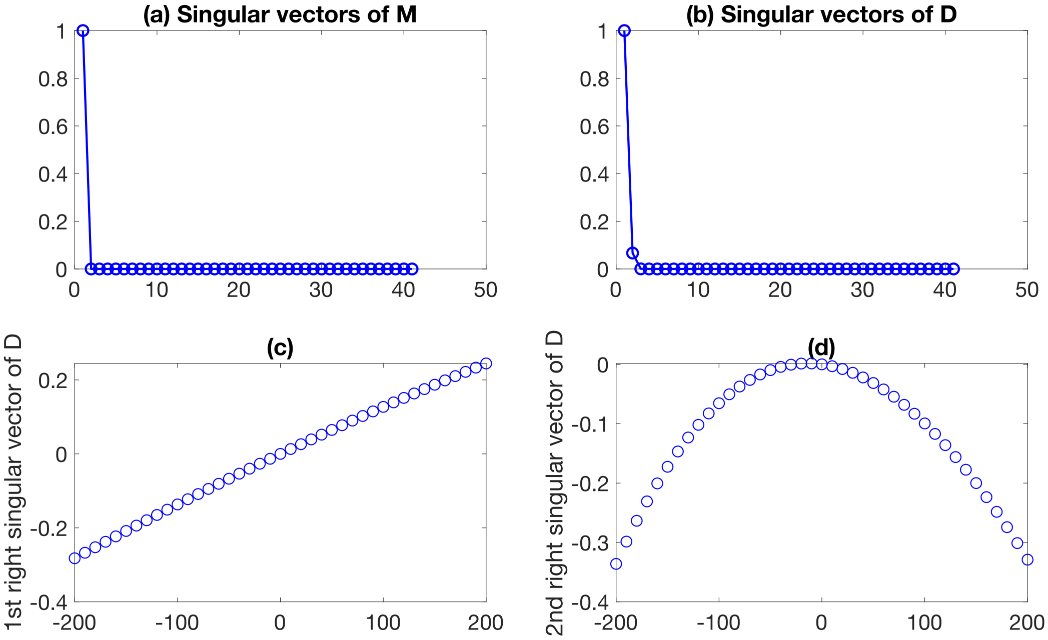

Figure 4.4: (a) & (b): Singular values for the matrices $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{D}}$ normalized by their maximum value. (c) & (d): Plots of time–lapse velocity perturbations $\delta\mathbf{m}$ used in $\boldsymbol{\mathcal{M}}$ against the two singular vectors of $\boldsymbol{\mathcal{D}}$.

4.5 Uncertainty and Dependance on the Background Model

4.5.1 Analytical Study

In section 4.3.3, in order to derive the 4D formulation of our posterior we assumed that the background model does not change significantly between model updates (equation 4.15). Therefore, we use a forward solver that is independent of the background model.

In this section, we try to verify this assumption both analytically and numerically. To start with the analytical study, we focus on what happens at the boundary of the local domain, when the background model (the one in the exterior Ω) changes. In the

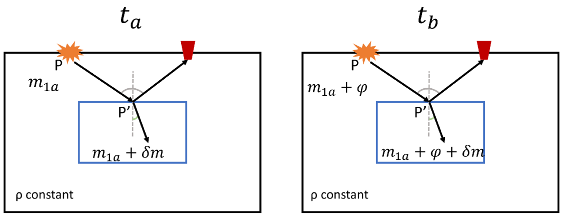

Figure 4.5: Illustration of the reflected and transmitted rays at the boundary of the local solver at time t_a and t_b.

raw seismic signal we have information about the time and amplitude of events. In a full wavefield scenario, this seismic signal can consist of direct, reflected, transmitted, and other smaller waves, with only the reflected and transmitted being important for the changes in the boundary of the local domain. We analyze the dependence of these waves on the background both in terms of amplitude and travel time.

Let us assume, that between time t_a and time t_b the background model m_1 changes by a perturbation ϕ. Hence, at time t_a we have a model m_{1a} and at time t_b we have a model $m_{1b} = m_{1a} + \phi$ (Figure 4.5). Here, we will also assume that models m_{1a} and m_{1b} represent velocity, and we are working in a constant density case. If we evaluate the zero–offset reflection coefficients exactly at the boundary of the local domain (Figure 4.5) at times t_a and t_b then

$$RC_a = \frac{m_{1a} - (m_{1a} + \delta m)}{m_{1a} + (m_{1a} + \delta m)} \leftrightarrow RC_a = \frac{-\delta m}{2m_{1a} + \delta m}, \tag{4.30}$$

and

$$RC_b = \frac{(m_{1a} + \phi) - (m_{1a} + \phi + \delta m)}{(m_{1a} + \phi) + (m_{1a} + \phi + \delta m)} \leftrightarrow RC_b = \frac{-\delta m}{2m_{1a} + 2\phi + \delta m}. \tag{4.31}$$

Therefore, we see that RC_b differs from RC_a by 2ϕ in the denominator. However ϕ is a fraction of the background model m_{1a} , and not the perturbation δm. This means that it has to be very big in order to create a big deviation between the amplitudes of the reflected waves at the two times. Hence, we can assume that $RC_a \approx RC_b$. Similarly, for the transmission coefficient we can assume that $TC_a \approx TC_b$

Having established that perturbations in the background model do not have a large impact on the amplitudes, let us now think what happens to the travel times. According to ray theory, the travel time through a medium with a velocity m_{1a} is

$$T_{1a} = \int_{ray} \frac{1}{m_{1a}} dS. \tag{4.32}$$

We are interested to know how different T_{1b} would be from T_{1a}. We thus define δT as the travel time anomaly between the travel time through the perturbed model $m_{1b} = m_{1a} + \phi$ minus the travel time at the reference background model m_{1a},

$$\delta T = \int_{ray(m_{1a}+\phi)} \frac{1}{m_{1a}+\phi} dS - \int_{ray(m_{1a})} \frac{1}{m_{1a}} dS. \tag{4.33}$$

According to Fermat's principle, *"for two points A and B on a ray, the ray itself is a path along which, in the velocity field $v(x)$, the travel time from A to B is stationary"* ([107], [135]). This means that the derivative of the travel time with respect to any small perturbation (here the ϕ perturbation of the velocity field m_{1a}) to the path is zero, which is important because it means that we get the correct answer (to first order) even if we have small perturbations along the ray path. Then, we can rewrite equation 4.33

$$\delta T = \int_{ray(m_{1a})} \frac{1}{m_{1a}+\phi} dS - \int_{ray(m_{1a})} \frac{1}{m_{1a}} dS \cong \int_{ray(m_{1a})} \frac{-\phi}{m_{1a}^2} dS, \tag{4.34}$$

where the term $m_{1a}\phi$ is small so we ignored it. Let the perturbation to be $\phi = 20\%$ and the background velocity model be $m_{1a} = 2000\ m/s$. Then the $m_{1a}\phi = 0.01\%$

difference in travel times. From equation 4.34 we can thus draw the conclusion that the difference in the travel time between a reference model (i.e. true velocity model) and a perturbed model (i.e. inverted velocity model) is linearly dependent on the perturbation.

4.5.2 Numerical Study

Thus far we have studied the impact of the incorrect background model analytically in terms of travel time differences and reflection and transmission coefficients at the boundary of the local domain. In this section, we study the impact of the incorrect background model following a numerical approach. [169] already mentioned that convergence to the true local model may not be achieved in the presence of an incorrect exterior. Therefore, we expect that there will be an impact on the time-lapse estimation when the background model is incorrect. This section does not only aim to prove that impact, but also understand and quantify the role of it in the time-lapse uncertainty estimation. Consequently, we want to answer the following questions:

1. What is the role of uncertainty or "incorrectness" of the background model in the estimation of the 4D uncertainty?
2. How good is the single model assumption?

To begin with, we consider six different initial models that were generated using the field expansion method [87]. In this method, a velocity model is estimated by a perturbed layer medium and then the analytical Helmholtz equation is quickly solved using a Taylor expansion [36]. Here, we do not approximate the analytical solution to the Helmholtz equation, but use the method only to generate the initial models. For each of the six initial models we perform an acoustic frequency—domain FWI using

Table 4.1: Mean absolute percentage error of inverted model

Inverted model	**Error**
1	8.9333
2	10.5202
3	10.6299
4	10.7580
5	11.4442
6	12.3989

64 shots and 651 receivers equally spaced at the surface of the model. We invert for six frequencies (3, 4, 5, 6.5, 8, 10 Hz) sequentially for 15 iterations at each frequency. We sort the models from one to six, based on their error measurement from the true Marmousi model, with model 1 being the lowest error (Figure 4.6). To measure the error, we use a mean absolute percentage error,

$$E_i = \left[\frac{1}{nz \cdot nx} \sum \frac{|model_i(:) - true(:)|}{true(:)} \right] \cdot 100, \tag{4.35}$$

where the subscript i represents each of the models, and nz, nx are the number of grid points in z- and x- directions respectively, and (:) represents the conversion of a matrix to a vector. Table 4.1 summarizes the percentage error for each of the inverted models.

We then use each of these inverted baseline models as a background model and run the MCMC algorithm. We use the same numerical setup as in section 3.1 and we want to retrieve the same time-lapse perturbation of $\delta m = 75\ m/s$. At each iteration of MCMC, a proposed δm_* is drawn from the Markov chain, which is added to the (incorrect) background model to create the estimated monitor model. We get the $F(\delta m_*)$ for the likelihood evaluation by subtracting the dataset computed on the background model from the dataset on the estimated monitor model. Obviously, the further away that background model is from the true, the more biased will be

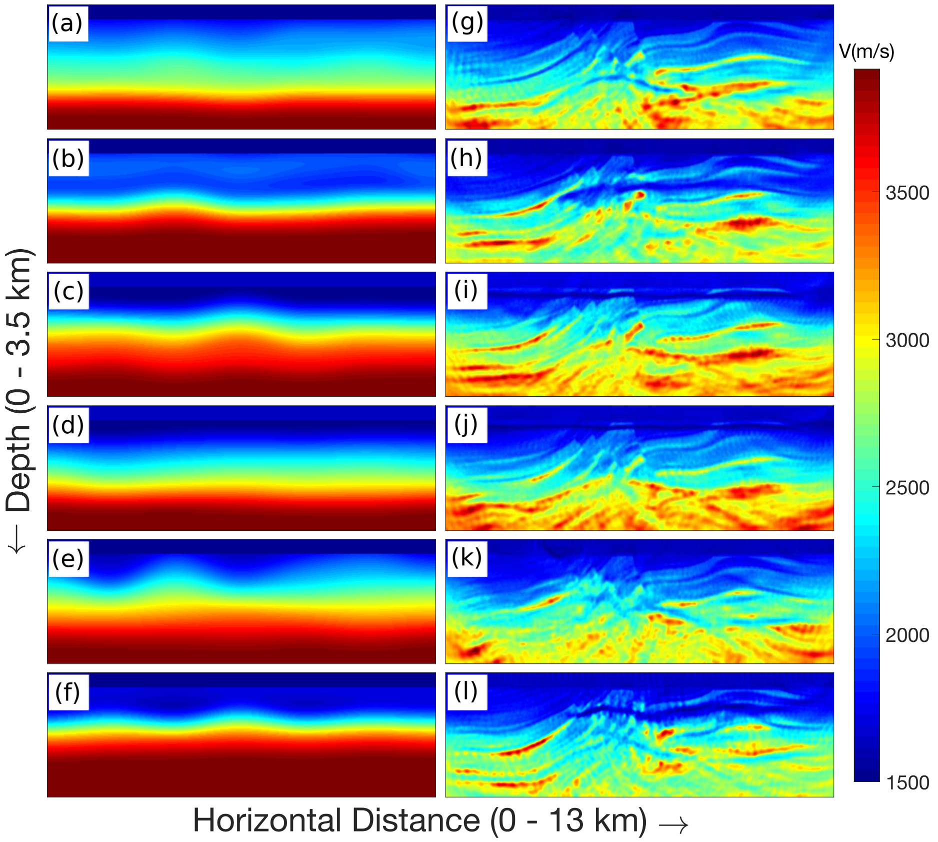

Figure 4.6: The six initial models (a -f) from the field expansion method (left panel) together with the respective inverted models (right hand side g- l)

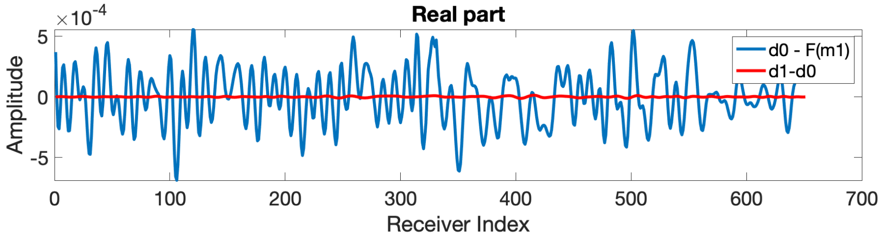

Figure 4.7: Dataset comparison of model 1's deviation from the true (blue line) and the true time-lapse change (red line) in a completely noise free environment; $d0$ is the true baseline data, $F(m1)$ is the synthetic data generated on model 1 (Figure 4.6), and $d1$ is the true monitor data.

the recovered density function. In simple words, it is possible that the incorrectness of the background model is greater than the time-lapse change itself, leading to the time-lapse change being "masked". If for example we consider model 1 that has the lowest error, we see that the model's residual from the true is higher by one order of magnitude from the true time-lapse residual (Figure 4.7). This difference may be more significant in the presence of noise, a scenario highly realistic. Therefore, it will be almost impossible to obtain a meaningful time-lapse distribution, by evaluating our current likelihood function where the covariance matrix Σ_d is based on the measurement noise. To combat this issue, we replace the standard covariance matrix with one that is generated based on the energy of the model residuals. In this way, we can account for the background model incorrectness while still recover useful distributions. We define the model residual energy as

$$\epsilon_i = \frac{2\left\|d_0 - F(m_i)\right\|}{numel(F(m_i))}, \tag{4.36}$$

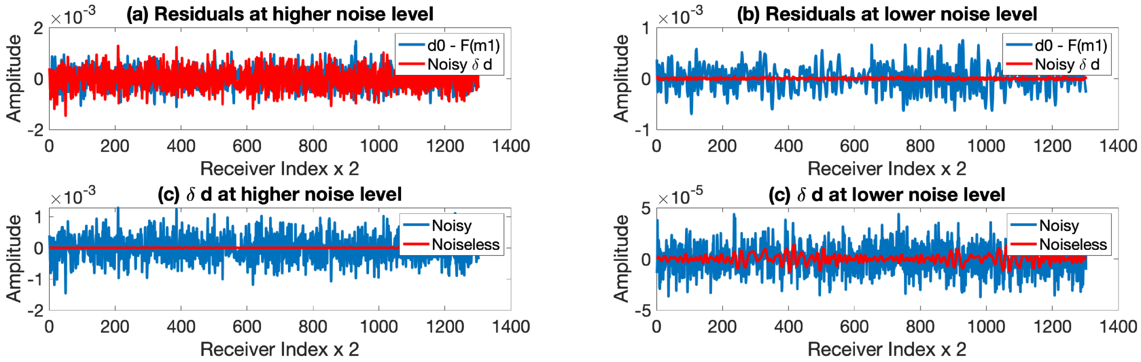

Figure 4.8: Model residual for model 1 (Figure 4.6) versus data residual. All data shown are after we concatenated the real and imaginary part. (a) Comparison of the model residual obtained by subtracting the synthetic $F(m_1)$ from the noisy baseline data d_0 in the high noise scenario (blue line) and the noisy data residual $\delta d = d_1 - d_0$ (red line). (b) Similar to (a) but for the low noise scenario. (c) Noisy data residual versus noiseless data residual in the high noise case. (d) Noisy data residual versus noiseless data residual in the low noise case.

and the covariance matrix as

$$\boldsymbol{\Sigma_i} = \begin{bmatrix} \epsilon_i & 0 & 0 & \dots \\ 0 & \epsilon_i & 0 & \dots \\ \vdots & & \ddots & \\ 0 & \dots & 0 & \epsilon_i \end{bmatrix}, \tag{4.37}$$

where the subscript i represents each of the background models, and *numel* represents number of vector elements.. Each inverted model will have a different covariance matrix. To set up a realistic numerical example we consider that the baseline (d_0) and monitor (d_1) were recorded in both a relatively high- and low-noise environment. In the relatively high noise environment, the noise is high enough to fully mask the true time-lapse signal (Figure 4.8 (c)). The difference in the noise level is more apparent if one notices the two orders of magnitude scale difference between Figure 4.8 (c) and Figure 4.8 (d).

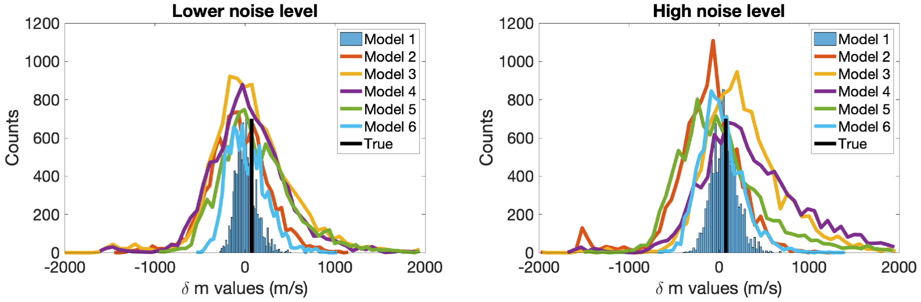

Figure 4.9: Recovered time-lapse change histograms using all six background models for the low noise case (left panel) and the high noise case (right panel).

We run the MCMC algorithm for 20,000 iterations using the six inverted models as background and their corresponding covariance matrices for both noise level scenarios. Figure 4.9 shows the recovered distributions for a single chain after we discard the 1st half to drop any dependancy on the starting model. Clearly, when we look at these histograms we cannot compare them explicitly since they were all retrieved using different covariance matrices. However, we use this type of visualization so we can better answer our original questions. We observe that the range of these histograms are significantly bigger compared to the histograms where the true Marmousi model was used as the background model (Figure 4.3). The histograms here are not exactly Gaussian, meaning that measures such as mean or standard deviation might not be representative anymore. Therefore, we could either have a qualitative interpretation by simply observing what the histograms could tell us, or use a quantitative measure such as the median. For the purpose of our questions, a qualitative interpretation is sufficient. In the lower noise case, all histograms show the same peak and the true value (solid black line in Figure 4.9) lies within the confidence range of one standard deviation of the data. On the contrary, when higher noise is present, the histograms' peaks do not align with each other and the true value is not within the confidence region for all of them. In a situation where more data are available, the recovered histograms are narrower. More specifically, when 5 shots were used the range of the histograms reduced to ±10% of its current range.

Therefore, we draw the following conclusions. The background model uncertainty has an impact on the 4D uncertainty estimation and it needs to be accounted for. In the presence of reasonable to low noise levels, all model cases converge to a meaningful distribution with a similar high peak, indicating that the single model assumption is valid. In the high–noise scenario where the noise completely masks the 4D change, the simple model assumption is no longer accurate. In this case, an alternative strategy might need to be adopted or pre-processing of the data to reduce the noise level.

4.6 Multiple Degrees of Freedom

Thus far, we have characterized the 4D uncertainty only in terms of velocity change. In time-lapse imaging, however, we are not only interested in the magnitude of a change but also in the extent of that change. In this section we want to characterize the 4D uncertainty in terms of both velocity and shape. We cannot do this for every pixel, even in this small local domain, because the model space would become too large; it is well known that faster convergence is possible with fewer unknowns (i.e. smaller dimensionality) [130] [106]. Adaptive Metropolis Hastings algorithms can generally perform well up to 200 degrees of freedom [51], while in velocity model building [37] could only go up to 41 degrees of freedom. It is logical then to try to reduce the dimensionality of the problem. The most straightforward parameterization to reduce dimensionality is to assume that the Earth can be described by a stack of layers, which could be either horizontal (i.e. [90], [59]) or with variable topography (i.e. [30], [37]). Even though these types of parameterizations can capture essential features of the subsurface, they are not able to represent small scale changes within layers (i.e. reservoir). For this purpose more advanced parameterizations such the discrete waveform transform [119] or the discrete cosine transform [78] may be more appropriate. As suggested in [68], we choose the discrete cosine transform (DCT) to parameterize the 4D model within the local domain. The reason we choose DCT is because it is a well–established technique that is widely used in image and signal processing to compress images [62] [48]. However we do not claim that this method is superior than others nor we expect that using a different method of compression will significantly effect the results. The DCT transformation can express a model in terms of a new set of parameters, with only a small subset of these parameters required to obtain a good approximation of the reconstructed model. This means fewer unknowns and better–posed inverse problems.

4.6.1 Discrete Cosine Transform (DCT)

Fourier based transforms use sinusoidal functions to represent spectral components of an input signal [16]. The DCT is a linear transformation that transforms an n-length vector of amplitudes to an n-length vector containing the coefficients of n different cosine functions [2]. In other words, it decomposes a signal into cosine functions. Its energy compaction efficiency (the ability to concentrate most of the input's signal energy to a few frequency coefficients) is greater than any other transformation. Further advantages of the DCT are that it can be computed in both 1D and 2D, and it is to date the most widely used transform in image and video compression standards [163] [75]. The DCT matrix is typically a non-symmetric matrix and the transformation is orthogonal. The 2D DCT coefficients for an m-by-n matrix for the k_{th} and l_{th} degree are:

$$DCT_{kl} = \alpha_k \alpha_l \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \frac{\cos \pi(2m+1)k}{2M} \frac{\cos \pi(2n+1)l}{2N}, \tag{4.38}$$

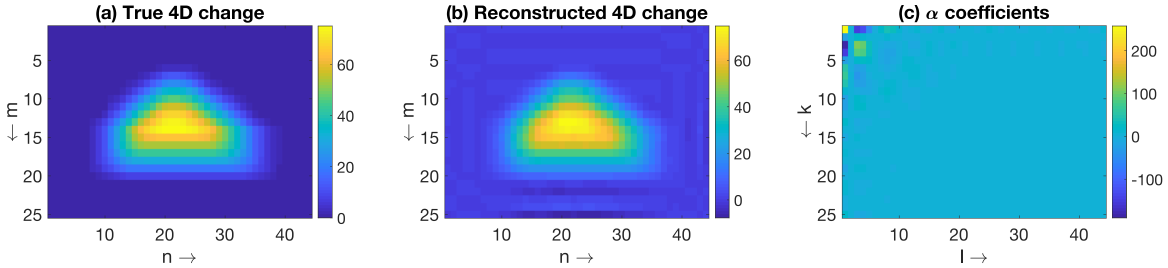

Figure 4.10: (a) The true time-lapse change with smoothed edges. (b) The reconstructed time-lapse change using the first 300 largest singular values. (c) $\vec{\alpha}$ coefficients.

where

$$\alpha_k = \begin{cases} \frac{1}{\sqrt{M}}, & k = 0 \\ \sqrt{\frac{2}{M}}, & 1 \leq k \leq M - 1, \end{cases} \tag{4.39}$$

and

$$\alpha_l = \begin{cases} \frac{1}{\sqrt{N}}, & l = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq l \leq M - 1. \end{cases} \tag{4.40}$$

The values of α_k and α_l act as normalizing constants. Here, we choose to normalize each component of the DCT matrix so that they have the same energy via:

$$DCT_{kl} = \frac{DCT_{kl}}{||D\vec{C}T_{kl}||} \tag{4.41}$$

Typically, this DCT matrix is then multiplied by an image A_{mn} to obtain the transformed image. Although we reconstruct the model during the MCMC process, we analyze convergence and interpret the results directly with the coefficients.

4.6.2 Numerical Example with 20 DOF

We now explain how we set up our numerical example. We take the 4D anomaly used in the single parameter estimation and we apply a moving average smoothing via convolution with a boxcar function to smooth the sharp edges of the anomaly (Figure 4.10). In 1D this is expressed as

$$\theta_n = \sum_{m=-M}^{M} |\phi|_{n-m} b_m, \tag{4.42}$$

where θ_n is the value of the 4D change at sample n, $|\phi|_{n-m}$ is the absolute value of the velocity model at sample $n - m$, and b_m is the value of the smoothing function at sample m. We want to emphasize that the reason we assume that the true time–lapse change is smooth, is simply because this scenario is closer to what we expect during real production.

The local domain here is exactly the same as the one in Figure 4.1, and hence $Nx_{sub} = 44$ and $Nz_{sub} = 25$. We then generate 1100 DCT matrices ($Nz_{sub} * Nx_{sub}$=1100) using Equation 4.38. Each of these DCT matrices has $m = Nz_{sub}$ rows and $n = Nx_{sub}$ columns. We then generate what we refer to as the Φ matrix by

$$\Phi = [DCT_1(:) \quad DCT_2(:) \quad ... \quad DCT_{1100}(:)]. \tag{4.43}$$

where each column of Φ is a vectorized DCT matrix. By doing so the Φ matrix is orthogonal. The time-lapse change $\delta\mathbf{m}$ is then decomposed with the DCT transformation to

$$\delta\mathbf{m} = \Phi\vec{\alpha} \tag{4.44}$$

where $\vec{\alpha}$ are the coefficients used to generate the DCT transformation.

To generate the true time lapse change, we first need to solve an inverse problem to recover all $\vec{\alpha}$ coefficients. We notice that the singular values of the Φ matrix decay fast after the first 300, which means that the Φ matrix is low-rank (Figure 4.10). Therefore, we perform a singular value decomposition (SVD) using only the first 300 largest singular values. Figure 4.10 shows the recovered time-lapse change using 300 $\vec{\alpha}$ coefficients. We are interested in setting up a Bayesian inversion in which we recover the $\vec{\alpha}$ coefficients. Trying to recover all coefficients, though, means that we have to solve a problem with 300 DoF, a number too high for convergence.

To reduce the dimensionality of the problem, we choose only a subset of the $\vec{\alpha}$ coefficients. If we plot $\vec{\alpha}$ as a function of the coefficients k and l we see that most of them are concentrated in the upper left corner (Figure 4.10, (c)). This is due to the energy compaction efficiency of the DCT. This visualization helps us identify which coefficients are essential to reconstruct the time-lapse change. Choosing the right number of coefficients for an accurate 4D reconstruction can be difficult; this number should be small enough for fast convergence but also big enough to sufficiently describe the data. In addition, in order to ensure consistency through the numerical examples, we choose the same simulation parameters as in the previous sections; a single shot and a single frequency of 8 Hz. In order to assure a meaningful time–lapse reconstruction, the DCT matrices have to comply with the minimum vertical and horizontal resolution for that frequency. For example, if we take 30 of the $\vec{\alpha}$ coefficients (30 DoF), the higher frequency coefficients offer a resolution below the minimum vertical resolution. This means that the thickness of the time–lapse will not be resolved. This issue could be mitigated by moving to higher frequencies. However since we choose to simulate all examples at 8 Hz, we choose a maximum of 20 DoF. At each iteration of the MCMC algorithm, a new set of $\vec{\alpha}$ coefficients are drawn. This set of coefficients are multiplied by their respective columns from the Φ matrix to generate a new proposed time–lapse change. Additionally, we add a smooth taper at the edges of the local domain. We do this so that we can avoid any artifacts that are produced because of the hard edge of the change when this is added to generate the monitor model. To ensure faster convergence we change our sampler to an Adaptive Metropolis Hastings (pseudocode in Algorithm 2) and we recover each of the $\vec{\alpha}$ coefficients. We first run the Non-Adaptive Metropolis Hastings for $N_c = 1000$ with a fixed step size, and then we turn on the tuned step size using the history of the proposals.

We run eight different Markov Chains for 100,000 iterations and we discard the first half to drop the dependency on the starting model. The acceptance rate for all chains ranges between 13% and 18%, which means that from 100,000 models only 13,000-18,000 were accepted. Figure 4.11 shows the histograms of each of the coefficients for all eight chains. All histograms seem to have converged with a mean in the confidence range of the true (green line). It is worth mentioning that for the same number of iterations the Non-Adaptive Metropolis Hastings algorithm did not converge. To determine an empirical relationship between the coefficients we create bivariate histogram plots between $\alpha_1 - \alpha_2$, $\alpha_2 - \alpha_3$, and so on. Of course, one could choose a different arrangement of pairs for these plots. Bivariate or 3D histograms are a combination of two histograms that show the dependencies between two variables by measuring their co-occurrences. Figure 4.12 top left panel shows a bivariate histogram between $\alpha_1 - \alpha_2$. Because interpretation could be challenging with this visualization, we show the top view of these histograms instead. The plots are from one of the Markov Chains and the colours of these plots relate to the frequency of observations. The majority of these plots (16 out of 19) show a diagonal concentration, which means that these coefficients are dependent on one another. The rest of the coefficients seem to follow a slightly blockier pattern, meaning that these coefficients are independent of one another. Given their overall relationship, we cannot measure convergence of these coefficients using the $\hat{\mathcal{R}}$ criteria [47]. For a meaningful use of the $\hat{\mathcal{R}}$ criteria, quantities/variables need to be independent.

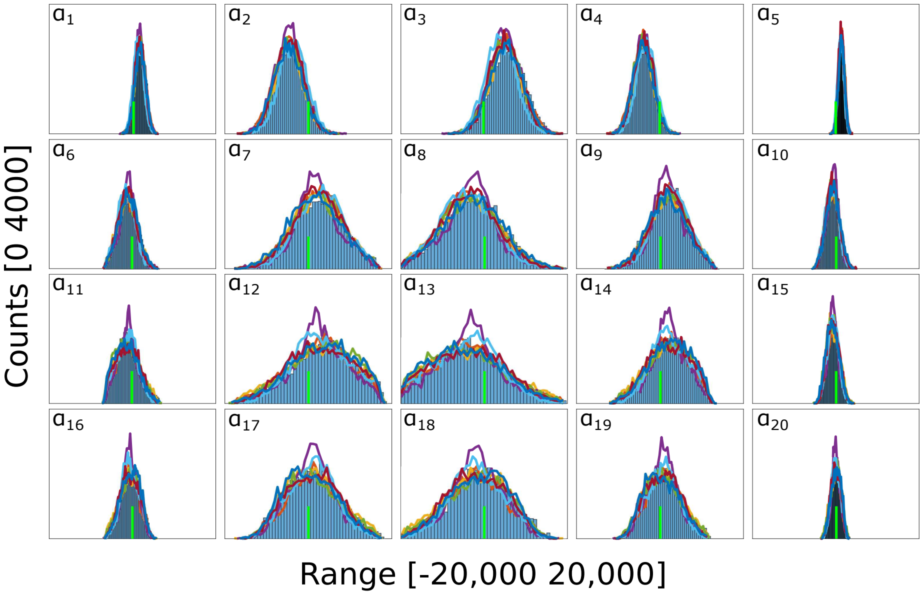

Figure 4.11: Histograms of the recovered α coefficients for the eight Markov chains. The green line represents the true value for each coefficient.

Therefore, for a meaningful interpretation, we look at histograms and measure convergence over quantities of interest. Typically, 4D changes are characterized by their magnitude and their extent. Thus, we define the following three quantities of interest:

1. vertical extent of the anomaly,
2. horizontal extent of the anomaly,
3. average velocity of the anomaly.

To perform the vertical extent calculation, we extract a vertical line passing through the middle of the anomaly (Figure 4.13, (a)). This extracted line looks similar to a bell curve, and we want to measure to width of that curve. To do so, we use MATLAB's function *"findpeaks"*, that finds local maxima in a given vector by searching for samples larger than its two neighbouring samples or equal to infinity.

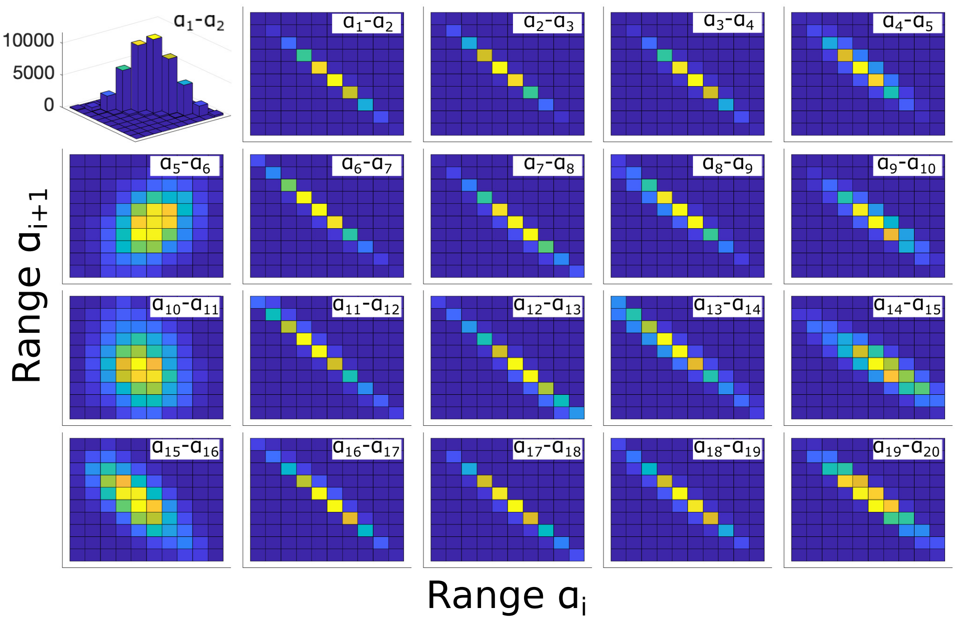

Figure 4.12: Bivariate histogram plots showing the correlation between the coefficients for one of the Markov Chains. The top left panel shows a bivariate histogram between $\alpha_1 - \alpha_2$. However, such a visualization is difficult, therefore we show the top view of these histograms. The brighter colours represent a higher number of observations. Tick marks are not shown just so each subplot looks clear. The axes and colorbar are not the same with each other, since each coefficient has a slightly different distribution range.

First the highest peak of the signal and its prominence are obtained, and then the width of the signal is calculated as the distance between two points located at the height of half the prominence. We do this for all recovered 4D models for each of the Markov Chains. Figure 4.13 (b) shows all recovered histograms with the green line representing the value of the vertical extent in the true model (Figure 4.13, (a)). Histograms appear Gaussian and all chains seem to have converged to the same distribution with the true being within the confidence range of one standard deviation. For reference, the dotted black line represents the vertical extent calculated using only 12 DoF. Fewer DoF lead to an overestimation of this quantity, which can also be seen at the maximum likelihood model extracted from one of the chains (Figure 4.14). We follow a similar workflow to calculate the horizontal extent of the anomaly. This quantity seems to be more stable between the 12 and 20 DoF, however the 20 DoF seem to be getting a mean more aligned with the true. To compute the average velocity we define a rectangle in the centre of the anomaly, in which we calculate the average of all values included. Figure 4.13 (a) shows the true anomaly with the white rectangle representing the area of calculation. Panel (d) of the same figure shows the recovered distributions from all Markov Chains. This quantity of interest is very well described by 20 DoF with almost perfect recovery of the true average velocity (+0.6 m/s) compared to a small overestimation from the 12 DoF (+5 m/s). Of course the maximum likelihood model (Figure 4.14, (b)) doesn't perfectly match the true time–lapse anomaly, however it sufficiently captures the three quantities of interest.

We can further assess convergence over the three quantities of interest using the $\hat{\mathcal{R}}$ criteria. In this criteria we compare the average variance of the quantity of interest within each chain (var_m) to the variance of all of the chains together (var_{mix}) by

$$\hat{\mathcal{R}} = \sqrt{\frac{mean(var_m)}{var_{mix}}}. \tag{4.45}$$

Figure 4.15 shows $\hat{R}$ as a function of iterations for the average time lapse velocity in the centre of the anomaly, and the vertical and horizontal extent of the anomaly. Generally convergence is declared if $\hat{R}$ is less than 1.1, and therefore we conclude that our chains have converged for the features of interest. Note that this convergence does not occur until close to 50,000 models have been sampled meaning that using a method in which you can only sample a few thousand models will not give a stable, converged solution.

4.7 Multiple Frequencies and Local Minima

FWI gradient–based minimization techniques (typically formulated as least squares local optimization) are susceptible to local minima in the presence of high noise or when a good starting model is not available [161] [21]. When complex and realistic velocity models are used it is possible that different frequencies will converge to different local minima [97]. To mitigate that issue hierarchical approaches have been proposed in the frequency domain moving from lower to higher frequencies [115] [23] [138].

On the other hand, global optimization techniques, such as the one used in this study, are less likely to be trapped in local minima. This property can make these techniques more desirable compared to local optimization techniques despite their computational expense. The choice of initial model in global optimization methods is usually random. The final distribution is independent of the initial guess when the

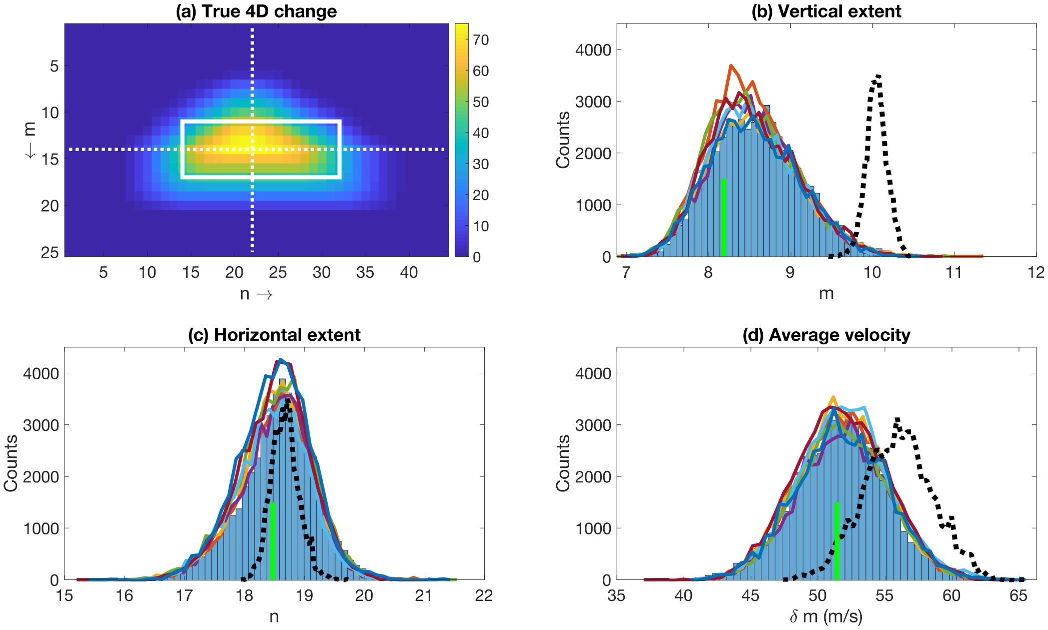

Figure 4.13: (a) The true time-lapse anomaly with the two dotted white lines indicating the extracted lines used for the vertical and horizontal extent calculations. The white box represents the area in which we computed the average velocity of the time-lapse change. (b) Recovered histograms for anomaly's vertical extent from the eight Markov Chains. (c) Recovered histograms for anomaly's horizontal extent from the eight Markov Chains. (d) Recovered histograms for anomaly's average velocity. In (b), (c), (d) the dotted black line represents the histogram from the 12 DoF while the solid green line is the true average velocity computed from the model in (a).

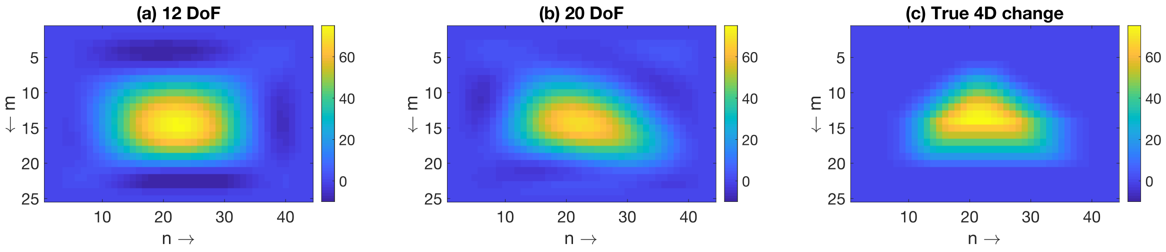

Figure 4.14: The maximum likelihood models for the 12 DoF example (a) and the 20 DoF (b) versus the true time–lapse change (c).

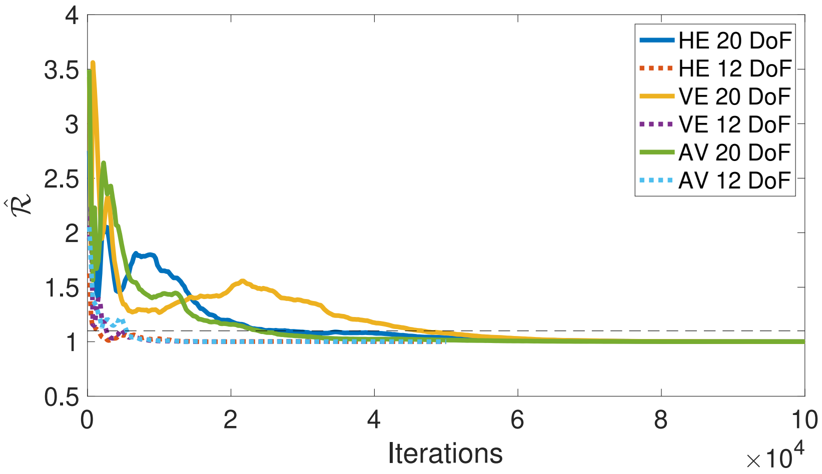

Figure 4.15: $\hat{\mathcal{R}}$ for the three quantities of interest for both the 20 and 12 DoFs.

algorithm converges.

Thus far we have considered only a single frequency, so it is reasonable to ask whether the issues with different frequencies will apply in the context of global optimization. We examine this in both single and 40 degrees of freedom examples. We begin with the single degree of freedom and use the noise level in the numerical example of that section as the lower noise case. We generate synthetic data for the frequencies of 3.0, 4.0, 5.0, 6.5, 8.0, 10.0 Hz. For the same frequency batch, we generate a higher noise case; Figure 4.16 shows a comparison of the data for the high- and low-noise cases at a frequency of 3 Hz. For both cases we evaluate the likelihood function (equation 5.5) for a range of perturbations —from 0 to 150 m/s with an interval of 10 m/s— for all frequencies in the batch. Figure 4.17 displays all results. We see that for the lower noise case all frequencies converge to the global minimum. This is not surprising considering the simplicity of the problem. The background model is the true Marmousi, we recover a single parameter (single DOF), and the noise level is relatively low. However, when the noise level gets higher (left panel in Figure 4.17) the different frequencies start deviating from the global minimum and converge to slightly different local minima. All of these local minima are near the global minimum most likely due to the simplicity of the single degree of freedom example. Perhaps, in a situation like this a hierarchical Bayesian approach similar to the one in local optimization might prove beneficial. One could, for example, evaluate the likelihood of a proposed model at a lower frequency, and if the model is accepted then move to higher frequencies. Such an approach could be characterized as a multi-fidelity framework and is outside the scope of this study. An alternative approach — and more straightforward to implement in an MCMC regime — is to compute all frequencies simultaneously and evaluate the likelihood function for the stacked data. [97] observe that stacking all frequencies together increases the chance of converging to a global minimum with a descent method (Figure 3 of their paper). If we have observed data $(\delta\mathbf{d}_1, \delta\mathbf{d}_2, ...\delta\mathbf{d}_N)$ at N frequencies (here $N = 6$) and the measurement of noise of each observation is $\Sigma_1, \Sigma_2, ...\Sigma_N$, then the likelihood function becomes

$$\mathcal{L}(\delta\mathbf{m}) \propto exp\left(-\frac{1}{2}\left[\mathcal{X}^1\mathcal{X}^2...\mathcal{X}^N\right]^T \mathbf{\Sigma}^{-1}\left[\mathcal{X}^1\mathcal{X}^2...\mathcal{X}^N\right]\right), \tag{4.46}$$

where $\mathcal{X} = F(\delta\mathbf{m}) - \delta\mathbf{d}$ with the superscript representing the different frequencies and $\mathbf{\Sigma}$ is the block diagonal matrix of $\Sigma_1, \Sigma_2, ...\Sigma_N$, i.e.

$$\mathbf{\Sigma} = \begin{bmatrix} \Sigma_1 & 0 & 0 & \dots \\ 0 & \Sigma_2 & 0 & \dots \\ \vdots & & \ddots & \\ 0 & \dots & 0 & \Sigma_N \end{bmatrix}. \tag{4.47}$$

Under this construction the noise is uncorrelated across frequencies. The right panel in Figure 4.17 shows this likelihood evaluation for the perturbation ranges of $\delta\mathbf{m} = 0 : 10 : 150 m/s$ in the presence of high noise. We see that unlike the situation where each

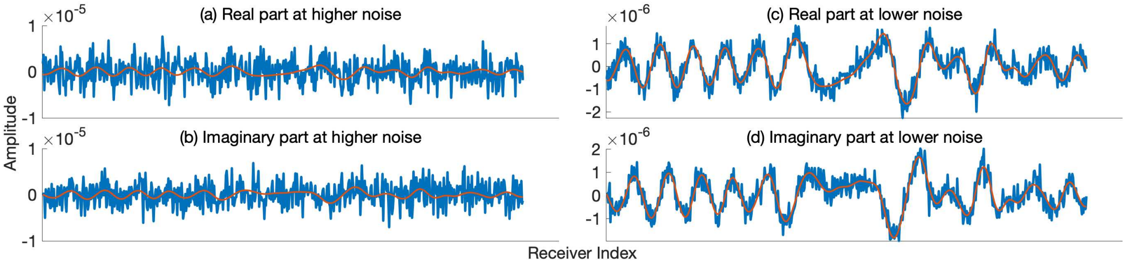

Figure 4.16: The data residuals δd at the frequency of 3 Hz. Panels (a) and (b) show the real and imaginary parts in the presence of higher noise, whereas panels (c) and (d) show the real and imaginary parts at lower noise. The blue and red lines represent the noisy and noiseless δd respectively.

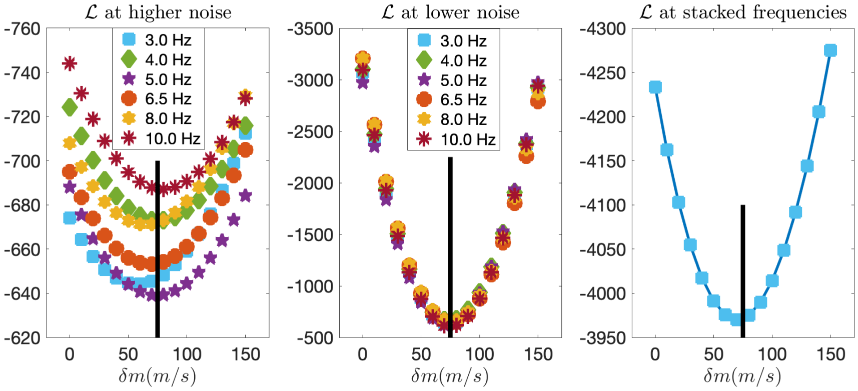

Figure 4.17: Likelihood evaluation for the perturbations $\delta\mathbf{m} = 0 : 10 : 150 m/s$ at different noise levels and frequencies. Left panel: likelihood computed for each frequency separately in the presence of high noise. Middle Panel: likelihood computed for each frequency separately in the presence of low noise. Right panel: likelihood computed for all frequencies at the same time in the presence of high noise.

frequency alone was trapped in a local minimum (left panel in Figure 4.17), evaluating all frequencies together we can converge to the basin of the global minimum. Based on the findings of Figure 4.17 we proceed with a single frequency and single shot for the rest of our paper.

To evaluate the presence of different local minima at different frequencies for the 20 DoF is somewhat more complicated than the single DoF. If, for example, we perturb by an amount all the DCT coefficients (i.e. perturb all coefficients by 10%,

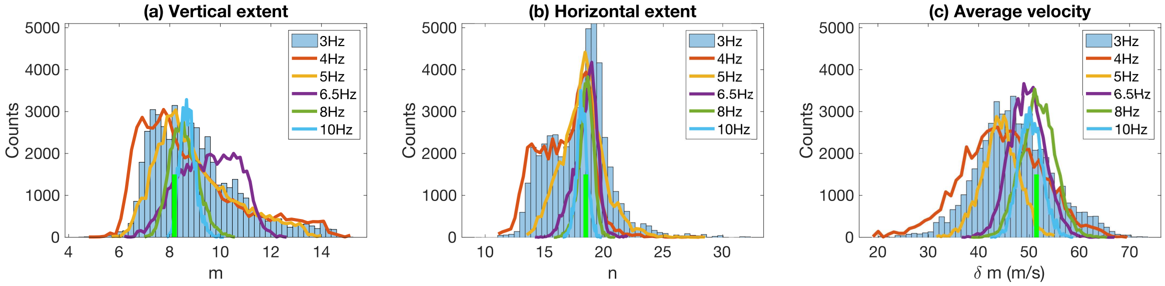

Figure 4.18: Recovered distributions of the three quantities of interest at different frequencies.

20%, and so on) and evaluate the likelihood function the result will be biased. This is because perturbing all coefficients by the same amount is equivalent to a single degree problem. To avoid this, we evaluate the presence of local minima, in terms of the mean of the distribution for the quantities of interest. To do so, we run the MCMC algorithm for one of the chains at the same frequencies used in the single DoF evaluation. The noise level is the same as the one in Section 4.6, with SNR = 1.95. Figure 4.18 shows the resulting distributions for the three quantities of interest. We observe different behaviour (either in terms of distribution shape or in terms of mean) for each frequency for all three quantities. For the vertical extent we see that frequencies 8 Hz and 10 Hz retrieve similar distributions, whereas lower frequencies seem to be more variable. This is likely due to the lower vertical resolution at these frequencies. For the horizontal extent, most frequencies show a similar behaviour and mean, however frequencies of 3 Hz and 4 Hz appear affected by limitations of horizontal resolution. The average velocity displays a Gaussian distribution for all frequencies, however each of them leads to a different local minima. We anticipate that in the presence of higher noise — similarly to the single DoF — the obtained means for each quantity might vary more significantly along the frequencies.

4.8 Validity of our assumptions

In this section we discuss a number of the assumptions behind our methodology and show how they influence our final results.

4.8.1 Gaussian assumption on the noise distribution

Assuming that the measurement noise is uncorrelated Gaussian with a perfectly known covariance matrix is a strong assumption. On the other hand, in field data sets the noise is typically coherent and spatially correlated. To investigate the performance of our methodology in such situation, we generate a correlated covariance matrix that resembles a toeplitz structure using the Matlab function `convmtx`. Figure 4.19(d) shows the noisy time–lapse data in the presence of correlated noise compared to the time–lapse data in the presence of uncorrelated noise. Both type of noises have the same energy. Since we change one of our assumptions, we expect that the resulting histograms will also change. Figure 4.19(a-c) shows the resulting histograms, which now appear to be broader compared to the ones from the uncorrelated noise scenario. It is very interesting to observe that the vertical extent is the most affected quantity, which the each noisy type converging to a different median. However, the horizontal extent and the average velocity are less affected with the medians of the two noise types in a good agreement. It is important to mention that in a situation where the noise is highly correlated, additional pre-processing steps may need to be applied before one can use this framework. When the covariance matrix is unknown, we could follow the standard practise of assuming a scaled identity matrix, with the scaling typically coming by estimating and computing the variance of the noise from the signal to noise ratio. Following the work of [15] and [1] future efforts will focus on investigating the

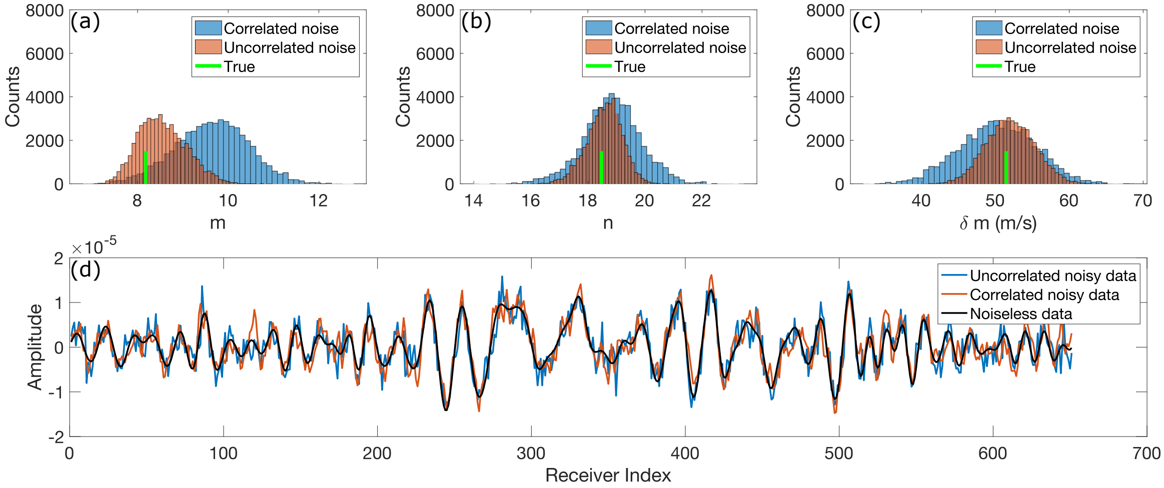

Figure 4.19: (a) Recovered histograms for the anomaly's vertical extent from one of the Markov Chains in the presence of correlated and uncorrelated noise. (b) Recovered histograms for the anomaly's horizontal extent from one of the Markov Chains in the presence of correlated and uncorrelated noise. (c) Recovered histograms for the anomaly's average velocity from one of the Markov Chains in the presence of correlated and uncorrelated noise. (d) The noisy time–lapse data for both correlated and uncorrelated noise compared to the noiseless ones.

potential of incorporating the covariance matrix into the unknown parameters.

4.8.2 The 2D acoustic approximation

All numerical examples here rely on the 2D acoustic approximation of the wave equation. We choose the acoustic wave equation because it is easier to implement compared to the elastic wave equation, while also allowing for fewer degrees of freedom. Having established an efficient acoustic UQ framework, we could consider two potential extensions. Following the work of [70] we could extend it to a 3D acoustic case, where we still can exploit the structure of the 4D problem by using a local domain, while also using the low rank approximation of the Green's functions. Using a low–rank approximation of the Green's functions, significantly decreases the cost of computing the background full–domain Green's functions, which can be prohibitively expensive when real data and large acquisitions are used. Alternatively, we could continue to work in 2D but shift our emphasis to more realistic estimations of the time–lapse change. Using more realistic physics models such as the elastic- or visco- or poro–elastic wave equation we can better image fluid flows in the subsurface. This could be done, for example, using a coupled acoutic–elastic local solver as proposed by [168] and [104].

4.8.3 Design of the hierarchical approach

In Section 4.7 we showed that different frequencies retrieve different posterior distributions. This is a clear sign of presence of local minima. The common practise in frequency domain gradient–based FWI is to start inverting for lower frequencies and subsequently invert for higher frequencies ([115], [23], [138]). A similar hierarchical approach could be employed here too. However, having to solve the wave equation hundreds or thousands of times for multiple frequencies can be a stumbling block, especially when typically $80 - 90\%$ of those models are rejected due to the low acceptance rate of the algorithm [3]. The best practise would be the implementation of a multi–level or multi-fidelity approach (i.e. [142]). The algorithm could sample the posterior distribution at lower frequency and accept or reject the proposed $\delta\mathbf{m}$ based on the Metropolis–Hastings criterion. We can think this low frequency evaluation as a cycle–skipping filter to which proposed models can be considered for further evaluation. If the proposed $\delta\mathbf{m}$ model is accepted, then we can sequentially move to evaluating the likelihood function at higher frequencies.

4.8.4 Tuning of the DCT parameterization

A huge advantage of working with time–lapse data is that they are repeated versions of the same experiment as a reservoir is actively producing. This means that the surrounding geology is well known, and the results of interest are focused to the region of the reservoir, making this an ideal situation for our proposed uncertainty analysis. In a typical field-data situation we would not have access to the true time-lapse change however. Here we explain how one could still use the DCT representation in this case. Typically, before any time–lapse analysis, the reservoir size and velocity response (prior to production) are well known. The first step would be to design a local domain that contains the reservoir. Knowing the exact size of the local domain, allows us to design the Φ matrix, which similar to the study here, will remain fixed throughout the whole process. Knowing the type of hydrocarbon production process that is used in the field, such as the type of fluid injection, allows for estimation of the changes to pore fluids, pore pressure, and effective stress in the reservoir. One can then use Gassmann's equations [43] for instance, to infer seismic velocity changes from different pore fluid saturations in a reservoir. Having a tentative $\delta\mathbf{m}$ at hand as well as a Φ matrix, one can easily follow the steps described in Section 3 to retrieve the $\vec{\alpha}$. When looking at Figure 4.20, we notice that between a 5-by-5 block (25 DoF) and a 6-by-6 (36 DoF) there is no significant change in the reconstructed time–lapse model, meaning that either of these choices for the number of DoF can accurately approximate a time–lapse change similar to the one used in this study. Obviously, using a 10-by-10 block (100 DoF) significantly improves the approximation, however the number of DoF is far larger, posing a challenge for the convergence of the algorithm.

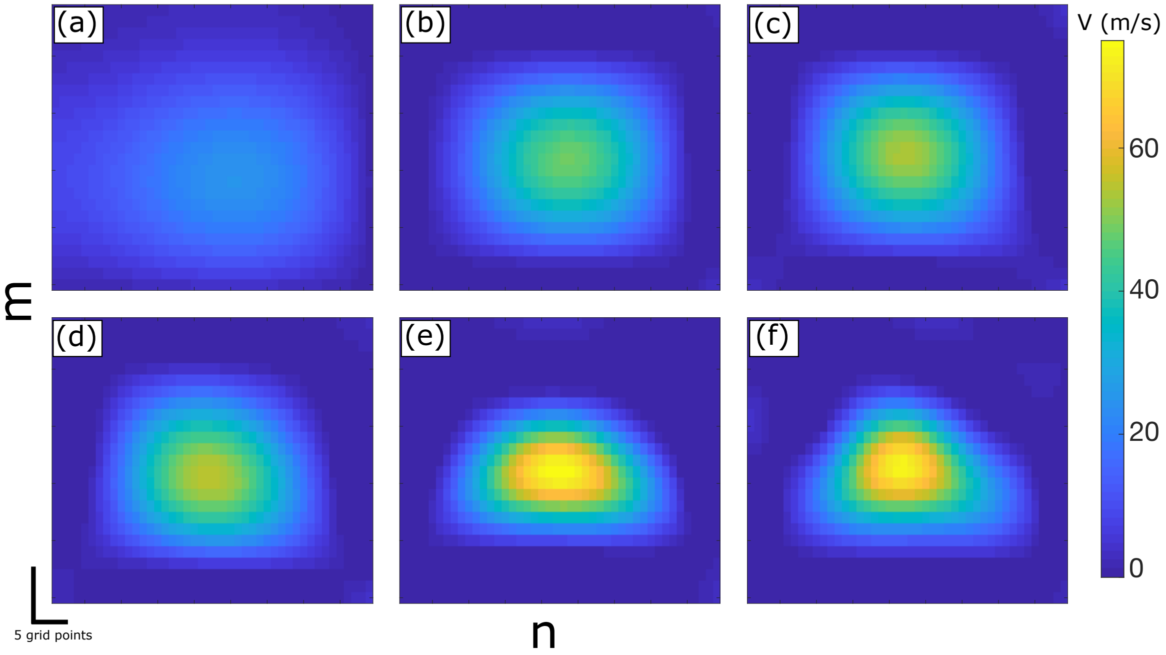

Figure 4.20: Time–lapse model reconstructions ($\delta\mathbf{m} = \Phi\vec{\alpha}$) as a function of the number of DCT coefficients. We pick different block sizes from the upper left corner of Figure 4.10c: (a) 3-by-3 block, (b) 4-by-4 block, (c) 5-by-5 block, (d) 6-by-6 block, (e) 7-by-7 block, and (f) 10-by-10 block.

4.8.5 Number of iterations in the burn–in process

It is well known that a Markov chain has “no memory”, meaning that any future state calculation solely relies on the current state and how it got to this state. Thus, if we discard a number of samples, the future calculations will be no different than if we have started from the last sampled we discarded. The so–called burn–in period typically refers to these number of samples that are discarded. We do this because after the burn–in period, the chain will be in a high probability region and hence more accurately sample the posterior distribution of interest. If, for instance, we know the region of high probability and we start from there, our chain would be immediately burned–in. Following the standard practise described in [19] and [37], we discard the first half of the samples in the chain as the burn–in samples. This means that we consider only the second half of the chain to represent the posterior distribution. We further investigate the number of iterations in the burn–in period by discarding the first 25%, 50%, and 75%, and looking at the histograms of the Quantities of Interest. We do not observe any significant difference, and all histograms converge to the same distributions for all three QoI. We also examine whether the samples for the three QoI after the burn–in period are correlated with each other; correlated samples typically lead to slower convergence, and risk deriving biased pdf estimations. We analyze the autocorrelation function of the sampled models as a function of lag and plotted the histograms of QoI using different lag values (the so–called “thinning” process), however we did not observe any changes in the retrieved distributions. This allows us to draw the conclusion that samples in the chain are not correlated, and we can use all samples after the burn–in period.

4.9 Discussion

Time-lapse uncertainty quantification is critical for decisions based upon these time-lapse changes. A successful uncertainty quantification relies not only on the mathematical models that express the problem, but also on the numerical and computational feasibility of the algorithm. For a Bayesian inversion to be meaningful, a fast forward solver is required so that the ten or hundreds of millions of models could be assessed within an acceptable time frame. Here, our choice of fast forward solver is a local acoustic solver. Such a solver is particularly ideal for time-lapse imaging because we can perform a wavefield solve only within a subset of our model (i.e. area around reservoir) and get exactly the same wavefield (for that subset) as would have been obtained with a wavefield solve in the full model. For the numerical setup of this study (Figure 4.1) and for a single shot and single frequency one wavefield solve in the local domain takes approximately 0.087 seconds. Therefore 20,000 iterations take approximately 30 minutes. By contrast, one wavefield solve in the full domain takes approximately 3 seconds, which in 20,000 iterations will be approximately 16.67 hours. This difference becomes significant when more iterations are needed or more shots and frequencies are included. We perform all computations on a MacBook Pro with a 2.6 GHz processor and 32 GB memory.

Deep neural networks are favourable for approximating physical phenomena that are described by underlying nonlinear physics, such as the wave equation. These techniques have been recently applied in seismic imaging by replacing the standard finite difference solution of the wave equation (e.g. [144]). [96] approximate seismic responses at multiple receivers for a horizontally layered medium with this theory. Even though this approach could be very promising, it might not be ideal (computationally) for the problem we study here with this particular size of local solver. They observe an order of magnitude reduction in computation time; 1sec for finite difference solution versus 0.1sec for deep neural network when there is no parallelization. This time does not include the training process for the neural network, which at times can be quite intense computationally. On the contrary, the size of local solver here led to a reduction of approximately one and a half order of magnitudes. A potential combination of the two techniques might lead to more significant computational time reduction. Furthermore, assuming that there is a linear relationship between models and data (which is commonly done in time–lapse problems) opens a possibility for combining the local solver with supervised algorithms such as Support Vector Regression (SVR) [156]. The idea would be that we can train our algorithm on current models and data, and then use a regressor to predict what the data could be for a given model or vice versa.

We showed that using an incorrect background model has an imprint on the retrieved time-lapse histograms, as also noted by [169]. When we tested six different background models against the single DoF example we observed a huge increase in the range of the recovered histograms. This is to be expected (to some extent) considering that we use a single shot and a single frequency. Traditional FWI techniques require multiple shots and frequencies to converge to an assumed global minimum. When we repeated the same test using 5 shots and a single frequency, we noticed a 10% reduction in the range of the histograms. In the case of multiple DoF, things could be more complicated since we essentially allow the algorithm to draw more variable models. To obtain meaningful quantities of interest we anticipate that more data will be needed; this means a multi–shot multi–frequency approach. Such an approach will be an easy extension to our current framework, however it will increase significantly the computational cost.

An MCMC algorithm can only sufficiently explore model space for relatively few DoF. The model parameterization needs to be in a way that also captures all important features that best represent the model. In this work, we parameterize our model using a Discrete Cosine Transformation (DCT). We choose DCT because it is easy to employ and we can describe our models using a set of cosine coefficients. Due to its energy compaction efficiency, we can reduce the degrees of freedom by using only a subset of these coefficients to accurately describe our models. Here, using 20 DoF was sufficient enough to recover the three chosen features of interest. There are many other ways to do this compression, which are thoroughly explored in the image processing literature [120] [117], however we do not expect that using a different method of compression will significantly effect the results.

4.10 Conclusions

We propose a local acoustic solver for a fast 4D Bayesian inversion. Calculating the full posterior pixel by pixel, even in a small local subdomain, would be both computationally challenging and potentially difficult to interpret. We have created a framework that calculates time-lapse uncertainty quantification in a targeted way that is computationally feasible. We show that our framework is robust for both single and multiple degrees of freedom examples. Metropolis Hastings is typically used to ground truth probability distributions. This is particularly useful in seismic imaging, where the distributions of most structures of interest are not well known. Now that this is done, our future work will focus on more advanced uncertainty quantification methods, such as Hamiltonian Monte Carlo (HMC), which can handle higher dimensions while providing faster convergence.

4.11 Acknowledgments

This work is supported by Chevron and with grants from the Natural Sciences and Engineering Research Council of Canada Industrial Research Chair Program and the InnovateNL and by the Hibernia Management and Development Corporation. In addition, we would like to thank Jean Virieux for interesting discussions.

Chapter 5

An introduction to Hamiltonian Monte Carlo for time–lapse seismic inversion and uncertainty quantification

5.1 Abstract

Uncertainty quantification is an important aspect of time–lapse imaging and is typically done using Bayesian inference. Traditional random–walk sampling methods are slow to converge and they fail to efficiently explore a high dimensional space. We propose using a local acoustic solver for an efficient 4D Hamiltonian Monte Carlo inversion. Using a local acoustic solver offers the advantage of quick and local gradient computations. For a meaningful HMC implementation the parameters of the Leapfrog simulator need to be tuned. Here, we use the strategy of trial and error to tune the

Leapfrog steps L and the method of dual averaging to tune the Leapfrog stepsize ϵ. Our simple numerical illustrations demonstrate the robustness of the method over the Metropolis–Hastings algorithm, and sets up the path towards a higher dimensional implementation.

5.2 Introduction

One of the critical aspects of reservoir monitoring is the analysis of time-lapse or 4D seismic data sets. These data sets are repeated surveys of the same location as a reservoir is undergoing production to characterize changes in fluid properties [79]. The first survey acquired is called the baseline survey, while all the following surveys are called monitor surveys. Full–Waveform Inversion (FWI) originated in the early 1980's [145] and aims to deliver high–resolution velocity models of the Earth using the entire content of seismic data. For a recent review, readers can refer to [160]. FWI is extended to time–lapse successfully with a variety of frameworks currently available [167] [182] [129] [84] [173] [6] [64].

Due to its inherent non–linearity, when local optimization strategies are used, such as the ones mentioned above, 4D FWI has two challenges. First, for a given initial model, FWI delivers a single model from the range of possible models that could equally describe the data. Therefore, it is not straightforward to draw any meaningful estimate of the uncertainty associated with that model. Second, FWI strongly depends on the initial model; for accurate model reconstruction, the predicted data for the starting model needs to be within half a wavelength of the observed data. In principle, both of these challenges can be handled by Monte Carlo sampling [95]. A fundamental characteristic of MCMC algorithms is that they do not assume the underlying structure of the posterior. [131] provide a helpful review of Monte Carlo methods in geophysical imaging. Solving an inverse problem using Markov Chain Monte Carlo (MCMC) means exploring and characterizing the ensemble of all plausible models. This is typically done using Bayes' theorem [10] and computing a posterior probability density function [147]. Even though this can sound naively simple, it can come with a significant computational burden given the dimension of the problem to be solved and the expense of the forward solver. For instance, for a standard Metropolis–Hastings implementation, one might need to run 10,000-100,000 iterations, which means 10,000 - 100,000 forward solves [37]. The expense of the forward solve is proportional to the size of the model, and hence to the number of Degrees of Freedom (DoF). Here, we overcome the latter issue by exploiting the structure of the 4D problem. Time–lapse data are typically collected over actively producing reservoirs; this means that the surrounding geology is well studied while the changes we are looking for are concentrated around the reservoir. This means that we can use localized wavefield techniques [155] [157] [169] [18] [177] that allow the solution of the forward problem in a smaller subdomain of the subsurface. In this work, we use the frequency–based local solver developed by [169].

Uncertainty quantification is an essential topic of geophysical imaging. Since the early study of [63], work has continued to understand these uncertainties and quantify them. Some of the most recent studies include [119], [37], [179], [140], and [152]. In our recent work [69], we derived a Bayesian formulation of the 4D–FWI problem, and performed random–walk and Adaptive Metropolis–Hastings inversions [51]. Metropolis–Hastings algorithms are typically used to ground–truth probability distributions. This is particularly useful in seismic imaging, where the distributions of most structures of interest are not well known. However, in addition to being slow to converge, these algorithms are also in general limited in the number of DoF that they can handle. This is because as the number of dimensions grow, there are exponentially more possible guesses the algorithm can make, while only a single one that can pass the check (meaning that is accepted). This could be addressed by utilizing reduced parameterization approaches, but there is the potential risk of oversimplifying the model representation. In this context, more advanced uncertainty quantification methods such as Hamiltonian Monte Carlo (HMC) could be beneficial because they can handle higher dimensions while also providing faster convergence. Originating with the title "Hybrid Monte Carlo" [32] in a quantum chromodynamics study, HMC has seen a widespread application in numerous research fields such as neural networks [99] and molecular simulations [33] just to name a few. HMC is particularly favourable in the case where the partial derivatives of the target distribution are easy to compute. In the case of FWI this can be easily achieved using the adjoint state method [109], Hamiltonian Full–Waveform Inversion has only been applied recently [133] [44] [27]. Building upon our earlier work, we employ a 4D–HMC algorithm and illustrate its performance on a simple 4D problem. This opens up the possibility of allowing the whole 4D model in the local domain to be uncertain, meaning more than 1,000 DoF to be considered which is not possible with standard MCMC methods. Even though we assume that the measurement noise is Gaussian, we make no assumptions about the posterior distribution which could be any type, including multi–modal and non–Gaussian. In the following sections, we revisit the theory of the local domain and the 4D problem setup, and explain the concept of HMC. We end with two simple seismic examples and discussion of current and future work.

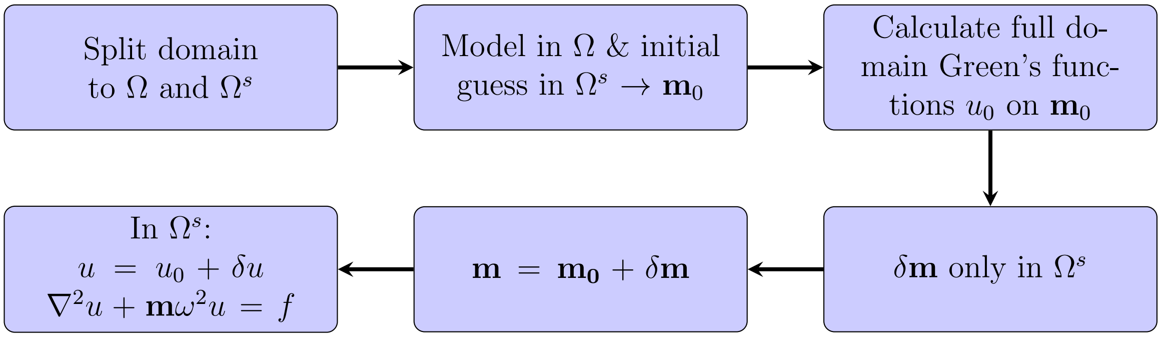
Split domain to Ω and Ω^s
Model in Ω & initial guess in $\Omega^s \to \mathbf{m}_0$
Calculate full domain Green's functions u_0 on $\mathbf{m}_0$
$\delta\mathbf{m}$ only in Ω^s
$\mathbf{m} = \mathbf{m_0} + \delta\mathbf{m}$
In Ω^s:
$u = u_0 + \delta u$
$\nabla^2 u + \mathbf{m}\omega^2 u = f$

Figure 5.1: Steps of local domain solver.

5.3 Theoretical Background

5.3.1 Local Acoustic Solver

Here, we briefly describe the local wave solver; a detailed explanation can be found in [169]. This is a frequency–based acoustic local solver, that accurately computes wavefields within only a subdomain of the region covered by the survey, providing a significant computational savings when compared to standard forward solvers. To successfully apply the local solver, we follow the steps in Figure 5.1. It is important to mention that the background Green's functions computation happens (all steps in the first line of Figure 5.1) only once, and then we can update the model and recompute the wavefield in only the local domain as many times as we need (all steps in the second line of Figure 5.1).

5.3.2 4D Bayesian Inference

A probabilistic inversion approach can be described using Bayes' theorem [10] [146], where a hypothesis (given some evidence) is expressed in terms of a probability distribution, typically called the posterior distribution. In the context of seismic imaging, a hypothesis could be, for example, represented in terms of velocity models ($\mathbf{m}$) given some observed data ($\mathbf{d}$) [146],

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{p(\mathbf{d})}. \tag{5.1}$$

In Equation 5.1, $p(\mathbf{d}|\mathbf{m})$ is the likelihood function calculated by [146]

$$\begin{aligned} \mathcal{L}(\mathbf{m}) &\equiv p(\mathbf{d}|\mathbf{m}) \qquad (5.2)\\ &\propto exp\left[-\frac{1}{2}(F(\mathbf{m})-\mathbf{d})^T\Sigma^{-1}(F(\mathbf{m})-\mathbf{d})\right], \end{aligned}$$

where F is the forward solver and Σ^{-1} is the inverse covariance matrix of the data noise. The variable $p(\mathbf{m})$ is the input prior model distribution and contains any prior information available on the models. The variable $p(\mathbf{d})$ — the so–called evidence — is considered to be a normalization constant [148] and is given by

$$p(\mathbf{d}) = \int p(\mathbf{d}|\mathbf{m})p(\mathbf{m})d\mathbf{m}. \tag{5.3}$$

The evidence ensures that the integral of the left hand side of Equation 5.1 is equal to unity and therefore it is a valid probability distribution.

Since the focus of this paper is on time–lapse seismic application, we are interested in having an expression only in terms of model differences $\delta\mathbf{m}$ and data differences $\delta\mathbf{d}$. Following [67], we can express Bayes' theorem as

$$p(\delta\mathbf{m}|\delta\mathbf{d}) = \frac{p(\delta\mathbf{d}|\delta\mathbf{m})p(\delta\mathbf{m})}{p(\delta\mathbf{d})}, \tag{5.4}$$

and the respective likelihood function as

$$\mathcal{L}(\delta\mathbf{m}) \equiv p(\delta\mathbf{d}|\delta\mathbf{m}) \propto \tag{5.5}$$

$$exp\left[-\frac{1}{2}(F(\delta\mathbf{m}) - \delta\mathbf{d})^T\Sigma^{-1}(F(\delta\mathbf{m}) - \delta\mathbf{d})\right],$$

where Σ is the sum of known covariance matrices from the noise in the baseline and monitor data.

5.3.3 Hamiltonian Monte Carlo (HMC)

Theoretical concept

Hamiltonian Monte Carlo (HMC) explores a target distribution by incorporating information about its differential geometry into the search–proposal process. This means that the algorithm utilizes physical system dynamics (Hamiltonian dynamics) to propose new and distant jumps in the Markov Chain [100]. To better understand this, an excellent example of a comprehensive analogy is described by [13], where one can think of a planet, a gravitational field, and an orbit instead of a mode, a gradient, and a target distribution. Then, the probabilistic challenge of exploring a target distribution turns into the physical challenge of placing a satellite in orbit around a planet. Both these challenges are equivalent to the same mathematical problem, which means they will suffer from the same issues. If we place a satellite at rest in space, it will crash onto the planet due to the gravitational forces. Equivalently, gradient–trajectories on a target distribution can crash onto the mode. To place the satellite in orbit around the planet, we need to add momentum. If the momentum is too little, then the gravitational forces overpower it and eventually lead to the satellite crashing onto the planet. On the other hand, if the momentum is too big, the gravitational forces will not be able to keep the satellite around the planet, and the satellite will eventually drift away in space. To ensure a stable orbit around the planet, we need to add the right amount of momentum, where momentum and gravitational forces are balancing each other. This way, we ensure that the system has conservative dynamics. Equivalently the probabilistic system can be augmented with auxiliary momentum parameters, $\mathbf{p}$, to explore the target distribution efficiently. Similar to the physical system, if the momentum is too little the trajectories will crash into the mode, whereas too big of a momentum will lead to trajectories exploring areas away from the target distribution. The main idea behind Hamiltonian Monte Carlo is to introduce a Hamiltonian function $H(\delta\mathbf{m}, \mathbf{p})$, such that the resulting dynamics allow us to explore a target distribution efficiently. These Hamiltonian dynamics are used to describe how an object moves in space in terms of its location (i.e. time–lapse model $\delta\mathbf{m}$) and its momentum $\mathbf{p}$ at some time t. For each location of the object, there is an associated potential energy $U(\delta\mathbf{m})$, and for each momentum of the object there is an associated kinetic energy $K(\mathbf{p})$. To relate the potential energy to the target distribution we use the concept of canonical distribution from the field of statistical mechanics. For a set of variables θ and their energy function $E(\theta)$, we have a canonical distribution with probability density function $p(\theta) = \frac{1}{Z}e^{-E(\theta)}$. The variable Z is a positive normalizing constant — so–called partition function— ensuring that this function sums to one so that the resulting distribution is a valid probability distribution. One can obtain the energy function $E(\theta)$ by simply $E(\theta) = -\log p(\theta) - \log Z$. The Hamiltonian function is an energy function that combines the potential and kinetic energy such that $E(\theta) = H(\delta\mathbf{m}, \mathbf{p}) = U(\delta\mathbf{m}) + K(\mathbf{p})$. The canonical distribution for the Hamiltonian energy function will therefore be

$$\begin{aligned} p(\delta\mathbf{m},\mathbf{p}) &\propto e^{-H(\delta\mathbf{m},\mathbf{p})} \\ &= e^{-[U(\delta\mathbf{m})+K(\mathbf{p})]} \\ &= e^{-U(\delta\mathbf{m})}e^{-K(\mathbf{p})} \\ &\propto p(\delta\mathbf{m})p(\mathbf{p}). \end{aligned} \tag{5.6}$$

From Equation 5.6 we see that the joint canonical distribution simplifies to independent canonical distributions for $\delta\mathbf{m}$ and $\mathbf{p}$ respectively. This means that $\delta\mathbf{m}$ and $\mathbf{p}$ are independent of one another. Therefore, we use the joint canonical distribution to sample from, but we ignore the momentum variables afterward; this is because the momentum variables are only auxiliary to allow the use of Hamiltonian dynamics, while the main variables of interest reside in $\delta\mathbf{m}$. This means that the target distribution we are interested is $p(\delta\mathbf{m})$. Following the energy expression of a canonical distribution from above, we can obtain the potential energy function $U(\delta\mathbf{m})$ by defining an expression that when negated and taken the exponential of, will give the target distribution $p(\delta\mathbf{m})$ [100] [133] such that

$$U(\delta\mathbf{m}^*) = -\log\left[p(\delta\mathbf{m}^*)\mathcal{L}(\delta\mathbf{m}^*)\right], \tag{5.7}$$

where $p(\delta\mathbf{m}^*)$ is the prior time–lapse model distribution and $\mathcal{L}(\delta\mathbf{m}^*)$ is the likelihood computed via Equation 5.5. Here, we assume that the prior is a bounded uniform distribution. Because $\delta\mathbf{m}$ and $\mathbf{p}$ are independent, we can choose any distribution we want to sample from for the kinetic energy. A common choice is a zero–mean unit–variance Gaussian distribution such that $\mathbf{p}{\sim}\mathcal{N}(0,1)$, where $\mathbf{p}$ is a vector with size equal to the model parameters (number of DoF). This is equivalent to having a quadratic kinetic energy $K(\mathbf{p})$ in the system, calculated as

$$K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\mathbf{p}. \tag{5.8}$$

As mentioned earlier, Hamiltonian dynamics describe how an object moves in space at some time t. Quantitively, this is expressed with a set of differential equations known as the Hamilton equations [100]:

$$\frac{\partial \delta\mathbf{m}}{\partial t} = \frac{\partial H(\delta\mathbf{m}, \mathbf{p})}{\partial \mathbf{p}} = \frac{\partial K(\mathbf{p})}{\partial \mathbf{p}}, \qquad \frac{\partial \mathbf{p}}{\partial t} = \frac{\partial H(\delta\mathbf{m}, \mathbf{p})}{\partial \delta\mathbf{m}} = -\frac{\partial U(\delta\mathbf{m})}{\partial \delta\mathbf{m}}. \tag{5.9}$$

The algorithm

HMC treats a model as a virtual Hamiltonian particle that moves along a trajectory. Evolving a current state $[\delta\mathbf{m}_0, \mathbf{p}_0]$ over some time τ we get a new state $[\delta\mathbf{m}^*, \mathbf{p}^*]$. Due to the conservation of energy the Hamiltonian is equal in both states. The sequence of models and momenta map out positions on the so–called phase space (i.e. plots of trajectories). It is difficult to solve the Hamiltonian dynamics analytically, and therefore we typically approximate them by discretizing them in time.

To discretize Hamiltonian dynamics we use symplectic numerical integrators. Symplectic means that the solution of a system exists on a symplectic manifold [76] [52], which is a characterization of phase space $[\delta\mathbf{m}, \mathbf{p}]$. Hamilton's equations naturally have that their solutions reside on a symplectic manifold in phase space, with the natural splitting of position and momentum variables. These symplectic integrators should respect the properties of the Hamiltonian dynamics, which are time reversibility (Figure 5.2 (a)) and volume preservation (Figure 5.2 (b)). Suppose we start at a

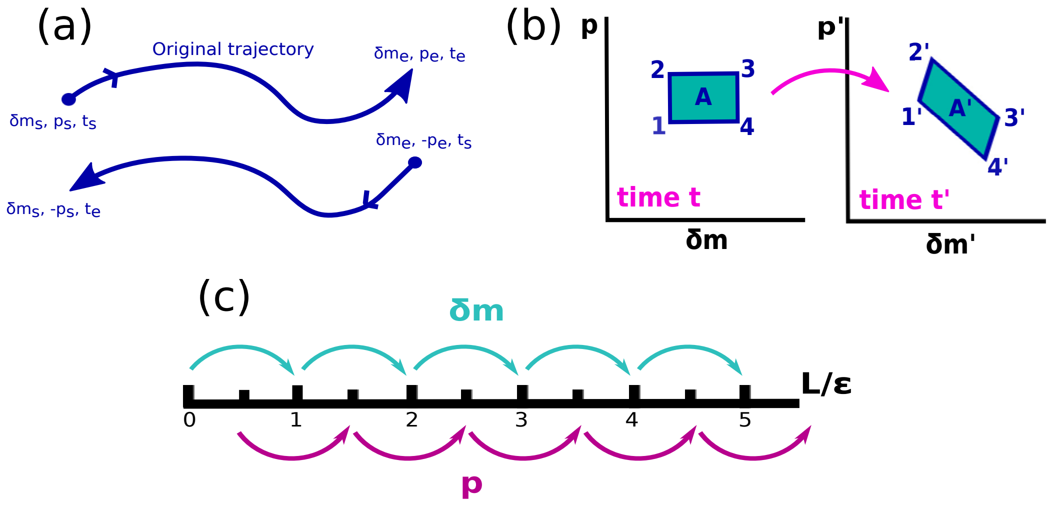
(a)
Original trajectory
δm_e, p_e, t_e
δm_s, p_s, t_s
δm_e, $-p_e$, t_s
δm_s, $-p_s$, t_e
(b)
p
2
3
A
1
4
time t
δm
p'
2'
A'
1'
3'
4'
time t'
δm'
(c)
δm
L/ε
0
1
2
3
4
5
p

Figure 5.2: (a) Illustration of an original and a time reversed trajectory. (b) Simplistic illustration of the volume preservation characteristic of the Hamiltonian dynamics. (c) Illustration of the structure of the leapfrog method.

position $\delta \mathbf{m}_s$ at some time t_s with momentum $\mathbf{p}_s$ and follow a trajectory to a later position $\delta \mathbf{m}_e$ with momentum $\mathbf{p}_e$ at time t_e. Considering the time reversed trajectory starting at time t_s, at position $\delta \mathbf{m}_e$ but with the opposite momentum $-\mathbf{p}_e$. Then, at time t_e, the particle will have reached the initial position $\delta \mathbf{m}_s$ and the momentum will be $-\mathbf{p}_s$. We think of this as a movie we run backwards; this describes the time reversibility (symplectic integrators satisfy this by negating the momentum at the end of the trajectory, as will be explained in the next paragraphs). The volume preservation (area in phase space) is proven using Liouville's theorem; [100] provides a detailed explanation on that. Here, we only show a simplistic illustration (Figure 5.2 (b)). Let the four corners of the square (1,2,3,4) represent four possible coordinates of a particle at time t. At a later time t' each of these four points will change to form the corners of a parallelogram. If we integrate Hamilton's equations for some finite time, this is essentially equivalent to designing a map in phase space. The area of the square A will be equal to the area of the parallelogram A', representing an area preserving map.

The most commonly used symplectic integrator in the context of HMC is the Leapfrog method (an integrator of 2nd order accuracy), where position and momentum are updated sequentially in an interleaving fashion such that they “leapfrog” each other (Figure 5.2 (c)). At each iteration of HMC, we simulate the Leapfrog integrator for L steps using a stepsize ϵ, which essentially leads to simulations over $L \times \epsilon$ units of time. Obviously, for a successful implementation, these two parameters need to be tuned; we will discuss this in more detail in the next subsection. Starting with a current position $\delta\mathbf{m}_0$, we randomly draw a momentum from a zero mean unit–variance Gaussian distribution. The Leapfrog method first updates the momentum dynamics for a small interval of time $\frac{\epsilon}{2}$, then updates the position and momentum for a slightly longer interval of time ϵ, and at the end updates the momentum for another small interval $\frac{\epsilon}{2}$, so that position and momentum are now at the same point in time [100]. Specifically, we firstly take a half step to update the momentum variable via

$$\mathbf{p}\left(t + \frac{\epsilon}{2}\right) = \mathbf{p}(t) - \left(\frac{\epsilon}{2}\right) \frac{\partial U}{\partial \delta\mathbf{m}(t)}. \tag{5.10}$$

Then, for L and $L-1$ steps we update position and momentum variables respectively with a full step via

$$\delta\mathbf{m}(t+\epsilon) = \delta\mathbf{m}(t) + (\epsilon)\frac{\partial K}{\partial \mathbf{p}\left(t+\frac{\epsilon}{2}\right)} \tag{5.11}$$

$$\mathbf{p}\left(t + \frac{\epsilon}{2}\right) = \mathbf{p}\left(t + \frac{\epsilon}{2}\right) - (\epsilon)\frac{\partial U}{\partial \delta\mathbf{m}(t+\epsilon)}. \tag{5.12}$$

We then take the remaining half step to update the momentum variable via

$$\mathbf{p}\left(t+\epsilon\right) = \mathbf{p}\left(t + \frac{\epsilon}{2}\right) - \left(\frac{\epsilon}{2}\right) \frac{\partial U}{\partial \delta\mathbf{m}(t+\epsilon)}, \tag{5.13}$$

and at the end we negate the momentum so that the proposal is symmetrical by ensuring reversibility ([13]; Figure 5.2 (a)). Let us take a moment and briefly explain why. Because we use a numerical integrator to approximate the Hamiltonian flow, this integration introduces an error. A usual trick to correct this error, is to consider the flow as a proposal and apply a Metropolis correction at the end. However, to apply this correction it is required that the flow is reversible, which in the case of HMC it is not. One way to make the flow reversible is consider that the kinetic energy is symmetric around $\mathbf{p}$, and apply a momentum flip at the end. [13] provides a detailed explanation of this in his Section 5.2. The pseudocode in Algorithm 3 summarizes the steps of the Leapfrog integration. The new proposed state $[\delta\mathbf{m}^{*}, \mathbf{p}^{*}]$ (where $\delta\mathbf{m}^{*} = \delta\mathbf{m}(t+\epsilon)$, and $\mathbf{p}^{*} = -\mathbf{p}\,(t+\epsilon)$) is accepted based on the Metropolis criterion, with a probability

$$\alpha = min\left[1, e^{-H(\delta\mathbf{m}^{*},\mathbf{p}^{*})+H(\delta\mathbf{m}_0,\mathbf{p}_0)}\right] = min\left[1, e^{-U(\delta\mathbf{m}^{*})+U(\delta\mathbf{m}_0)-K(\mathbf{p}^{*})+K(\mathbf{p}_0)}\right]. \qquad (5.14)$$

The flowchart in Figure 5.3 provides a summary of the algorithm. Optimal acceptance rates, α, of the algorithm are typically in the range of $60 - 80\%$.

Algorithm 3 Leapfrog integration

```
Require: $\delta\mathbf{m}_0$                                   ▷ initial position
Require: $\mathbf{p}_0$                                         ▷ initial momentum
Require: $L$                                                    ▷ number of leapfrog steps
Require: $\epsilon$                                             ▷ stepsize
$\mathbf{p}_1 = \mathbf{p}_0 - \frac{\epsilon}{2}\frac{\partial U}{\partial \delta\mathbf{m}_{i-1}}$     ▷ half step to update momentum
for $j = 1, .., L$ do
    $\delta\mathbf{m}_j = \delta\mathbf{m}_{j-1} + \epsilon\mathbf{p}_{j-1}$     ▷ full steps for position
    if $j \neq L$ then
        $\mathbf{p}_j = \mathbf{p}_{j-1} - \epsilon\frac{\partial U}{\partial \delta\mathbf{m}_{j-1}}$     ▷ full steps for momentum
    end if
end for
$\mathbf{p}_j = \mathbf{p}_j - \frac{\epsilon}{2}\frac{\partial U}{\partial \delta\mathbf{m}_j}$     ▷ remaining half step for momentum
$\mathbf{p}_j = -\mathbf{p}_j$     ▷ flip the momentum at the end of trajectory
```

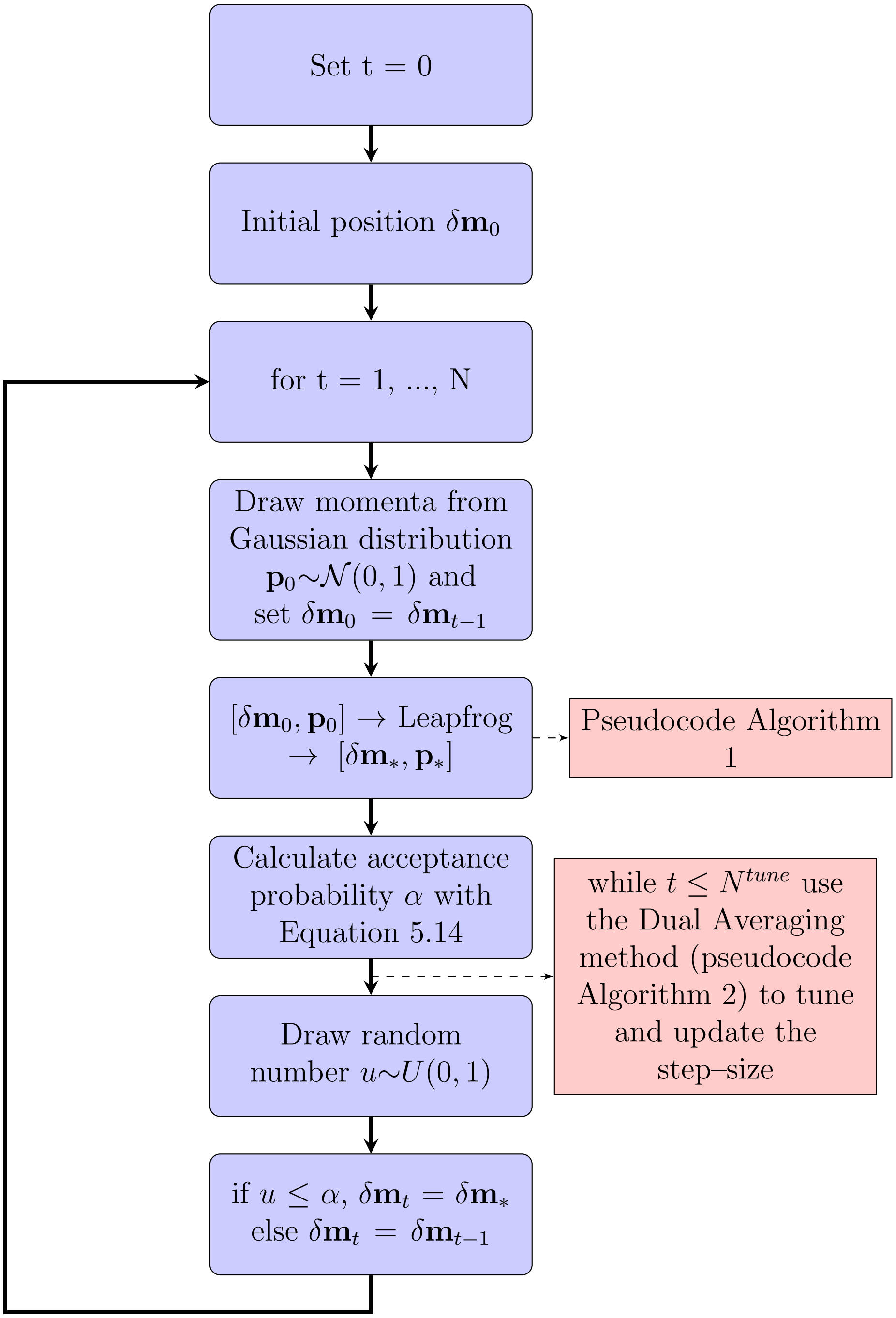
Set t = 0
Initial position $\delta\mathbf{m}_0$
for t = 1, ..., N
Draw momenta from Gaussian distribution $\mathbf{p}_0 \sim \mathcal{N}(0,1)$ and set $\delta\mathbf{m}_0 = \delta\mathbf{m}_{t-1}$
$[\delta\mathbf{m}_0, \mathbf{p}_0] \rightarrow$ Leapfrog $\rightarrow [\delta\mathbf{m}_*, \mathbf{p}_*]$
Pseudocode Algorithm 1
Calculate acceptance probability α with Equation 5.14
while $t \leq N^{tune}$ use the Dual Averaging method (pseudocode Algorithm 2) to tune and update the step–size
Draw random number $u \sim U(0,1)$
if $u \leq \alpha$, $\delta\mathbf{m}_t = \delta\mathbf{m}_*$ else $\delta\mathbf{m}_t = \delta\mathbf{m}_{t-1}$

Figure 5.3: 4D Hamiltonian Monte Carlo Flowchart.

Hyperparameters

Even though HMC is considered to be one of the most powerful sampling methods, its efficiency depends unavoidably on the choice of the hyper-parameters; here these hyper-parameters are the number of leapfrog steps L and the leapfrog stepsize ϵ, which together determine the length of the trajectory at some time $\tau = \epsilon L$. A poor choice of L and ϵ can lead to dramatic degradation in the performance of HMC [100] [58] [13]. There are different ways that one can choose to tune these parameters, and we elaborate on those in the next couple of paragraphs.

Number of Leapfrog steps L A correct choice of L will determine whether the space exploration is systematic and efficient, rather than a random walk. For example, if L is chosen too large, then the trajectories will be too long resulting in revisiting and resampling the same area. This translates to a waste in computational time. On the contrary, if L is chosen too small, then the consecutive samples will be too close to one another, behaving similarly to a random–walk Metropolis algorithm. Thankfully, the choice of L is independent of the choice of ϵ. For example, for the number of leapfrog steps L one can choose the method of trial and error, or more sophisticated approaches such as the one proposed in [58]. Here, we use the trial and error technique following the recommendations on L choices provided in [100], and in particular we find that $L = 10$ is good choice for the numerical tests of this study. However, much like all MCMC algorithms, HMC can also show pathological behaviour when it encounters areas of high curvature in parameter space. For instance, the most common way of illustrating this phenomenon is using the Funnel distribution [100], where a Markov chain can get stuck in the neighbourhood of high curvature, and only after infinite number of iterations can potentially escape (i.e. Figure 9 in [13]). In order for an HMC algorithm to avoid poor interactions in regions of high curvature when a fixed trajectory length τ is used in every iteration, [100] suggests to randomly vary τ. Here, we adopt this strategy by randomly jittering L at each iteration, via $L = 2 * np.random.rand() * L$.

Leapfrog stepsize ϵ Tuning the stepsize ϵ is, to some extent, similar to tuning the stepsize of Metropolis Hastings but with higher sensitivity; an incorrect choice of ϵ can have severe effects on the performance of the algorithm and sometimes even generate unstable trajectories. If ϵ is chosen too large the leapfrog method will generate inaccurate simulations, which will be reflected in low acceptance rates of the HMC algorithm. Otherwise, if ϵ is chosen too small, this will lead to taking too many steps in the integration and essentially wasting computational time. With respect to how to tune the stepsize, there are a few more possibilities in addition to trial and error. For example, one could choose ϵ randomly from a distribution at the beginning of each trajectory [100] or using adaptive strategies such as the ones described in [5]. Here, we choose an adaptive strategy called the Dual Averaging Scheme proposed by [58]. This method combines the concept of vanishing adaptation (this means that the step–size adaptation vanishes gradually as more samples are drawn) proposed by [123] with the primal–dual algorithm of [101]. Here, we will briefly describe the algorithm, but for more details please refer to [58]. We assume that we have a statistic H_t that contains some information about the MCMC process at some iteration t. This statistic could be, for example, described in terms of the acceptance probability at iteration α_t and the desired average acceptance probability δ such that

$$H_t = \delta - \alpha_t. \tag{5.15}$$

The expectation of H_t is

$$h(\epsilon) = E_t[H_t|\epsilon]. \tag{5.16}$$

We are interested in finding all $\epsilon \in \mathbb{R}$ such that Equation 5.16 will converge to 0. This can be accomplished by performing the following updates

$$\epsilon_{t+1} \leftarrow \mu - \frac{\sqrt{t}}{\gamma} \frac{1}{t + t_0} \sum_{i=1}^{t} H_i, \tag{5.17}$$

$$\bar{\epsilon}_{t+1} \leftarrow \eta_t \epsilon_{t+1} + (1 - \eta_t)\bar{\epsilon}_t,$$

with the second equation being an evaluation of the average stepsize ϵ_t. In the above equations, μ is a randomly chosen point towards which ϵ_t is shrunk, and the amount of that shrinkage is controlled by $\gamma > 0$. The term $t_0 \geq 0$ conveys stability of the algorithm at initial iterations; theoretically larger values of t_0 will stabilize the stepsize exploration early. The parameter $\eta_t = t^{-\kappa}$ with $\kappa \in (0.5, 1]$ is a stepsize ensuring that $h(\epsilon)$ will converge to 0. The smaller κ is, the faster the algorithm forgets its earlier stepsize iterates. In addition, η_t needs to satisfy

$$\sum_t \eta_t = \infty, \qquad \sum_t \eta_t^2 < \infty. \tag{5.18}$$

From equation 5.17 the per–iteration update in the stepsize ϵ is on the order of

$$\epsilon_{t+1} - \epsilon_t = \mathcal{O}(-H_t t^{-0.5}), \tag{5.19}$$

which validates the vanishing nature of the adaptation. In this work, we choose the default parameters of $\gamma = 0.05$, $t_0 = 10$, and $\kappa = 0.75$ from [58]. We also set the desired average acceptance probability to $\delta = 0.65$ [100]. The pseudocode in Algorithm 4 shows the Dual Averaging algorithm for stepsize tuning. During the tuning process, the HMC algorithm uses a stochastic stepsize that comes from

Equation 5.17 at each iteration. Once the tuning is done, the HMC uses the dual–averaged stepsize for sampling.

Algorithm 4 Dual Averaging Algorithm

```
Require: $N^{tune}$                       ▷ number of iterations to run tuning
Require: $\kappa = 0.75$                  ▷ default value, Hoffman & Gelman (2014)
Require: $\gamma = 0.05$                  ▷ default value, Hoffman & Gelman (2014)
Require: $t_0 = 10$                       ▷ default value, Hoffman & Gelman (2014)
Require: $\mu = \log(10\epsilon_0)$       ▷ default value, Hoffman & Gelman (2014)
Require: $\delta = 0.65$                  ▷ desired acceptance rate, Neal (2011)
Require: $\overline{H}_0 = 0$
Require: $\epsilon_0$                     ▷ initial stepsize
Require: $\bar{\epsilon}_0$
if $i \leq N^{adapt}$ then
    $\alpha_i = min(1, exp(-H(\delta\mathbf{m}^*, \mathbf{p}^*) + H(\delta\mathbf{m}_{i-1}, \mathbf{p}_{i-1})))$    ▷ acceptance probability
    $\overline{H}_i = \left(1 - \frac{1}{i+t_0}\right)\overline{H}_{i-1} + \frac{1}{i+t_0}(\delta - \alpha_i)$
    $\log\epsilon_i = \mu - \frac{\sqrt{i}}{\gamma}\overline{H}_-$
    $\log\bar{\epsilon}_i = i^{-\kappa}\log\epsilon_i + (1 - i^{-\kappa})\log\bar{\epsilon}_{i-1}$
else
    $\epsilon_i = \bar{\epsilon}_{N^{adapt}}$    ▷ Set to dual–averaged stepsize for sampling
end if
```

Algorithm illustration with Gaussian examples

To illustrate how the algorithm works we consider a couple of simple Gaussian examples that we compare to the examples shown in [25]. Let us assume that we want to sample from a posterior distribution $p(\mathbf{x})$ that has a normal distribution with the following parameterization

$$p(\mathbf{x}) \sim \mathcal{N}(\mu, \sigma), \tag{5.20}$$

with a mean of $\mu = 0$ and a standard deviation of $\sigma = 0.1$. Using the kinetic energy from equation 5.8, we need to define the expressions for the potential energy and its partial derivative. The potential energy, as mentioned earlier, can be defined as the negative logarithm of the posterior distribution, $U(\mathbf{x}) = -\log[p(\mathbf{x})]$. Taking the negative logarithm of a normal distribution results in

$$U(\mathbf{x}) = \frac{1}{2}\left(\log(2\pi\sigma^2) + \left(\frac{\mathbf{x}-\mu}{\sigma}\right)^2\right). \tag{5.21}$$

To calculate the partial derivative of Equation 5.21 with respect to $\mathbf{x}$, we use a Python–based automatic differentiation package called `autograd`[1]. This package offers a way of taking a code that computes a function, and automatically constructs a code that computes the derivative of that function [82] [9]. We, therefore, define the gradient of potential energy as `dUdx = grad(U(x))`. We start from an initial position of $\mathbf{x}_0 = 0.0$, and draw initial momentum from $\mathbf{p}_0 \sim \mathcal{N}(0, 1)$. We then run the HMC algorithm for 2,000 iterations with the hyper-parameters of $L = 1.0$ and $\epsilon = 0.01$. Figure 5.4 (a) and (b) show the recovered histograms and the phase space respectively. We see that the single Gaussian distribution maps out ellipses in the phase space, where the radius of the ellipses depends on the energy of the system, defined by the initial conditions. Recall, that these ellipses in the phase space represent the Hamiltonian trajectories (defined by plotting positions and momenta) along the integration length.

To progress this example further, we consider a target distribution that is a mixture of three normal distributions. Specifically, we consider

$$p(\mathbf{x}) = [p_1(\mathbf{x}); p_2(\mathbf{x}); p_3(\mathbf{x})], \tag{5.22}$$

where $p_1(\mathbf{x}) \sim \mathcal{N}(-1.0, 0.3)$, $p_2(\mathbf{x}) \sim \mathcal{N}(0.0, 0.2)$, and $p_3(\mathbf{x}) \sim \mathcal{N}(1.0, 0.3)$, where each of the individual normal distributions have different weights (specifically $p_1(\mathbf{x})$ has a 10% weight, $p_2(\mathbf{x})$ has 50%, and $p_3(\mathbf{x})$ has 40%). We run the HMC algorithm for 2,000 iterations using exactly the same parameters as in the previous example. Figure 5.4 (c) and (d) show the recovered histograms and the phase space respectively. We see

[1] https://github.com/hips/autograd

that the phase space is different than before, with three ellipses with their centres at each of the different means. The density of these ellipses is proportional to the weight given to each normal distribution; for example, $p_2(\mathbf{x})$ has higher weight than $p_3(\mathbf{x})$ and $p_1(\mathbf{x})$, which is reflected in how dense the sampling is for this one compared to the other two.

Now, to make things a bit more exciting we set up a target distribution that is a mixture of two, $p(\mathbf{x}) = [p_1(\mathbf{x}); p_2(\mathbf{x})]$, where the mean of the 2nd is within the second standard deviations of the first and the areas of their first standard deviation are overlapping. In particular, $p_1(\mathbf{x}){\sim}\mathcal{N}(0.0, 0.5)$ and $p_2(\mathbf{x}){\sim}\mathcal{N}(0.6, 0.3)$, and we give both equal weight. This arrangement of distributions will most likely lead to a skewed target distribution. Once again, we run the HMC algorithm for 2,000 iterations with the same parameters as in the previous two examples. As expected, the resulting histogram (Figure 5.4 (e)) is a negative skewed distribution with the mode being in the vicinity of the $p_2(\mathbf{x})$ mean. This is also apparent in the phase space (Figure 5.4 (f)).

5.4 Simple Seismic Example

In order to be able to perform a direct comparison between the Metropolis–Hastings and the HMC algorithm, we use the same example as in [69]. For both the true baseline and background model $\mathbf{m_0}$ (on which we compute the full subsurface Green's functions), we use the standard Marmousi model [159] (Figure 5.5 (a)). The white box shows the location of the subdomain in which we evaluate the Hamiltonian energy function at each iteration i. The number of grid points in the local subdomain is significantly smaller than the number of grid points in the full domain; in this example

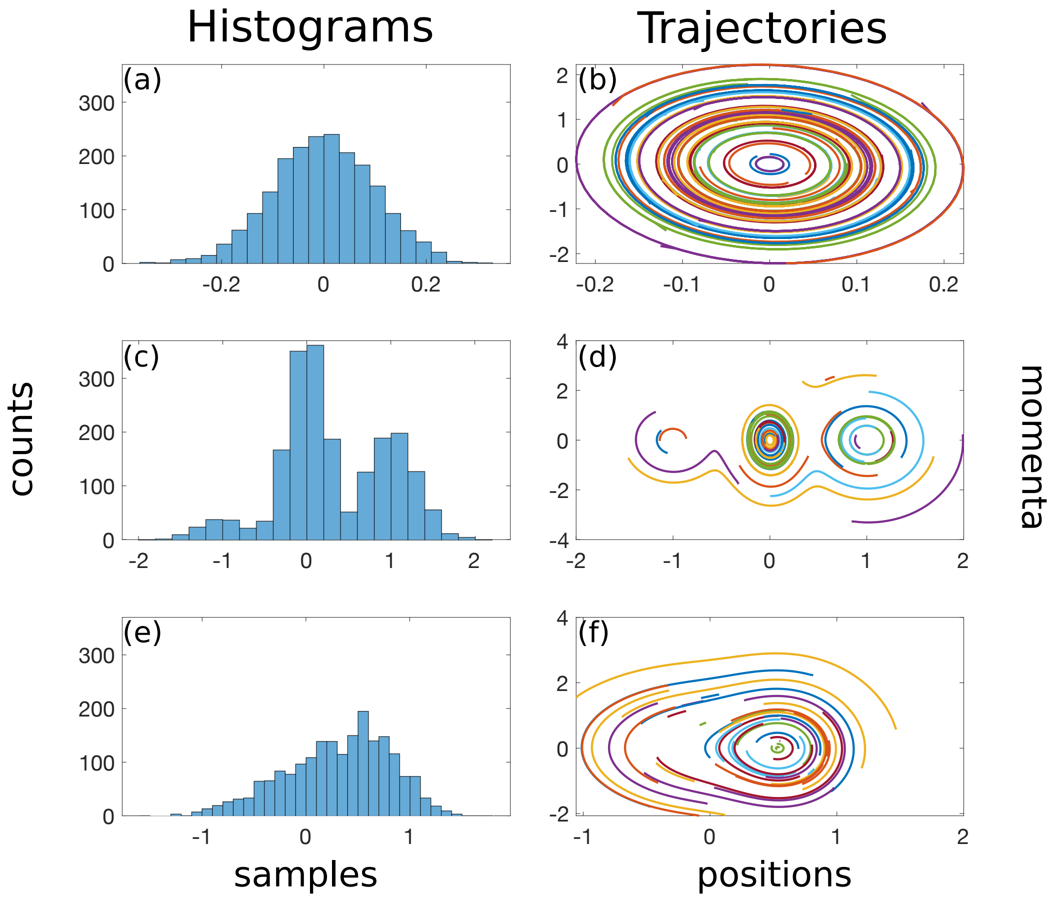

Figure 5.4: 1D Gaussian examples (left panel) and their respective trajectories in phase space (right panel). The first row shows the results of the single normal distribution example, the second row the results of the mixture of three normal distributions, and the third row the results of the mixture of two normal distributions. All histograms are retrieved after 2,000 iterations of the HMC algorithm, using the exact same hyper-parameters for all. All trajectories are plotted every 50th iteration.

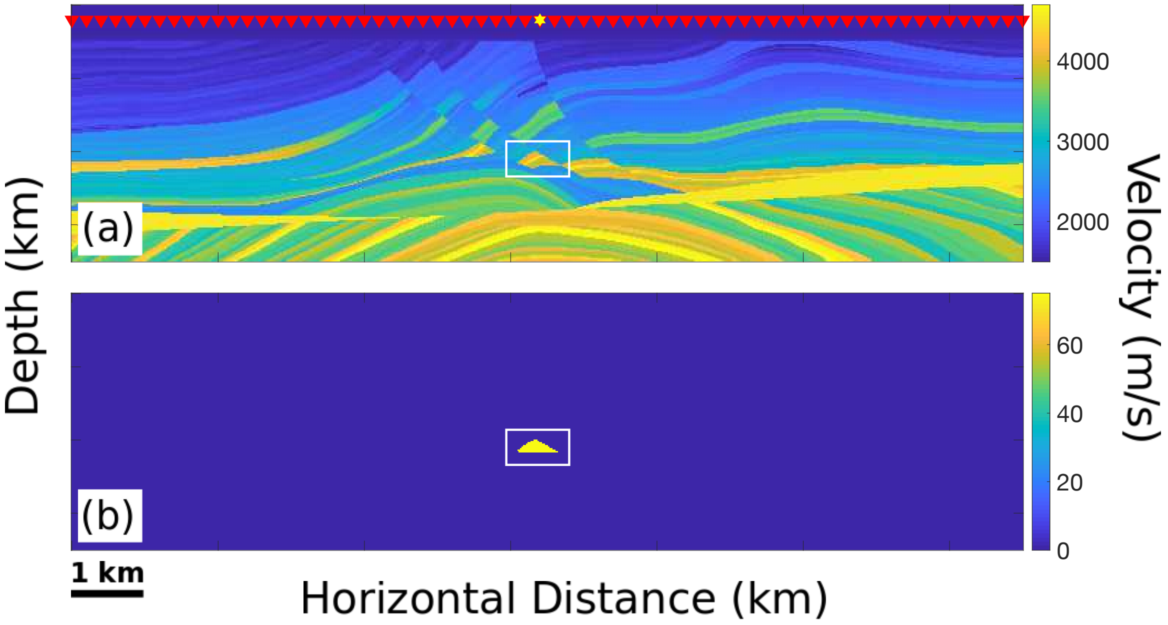

Figure 5.5: (a) The true Marmousi model, with the yellow star and red triangles indicating the location of the source and receivers respectively. (b) The true time–lapse change with a magnitude of 75 m/s that was was added in one of the layers. In both images the white box shows the location of the local domain.

$Nx_{sub} = 44$ and $Nz_{sub} = 25$, whereas $Nx_{full} = 651$ and $Nz_{full} = 176$.

To generate the monitor velocity model, we add a perturbation of 75 m/s as shown in Figure 5.5 (b). The acquisition geometry consists of a single shot and 651 receivers as is shown in Figure 5.5 (a). We use a Ricker wavelet with a peak frequency of 6 Hz as the source, while we perform the 4D HMC inversion for the frequency of 8 Hz. In the inversions we use noisy $\delta\mathbf{d}$ to which we add Gaussian noise with covariance matrix Σ_d. The signal to noise ratio is 1.9, calculated as the ℓ_2 norm of the noiseless signal over the ℓ_2 norm of the noise.

Since this is a single DoF example, computing the gradient of the potential energy (Equation 5.7) with respect to the time-lapse perturbation using the adjoint state method will be complicated and unnecessary. We instead use the central difference method such that

$$\nabla_{\delta\mathbf{m}} U = \frac{U(\delta\mathbf{m} + h) - U(\delta\mathbf{m} - h)}{2h}, \tag{5.23}$$

where we choose $h = 1 \; m/s$. Using the above equation is quite fast since it only requires two wavefield solves per gradient calculation. To initially validate the algorithm we run the HMC starting from three different $\delta\mathbf{m}_0$, in particular for $\delta\mathbf{m}_0 =$ 0 m/s, 75 m/s, and 1000 m/s. In principle, if the algorithm is set up correctly, when we start from the true perturbation of 75 m/s, it should keep exploring the vicinity of that value, instead of doing random jumps as in MH for example. Figure 5.6 (a) shows the results of only the first 10 iterations, as we are only interested in seeing how fast the algorithm reaches the neighbourhood of high probability. As expected, when we start from the true answer, HMC keeps exploring the vicinity of the true answer. When we start from 0 m/s the algorithm needs approximately two iterations to reach the proximity of the correct answer, in contrast with the scenario of $\delta\mathbf{m}_0 = 1000$ m/s where it needs approximately 6. We chose this example to allow direct comparison with our previous work. Figure 5.6 (b) compares the results of the first 50 iterations between the HMC and MH algorithm when we start from an initial perturbation of $\delta\mathbf{m}_0 = 0$ m/s. It is not surprising to see that the MH algorithm needs approximately 30 iterations before it reaches the neighbourhood of the correct perturbation; this number is vastly different from the two iterations needed for HMC. Obviously, the small number of required iterations is proportional to the simplicity of the problem. Once we move to higher dimensional problems, a larger number of iterations will be needed before the HMC reaches the proximity of the high density area. For completeness we also show the trajectories (plot of positions and momenta along the integration path at each iteration) of the HMC algorithm (Figure 5.6 (c)), and we also compare the retrieved histograms (Figure 5.6 (d)) after running the MH algorithm for 20,000 iterations (and discarding the 1st half) and the HMC algorithm

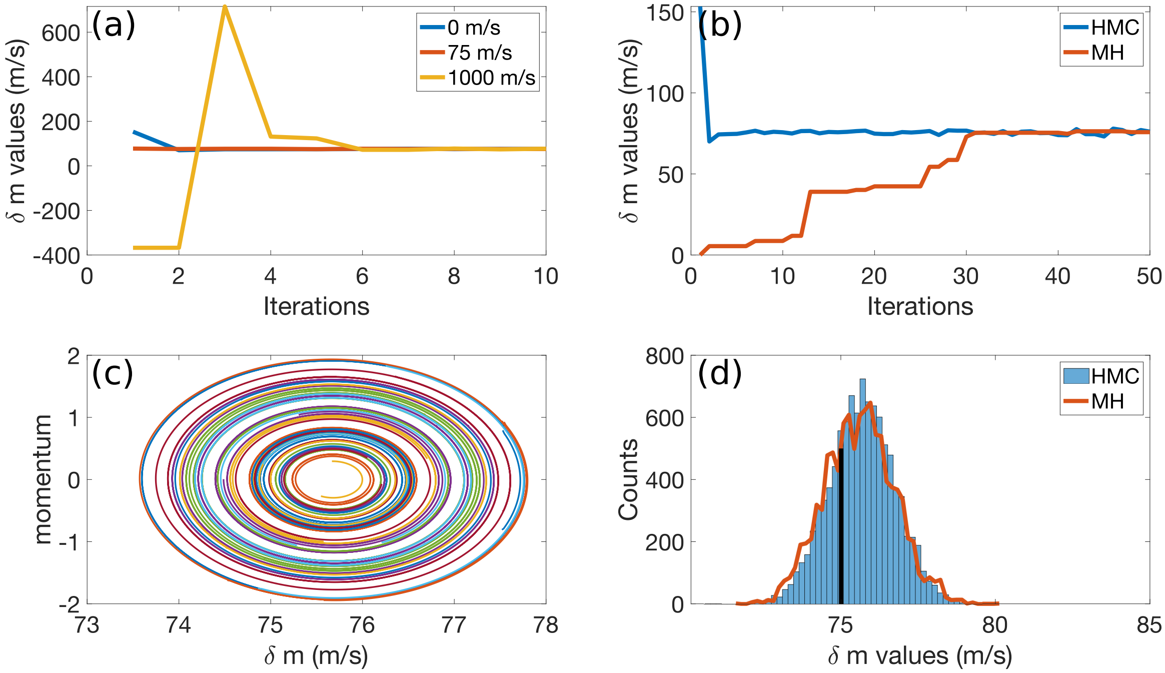

Figure 5.6: (a) Comparison of the behaviour of HMC with respect to different initial models given the same hyper–parameters L and ϵ. (b) Comparison of the performance of HMC versus MH given a starting model of $\delta\mathbf{m}_0 = 0$ m/s. (c) Trajectories in the phase space. (d) Retrieved histograms for both Metropolis–Hastings and Hamiltonian Monte Carlo inversions; the straight black line represents the value of the true time-lapse change.

for 10,000 iterations (without discarding any of the samples drawn).

5.5 2D Local Domain HMC Inversion

Moving towards a more exciting and realistic numerical example, we consider a different local domain than in the previous example and we attempt a full local domain retrieval. Without having the limitations of dimensionality nor the need to use reduced parameterization strategies, we could allow the whole model within the local–domain to be updated. This will mean a pixel–by–pixel uncertainty quantification, which will result in a problem with the number of DoF being equal to the total number of grid points that comprise the local domain. Figure 5.7 (a) shows the Marmousi model (that serves as our true baseline model) with the location of the local domain represented by the white box. We design this local domain to be shallower, as this allows for better illumination and imaging. We generate the monitor model by adding a velocity change of 200 m/s in one of the layers within this local domain (Figure 5.7 (b)). The number of grid points in the local domain are $Nx_{sub} = 30$ and $Nz_{sub} = 20$, accounting for a total of 600 DoF. For the simulations, we use five shots and 651 equally spaced receivers at the surface (Figure 5.7 (a)). The source is a Ricker wavelet with a peak frequency of 6 Hz. We perform the HMC inversion using a single frequency of 5 Hz.

Having a local acoustic solver at hand, means that we can compute gradients locally very fast using the adjoint state method [109] [89]. However, the time–lapse data differences $\delta\mathbf{d}$ do not satisfy the wave equation, and therefore any gradient computation will be extremely difficult if not impossible. To make this feasible, we have to adopt the same practise as several authors do in Double Difference FWI [180] [171] where we use a synthesized monitor data set called $\mathbf{d}_{comp}$. This new data set is generated by

$$\mathbf{d}_{comp} = \mathbf{b}_{syn} + (\mathbf{d}_1 - \mathbf{d}_0), \tag{5.24}$$

where $\mathbf{b}_{syn}$ is the synthetic data simulated on the inverted baseline model (that is used as a starting model), and $\mathbf{d}_1$ and $\mathbf{d}_0$ are the observed monitor and baseline data respectively. By doing this trick, we use a standard FWI gradient via the adjoint state method, while still respecting our Bayesian formulation of model differences and data differences in the likelihood function. If we have observed data $(\mathbf{d}^1_{comp}, \mathbf{d}^2_{comp}, ...\mathbf{d}^N_{comp})$ for N number of shots (here $N = 5$) and the measurement noise of each observation has a covariance matrix $\Sigma_1, \Sigma_2, ...\Sigma_N$ respectively, then the likelihood function becomes

$$\mathcal{L}(\delta\mathbf{m}) \propto exp\left(-\frac{1}{2}\left[\mathcal{X}^1\mathcal{X}^2...\mathcal{X}^N\right]^T \mathbf{\Sigma}^{-1}\left[\mathcal{X}^1\mathcal{X}^2...\mathcal{X}^N\right]\right), \tag{5.25}$$

where $\mathcal{X} = F(\mathbf{m}_1) - \mathbf{d}_{comp}$ with the superscript representing the different shots and $\mathbf{\Sigma}$ is the block diagonal covariance matrix of $\Sigma_1, \Sigma_2, ...\Sigma_N$, i.e.

$$\mathbf{\Sigma} = \begin{bmatrix} \Sigma_1 & 0 & 0 & \dots \\ 0 & \Sigma_2 & 0 & \dots \\ \vdots & & \ddots & \\ 0 & \dots & 0 & \Sigma_N \end{bmatrix}. \tag{5.26}$$

Each shot has a different realization of Gaussian measurement noise, however for convenience all of the distributions are described by the same covariance matrix. We generate noise with respect to the time–lapse differences $\delta\mathbf{d}$, and the signal to noise ratio is 3.9, calculated as the ℓ_2 norm of the noiseless signal $\delta\mathbf{d}$ over the ℓ_2 norm of the noise. This noise level is quite low, however our future work is aiming to address this by testing the algorithm in the presence of higher noise.

To make this numerical test as realistic as possible, we consider that our initial background model is an inverted baseline model. Specifically, we perform a frequency–based, constant–density acoustic FWI using 64 shots and 651 receivers that span the entire subsurface of the model using the PySIT library [56]. We invert for six frequencies (3, 4, 5, 6.5, 8, 10 Hz) sequentially for 15 iterations per frequency. Figure 5.7 (c) shows the obtained baseline velocity model. Since the HMC inversion will take place only within the local domain, we first need to compute the full domain Green's functions on the inverted baseline model of Figure 5.7 (c).

It is known that HMC typically has high acceptance rates that range between

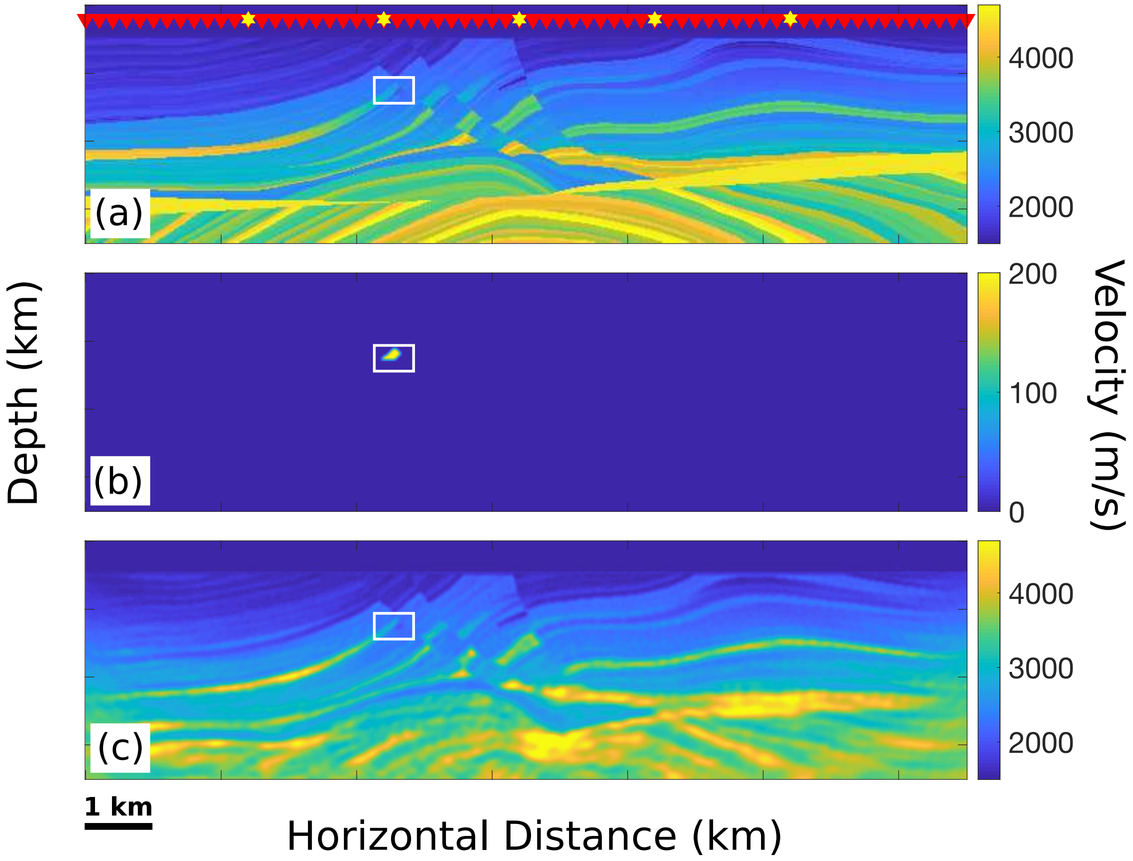

Figure 5.7: (a) The true baseline model with the white box indicating the location of the local domain. The yellow stars and red triangles represent the positions of the sources and receivers respectively. (b) The true time–lapse velocity change. (c) The inverted baseline model that serves as the starting and background model on which the full Green's functions are calculated.

60% to 80%. However, a high acceptance rate is not necessarily an indicator of good performance (in terms of successful exploration of the posterior distribution). In this example, we tune the step size for a desirable acceptance rate, however there are always trade offs between this and successful exploration of model space, and ideally a proper balance is needed for good performance. To assess the performance as well as the convergence of the algorithm, we look at the evolution of the potential energy $U(\delta\mathbf{m})$ (Figure 5.8 (a)), simply because it contains the likelihood function and hence the data misfit. We run the 4D HMC algorithm (Figure 5.3) for 7,000 iterations, with the Dual Averaging tuning for the stepsize only for the first 10 iterations. One can choose to tune for more or fewer iterations depending on the trust one has in the initial stepsize guess. Figure 5.8 (b) shows the evolution of the stepsize choices while the Dual Averaging is taking place; we start with an initial stepsize of $\epsilon_0 = 0.0032$, and we get an adapted stepsize of $\epsilon_{adapt} = 0.1968$ which we keep fixed and use throughout the 7,000 iterations. The acceptance rate is 51%; even though it is a bit lower than the optimal range, we consider it as a good performance given the convergence of the potential energy function. It will be quite challenging to show histograms at such a high dimensional space, because we will potentially need to look at histograms of each variable, meaning of each pixel. We choose, instead, for visualization purposes to look at the median model obtained from all models in the chain. The median model provides a good approximation of the true model in terms of the shape of the anomaly, however they seem to underestimate the magnitude of it. Incorporation of more information (e.g. adding more frequencies) could potentially improve this.

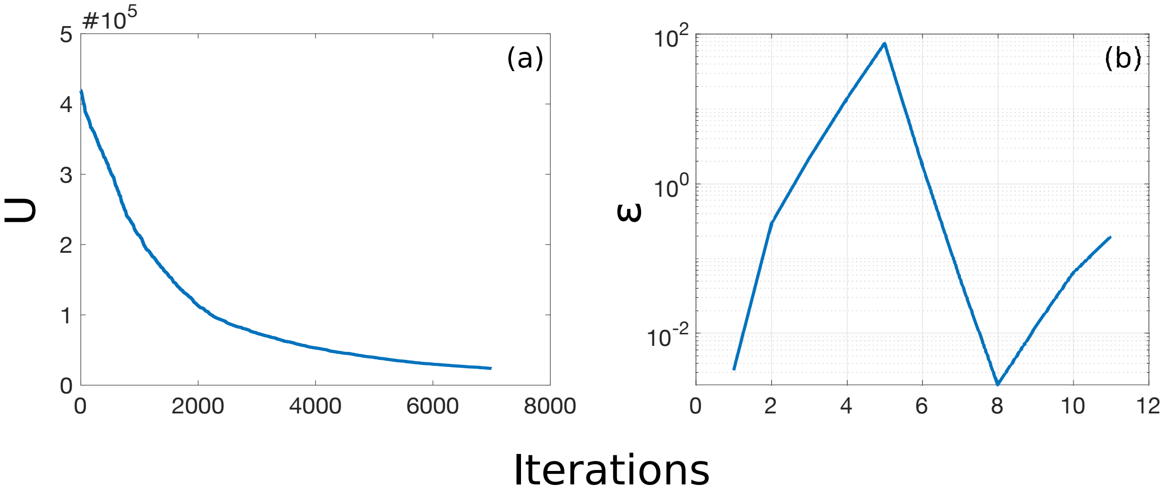

Figure 5.8: (a) Evolution of the potential energy function $U(\delta\mathbf{m})$ as a function of iterations. (b) Stepsize estimates during the Dual Averaging tuning process.

5.6 Discussion

Thus far, we have provided a proof of principle that HMC can be successfully applied in 4D uncertainty quantification. While HMC algorithms are computationally expensive, using a local solver as in this study proved to be a significant reduction in the cost. For instance, running the 600 DoF example for 7,010 iterations (including the iterations for tuning) took approximately 48 hours. This means that we were able to perform the computation locally on a MacBook Pro with a 2.6 GHz processor and 32 GB memory. Our results thus far using only five shots and a single frequency seem promising. We are currently working on extending our framework to multiple frequencies.

Any global global optimization technique is supposed to be independent of the choice of starting model. This means that the target distribution will be efficiently explored after a finite number of iterations. However, it is obvious that the algorithm will reach the equilibrium stage much faster if the starting model is in the neighbourhood of the global minimum. This is what Figure 5.10 illustrates. Both HMC inversions became localized in the region of the correct model, however one of them took 2,000 iterations more. In other words, starting from a "better" initial model can considerably reduce the computational cost of the HMC inversion. In addition, the current acceptance rate of our inversion is 51%, even though the ideal rate should be between 60% - 80%. When we run the same inversion with a much smaller stepsize, we observe that even though we get an acceptance rate of 65%, for the same number of iterations, the algorithm is significantly slower to explore the model space in this case. This means that we wasted a lot of computational time. This begs the question: is there a trade–off? And if so, what is the acceptable balance? We are currently investigating this topic.

The uncertainty quantification in this study took place using a pixel–by–pixel parameterization. However, to what extent does a pixel–by–pixel versus a basis function parameterization provide a more meaningful result is still a debate. This could potentially be problem dependent. [78] provide an interesting comparison between a grid (pixel–by–pixel) and a DCT parameterization for MCMC on Ground Penetrating Radar (GPR) data. They observe that the DCT parameterization is better able to capture the smooth variations in the soil moisture compared to the grid parameterization which resulted in meaningless randomly perturbed realizations. To some extent, we observed a similar pattern to our results, which could be either a result of the single frequency in the inversion or simply because we randomly perturb every pixel independently. We are currently exploring two different avenues to address this. Firstly, we are interested in whether we can use an off-diagonal covariance, which will essentially introduce a prior constraint on how correlated pixels are with each other. By doing so, we almost avoid randomly perturbing neighbouring pixels, which is what leads to difficult interpretations. Secondly, following our earlier work [68] we would like to test the possibility of using DCT parameterization instead of the uniform grid that we have used so far.

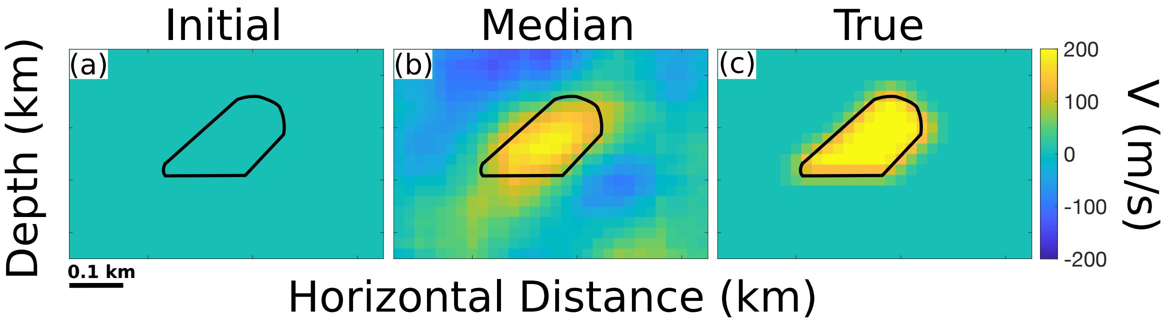

Figure 5.9: (a) The initial $\delta\mathbf{m}_0$ model used for the HMC inversion. (b) The median model computed from all 7,000 models in the chain. (c) The true time–lapse change.

When working with hybrid methods such as HMC, it might be interesting to implement and compare alternative hybrid approaches. For example [181] incorporate information from the local gradient and approximate Hessian into a Bayesian formulation that allows them to perform FWI and uncertainty analysis of the full Marmousi model. Our suggestion would be to combine local–optimization and Monte–Carlo sampling into a single scheme. This arises from the fact that sampling methods can provide a range of possible solutions, while local–optimization delivers the "best–fit" model. The question is, can we do better? One potential way to do so could be the following. Starting with a current model perturbation, we can propose a new model $\delta\mathbf{m}_*$ by adding a zero mean perturbation to the current such that $\delta\mathbf{m}_* = \delta\mathbf{m} + n, n \sim N(0, C)$. Use that model to run a few iterations of FWI using local–optimization. Then use the reconstructed model to evaluate the likelihood function and accept or reject the model. This is a simple idea, but it is yet to be tested.

Last but not least, in the Gaussian examples we used an automatic differentiation package called `autograd`. We are considering the possibility of adopting `autograd` or other deep–learning tools for automatic differentiation instead of the classic adjoint state method. An interesting application of such an approach is described in [121].

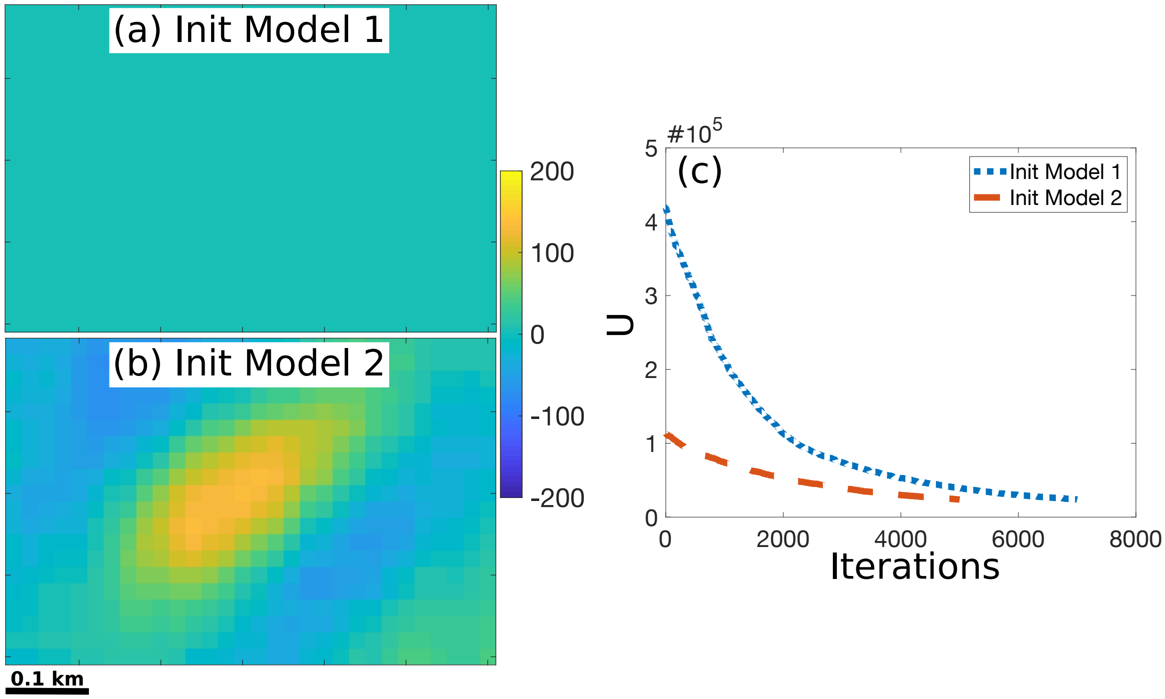

Figure 5.10: (a) - (b) Two different initial time–lapse starting models (c) Evaluation of the potential energy as a function of iteration for the two HMC inversions using as a starting model the models from (a) and (b).

5.7 Conclusions

Following our earlier work, we propose using a local acoustic solver for an efficient 4D Hamiltonian Monte Carlo inversion. Our framework can compute the partial derivatives of the potential energy in a targeted way that is computationally feasible. In addition, we use a dual averaging algorithm in order to tune the stepsize to the desired acceptance rate of 65%. Our single DoF illustration shows the efficiency and superior performance of the algorithm when compared to 4D Metropolis Hastings inversion. We successfully apply our framework to a full local domain inversion, that equals to 600 DoF, using only a single frequency and 5 shots. Our future work will be focusing on extending our framework a multi–frequency approach while also exploring the possibility of using deep learning algorithms for the gradient computation.

5.8 Acknowledgements

This work is supported by Chevron and with grants from the Natural Sciences and Engineering Research Council of Canada Industrial Research Chair Program and the InnovateNL and by the Hibernia Management and Development Corporation. In addition, we would like to thank Oliver Stueker from Compute Canada with his help on building Singularity Images on cluster.

Chapter 6

Conclusions and Future Work

Each individual chapter of this thesis has its own specific conclusions based on the focus of the respective chapter. Here, we only provide an overall summary of the whole thesis. This thesis proposes frameworks for time–lapse imaging using both deterministic and stochastic approaches. Our novel and hybrid Dual Domain Waveform Inversion (DDWI) is able to offer high resolution 4D images in complex scenarios by combining a standard 4D FWI with image domain constrains. However, even the most accurate and highest resolution 4D image (that is delivered through deterministic FWI, such as for example DDWI) will never be complete without estimates of the associated uncertainty. Calculating the full posterior pixel by pixel using expensive forward solvers would be computationally infeasible. We created a framework that is fast and calculates the time–lapse uncertainty in a targeted way. Using an exact local solver not only retains the physics of seismic imaging, but also requires no restrictions on the structure of the velocity model (i.e. layered Earth). Even though our derived framework is sensitive to uncertainties inherent in the background model, it still performs accurately for small perturbations in the wave path. Adding more redundancy in the data can potentially overcome this, but on the other side the computational cost would likely increase. Having such a fast and accurate forward solver at hand, allowed us to bench mark Metropolis–Hastings against Hamiltonian Monte Carlo, and show that the latter is more efficient for time–lapse UQ.

6.1 Future Directions

Although our proposed scheme finds immediate application in reservoir monitoring, it can be easily extended or applied to any problem that is localized in region and relies on sound wave propagation. For example, much like reservoir monitoring in seismic imaging, tumour monitoring in medical imaging can be considered a localized problem. Ultrasound waves together with a local domain that contains the affected organ can be employed to monitor the tumour progress while also providing estimates of the associated uncertainty. Alternative applications can be found in both environmental and engineering problems. One could use the newly popular Distributed Array Sensing (DAS) technology and apply our framework to hydraulic fracturing, CO_2 sequestration and near surface geohazard applications. Last but not least, a similar workflow can be employed to large scale seismological problems as for example estimation of Moho depth.

The framework presented in this thesis, sets the basis for simple and straightforward time–lapse UQ and allows for testing different assumptions. However, there are still various improvements and validation tests that could be made. We summarize those in the next two subsections.

6.1.1 Algorithmic Improvements & Reproducibility

The UQ framework in Chapter 4, even though it relies on a single source and a single frequency estimation, provides a rigorous uncertainty estimation and sets the baseline for more advanced implementations. Since most seismic observations are incomplete and noisy, adding more redundancy is critical. For example, a hierarchical Bayesian approach where the likelihood of a proposed model is first evaluated at a lower frequency, and if the model is accepted, then move to higher frequencies. The acoustic approximation is typically considered adequate when dealing with data collected in marine environments. However, a realistic Earth model representation requires more than just the P–wave velocity. An elastic FWI approach increases not only the number of degrees of freedom but also the computational cost. This issue is particularly significant in the Metropolis–Hastings scheme, where most of the proposed models are rejected. The best practice approach would be to employ a multi-fidelity algorithm, in some way similar to frequency stepping described above. For instance, a proposed model can be first evaluated using the coarse acoustic wave equation, and only if it's accepted to then evaluated in the finer elastic wave equation.

A significant challenge in geophysical algorithms is that the majority of them are not open source, and thus, the rest of the academic community cannot reproduce them or even advance them. One possible solution that we particularly favor and use in this thesis is using containers such as Docker, for example. This container is simply a virtual machine image that packages a seismic toolbox together with all its libraries and dependencies. Once an image like this is made, it can be pulled and run by any machine without installing any libraries or being dependent on machine software updates. If more computing resources are needed to run an algorithm, a Docker image can be used to build a Singularity image (`https://www.sylabs.io/docs/`) that can be commonly run on a computer cluster. The proposed framework of this thesis is currently in preparation as an open-source 4D UQ toolbox on a Docker image.

We have compared two different UQ methods on the time–lapse problem. A logical extension would be to compare the current forward solver to other MCMC approaches, such as trans-dimensional sampling [119] or polynomial chaos expansion [29] [35]. Having a complete 4D UQ toolbox available allows for a direct comparison of performance, which algorithm is better than another in terms for example of number of DoF, convergence, or include more sources of uncertainty.

6.1.2 Establishing Case Studies

Survey repeatability is a significant issue for an auspicious time–lapse analysis. In practice though, this is very difficult to accomplish. Therefore, acquisition geometry has its underlying uncertainty. An ideal and comprehensive framework would be able to account for that and provide uncertainty analysis considering various sources.

One could argue that any framework is not fully proved unless it is applied on real–data. However, getting prestack seismic data is not an easy task since they tend to be proprietary information owned by oil companies. Equinor's latest release of the Volve data set is a possible solution. The particular release includes all subsurface and production data from the field, making it an ideal candidate for both research and algorithm development purposes. Also, laboratory data are typically acquired in well–controlled environments, making them excellent candidates for time–lapse analysis. Under these circumstances, different time-lapse scenarios can be easily tested and verified, such as pressure effects and nonlinearity.

Bibliography

[1] N. P. Agostinetti and A. Malinverno. Assessing uncertainties in high-resolution, multifrequency receiver-function inversion: A comparison with borehole data. *Geophysics*, 83(3):KS11–KS22, 2018.

[2] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete Cosine Transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974.

[3] M. Akbarabadi, M. Borges, A. Jan, F. Pereira, and M. Piri. A bayesian framework for the validation of models for subsurface flows: synthetic experiments. *Computational Geoscience*, 19:1231–1250, 2015.

[4] T. Alkhalifah and Y. Choi. From tomography to full-waveform inversion with a single objective function. *Geophysics*, 79(2)(2):R55–R61, 2014.

[5] C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18:343–373, 2008.

[6] R. Asnaashari, R. Brossier, S. Garambois, F. Audebert, P. Thore, and J. Virieux. Time-lapse seismic imaging using regularized full-waveform inversion with a prior model: which strategy? *Geophysical Prospecting*, 63(1)(1):78–98, 2015.

[7] G. E. Backus and J. F. Gillbert. Uniqueness in the inversion of inaccurate gross earth data. *Philos. Trans. R. Soc. London, Ser. A.*, 266:123–192, 1970.

[8] O. Barkved and T. Kristiansen. Seismic time-lapse effects and stress changes: Examples from a compacting reservoir. *The Leading Edge*, 24:1244–1248, 2005.

[9] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic Differentiation in Machine Learning: a Survey. *Journal of Machine Learning Research*, 18:1–43, 2018.

[10] T. Bayes and R. Price. An essay towards solving a problem in the doctrine of chance. *Phil. Trans. R. Soc.*, 53:370–418, 1763.

[11] E. Baysal, D. D. Kosloff, and J. W. C. Sherwood. Reverse time migration. *Geophysics*, 48:1514–1524, 1983.

[12] R. E. Bellman. Adaptive control processes. *Princeton University Press, Princeton, NJ*, 1961.

[13] M. Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv: 1701.02434 [stat.ME]*, 2017.

[14] B. Biondi and A. Almomin. Simultaneous inversion of full data bandwidth by tomographic full-waveform inversion. *Geophysics*, 79(3):WA129–WA140, 2014.

[15] T. Bodin, M. Sambridge, H. Tkalčić, P. Arroucau, K. Gallagher, and N. Rawlinson. Transdimensional inversin of receiver functions and surface wave dispersion. *Journal of Geophysical Research: Solid Earth*, 117(B2), 2012.

[16] R. Bracewell. The Fourier Transform and Its Applications. *McGRaw Hill*, 2000.

[17] E. Brechet, H. Jorge, P. Turpin, and S. Maultzsch. 4d fast track advanced interpretation on dalia field, block 17, angola. *72nd Conference and Exhibition, EAGE*, 2010.

[18] F. Broggini, M. Vasmel, J. O. A. Robertsson, and D. J. van Manen. Immersive boundary conditions: Theory, implementation, and examples. *Geophysics*, 82(3):T97–T110, 2017.

[19] S. Brooks, A. Gelman, G. Jones, and X. L. Meng. Handbook of markov chain monte carlo. *CRC Press*, 2011.

[20] R. Brossier, S. Operto, and J. Virieux. Robust elastic frequency-domain full-waveform inversion using the l_1 norm. *Geophysical Research Letters*, 36, 2009.

[21] R. Brossier, S. Operto, and J. Virieux. Which data residual norm for robust elastic frequency-domain full waveform inversion? *Geophysics*, 75(3):R37–R46, 2010.

[22] K. P. Bube and R. T. Langan. Hybrid l_1/l_2minimization with applications to tomography. *Geophysics*, 62:1183–1195, 1997.

[23] C. Bunks, F. M. Saleck, S. Zaleski, and G. Chavent. Multiscale seismic waveform inversion. *Geophysics*, 60:1457–1473, 1995.

[24] Y. Capdeville, L. Guillot, and J. J. Marigo. 1-D non-periodic homogenization for the seismic wave equation. *Geophysical Journal International*, 181(2):897–910, 2010.

[25] C. Carroll. Personal blog: https://colindcarroll.com. 2012.

[26] P. W. Cary and C. H. Chapman. Automatic 1-D waveform inversion of marine seismic refraction data. *Geophys. J.*, 93:527–546, 1988.

[27] K. M. Conrad, F. Kazunori, and M. Akira. Adjoint Hamiltonian Monte Carlo algorithm for the estimation of elastic modulus through the inversion of elastic wave propagation data. *International Journal of Numerical Methods in Engineering*, page https://doi.org/10.1002/nme.6256, 2019.

[28] E. Crase, A. Pica, M. Noble, J. McDonald, and A. Tarantola. Robust elastic non-linear waveform inversion: Application to real data. *Geophysics*, 55(527-538), 1990.

[29] T. Crestaux, O. L. MaiEEtre, and J.-M. Martinez. Polynomial chaos expansion for sensitivity analysis. *Reliability Engineering & System Safety*, 94(7):1161–1172, 2009.

[30] D. Datta and M. K. Sen. Estimating a starting model for full-waveform inversion using a global optimization method. *Geophysics*, 81(4):R211–R223, 2016.

[31] H. Denli and L. Huang. Double difference elastic waveform inversion tomography in the time domain. *79th Annual International Meeting, SEG, Expanded Abstracts*, 28:2302–2306, 2009.

[32] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B.*, 195(216–222), 1987.

[33] D. Dubbledam, S. Calero, D. E. Ellis, and R. Q. Snurr. Raspa: molecular simularion software for adsorption and diffusion in flexible nanoporous materials. *Molecular Simulations*, 42:81–101, 2016.

[34] J. A. Edgar and N. Mastio. Time-lapse velocity change tomography. *79th EAGE Conference & Exhibition, Paris June 12th-15th*, DOI: 10.3997/2214-4609.201700810, 2017.

[35] M. Eldred and J. Burkardt. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 976, 2009.

[36] G. Ely, A. Malcolm, and D. P. Nicholls. Combining global optimization and boundary integral methods to robustly estimate subsurface velocity models. *85th Annual International Meeting, SEG, Expanded Abstractsth Annual International Meeting, SEG, Expanded Abstracts*, pages 3729–3733, 2015.

[37] G. Ely, A. Malcolm, and O. V. Poliannikov. Assessing uncertainties in velocity models and images with a fast nonlinear uncertainty quantification method. *Geophysics*, 83:R63–R75, 2018.

[38] G. Evensen. Sequential data assimilation with nonlinear quasi–geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research*, C5(99):143–162, 1994.

[39] R. Falahat, A. Shams, and C. Macbeth. Towards quantitative evaluation of gas injection using time-lapse seismic data. *Geophysical Prospecting*, 59:310–322, 2011.

[40] W. Feller. An introduction to probability theory and its applications. *Wiley Series*, Volume I, 3rd Edition, 1986.

[41] A. Fichtner. Full seismic waveform modeling and inversion. *Advances in Geophysical and Environmental Mechanics and Mathematics (ACEM$\hat{2}$), Springer*, 2011.

[42] A. Fichtner, A. Zunino, and L. Gebraad. Hamiltonian Monte Carlo solution

of tomographic inverse problems. *Geophysical Journal International*, 216:1344–1363, 2019.

[43] F. Gassmann. Über die elastizität poröser medien. 1951.

[44] L. Gebraad, C. Boehm, and A. Fichtner. Bayesian elastic Full–Waveform Inversion using Hamiltonian Monte Carlo. *EarthArXiv*, 2019.

[45] J. Geertsma. Land subsidence above compacting oil and gas reservoirs. *Journal of Petroleum Technology*, 25:734–744, 1973.

[46] H. D. Geiger and P. T. Daley. Finite difference modeling of the full acoustic wave equation in matlab. *CREWES Research Report*, 15, 2003.

[47] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–472, 1992.

[48] R. C. Gonzalez and R. E. Woods. Digital image processing 2nd edition. *Prentice Hall*, 2002.

[49] R. J. Greaves and T. J. Fulp. Three-dimensional seismic monitoring of an enhanced oil recovery process. *Geophysics*, 52(9):1175–1187, 1987.

[50] A. Guitton and W. W. Symes. Robust inversion of seismic data using the Huber norm. *Geophysics*, 68:1310–1319, 2003.

[51] H. Haario, E. Saksman, and J. Tamminen. An adaptive metropolis algorithm. *Bernouli*, 7(2):223–242, 2001.

[52] E. Hairer, C. Lubich, and G. Wanner. Geometric Numerical Integration: Structure - Preserving Algorithms for Ordinary Differential Equations. *Springer New York*, 2006.

[53] D. Hale. Dynamic warping of seismic images. *Geophysics*, 78(2):S105–S115, 2013.

[54] T. M. Hansen and K. S. Cordua. Efficient monte carlo sampling of inverse problems using a neural network-based forward – apllied to GPR crosshole traveltime inversion. *Geophysical Journal International*, 211(3):1524–1533, 2017.

[55] W. K. Hastings. Monte Carlo sampling methods using Markov chain and their applications. *Biometrika*, 57(1):97–109, 1970.

[56] R. Hewett and L. Demanet. pysit: Python seismic imaging toolbox v0.5: Release 0.6. 2013.

[57] E. Hicks, H. Hoeber, M. Houbiers, S. P. Lescoffit, A. Ratcliffe, and V. Vinje. Time-lapse full waveform inversion as a reservoir-monitoring tool — A North Sea case study. *The Leading Edge*, 35(10):850–858, 2016.

[58] M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1593–1623), 2014.

[59] T. Hong and M. K. Sen. A new MCMC algorithm for seismic waveform inversion and corresponding uncertainty analysis. *Geophysical Journal International*, 177:14–32, 2009.

[60] G. Huang, R. Nammour, and W. Symes. Full-Waveform Inversion via source-receiver extension. *Geophysics*, 82(3)(3):R153–R171, 2017.

[61] J. Hunziker, E. Laloy, and N. Linde. Bayesian full-waveform tomography with application to crosshole ground penetrating radar data. *Geophysical Journal International*, 218:913–931, 2019.

[62] A. K. Jain. Fundamentals of digital image processing. *Prentice Hall*, 1989.

[63] V. I. Keilis-Borok and T. B. Yanovskaja. Inverse Problems of Seismology (Structural Review). *Geophys. J. R. astr. Soc.*, 13:223–234, 1967.

[64] M. Kotsi, J. Edgar, A. Malcolm, and S. de Ridder. Combining reflection and transmission information in time – lapse velocity inversion: A new hybrid approach. *Geophysics*, 84(4), 2019.

[65] M. Kotsi and A. Malcolm. Estimating the error distribution of recovered changes in earth properties with full – waveform inversion. *13th International Conference on Mathematical and Numerical Aspects of Wave Propagation (Extended Abstract)*, pages 209–210, 2017.

[66] M. Kotsi and A. Malcolm. A statistical comparison of three 4d full-waveform inversion schemes. *SEG International Exposition and 87th Annual Meeting, Expanded Abstracts*, pages 1434–1438, 2017.

[67] M. Kotsi, A. Malcolm, and G. Ely. 4D Full-Waveform Metropolis Hastings Inversion Using a Local Acoustic Solver. *SEG International Exposition and 88th Annual Meeting, Expanded Abstracts*, pages 5323–5327, 2018.

[68] M. Kotsi, A. Malcolm, and G. Ely. 4D Multiparameter Adaptive Metropolis Hastings Inversion. *SEG International Exposition and 89th Annual Meeting, Expanded Abstracts*, pages 5290–5294, 2019.

[69] M. Kotsi, A. Malcolm, and G. Ely. Uncertainty quantification in 4d seismic imaging: a full–wavefield approach. *Geophysical Journal International*, Under review, 2019.

[70] R. Kumar, B. Willemsen, F. Herrmann, and A. Malcolm. Enabling numerically exact local solver for waveform inversion – a low-rank approach. *Computational Geosciences*, 23(4):829–847, 2019.

[71] P. Lailly. The seismic inverse problem as a sequence of before–stack migrations. *Presented at the Conference on Inverse Scattering: Theory and Application, SIAM*, 1983.

[72] J. Landa and D. Kumar. Joint inversion of 4D seismic and production data. *SPE Annual Technical Conference and Exhibition*, DOI: https://doi.org/10.2118/146771-MS, 2011.

[73] M. Landrø and J. Stammeijer. Quantitative estimation of compaction and velocity changes using 4d impedance and traveltime changes. *Geophysics*, 69:949–957, 2004.

[74] K. Lee, S. Jung, and J. Choe. Ensember smoother with clustered covariance for 3D channelized reservoirs with geological uncertainty. *Journal of Petroleum Science and Engineering*, 145:423–435, 2016.

[75] D. LeGall. A video compression standard for Multimedia Applications. *Communications of the ACM*, 34(4):46–58, 1991.

[76] B. Leimkuhler and S. Reich. Simulating Hamiltonian Dynamics. *Cambridge University Press*, page doi:10.1017/CBO9780511614118, 2005.

[77] W. Lewis, B. Starr, and D. Vigh. A level set approach to salt geometry inversion in full–waveform inversion. *SEG International Exposition and 82nd Annual Meeting, Expanded Abstracts*, pages 1–5, 2012.

[78] N. Linde and J. A. Vrugt. Distributed soil moisture from crosshole ground-penetrating radar travel times using stochastic inversion. *Vadose Zone J.*, 12:1–16, 2013.

[79] D. E. Lumley. Time-lapse reservoir monitoring. *Geophysics*, 66(1)(1):50–53, 2001.

[80] D. E. Lumley. Nonlinear uncertainty analysis in reservoir seimsic modeling and inverse problems. *SEG International Exposition and 76th Annual Meeting, Expanded Abstracts*, pages 2037–2041, 2006.

[81] D. E. Lumley, D. Adams, M. Meadows, S. Cole, and R. Ergas. 4D seismic pressure - saturation inversion at Gullfaks field, Norway. *First Break*, 21, 2003.

[82] D. Maclaurin, D. Duvenaud, and R. P. Adams. Autograd: Effortless Gradients in Numpy. *ICML AutoML Workshop*, 2015.

[83] M. Maharramov and U. Albertin. Localized image-difference wave-equation tomography. *77th Annual International Meeting, SEG, Expanded Abstracts*, pages 3009–3013, 2007.

[84] M. Maharramov and B. Biondi. Joint full waveform inversion of time-lapse seismic data sets. *84th Annual International Meeting, SEG, Expanded Abstracts*, pages 1–5, 2014.

[85] M. Maharramov, B. Biondi, and M. Meadows. Simultaneous TV-regularized time-lapse FWI with application to field data. *85th Annual International Meeting, SEG, Expanded Abstracts*, pages 1236–1241, 2015.

[86] M. Maharramov, G. Chen, P. Routh, A. I. Baumstein, S. Lee, and S. Lazaratos.

Multiscale time-domain time-lapse Full-Waveform Inversion with a model-difference regularization. *87th Annual International Meeting, SEG, Expanded Abstracts*, pages 1428–1433, 2017.

[87] A. Malcolm and D. P. Nicholls. A field expansion method for scattering by periodic multilayered media. *The Journal of the Acoustical Society of America*, 129:1783–1793, 2011.

[88] A. Malcolm and B. Willemsen. Rapid 4D FWI using a local solver. *The Leading Edge*, 35:1053–1059, 2017.

[89] A. Malcolm and B. Willemsen. Rapid 4d fwi using a local solver. *The Leading Edge*, 35:1053–1059, 2017.

[90] S. Mallick. Some practical aspects of prestack waveform inversion using a genetic algorithm: an example from the east Texas Woodbine gas sand. *Geophysics*, 64(2):326–336, 1999.

[91] G. Mavko, T. Mukerji, and J. Dvorkin. The rock physics handbook: Tools for seismic analysis of porous media. *Cambridge University Press*, 2009.

[92] G. A. McMechan. Migration by extrapolation of time-dependent boundary values. *Geophysical Prospecting*, 31:413–420, 1983.

[93] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[94] N. Metropolis and S. Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[95] K. Mosegaard and A. Tarantola. Monte Carlo sampling of solutions to inverse problems. *Journal of Geophysical Research: Solid Earth*, 100(B7):12431–12447, 1995.

[96] B. Moseley, A. Markham, and T. Nissen-Meyer. Fast approximate simulation of seismic waves with deep learning. *arXiv:1807.06873v1 [physics.geo-ph]*, 2018.

[97] W. A. Mulder and R. E. Plessix. Exploring some issues in acoustic full waveform inversion. *Geophysical Prospecting*, 56:827–841, 2008.

[98] I. M. Navon. Data assimilation for numerical weather prediction: a review. *Data assimilation for atmospheric, oceanic and hydrologic applications*, pages 21–65, 2009.

[99] R. M. Neal. Bayesian learning for neural networks. *Springer New York*, 1996.

[100] R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, page Chapter 5, 2011.

[101] Y. Nesterov. Primal–dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009.

[102] F. Oghenekohwo, H. Wason, E. Esser, and F. Herrmann. Low-cost time-lapse seismic with distributed compressive sensing — part 1: Exploiting common information among the vintages. *Geophysics*, 82(3):P1–P13, 2017.

[103] A. Orange, K. Key, and S. Constable. The feasibility of reservoir monitoring using time–lapse marine CSEM. *Geophysics*, 74(2), 2009.

[104] L. E. J. Osorio and A. Malcolm. Amplitude variation with offset (avo) inversion modeling with a local elastic solver. *SEG International Exposition and 88th Annual Meeting, Expanded Abstracts*, 2018.

[105] K. Osypov, Y. Yang, A. Fournier, N. Ivanova, R. Bachrach, C. E. Yarman, Y. You, D. Nichols, and M. Woodward. Model-uncertainty quantification in seismic tomography: method and applications. *Geophysical Prospecting*, 61:1114–1134, 2013.

[106] W. Pan, K. A. Innanen, and Y. Geng. Elastic full–waveform inversion and parametirization analysis applied to walk–away vertical seismic profile data for unconventional (heavy oil) reservoir characterization. *Geophysical Journal International*, 213(3):1934–1968, 2018.

[107] B. C. Papazachos. Introduction to Applied Geophysics. *Ziti Editions, Thessaloniki*, 1996.

[108] A. Pica, J. P. Diet, and A. Tarantola. Nonlinear inversion of seismic reflection data in a laterally invariant medium. *Geophysics*, 55(3)(3):284–292, 1990.

[109] R. E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167:495–503, 2006.

[110] R. E. Plessix, S. Michelet, H. Rynja, H. Kuehl, C. Perkins, J. W. de Maag, and P. Hatchell. Some 3D applications of full waveform inversion. *72nd Conference and Exhibition, EAGE, Workshops and Fieldtrips, Session: WS6 3D Full Waveform Inversion – A Game Changing Technique?*, 2010.

[111] E. Polak and G. Ribiere. Note sur la convergence de directions conjuguee. *Revue Francaise D'Informatique et de Recherche Operationelle*, 3:35–43, 1969.

[112] O. V. Poliannikov and A. E. Malcolm. The effect of velocity uncertainty on migrated reflectors: Improvements from relative-depth imaging. *Geophysics*, 81:S21–S29, 2015.

[113] R. G. Pratt. Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model. *Geophysics*, 64(3)(3):888–901, 1999.

[114] R. G. Pratt, C. Shin, and G. J. Hick. Gauss-Newton and Full Newton methods in frequency-space seismic waveform inversion. *Geophysical Journal International*, 133:341–362, 1998.

[115] R. G. Pratt and M. H. Worthington. Inverse theory applied to multi-source cross-hole tomography. part I: Acoustic wave-equation method. *Geophysical Prospecting*, 38:287–310, 1990.

[116] S. Qu and D. J. Verschuur. Simultaneous joint migration inversion for accurate time-lapse analysis of sparse monitor surveys. *First EAGE Workshop on Practical Reservoir Monitoring*, 2017.

[117] M. Rabbani and P. W. Jones. Digital Image Compression Techniques. *Society of Photo-Optical Instrumentation Engineers (SPIE) Bellingham, WA, USA*, 1991.

[118] E. B. Raknes, W. Weibull, and B. Arntsen. Time-lapse full waveform inversion: Synthetic and real data examples. *83rd Annual International Meeting, SEG, Expanded Abstracts*, 2013.

[119] A. Ray, S. Kaplan, J. Washbourne, and U. Albertin. Low frequency full waveform seismic inversion within a tree based Bayesian framework. *Geophysical Journal International*, 212(1):522–542, 2017.

[120] J. Rebaza. A First Course in Applied Mathematics, chapter 5. *John Wiley & Sons, Inc.*, pages 235–275, 2012.

[121] A. Richardson. Seismic Full–Waveform Inversion Using Deep Learning Tools and Techniques. *arXiv:1801.07232v2*, 2018.

[122] J. Rickett, L. Duranti, T. Hudson, B. Regel, and N. Hodgson. 4-d time strain and the seismic signature of geomechanical compaction at genesis. *The Leading Edge*, 26:644–647, 2007.

[123] H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[124] C. Robert and G. Casella. A Short History of Markov Chain Monte Carlo: Subjective Recollections from Incomplete Data. *Statistical Science*, 26(1):102–115, 2011.

[125] C. P. Robert and G. Casella. Monte Carlo statistical methods. *Springer New York*, 2nd Edition, 2004.

[126] G. O. Roberts and J. S. Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351–367, 2001.

[127] M. Rodell, P. Houser, U. Jambor, J. Gottschalck, K. Mitchell, C. J. Meng, K. Arsenault, B. Cosgrove, J. Radakovich, and M. Bosilovich. The global land data assimilation system. *Bulletin of the American Meteorological Society*, 85(3):381–394, 2004.

[128] G. Röth and A. Tarantola. Neural networks and inversion of seismic data. *Journal of Geophysical Research: Solid Earth*, 99(B4):6753–6768, 1994.

[129] P. Routh, G. Palacharia, I. Chikichev, and S. Lazaratos. Full Wavefield Inversion of time-lapse data for improved imaging and reservoir characterization. *82nd Anual International Meeting, SEG, Expanded Abstracts*, pages 1–6, 2012.

[130] M. Sambridge, K. Gallagher, A. Jackson, and P. Rickwood. Trans–dimensional inverse problems, model comparison and the evidence. *Geophysical Journal International*, 167:528–542, 2006.

[131] M. Sambridge and K. Mosegaard. Monte Carlo methods in geophysical inverse problems. *Reviews of Geophysics*, 40(3):1–29, 2002.

[132] M. K. Sen and R. Biswas. Choice of regularization weight in basis pursuit reflectivity inversion. *Journal of Geophysics and Engineering*, 12:70–79, 2015.

[133] M. K. Sen and R. Biswas. Transdimensional seismic inversion using the reversible jump Hamiltonian Monte Carlo algorithm. *Geophysics*, 82(3):R119–R134, 2017.

[134] M. K. Sen, R. Biswas, P. Mandal, and P. Kumar. Basis pursuit receiver function. *Bulletin of the seismological society of America*, 2014.

[135] P. M. Shearer. Introduction to Seismology, Second Edition. *Cambridge University Press*, 2012.

[136] J. Shragge and D. E. Lumley. Time-lapse wave-equation migration velocity analysis. *SEG International Exposition and 82nd Annual Meeting, Expanded Abstracts*, pages 1–5, 2012.

[137] M. Sinha and G. T. Schuster. Seismic time-lapse imaging using interferometric least-squares migration: Case study. *Geophysical Prospecting*, 66:1457–1474, 2018.

[138] L. Sirgue and R. G. Pratt. Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies. *Geophysics*, 69(1):231–248, 2004.

[139] R. Souza, D. E. Lumley, and J. Shragge. Estimation of reservoir fluid saturation from 4d seismic data: effects of noise on seismic amplitude and impedance attributes. *Journal of Geophysics and Engineering*, 14:51–68, 2017.

[140] G. Stuart, S. E. Minkoff, and F. Pereira. Enhanced neural network sampling for two–stage Markov Chain Monte Carlo seimic inversion. *SEG International Exposition and 89th Annual Meeting, Expanded Abstracts*, pages 1670–1674, 2019.

[141] G. Stuart, W. Yang, S. E. Minkoff, and F. Pereira. A two-stage Markov chain Monte Carlo methd for velocity estimation and uncertainty quantification. *SEG International Exposition and 86th Annual Meeting, Expanded Abstracts*, pages 3682–3687, 2016.

[142] G. K. Stuart, S. E. Minkoff, and F. Pereira. A two-stage Markov chain Monte Carlo for seismic inversion and uncertainty quantification. *Geophysics*, 84(6), 2019.

[143] T. J. Sullivan. Introduction to uncertainty quantification. *Springer*, 63, 2015.

[144] J. Sun, Z. Niu, K. A. Innanen, J. Li, and D. O. Trad. A theory-guided deep-learing formulation and optimization of seismic waveform inversion. *Geophysics*, 85(2):R87–R99, 2020.

[145] A. Tarantola. Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266, 1984.

[146] A. Tarantola. Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation. *Elsevier, New York*, 1987.

[147] A. Tarantola. Inverse problem theory: Methods for data fitting and parameter estimation. *Elsevier Science*, 1987.

[148] A. Tarantola and B. Valette. Generalized nonlinear inverse problems solved using the least squares criterion. *Reviews of Geophysics*, 20:219–232, 1982.

[149] M. Tatanova and P. Hatchell. Time-lapse avo on deepwater OBN seismic at the mars field. *SEG International Exposition and 82nd Annual Meeting, Expanded Abstracts*, 2012.

[150] K. Terzaghi. Relation Between Soil Mechanics and Foundation Engineering. *Proceedings, 1st International Conference on Soil Mechanics and Foundation Engineering, Boston*, 3(13–18), 1936.

[151] J. Thurin, R. Brossier, and L. Metivier. An ensemble-transform Kalman filter: Full - waveform inversion scheme for uncertainty quantification. *SEG International Exposition and 87th Annual Meeting, Expanded Abstracts*, pages 1307–1313, 2017.

[152] J. Thurin, R. Brossier, and L. Metivier. Ensemble–based uncertainty estimation in full waveform inversion. *Geophysical Journal International*, 219(3):1613–1635, 2019.

[153] E. Tolstukhin, B. Lyngnes, and H. H. Sudan. Ekofisk 4d seismic-seismic history matching workow. *SPE Europec/EAGE Annual Conference*, 2012.

[154] A. Tura, T. Barker, P. Cattermole, C. Collins, J. Davis, P. Hatchell, K. Koster, P. Schutjens, and P. Wills. Monitoring primary depletion reservoirs using amplitudes and time shifts from high–repeat seismic surveys. *The Leading Edge*, 24:1214–1221, 2005.

[155] D. J. van Manen, J. O. A. Robertsson, and A. Curtis. Exact wave field simulation for finite-volume scattering problems. *The Journal of the Acoustical Society of America*, 122(4):EL115–EL121, 2007.

[156] V. Vapnik. The Nature of Statitstical Learning Theory. *Springer New York*, 1995.

[157] M. Vasmel and J. O. A. Robertsson. Exact wavefield reconstruction on finite-difference grids with minimal memory requirements. *Geophysics*, 81(6):T303–T309, 2016.

[158] T. Vdovina and S. Minkoff. A priori error analysis of operator upscaling for the acoustic wave equation. *International Journal of Numerical Analysis and Modeling*, 5(543–569), 2008.

[159] R. Versteeg. The marmousi experience: Velocity model determination on a synthetic complex data set. *The Leading Edge*, 13:927–936, 1994.

[160] J. Virieux, A. Asnaashari, R. Brossier, L. Metivier, A. Ribodetti, and W. Zhou. An introduction to full waveform inversion. *Encyclopedia of exploration geophysics: SEG*, pages R1–1 – R1–40, 2017.

[161] J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74:WCC1–WCC26, 2009.

[162] F. Waldhauser and W. L. Ellsworth. A double difference earthquake location algorithm: method and application to the northern hayward fault, california. *Bulletin of the seismological society of America*, 90:1353–1368, 2000.

[163] G. K. Wallace. The JPEG Still Picture Compression Standard. *Communications of the ACM*, 34(4):30–40, 1991.

[164] Z. Wang. Fundamentals of seismic rock physics. *Geophysics*, 66:398–412, 2001.

[165] C. P. A. Wapenaar and A. J. Berkhout. Elastic wave field extrapolation, redatuming of single- and multi- component seismic data. *Advances in Exploration Geophysics, Elsevier*, 1989.

[166] K. Wapenaar, J. Thorbecke, J. van der Neut, F. Broggini, E. Slob, and R. Snieder. Marchenko imaging. *Geophysics*, 79(3)(3):WA39–WA57, 2014.

[167] T. Watanabe, S. Shimizu, E. Asakawa, and T. Matsuoka. Differential waveform tomography for time-lapse crosswell seismic data with application to gas hydrate production monitoring. *SEG Technical Program Expanded Abstracts*, 2005.

[168] B. Willemsen and A. Malcolm. Coupling a local elastic solver to a background acoustic model to estimate phase variation. *SEG International Exposition and 86th Annual Meeting, Expanded Abstracts*, 2016.

[169] B. Willemsen, A. Malcolm, and W. Lewis. A numerically exact local solver applied to salt boundary inversion in seismic full waveform inversion. *Geophysical Journal International*, 204:1703–1720, 2016.

[170] P. R. Williamson, A. J. Cherrett, and P. A. Sexton. A new approach to warping for quantitative time–lapse characterisation. *69th EAGE Conference and Exhibition incorporating SPE EUROPEC 2007*, 2007.

[171] D. Yang, M. Fehler, A. Malcolm, F. Liu, and S. Morton. Double difference waveform inversion of 4d ocean bottom cable data: Application to valhall, north sea. *SEG Technical Program Expanded Abstracts*, pages 4966–4970, 2013.

[172] D. Yang, F. Liu, S. Morton, A. Malcolm, and M. Fehler. Time-lapse full-waveform inversion with ocean-bottom-cable data: Application on Valhall field. *Geophysics*, 81(4):R225–R235, 2016.

[173] D. Yang, A. Malcolm, and M. Fehler. Time-lapse full waveform inversion and uncertainty analysis with different survey geometries. *76th EAGE Conference and Exhibition*, 2014.

[174] D. Yang, A. Malcolm, and M. Fehler. Using image warping for time-lapse image domain wavefield tomography. *Geophysics*, 79(3)(3):WA141–WA151, 2014.

[175] D. Yang, M. Meadows, P. Inderwiesen, J. Landa, A. Malcolm, and M. Fehler. Double difference waveform inversion: Feasibility and robustness study with pressure data. *Geophysics*, 80:M129–M141, 2015.

[176] T. Yang, J. Shragge, and P. Sava. Illumination compensation for image-domain wavefield tomography. *Geophysics*, 78(5)(5):U65–U76, 2013.

[177] S. Yuan, N. Fuji, S. Singh, and D. Borisov. Localized time-lapse elastic inversion using wavefield injection and extrapolation: 2-D parametric studies. *Geophysical Journal International*, 209(3):1699–1717, 2017.

[178] R. Zhang and J. Castagna. Seismic sparse-layer reflectivity inversion using basis pursuit decomposition. *Geophysics*, 76(6):R145–R158, 2011.

[179] X. Zhang, A. Curtis, E. Galetti, and S. de Ridder. 3-D Monte Carlo surface wave tomography. *Geophysical Journal International*, 215:1644–1658, 2018.

[180] Z. Zhang and L. Huang. Double-difference elastic-waveform inversion with prior information for time-lapse monitoring. *Geophysics*, 778:R259–R273, 2013.

[181] Z. Zhao and M. K. Sen. A gradient based MCMC method for FWI and uncertainty analysis. *SEG International Exposition and 89th Annual Meeting, Expanded Abstracts*, pages 1465–1469, 2019.

[182] Y. Zheng, P. Barton, and S. Singh. Strategies for elastic full waveforms inversion of time-lapse ocean bottom cable (obc) seismic data. *81st Annual International Meeting, SEG, Expanded Abstracts*, pages 4195–4200, 2011.

[183] W. Zhou, R. Brossier, S. Operto, and J. Virieux. Full waveform inversion of diving & reflected waves for velocity model building with impedance inversion based on scale separation. *Geophysical Journal International*, 202(3):1535–1554, 2015.

Appendix A

Automatic Gain Control (AGC) application on the misfit function from Chapter 3

The objective function with the AGC is

$$\hat{w}(m|m_0;x,z) = \underset{w}{\operatorname{argmin}}||AGC(I_1(m;x,z)) - AGC(I_0(m_0;x,z+w(x,z)))||_2^2, \quad \text{(A.1)}$$

where $I_1(m)$ is the monitor image that updates at each iteration and $I_0(m)$ is the baseline image that remains constant. If we now take the gradient of Equation (A.1)

$$\frac{\partial \hat{w}}{\partial m} = 2\left(AGC(I_1(m)) - AGC(I_0(m))\right)\frac{\partial AGC(I_1(m))}{\partial m}, \quad \text{(A.2)}$$

and apply the chain rule to the last term we get

$$\frac{\partial AGC(I_1(m))}{\partial m} = \frac{\partial AGC(I_1(m))}{\partial I_1(m)}\frac{\partial I_1(m)}{\partial m}. \quad \text{(A.3)}$$

We argue that $\frac{\partial AGC(I_1(m))}{\partial I_1(m)}$ is approximately constant (or varies only slowly as a function of space), and therefore does not have a big effect on the final gradient of the objective function. In a Taylor series approach

$$AGC(I_1(m) + \delta I) = AGC(I_1(m)) + \frac{\partial AGC(I_1(m))}{\partial I_1(m)} \delta I + ..., \tag{A.4}$$

where we neglect higher order terms. Then

$$\frac{\partial AGC(I_1(m))}{\partial I_1(m)} \approx \frac{AGC(I_1(m) + \delta I) - AGC(I_1(m))}{\partial I_1(m)}. \tag{A.5}$$

is small when the window size is large. In other words, if the window size is large enough the AGC is approximately a constant scaling. If the image change is small enough, that constant will be the same in the perturbed and unperturbed images. More formally when $\frac{\partial I_1(m)}{\partial m}$ varies on a faster scale than $\frac{\partial AGC(I_1(m))}{\partial I_1(m)}$, we expect that the AGC value between the two images will be similar and that this factor varies slowly across the image (this factor is approximately constant).