Time-Lapse Full Waveform Inversion Methods

by

Wubshet M. Alemie

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Geophysics

Department of Physics
University of Alberta

© Wubshet M. Alemie, 2017

Abstract

In the time-lapse seismic method, often referred to as 4D seismic, a series of seismic data sets are acquired at different times to study the temporal variation of a target subsurface reservoir. This technique is being used as a subsurface monitoring tool in the oil and gas industry for fluid driven production enhancement as well as for environmental monitoring purposes in CO_2 sequestration.

Assuming that the geological structures are time-invariant, the expected time-lapse changes are due to temporal variations in the reservoir. These changes, for instance, occur due to injection or depletion of fluid into or from the reservoir that creates a dynamic process. However, the time-lapse signature that can be obtained from time-lapse seismic data does not always reflect lithological changes. Time-lapse data are also liable to contamination by acquisition and processing artifacts. A major contributor to artifacts in the time-lapse signature is the non-repeatability of the seismic experiment.

The qualitative interpretation that can be made by analyzing amplitude changes and time shifts on post-stack time-lapse seismic data are often not sufficient for a complete understanding of detailed reservoir conditions. Thus, to quantitatively characterize the dynamic of a reservoir, one should adopt pre-stack imaging and inversion algorithms that yield high-resolution images of physical parameters. Traditionally, time-lapse seismic inversion is carried out by processing baseline and monitor data-sets independently. Then, the difference between baseline and monitor surveys are computed.

Artifacts that accumulate as a result of acquisition and data processing errors may often have a strength that is comparable to the actual time-lapse signal of interest. Consequently, artifacts can have negative impacts on the interpretation of time-lapse signals. Therefore,

developing seismic inversion methods that are capable of suppressing the aforementioned artifacts is a problem of practical interest.

In this thesis, the current full waveform inversion algorithms are reviewed. The performance of four optimization methods (gradient descent, approximate diagonal Hessian, Gauss-Newton, and Quasi-Newton) are assessed with a variety of numerical examples. The primary goal is to extend the traditional full waveform inversion algorithm into a time-lapse inversion algorithm. In particular, a new time-lapse full waveform inversion algorithm, namely, Joint Reparameterized Time-Lapse Full Waveform Inversion (JRTL FWI) is proposed in comparison with the traditional algorithms that include independent and double difference inversions. This algorithm allows passing a prior information from baseline to the monitor velocity models. Therefore, it plays a role of regularization.

To demonstrate the JRTL FWI with synthetic examples, the seismic data are modeled via a 2D frequency domain acoustic wave equation with Perfectly Matched Layer (PML) boundary conditions. The inversion algorithm is derived using a least squares data misfit that is minimized via the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization method. Two synthetic velocity models are considered in various time-lapse seismic monitoring settings. The examples show that when models have high complexity and the seismic experiments are highly non-repeatable, the proposed algorithm improves the quality of time-lapse velocity signature as compared to the traditional ones.

This thesis is dedicated to

My Brother, Zewdu M. Alemie (1968-2016), Rest In Peace

Acknowledgements

My special thanks go to my supervisor, Mauricio D. Sacchi, for his unreserved help and encouragement throughout the PhD research. The successful completion of the thesis would not be possible without his sustained follow-ups and support.

I would like to express my warmest thanks to all the members of Signal Analysis and Imaging Group (SAIG) for the valuable discussions in our research as well as for their kind cooperation during my stay in the group.

I would like to thank members of my supervisory committee, Douglas R. Schmitt and Mirko van der Baan, for their helpful commentaries and suggestions in the completion of my thesis.

I would like to thank the graduate student advisor, Sarah MacKinnon, and the associate chair of graduate studies, Richard Marchand, for their continued assistance throughout the program.

I am also indebted to Ayesheshim Kebie and Tadela Chekol for their encouragement and unreserved hospitality.

Finally, I would like to thank my parents, sisters, and brothers who propel me to dream and do things bigger than myself. It would not be possible without their absolute comfort and motivation.

Contents

List of Tables

List of Figures

List of Symbols

$\mathbf{A}$	Discrete matrix form of Helmholtz operator
$\mathbf{d}$	Seismic data
$\mathbf{D}_x$	Differencing operator in the x-direction
$\mathbf{D}_z$	Differencing operator in the z-direction
f	Seismic source
$\mathbf{F}$	Jacobian matrix
$\mathbf{g}$	Gradient
$\mathbf{H}$	Hessian
$\mathbf{H}_a$	Approximate Hessian
$\mathbf{H}_{diag}$	Diagonal Hessian
J	Objective (cost) function
K	Bulk modulus
L	Lagrangian
$\mathbf{m}$	Model parameter
P	Pressure wave-field
$\mathbf{r}$	Reflectivity
$\mathbf{R}$	Receiver sampling operator
s	slowness
V_p or c	P-wave velocity
V_s	S-wave velocity
ρ	Density
κ	Bulk modulus
μ and λ	Lame parameters
μ	Regularization parameter
γ	Step length
ω	temporal frequency

List of Abbreviations

1D	One spacial dimension
2D	Two spacial dimension
3D	Three spacial dimension
4D	Four dimensional (three spacial coordinate and time)
AVO	Amplitude Variation with Offset
ABC	Absorbing Boundary Conditions
CCS	Carbon Capture and Storage
CPU	Central Processing Unit
CG	Conjugate Gradient
DH	Diagonal Hessian
ECMB	Enhanced Coal-bed Methane Recovery
EOR	Enhanced Oil Recovery
FWI	Full Waveform Inversion
GD	Gradient Descent
GN	Gauss Newton
JRTL FWI	Joint Reparametrized Time-Lapse Full Waveform Inversion
L-BFGS	Limited Memory - Broyden-Fletcher-Goldfarb-Shanno
LU	Lower-Upper triangular matrix decomposition
MUMPS	MUltifrontal Massively Parallel sparse direct Solver
PML	Perfectly Matched Layers
PSDM	Pre-Stack Depth Migration
PSTM	Pre-Stack Time Migration
RTM	Reverse Time Migration
SM	Smooth
SNR	Signal to Noise Ratio
TV	Total Variation

CHAPTER 1

Introduction

1.1 4D Seismic

Injection or depletion of fluids into or from subsurface reservoirs over time creates a time-variant process. This dynamic process changes the saturation, pressure, and temperature of the reservoir, which leads to changes in the seismic velocity and density of underlying formations. Time-lapse seismic, also known as 4D seismic, is a monitoring method of such dynamic reservoirs by acquiring time-lapse data from repeated 3D seismic surveys (Lumley, 1995). The data obtained from repeated surveys of the same reservoir undergo seismic processing steps followed by qualitative and quantitative interpretation. In addition to 4D seismic, time-lapse surface-to-borehole VSP and borehole-to-borehole cross-well seismic geometries are also seismic methods, which can be used for monitoring purposes (O'Brien et al., 2004; Daley et al., 2008)

Time-lapse seismic signals provide information about the change in saturation, pressure, and temperature of reservoirs. Extraction of useful information is possible only if the signals are detectable and the noise is insignificant. However, this is not always the case. The signals could be small and contaminated with noise, which makes reservoir monitoring uncertain. Before taking expensive 4D data acquisition risks, the first phase in 4D seismic is carrying out a careful feasibility study on the target reservoir. The later is needed to provide insight into whether or not 4D seismic monitoring is applicable in the region of interest before deploying 4D data acquisition. Feasibility for successful time-lapse studies depends on the fluids in the reservoir, rock types, reservoir thickness, depth and structure and the cost-effectiveness of acquiring high-quality data (Holberg, 1997).

Time-lapse studies require a multi-disciplinary approach such as reservoir simulation, rock physics and zero-dimensional and one-dimensional seismic analysis from well-logs followed by 3D baseline survey to estimate detectability and interpretability of expected time-lapse signal (Lumley, 2001). Next, I briefly describe each of the four most important elements towards an integrated feasibility study: geological modeling, reservoir simulation, rock physics, and seismic modeling.

1.1.1 Geological modelling

Building a geological model of a region of interest is the first step in the course of the feasibility study. It mainly takes into account the stratigraphy, porosity, and facies, which represents the structure and physical properties of the reservoir rocks. The model can be built from well-log, available seismic data, laboratory core measurements, and geologic information.

1.1.2 Reservoir Simulation

Reservoir simulation is the widely used method of understanding how fluids flow with in a reservoir given a geologic model and necessary fluid properties. The simulation is based on a mass conservation mathematical model via a numerically solvable differential equation. The simulation provides time dependent pressure and fluid saturation of the reservoir.

A coupled geomechanics and reservoir simulation has also been suggested. Several authors (Rhett, 1998; Chin et al., 2002; Sen and Settari, 2005) have created geomechanical and fluid flow models of compacting reservoirs based on laboratory studies of reservoir rocks. Minkoff et al. (2004) tried to predict production in compacting reservoirs by linking geomechanics and flow models with time-lapse seismic. The computed pore pressure from reservoir simulation is used in geomechanical modeling. The geomechanics, in turn, provides porosity feedback to the flow simulation (Vidal et al., 2001; Minkoff et al., 2004). The outcome of a refined coupled simulation can be used in rock physics to estimate more realistic elastic parameters.

1.1.3 Rock Physics

Rock physics is an integral part of reservoir monitoring feasibility studies. It delivers the relationship between known static and dynamic reservoir properties of the reservoir rock. The time-dependent pressure and fluid saturation resulted from reservoir simulation are input to the rock physics analysis. Usually, Gassmann's fluid substitution model that links saturated bulk modulus with dry bulk modulus is used (Gassmann, 1951).

Changes in saturated bulk modulus and shear modulus result in changes in seismic response and physical properties of the reservoir namely P-wave velocity (V_p), S-wave velocity (V_s), bulk density (ρ), and Poisson's ratio that give insight to different reservoir scenarios. Gassmann's fluid substitution model, however, has limitations (Mavko and Jizba, 1991; Mavko et al., 1998; Batzle et al., 2001; Smith et al., 2003). The assumptions and the limitations of the fluid substitution model are presented in detail in Chapter 6.

1.1.4 Synthetic Seismic Modelling

The seismic signal due to the reservoir condition can be significant enough to be detected or inadequate. The strength of the seismic response depends on the rock properties, depth of the reservoir, fluid saturation, pressure, and temperature. Therefore, it is always important to evaluate the magnitude and interpretability of the expected 4D seismic response by generating synthetic seismic data based on the results of reservoir simulation and rock physics analysis. By doing so, one can create multiple scenarios by changing reservoir parameters and fluid saturation before and after fluid substitution.

Well-logs are most reliable and primary sources of information about the underlying physical parameters such as V_p, V_s, and ρ. Using the Gassmann's fluid substitution model, one can generate corresponding V_p, V_s, and ρ models by perturbing the baseline well logs with various fluid saturation levels. Typical examples for such a case are oil reservoir swept with steam flood and CO_2 injection into a saline aquifer (brine). In the first instance, the steam (water) replacing oil. In the second case, CO_2 is replacing brine. The direct comparison of the parameters at different saturation level provides insight into the behavior of the reservoir. One can also derive synthetic data out of V_p, V_s, and ρ based on convolution and ray tracing by generating reflection coefficient using Zoeppritz equations or with sophisticated seismic modeling techniques based on finite difference or finite element methods.

Although well-logs provide information about the reservoir, the information is limited to a 1D model. In other words, lateral reservoir variations between wells can not be captured by well-log information. Therefore, 4D synthetic modeling is needed by integrating the geological model, time-dependent parameters resulting from reservoir simulation and rock physics. Then, generate time-dependent V_p, V_s, and ρ and use this time-dependent reservoir information to model repeated seismic surveys. This 4D synthetic seismic model gives us insight into the temporal dynamics of the reservoir.

1.2 Application of 4D seismic

In the last couple of decades, time-lapse (4D) seismic has transformed significantly from an academic research topic to a standard industry workflow for monitoring reservoirs worldwide.

Time-lapse data provide information in inter-well regions; regions which are not sampled by wells. It has been used successfully in oil and gas industry for enhanced recovery to improve quantitative reservoir characterization, identifying movement of fluid interfaces, and locating bypassed reserves. Apart from Enhanced Oil Recovery (EOR), 4D seismic is also being applied to monitor CO_2 geo-sequestration reservoirs, which aims to reduce greenhouse gas (GHG) emissions.

To mention few successful case histories: in the North Sea this has been achieved largely by the acquisition of repeated marine streamer data and has been applied most widely on fields with aquifer drive (e.g. the Gannet fields; Kloosterman et al. (2003)) or water injection drive (e.g. Draugen Field; Koster et al. (2000)). Other saturation change signals such as gas flood (e.g. Troll West; Eikeberg and Elde (2002)), solution gas breakout (e.g. Foinaven Field; Jack and Andrew (2000)), and steam injection (e.g. Duri Field; Sigit et al. (1999)) have also proved strong signals for monitoring. The method is now being used to provide evidence for the containment of CO_2 in the Sleipner CO_2 sequestration project (Chadwick et al., 2008). 4D seismic has been successfully used as a tool to monitor reservoir in the case of enhanced oil recovery process in many reservoirs. These include Gullfaks Field, Norway; Duri Field, Indonesia; Northern Alberta Field, Canada; Weyburn Field, Saskatchewan, Canada; Fulmar Field, UK; Holt Field, USA, and Balol Field; India (Brown, 2004).

A successful 4D seismic monitoring comes with a lot of challenges throughout each phase of its implementation: feasibility study, data acquisition and processing, and seismic imaging. Improving the available methods and developing new techniques, which increase the capability of 4D seismic monitoring, is crucial. The next sections are devoted to reviewing the basics of 4D seismic and works that have been done to date.

1.3 4D Seismic Data Acquisition and Processing

The value of the 4D seismic data and their contribution to reservoir monitoring depends on resolution and signal-to-noise ratio. These are, in turn, controlled by data acquisition and processing and by the specific geological environment for a specific reservoir.

1.3.1 4D Seismic Data Acquisition

The time-lapse seismic acquisition is 3D seismic surveys taken in the same geographic location at different times. The acquired seismic data are used to determine the changes occurring in the reservoir as a result of hydrocarbon production or injection of water or gas into the reservoir by comparing the repeated data sets. However, the changes in the repeated

seismic data are not purely from reservoir changes; the data may have changes unrelated to reservoir changes. This issue leads us to the most important subject of 4D data acquisition, repeatability. It is a measure of similarity in the data acquired from repeated surveys; the lesser is the noise from one survey to the other, the higher is the repeatability and vice versa. The main factors that likely affect repeatability include: (Johnston, 2013)

- **Survey Acquisition Geometries:** sail line orientation and heading, source-receiver spacing, streamer feathering, source and receiver depths, coverage influenced by obstructions.
- **Near-Surface Conditions:** variations in statics and receiver coupling.
- **Environment:** sea level, sea state and swell noise, water temperature, salinity, and ground water level.
- **Noise:** shot-generated noise, residual multiples, and shallow gas.
- **Geology:** steep dips, faults, shadows, and producing reservoirs.

In general, both onshore and offshore 4D data acquisition have advantages and disadvantages. Onshore 4D data acquisition is less expensive as compared to offshore acquisition, and geometry noise can be reduced. But, the data can be noisy due to surface generated noise. Offshore 4D data acquisition has high geometry noise relative to onshore, and it is expensive.

Furthermore, 4D seismic data are more likely to be repeatable in clastic sedimentary rocks than in carbonate reservoirs. Because the acoustic response of carbonates is highly variable (heterogeneous) and the changes in seismic parameters are smaller than the background noise (Raef et al., 2005; Landrø, 2006). As a result, most successful 4D seismic applications are reported in high porosity clastic reservoirs than stiff or low porosity carbonate reservoirs (Lumley, 2001; Raef et al., 2005). Regardless of the location of the time-lapse data acquisition and the kind of reservoir rocks under consideration, a feasibility study is the fundamental step for deciding a 4D acquisition.

1.3.2 4D Seismic Data Processing

The goal of 4D seismic processing is to maximize the repeatability of time-lapse data and preserve the true time-lapse difference signals. Conventional data processing techniques use cross-matching or cross-equalization tools either at pre-imaging or post-imaging phase to suppress time-lapse difference artifacts (Lumley, 2001). The cross-equalization tools include:

- survey alignment to a common geometric grid, bandwidth, and phase matching to compensate for different source wavelets,

- amplitude normalization to scale the data to the same average levels,
- residual migration to correct for lateral and vertical misalignment resulting from different migration velocities and imaging algorithms, and
- residual time alignment or warping to correct for different statics, local migration differences, and velocity changes in the overburden or reservoir.

Advanced pre-stack and post-stack seismic 3D processing algorithms can also be used to attenuate time-lapse artifacts on time-lapse data by applying them, consistently (Ross et al., 1996; Rickett and Lumley, 2001). However, processing time-lapse data with the existing non-time-lapse 3D algorithms may not be accurate. A failure in understanding image events on the baseline survey leads us to categorize them as noise. If those events are highly repeatable on multiple time-lapse data, they should be regarded as real signals (Lumley, 2001).

The cross-equalization tools and the existing 3D processing algorithms are not adequate, and they have their limitations. It is necessary to extend the traditional migration and inversion algorithms to time-lapse algorithms that can handle time-lapse difference artifacts. Thus, developing new processing techniques unique to time-lapse seismic data is a problem of practical interest.

1.4 Migration and Inversion of 4D Seismic Data

Seismic migration and inversion are an integral part of seismic processing and interpretation. They provide detailed subsurface information from structural imaging to elastic parameter estimation. These techniques are well established in oil and gas industry toolkits and are being used with success. 2D/3D migration and inversion algorithms, which are used for non-time-lapse data, can also be used directly for time-lapse data. Next, I briefly discuss the applications of seismic migration and inversion in the framework of 4D seismic.

1.4.1 4D AVO Inversion

Amplitude Variation with Offset (AVO) analysis and AVO inversion are useful component of qualitative and quantitative interpretation of 4D seismic monitoring. AVO inversion, in particular, is based on an approximation of Zoeppritz equations, which can be used to invert P-wave and S-wave impedance or other parameters from seismic data. The changes in P-wave and S-wave impedance in a 4D seismic scenario can be linked to a change in saturation and pressure.

Under the restrictive assumptions of Gassmann, the S-wave impedance is insensitive to fluid saturation. Therefore, AVO inversion attributes can be used to differentiate whether the change in AVO is due to saturation and pressure effects (Cole et al., 2002). The saturation and pressure information, in turn, provides useful information about the dynamics of the reservoir. Landrø et al. (2003) pointed out that a combined use of PP and PS time-lapse data help to reduce uncertainty in saturation-pressure discrimination. Chu and Gist (2010) also proposed a method for estimating changes in pressure and saturation that uses a model-based inversion by combining rock-physics and statistical approaches. Other related works on elastic impedance inversion can also be found in Sarkar et al. (2003), Buland and El Ouair (2006), and Lafet et al. (2009).

1.4.2 4D Seismic Data Migration

Pre-Stack Time Migration (PSTM) is the most common imaging method for 4D projects. Many case studies have published migration results based on PSTM. Pre-Stack Depth Migration (PSDM) is also required for structural imaging and 4D processing. A comparison of PSTM and PSDM was done by Kvalheim et al. (2007) for a faulted North Sea field; they found that 3D amplitudes and positioning in the PSDM data superior to the PSTM results and the 4D differences have better consistency and coherency and are easier to interpret. Brain et al. (2009) presented an example from a North Sea field where they carefully constructed a velocity model that incorporated laterally varying anisotropy. The PSDM showed a better performance in structural imaging and a 4D interpretation of water sweep beneath a gas chimney where PSTM had failed. Miller and Helgerud et al. (2011) also showed the effect of a post-migration processing step of residual-curvature correction in changing the normalized root mean square value of PSTM and PSDM. They found that PSDM significantly improved repeatability in contrast to PSTM.

Ray based migrations such as Kirchhoff migration and beam migration are limited in their ability to handle sharp velocity contrasts and rapid lateral variations in the overburden. Wave-equation migration such as Reverse Time Migration (RTM) is used in 4D projects particularly for ocean bottom node technology in deep water (Stopin et al., 2011). Linearized time-lapse least-square inversion by Ayeni and Biondi (2011) introduced a time-lapse inversion algorithm by incorporating temporal and spatial regularization to invert baseline and monitor data jointly.

1.4.3 4D Full Waveform Inversion

Full waveform inversion (FWI) provides high-resolution complex geological information from seismic data to characterize the subsurface qualitatively and quantitatively. The

current advanced class of imaging method is the least squares imaging based on seismic wave equation in the framework of inverse theory. It is a means of extracting model parameters by minimizing residuals between experimentally observed data and data obtained from physical modeling (Lailly, 1983; Tarantola, 1984a). It has been used in exploration geophysics with success for high-resolution subsurface parameter estimation (Pratt, 1999; Hu et al., 2009; Brossier, 2011; Ma and Hale, 2012).

Acoustic FWI allows us to retrieve the P-wave velocity directly from observations. The elastic version of FWI provides P-wave velocity, S-wave velocity, and density. As pointed out above, the changes in elastic parameters in a 4D sense can directly be linked to changes in saturation and pore pressure of the reservoir due to fluid flow. Therefore, FWI can be an excellent tool for reservoir monitoring. However, reports on the application of FWI for 4D data are limited. Barkved and Kristiansen (2005) showed the application of FWI to improve imaging at the Valhall Field. Queißer and Singh (2013) inverted 2D lines from 4D surveys acquired over the Sleipner CO_2-sequestration project to estimate gas saturation. Routh et al. (2012) demonstrated that broadband FWI could appropriately image the subsurface when the overburden changes and can recover high-resolution 4D anomalies.

1.5 Motivation and Research Goals

A qualitative interpretation that can be made by analyzing amplitude changes and time shifts on post stack time-lapse seismic data (Watts et al., 1996; Burkhart et al., 2000; Vedanti and Sen, 2009; Isaac and Lawton, 2006) are often not sufficient for a complete understanding of detailed reservoir conditions. Thus, to quantitatively characterize the dynamic of reservoirs, one should adopt pre-stack imaging and inversion algorithms that yield high-resolution images of physical parameters (Pratt et al., 1998; Pratt, 1999; Hu et al., 2009; Brossier, 2011; Ma and Hale, 2012; Virieux and Operto, 2009). Traditionally, time-lapse seismic inversion is carried out by processing baseline and monitor data sets independently. Then, the difference between baseline and monitor surveys are computed.

Artifacts that accumulate as a result of acquisition and data processing errors may often have a strength that is comparable to the actual time-lapse signal of interest. Consequently, artifacts can have an adverse impact on the interpretation of time-lapse experiments. Therefore, developing seismic inversion methods that are capable of suppressing the aforementioned artifacts is a problem of practical interest. Various authors have extensively studied time-lapse seismic inversion for monitoring. For instance, time-lapse AVO inversion has been addressed by Buland and El Ouair (2006), Kato and Stewart (2010), Saeed (2014), and Jabbari et al. (2015). Similarly, linearized time-lapse inversion has been addressed by Ayeni and Biondi (2011) and Innanen et al. (2014). Double difference tomography (Zhang and

Thurber, 2003; Watanabe et al., 2004; Denli and Huang, 2009; Tao et al., 2013) and joint and double difference full waveform inversion (FWI) (Denli and Huang, 2009; Queißer and Singh, 2010; Zheng et al., 2011; Zhang and Huang, 2013; Brossier, 2011; Yang et al., 2013; Alemie, 2014; Maharramov et al., 2015) are also viable ways of tackling time-lapse inversion.

In this thesis, we review in detail the current full waveform inversion algorithms. We assess the performance of four optimization methods (gradient descent, approximate diagonal Hessian, Gauss-Newton, and Quasi-Newton) with enough numerical examples. Our primary goal is to extend traditional full waveform inversion algorithms to time-lapse inversion algorithms. In particular, we present a new time-lapse full waveform inversion algorithm namely joint reparameterized time-lapse full waveform inversion and compared it to the traditional time-lapse inversion algorithms such as independent inversion and double difference inversion. We demonstrate the performance of the proposed algorithm with various examples in different time-lapse seismic monitoring settings.

1.6 Organization of the Thesis

The thesis is organized as follows. In chapter two, I cover modeling of seismic data by deriving the general seismic wave equation. Since the inverse problems in the next subsequent chapters are based on a particular case, acoustic wave equation, I emphasize on the acoustic wave equation in the frequency domain. The discrete form of the acoustic wave equation with finite difference discretization is implemented with Absorbing Boundary Conditions (ABC) and Perfectly Matched Layer (PML) boundary conditions (Clayton and Engquist, 1977; Berenger, 1994). Synthetic data modeling examples are given using the resulting linear matrix form of the wave equation via a sequential version of the MUltifrontal Massively Parallel sparse direct Solver (MUMPS) (Amestoy et al., 2001).

In chapter three, the general mathematical formulation of frequency domain acoustic full waveform inversion is covered. In particular, efficient ways of computation of gradient and Hessian for gradient-based optimization are derived. Besides, derivation of reverse time migration in the framework of full waveform inversion is discussed. Finally, numerical examples are given to demonstrate how the framework of full waveform inversion can be used for reverse time migration. In short, this chapter provides necessary tools for the full waveform inversion optimization techniques covered in chapter four, five and six.

In chapter four, detailed optimization methods for Full waveform inversion are covered. Four gradient-based optimizations for full waveform inversion are discussed: gradient descent, diagonal Hessian, Gauss Newton and Quasi-Newton. In particular, the Quasi-Newton method is covered in depth. In addition, the full waveform inversion is extended with regularization

techniques via the limited memory BFGS: 2D smoothing regularization and total variation regularization. Finally, comparisons of the four optimization procedures with three velocity models are given. This chapter provides a framework for next two chapters where time-lapse inversion strategies are covered.

In chapter five, time-lapse full waveform inversion strategies are presented: traditional independent inversion and double difference inversion. In addition, a new time-lapse inversion method, namely joint reparameterized time-lapse full waveform inversion, is developed. Four time-lapse inversion strategies (independent, double difference, joint, and joint reparameterized are explored with a modified Marmousi velocity model that mimic non-repeatable seismic experiments.

In chapter six, typical CO_2 sequestration synthetic velocity models are generated taking into account reservoir conditions with Gassman's fluid substitution model. These models are also explored with the time-lapse inversion strategies covered in Chapter 5. In addition to acquisition differences in the data, two more examples with data decimation are added to mimic the issue of non-repeatability.

In chapter seven, general discussions and conclusions are provided. Following the conclusions, four appendices that support the various topics involved in the thesis are presented.

CHAPTER 2

Seismic Data Modeling: Acoustic Wave

In this chapter, I present the derivation of the general seismic wave equation. Since the inverse problems in the next subsequent chapters are based on the acoustic wave equation, I emphasize on the 2D acoustic wave equation in the frequency domain. Finite difference methods with ABC and PML boundary conditions are used to discretize the wave equation. The LU decomposition with the help of MUMPS is used to solve the resulting discretized linear system of equations. Finally, numerical examples that show the computed wave-fields and seismic data are given to demonstrate the accuracy of the simulations and the effect of boundary conditions.

2.1 Seismic Waves

Seismic waves are created when a force in the form of compression and shear is applied on material bodies. When work is done on a material body, the input energy propagates through it. The seismic energy gets modified according to the physical properties of the body. In other words, the seismic wave acquires the signature of the physical properties of the material body. By recording and analyzing the seismic data, it is possible to study the interior of the body. Thus, seismic waves can be considered as an extension of our senses to look deep into the subsurface.

Understanding of the model of the earth is possible with the help of seismic waves particularly, from naturally occurring phenomena such as earthquakes, volcanic eruption, and landslides under the field of global seismology. Seismology is synonymous with the study of the

Earth's interior with the aid of seismic waves. These days, however, it is an understatement because the principle of seismology is being used to study other heavenly bodies such as the sun's interior in the field of Helio-seismology. The seismic techniques developed to study the Earth's interior through naturally occurring phenomena paved the way to reflection seismology. Reflection seismology uses controlled sources (artificial seismic sources) to generate seismic energy. The seismic energy propagates into the subsurface, and the reflected seismic energy is recorded by receivers to create seismic traces. Seismic traces are packed with information of the subsurface.

The structure and property of the Earth's interior are studied by analyzing and understanding the propagation of elastic waves. From a given source, the generated waves travel through the earth's interior and just under the surface. Those elastic waves which propagate in the interior of the Earth are called body waves (P-waves and S-waves), and those travel just near the surface are called surface waves (Love and Rayleigh waves). Seismic exploration mainly focuses on body waves which contain much information about the deeper structure of the earth. Before I go into the main topic of this chapter, it is worthwhile to define the P and S-waves.

P-waves

These type of waves are generated due to alternate compression and relaxation of the media. The motion of the waves is in the direction of the motion of the particles of the underground for this reason they are longitudinal waves. Most seismic experiments in seismic exploration use compressional wave sources to generate P-waves.

S-waves

Contrary to P-waves, S-waves (shear waves) are transverse waves. The particles motion are perpendicular to the direction of propagation of the waves. The most distinctive feature of this wave is that it travels only through solids because liquids and gases do not support the shear stress.

Modeling seismic wave propagation is an integral part of seismology. The complexity of the underlying physics of seismic waves depends on the physical medium under consideration: isotropic, anisotropic, homogeneous, heterogeneous, etc. In this section, I present the general wave equation starting from the basic underlying physics and I narrow the derivation down to the acoustic wave equation in an isotropic medium.

2.2 Derivation of the Acoustic Wave Equation

To derive the seismic wave equation, it is assumed that the medium of propagation is a continuum. That is to say, one can apply continuum mechanics in the medium where

differential calculus is applicable by disregarding the particle nature of the medium at the micro level by applying conservation of mass, momentum and energy.

2.2.1 Acoustic Waves in Fluids

Hooke's law in 3D states that the pressure applied to a small volume element $V(x, y, z)$ is proportional to the volume change $dV(x, y, z)$ due to the pressure. Mathematically,

$$P(x, y, z) = -K(x, y, z)\frac{dV(x, y, z)}{V(x, y, z)}, \tag{2.1}$$

where $K(x, y, z)$ is the bulk modulus. We assume that the volume change is very small such that the second order relative volume change is negligible, Equation 2.1 can also be expressed using the local particle displacement $\mathbf{u}$

$$P(x, y, z) = -K(x, y, z)\nabla \cdot \mathbf{u}. \tag{2.2}$$

If external time-dependent pressure source (f_e) term is included, Equation 2.3 can be written as,

$$P(x, y, z) = -K(x, y, z)\nabla \cdot \mathbf{u} + f_e(x, y, z, t). \tag{2.3}$$

Employing Newton's second law, it is possible to show that,

$$\nabla P(x, y, z) = -\rho(x, y, z)\frac{\partial^2 \mathbf{u}}{\partial t^2}, \tag{2.4}$$

where $\rho(x, y, z)$ is the density. Equation 2.4 implies the pressure per uint volume is equal to the negative of the product of the density and particle acceleration i.e equivalent to force is equal to mass times acceleration, $F = ma$.

From Equation 2.3 and 2.4, it can be shown that

$$\rho(x, y, z)\nabla \cdot \left(\frac{1}{\rho(x, y, z)}\nabla P(x, y, z)\right) - \frac{1}{v_p(x, y, z)^2}\frac{\partial^2 P(x, y, z, t)}{\partial t^2} = -f(x, y, z, t) \tag{2.5}$$

where

$$f = \frac{\rho(x, y, z)}{K(x, y, z)}\frac{\partial^2 f_e(x, y, z, t)}{\partial t^2} \tag{2.6}$$

and c is the P-wave velocity. The constant density version of the acoustic wave equation, can also be written as,

$$\nabla^2 P(x, y, z) - \frac{1}{v_p(x, y, z)^2}\frac{\partial^2 P(x, y, z, t)}{\partial t^2} = -f(x, y, z, t). \tag{2.7}$$

2.2.2 Acoustic Approximation of Elastic Waves

In a general homogeneous isotropic elastic medium, the seismic wave equation can be written as

$$\rho\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla\lambda(\nabla.\mathbf{u}) + \nabla\mu[\nabla\mathbf{u} + (\nabla\mathbf{u})^T] + (\lambda + 2\mu)\nabla(\nabla.\mathbf{u}) - \mu\nabla\times\nabla\times\mathbf{u}, \tag{2.8}$$

where $\rho = \rho(x, y, z)$ is density, $\mathbf{u}$ is displacement, and $\lambda(x, y, z)$ and $\mu(x, y, z)$ are the Lamé parameters. Under the assumption that the Lamé parameters are slowly varying ($\nabla\mu \approx \mathbf{0}$ and $\nabla\lambda \approx \mathbf{0}$), the homogenous isotropic medium, the wave equation reduces to

$$\rho(x, y, z)\frac{\partial^2\mathbf{u}}{\partial t^2} = (\lambda(x, y, z) + 2\mu(x, y, z))\nabla(\nabla.\mathbf{u}) - \mu(x, y, z)\nabla\times\nabla\times\mathbf{u}. \tag{2.9}$$

The wave equation given by Equation 2.9 comprises both the compressional wave (v_p(x,y,z)) and shear wave ($v_s(x, y, z)$), which are longitudinal and transverse waves, respectively. The beauty of this equation is that we can decompose it to the P-wave and S-wave equations by taking the divergence and curl of the equation, respectively. Taking the divergence of Equation 2.9, using the vector identity $\nabla\times\nabla\times\mathbf{u} = \mathbf{0}$ and rearranging terms, we have

$$\nabla^2(\nabla.\mathbf{u}) - \frac{1}{{v_p}^2(x, y, z)}\frac{\partial^2\nabla.\mathbf{u}}{\partial t^2} = \mathbf{0} \tag{2.10}$$

where

$$v_p(x, y, z) = \sqrt{\frac{(\lambda(x, y, z) + 2\mu(x, y, z))}{\rho(x, y, z)}}. \tag{2.11}$$

The acoustic approximation of elastic wave equation in fluids can be derived from Equation 2.10 as special case in which the fluid is inviscid, it does not support shear stress. Multiplying both side of Equation 2.9 by $\frac{1}{\rho(x,y,z)}$ and taking the divergence of both side, we have

$$\nabla\cdot\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla\cdot\left(\frac{(\lambda(x, y, z) + 2\mu(x, y, z))}{\rho(x, y, z)}\nabla(\nabla.\mathbf{u}) - \mu(x, y, z)\nabla\times\nabla\times\mathbf{u}\right). \tag{2.12}$$

Letting $\nabla\times\nabla\times\mathbf{u} = \mathbf{0}$, setting $\nabla\cdot\mathbf{u} = -\frac{1}{\lambda(x,y,z)}P(x, y, z, t)$ and $\mu(x, y, z) = (0, 0, 0)$ in the right hand side of Equation 2.12, we obtain

$$\nabla\cdot\frac{\partial^2\mathbf{u}}{\partial t^2} = \nabla\cdot\left(\frac{1}{\rho(x, y, z)}\nabla P\right) \tag{2.13}$$

Equation 2.13 can further reduces to

$$\frac{\partial^2\nabla\cdot\mathbf{u}}{\partial t^2} = \nabla\cdot\left(\frac{1}{\rho(x, y, z)}\nabla P(x, y, z, t)\right) \tag{2.14}$$

and substituting again $\nabla \cdot \mathbf{u} = -\frac{1}{\lambda(x,y,z)} P(x,y,z,t)$, we have

$$\frac{1}{\lambda(x,y,z)} \frac{\partial^2 P(x,y,z,t)}{\partial t^2} = \nabla \cdot \left(\frac{1}{\rho(x,y,z)} \nabla P(x,y,z,t) \right). \tag{2.15}$$

Substituting $\lambda(x,y,z) = v_p^2(x,y,z)\rho(x,y,z)$ and rearranging terms, we obtain

$$\rho(x,y,z)\nabla\left(\frac{1}{\rho(x,y,z)} \nabla P(x,y,z,t) \right) - \frac{1}{v_p(x,y,z)^2} \frac{\partial^2 P(x,y,z,t)}{\partial t^2} = 0 \tag{2.16}$$

or with source funciton f included

$$\nabla\left(\frac{1}{\rho(x,y,z)} \nabla P(x,y,z,t) \right) - \frac{1}{K(x,y,z)} \frac{\partial^2 P(x,y,z,t)}{\partial t^2} = -f(x,y,z,t), \tag{2.17}$$

where $K(x,y,z)$ be bulk modulus in a fluid i.e. the product of density and velocity squared.

2.3 Solution of the Acoustic Wave equation

The solution to the hyperbolic wave equation derived above can be solved via finite difference approximations, which is one of the techniques to solve problems that involve partial differential equations numerically. If treated properly, a finite difference approximation can provide a close simulation of the real seismic wave. A variety of finite difference methods have been proposed and used: explicit time marching method, implicit method, frequency domain method, a different order of approximations, and dynamic grid size methods, and staggered grid method.

The two important factors that determine the accuracy of the finite difference method to simulate seismic data are the numerical dispersion and stability (Lines et al., 1999a; Alford et al., 1974). Also, boundary conditions are needed to model the 'infinite medium' and free surface. The two most exploited boundary conditions are the Absorbing Boundary Condition (ABC) and the Perfectly Matched Layer (PML)(Clayton and Engquist, 1977; Berenger, 1994).

Note that Equation 2.17 is 3D in space and in the time domain. By reducing Equation 2.17 to 2D, and then by taking the Fourier transform, one can obtain the following 2D frequency domain version of the equation,

$$\nabla\left(\frac{1}{\rho(x,z)} \nabla \right) \mathcal{F}\{P(x,z,t)\} - \frac{1}{K(x,z)} \mathcal{F}\{ \frac{\partial^2 P(x,z,t)}{\partial t^2} \} = -\mathcal{F}\{f(x,z,t)\}. \tag{2.18}$$

For,

$$\mathcal{F}\{P(x,z,t)\} = P(x,z,\omega), \tag{2.19}$$

$$\mathcal{F}\{\frac{\partial^2 P(x,z,t)}{\partial t^2}\} = -\omega^2 P(x,z,\omega), \tag{2.20}$$

and

$$\mathcal{F}\{f(x,z,t)\} = f(x,z,\omega), \tag{2.21}$$

the frequency domain equation takes the form,

$$\nabla\left(\frac{1}{\rho(x,z)}\nabla P(x,z,\omega)\right) + \frac{\omega^2}{K(x,z)}P(x,z,\omega) = -f(x,z,\omega). \tag{2.22}$$

2.3.1 Second order accurate central difference approximation ($O(2)$)

First, the finite difference approximation of the constant density version of the acoustic wave equation ($\rho(x,z) = \rho_0$ and $v_p(x,z) = c(x,z)$) for a source located at x_s and z_s, is given by

$$\frac{\partial^2 P(x,z,\omega)}{\partial x^2} + \frac{\partial^2 P(x,z,\omega)}{\partial z^2} + \frac{\omega^2}{c^2(x,z)}P(x,z,\omega) = -f(x,z,\omega)\sigma(x-x_s, z-z_s). \tag{2.23}$$

To derive a numerical solution to the wave equation given by Equation 2.23 via a second order accurate central finite difference approximation, the following Taylor series expansion of a function $P(x,z,\omega)$ of the first oder central difference can be used,

$$P_{i+1,j} = P_{i,j} + \Delta x\frac{\partial P}{\partial x}|_{x_i} + \frac{\Delta x^2}{2}\frac{\partial^2 P}{\partial x^2}|_{x_i} + \frac{\Delta x^3}{3!}\frac{\partial^3 P}{\partial x^3}|_{x_i} + \frac{\Delta x^4}{4!}\frac{\partial^4 P}{\partial x^4}|_{x_i} + \cdots \tag{2.24}$$

and

$$P_{i-1,j} = P_{i,j} - \Delta x\frac{\partial P}{\partial x}|_{x_i} + \frac{\Delta x^2}{2}\frac{\partial^2 P}{\partial x^2}|_{x_i} - \frac{\Delta x^3}{3!}\frac{\partial^3 P}{\partial x^3}|_{x_i} + \frac{\Delta x^4}{4!}\frac{\partial^4 P}{\partial x^4}|_{x_i} + \cdots . \tag{2.25}$$

By subtracting Equations 2.25 from Equation 2.24, we find

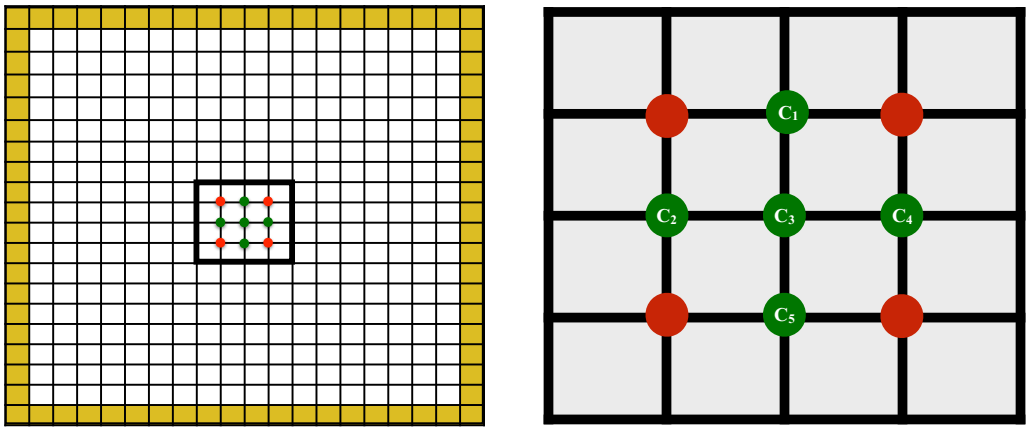
$$P_{i+1,j} - P_{i-1,j} = 2\Delta x\frac{\partial P}{\partial x}|_{x_i} + 2\frac{\Delta x^3}{3!}\frac{\partial^3 P}{\partial x^3}|_{x_i} + \cdots . \tag{2.26}$$

from which can derive central difference first order derivative

$$\frac{\partial P}{\partial x}|_{x_i} = \frac{P_{i+1,j} - P_{i-1,j}}{2\Delta x} - \frac{\Delta x^2}{3!}\frac{\partial^3 P}{\partial x^3}|_{x_i} + \cdots . \tag{2.27}$$

or

$$\frac{\partial P}{\partial x}|_{x_i} = \frac{P_{i+1,j} - P_{i-1,j}}{2\Delta x} + \mathcal{O}(\Delta x^2), \tag{2.28}$$



(a) Computational Grid (b) Sample nodes

Figure 2.1: A computational grid with sample nodes for the second order finite difference approximation.

which is the second order accurate first order central derivative. The second order accurate second order derivative can also be obtained by summing Equations 2.24 and Equation 2.25. It results,

$$P_{i+1,j} + P_{i-1,j} = 2P_{i,j} + \Delta x^2 \frac{\partial^2 P}{\partial x^2}|_{x_i} + 2\frac{\Delta x^4}{4!}\frac{\partial^4 P}{\partial x^4}|_{x_i} + \cdots . \tag{2.29}$$

from which we can derive the central difference second order derivative

$$\frac{\partial^2 P}{\partial x^2}|_{x_i} = \frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{\Delta x^2} - \frac{\Delta x^2}{4!}\frac{\partial^4 P}{\partial x^4}|_{x_i} + \cdots . \tag{2.30}$$

or

$$\frac{\partial^2 P}{\partial x^2}|_{x_i} = \frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{\Delta x^2} + \mathcal{O}(\Delta x^2). \tag{2.31}$$

which is to the second order accurate. Similar second order derivative can also be obtained in the z direction,

$$\frac{\partial^2 P}{\partial z^2}|_{z_j} = \frac{P_{i,j+1} - 2P_{i,j} + P_{i,j-1}}{\Delta z^2} + \mathcal{O}(\Delta z^2). \tag{2.32}$$

By inserting Equations 2.31 and 2.32 into Equation 2.23 and letting $P(x, z, \omega) = P_{i,j}$ and $f(x, z, \omega) = f_{i,j}$, the discrete form of the wave equation at an agular frequency ω becomes,

$$\frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{\Delta x^2} + \frac{P_{i,j+1} - 2P_{i,j} + P_{i,j-1}}{\Delta z^2} + \frac{\omega^2}{c_{i,j}^2}P_{i,j} = -f_{i,j}\sigma(x_i - x_{is}, z_j - z_{js}). \tag{2.33}$$

This is a standard finite difference discretization with five point stencils. With this standard

discretization, the finite difference coefficients for each grid points turn out to be

$$\begin{aligned} C_1 &= \frac{1}{\Delta z^2}, \\ C_2 &= \frac{1}{\Delta x^2}, \\ C_3 &= \frac{\omega^2}{c_{i,j}^2} - \frac{2}{\Delta x^2} - \frac{2}{\Delta z^2}, \\ C_4 &= \frac{1}{\Delta x^2}, \\ C_5 &= \frac{1}{\Delta z^2}. \end{aligned} \tag{2.34}$$

In some physical phenomena, hard-wall type boundary conditions such as Dirichlet or Neumann can be useful. When the computational grid space has to be limited because of the limited power of our computer, the hard-wall boundary conditions have to be replaced by special kind of boundary conditions that simulate the space that extends beyond the target of the experiment. In seismic experiments, seismic waves propagate indefinitely. Thus, we need to incorporate boundary conditions that simulate the 'infinite' medium of the subsurface. In the next two sections, I review two boundary conditions that are used to simulate wave propagation in elastic media: the ABC boundary condition and the PML boundary condition.

ABC Boundary Condition

Absorbing boundary conditions are derived via a paraxial approximation. The ABC boundary conditions with a paraxial approximation for a general differential wave equation was developed by Engquist and Majda (1977). Later, this technique was implemented in scalar wave equations by Claerbout (1976). Clayton and Engquist (1977) also developed the ABC boundary conditions with three orders of approximation for acoustic and elastic wave equations. The paraxial approximation allows propagation of waves in a preferred direction by separating the incoming and outgoing waves. The first oder approximation of the one-way wave equation normal to the boundary is given by,

$$\frac{\partial p}{\partial n} - \frac{\omega}{c} p = 0, \tag{2.35}$$

where n stands for normal direction. This implies that simple boundary condition allows complete reflection only at normal incident. The order of the approximation is related to the angle of incidence. The higher is the order of approximation, waves with larger angles of incidence could also be absorbed but the complexity of the computation increases. Using first order forward and backward finite difference approximations, the following top (tb), bottom (bb), left (lb), and right (rb) boundary conditions can be obtained,

Top Boundary:

$$C_{t1} = 0, \quad C_{t2} = 0, \quad C_{t3} = -\frac{1}{\Delta z} - \imath\frac{\omega}{c_{i,1}}, \quad C_{t4} = 0, \quad C_{t5} = \frac{1}{\Delta z}. \tag{2.36}$$

Bottom Boundary:

$$C_{b1} = \frac{1}{\Delta z}, \quad C_{b2} = 0, \quad C_{b3} = -\frac{1}{\Delta z} - \imath\frac{\omega}{c_{i,m}}, \quad C_{b4} = 0, \quad C_{b5} = 0. \tag{2.37}$$

Left Boundary:

$$C_{l1} = 0, \quad C_{l2} = 0, \quad C_{l3} = -\frac{1}{\Delta x} - \imath\frac{\omega}{c_{1,j}}, \quad C_{l4} = \frac{1}{\Delta x}, \quad C_{l5} = 0. \tag{2.38}$$

Right Boundary:

$$C_{r1} = 0, \quad C_{r2} = \frac{1}{\Delta x}, \quad C_{r3} = -\frac{1}{\Delta x} - \imath\frac{\omega}{c_{n,j}}, \quad C_{r4} = 0, \quad C_{r5} = 0. \tag{2.39}$$

Note that $\imath$ is used as $\sqrt{-1}$ to avoid confusion with the grid index i. Similarly, the top left (tl), top right (tr), bottom left (bl) and bottom right (br) corner boundary conditions,

Top Left Corner:

$$C_{tl1} = 0, \;\; C_{tl2} = 0, \;\; C_{tl3} = -\frac{1}{\Delta x} - \frac{1}{\Delta z} - 2\imath\frac{\omega}{c_{1,1}}, \;\; C_{tl4} = \frac{1}{\Delta x}, \;\; C_{tl5} = \frac{1}{\Delta z}. \tag{2.40}$$

Top Right Corner:

$$C_{tr1} = 0, \;\; C_{tr2} = \frac{1}{\Delta x}, \;\; C_{tr3} = -\frac{1}{\Delta x} - \frac{1}{\Delta z} - 2\imath\frac{\omega}{c_{n,1}}, \;\; C_{tr4} = 0, \;\; C_{tr5} = \frac{1}{\Delta z}. \tag{2.41}$$

Bottom Left Corner:

$$C_{bl1} = \frac{1}{\Delta z}, \;\; C_{bl2} = 0, \;\; C_{bl3} = -\frac{1}{\Delta x} - \frac{1}{\Delta z} - 2\imath\frac{\omega}{c_{1,m}}, \;\; C_{bl4} = \frac{1}{\Delta x}, \;\; C_{bl5} = 0. \tag{2.42}$$

Bottom Right Corner:

$$C_{br1} = \frac{1}{\Delta z}, \;\; C_{br2} = \frac{1}{\Delta x}, \;\; C_{br3} = -\frac{1}{\Delta x} - \frac{1}{\Delta z} - 2\imath\frac{\omega}{c_{n,1}}, \;\; C_{br4} = 0, \;\; C_{br5} = 0. \tag{2.43}$$

It is important to note that this boundary condition works perfectly for one-dimensional wave propagation. It perfectly prevents reflections of incident waves normal to the boundary. For higher dimensional space (2D and 3D), higher order version of this boundary condition should be considered to sufficiently minimize reflection from the boundary at any arbitrary angle.

PML Boundary Condition

The first instance of PML boundary condition was used in electromagnetic wave simulation (Maxwell's equations) by Berenger (1994). Later, this technique was adopted in elastodynamics by Chew and Liu (1996). The PML boundary conditions can be thought of as a workaround on boundary conditions in a sense that it is an informal way of incorporating boundary conditions into wave equations. But it is easy to implement with the finite difference numerical schemes. In this method, additional fictitious grid space in addition to the main computational grid space is added. In this extended fictitious space, the wave is made to decay such that the reflection into the main computational grid space is within a certain level of tolerance.

There is at least two version of PML formulations: unspilt-field and split-field. The unspilt-field approach is formulated by casting the regular coordinate system into a complex 'stretched' coordinate system to incorporate a decaying term outside the region of interest (Roden and Gedney, 2000; Komatitsch and Martin, 2007; Drossaert and Giannopoulos, 2007). The split-field approach is derived by reformulating the acoustic wave equation by splitting the pressure wavefield into x and z components, p_x and p_z respectively such that direct sum of these unphysical wave-fields gives rise to the actual pressure wavefield. Both approaches lead to similar PML boundary condition. The wave equation in Equation 2.22 are put into in the following system of hyperbolic equations (Hustedt et al., 2004),

$$\begin{aligned}
\frac{\partial P_x(x,z,t)}{\partial t} + \gamma_x(x) P_x(x,z,t) &= K(x,z)\frac{\partial Q(x,z,t)}{\partial x} + s(x,z,t), \\
\frac{\partial P_z(x,z,t)}{\partial t} + \gamma_z(z) P_z(x,z,t) &= K(x,z)\frac{\partial R(x,z,t)}{\partial z}, \\
\frac{\partial Q(x,z,t)}{\partial t} + \gamma_x(x) Q(x,z,t) &= \frac{1}{\rho(x,z)}\left(\frac{\partial P_x(x,z,t)}{\partial x} + \frac{\partial P_z(x,z,t)}{\partial x}\right), \\
\frac{\partial R(x,z,t)}{\partial t} + \gamma_z(z) R(x,z,t) &= \frac{1}{\rho(x,z)}\left(\frac{\partial P_x(x,z,t)}{\partial z} + \frac{\partial P_z(x,z,t)}{\partial z}\right).
\end{aligned} \tag{2.44}$$

Note that two wave-fields Q and R are introduced in the hyperbolic system of equations. They are the particle velocity components that involve the first partial derivative of the pressure wave-field in space. The functions γ_x and γ_z are damping functions that control the wave-field in the PML region. Transforming the terms in Equation 2.44 into the frequency

domain, we get

$$\begin{aligned}
-\frac{\imath\omega\xi_x(x)}{K(x,z)}P_x(x,z,\omega) &= \frac{\partial Q(x,z,\omega)}{\partial x} + f(x,z,\omega),\\
-\frac{\imath\omega\xi_z(z)}{K(x,z)}P_z(x,z,\omega) &= \frac{\partial R(x,z,\omega)}{\partial z},\\
-\imath\omega Q(x,z,\omega) &= \frac{1}{\xi_x(x)\rho(x,z)}\frac{\partial P_x(x,z,\omega)}{\partial x},\\
-\imath\omega R(x,z,\omega) &= \frac{1}{\xi_z(z)\rho(x,z)}\frac{\partial P_x(x,z,\omega)}{\partial z},
\end{aligned} \tag{2.45}$$

where new functions $\xi_x(x) = 1 + \imath\gamma_x(x)/\omega$ and $\xi_z(z) = 1 + \imath\gamma_z(z)/\omega$ are defined for simplicity. The second order centred finite difference that we discussed in Section 2.3.1 can be applied on Q and R with the following approximation,

$$\begin{aligned}
\left[\frac{\partial Q(x,z,\omega)}{\partial x}\right]_{i,j} &\approx \frac{1}{\Delta x}(Q_{i+1/2,j} - Q_{i-1/2,j}),\\
\left[\frac{\partial R(x,z,\omega)}{\partial x}\right]_{i,j} &\approx \frac{1}{\Delta z}(R_{i,j+1/2} - R_{i,j-1/2})
\end{aligned} \tag{2.46}$$

By employing Equations 2.46 in Equation 2.45 and taking the first order finite difference approximation for first order derivative in P, one can show that

$$-\frac{\imath\omega\xi_x(x)}{K(x,z)}P_{xi,j} = \frac{1}{\Delta x}(Q_{i+1/2,j} - Q_{i-1/2,j}) + s_{i,j}, \tag{2.47a}$$

$$-\frac{\imath\omega\xi_z(z)}{K(x,z)}P_{zi,j} = \frac{1}{\Delta z}(R_{i,j+1/2} - R_{i,j-1/2}), \tag{2.47b}$$

$$-\imath\omega Q_{i+1/2,j} = \frac{1}{\xi_{x_{i+1/2}}\rho_{i+1/2,j}}\frac{1}{\Delta x}(P_{i+1,j} - P_{i,j}), \tag{2.47c}$$

$$-\imath\omega Q_{i-1/2,j} = \frac{1}{\xi_{x_{i-1/2}}\rho_{i-1/2,j}}\frac{1}{\Delta x}(P_{i,j} - P_{i-1,j}), \tag{2.47d}$$

$$-\imath\omega R_{i,j+1/2} = \frac{1}{\xi_{z_{j+1/2}}\rho_{i,j+1/2}}\frac{1}{\Delta z}(P_{i,j+1} - P_{i,j}), \tag{2.47e}$$

$$-\imath\omega R_{i,j-1/2} = \frac{1}{\xi_{z_{j-1/2}}\rho_{i,j-1/2}}\frac{1}{\Delta z}(P_{i,j} - P_{i.j-1}), \tag{2.47f}$$

where $\rho_{i\pm1/2,j} = \frac{1}{2}(\rho_{i\pm1,j} + \rho_{i,j})$ and similarly $\xi_{i\pm1/2,j} = \frac{1}{2}(\xi_{i\pm1,j} + \xi_{i,j})$ are arithmetic averages of a two most nearest grid points. By adding Equation 2.47a and 2.47b and substituting Equations 2.47c to 2.47d, we can get the second order finite difference equation

with the staggered grid and PML boundary conditions intact. For $P = P_x + P_z$, we have

$$\begin{aligned}
&\frac{1}{\xi_{x_i}}\frac{1}{\Delta x^2}\left(\frac{1}{\xi_{x_{i+1/2}}\rho_{i+1/2,j}}(P_{i+1,j}-P_{i,j}) - \frac{1}{\xi_{x_{i-1/2}}\rho_{i-1/2,j}}(P_{i,j}-P_{i-1,j})\right) \quad + \\
&\frac{1}{\xi_{z_i}}\frac{1}{\Delta z^2}\left(\frac{1}{\xi_z i+1/2\rho_{i+1/2,j}}(P_{i,j+1}-P_{i,j}) - \frac{1}{\xi_{z_{j-1/2}}\rho_{i,j-1/2}}(P_{i,j}-P_{i,j-1})\right) \quad + \\
&\qquad\qquad \frac{\omega^2}{K_{i,j}}P_{i,j} = -s_{i,j}. \qquad (2.48)
\end{aligned}$$

For the constant density case, Equation 2.48 can be rewritten as

$$\begin{aligned}
&\frac{1}{\xi_{x_i}}\frac{1}{\Delta x^2}\left(\frac{1}{\xi_{x_{i+1/2}}}(P_{i+1,j}-P_{i,j}) - \frac{1}{\xi_{x_{i-1/2}}}(P_{i,j}-P_{i-1,j})\right) \quad + \\
&\frac{1}{\xi_{z_i}}\frac{1}{\Delta z^2}\left(\frac{1}{\xi_z i+1/2}(P_{i,j+1}-P_{i,j}) - \frac{1}{\xi_{z_{j-1/2}}}(P_{i,j}-P_{i,j-1})\right) \quad + \quad \frac{\omega^2}{c_{i,j}^2}P_{i,j} = -f_{i,j}. \qquad (2.49)
\end{aligned}$$

For $\gamma_x = \gamma_z = 0$ or $\xi_x = \xi_z = 1$ in the main computational region, Equation 2.49 is the ordinary 2nd order central difference given by Equation 2.33. The general finite difference coefficients including the PML parameters are

$$\begin{aligned}
C_1 &= \frac{1}{\xi_{z_j}\xi_{z_{j-1/2}}}\frac{1}{\Delta z^2}, \\
C_2 &= \frac{1}{\xi_{x_i}\xi_{x_{i-1/2}}}\frac{1}{\Delta x^2}, \\
C_3 &= \frac{\omega^2}{c_{i,j}^2} - \frac{1}{\xi_{x_i}}\frac{1}{\Delta x^2}\left(\frac{1}{\xi_{x_{i+1/2}}} + \frac{1}{\xi_{x_{i-1/2}}}\right) - \frac{1}{\xi_{z_j}}\frac{1}{\Delta z^2}\left(\frac{1}{\xi_{z_{j+1/2}}} + \frac{1}{\xi_{z_{j-1/2}}}\right), \\
C_4 &= \frac{1}{\xi_{x_i}\xi_{x_{i+1/2}}}\frac{1}{\Delta x^2}, \\
C_5 &= \frac{1}{\xi_{z_j}\xi_{z_{j+1/2}}}\frac{1}{\Delta z^2}, \qquad (2.50)
\end{aligned}$$

which similarly turn into the same coefficients as in Equation 2.34 for $\gamma_x = \gamma_z = 0$ or $\xi_x = \xi_z = 1$ in the main computational grid. The damping function γ_x and γ_z are functions that should be chosen such that they have a value of zero at the beginning of the PML region and one at the outermost PML region i.e $1 > \gamma_x(\gamma_z) > 0$. For example, a quadratic function $\gamma_x = C_{PML}(x/L_x)^2$, a cosine function $\gamma_x = C_{PML}\cos(\frac{\pi}{2}(x/L_x))$ and so on, where $x = i_{pml}\Delta x$ is the location PML region and $L_x = N_{pml}\Delta x$ is the total length of the PML region in one side of the PML region. Interested readers can refer Hustedt et al. (2004) for a detailed coverage of the staggered grid and PML implementation of the frequency domain acoustic wave equation both in second order and forth order spatial approximations.

2.3.2 Stability and Dispersion Analysis

Depending on the physical model and the numerical scheme used to simulate seismic wave or wave in general, two important criteria should be considered: the stability of the numerical scheme and grid dispersion (Alford et al., 1974; Marfurt, 1984; Sei and Symes, 1995; Lines et al., 1999b). Stability is controlled by temporal sampling and dispersion is controlled by spatial sampling.

Van Neumann Stability Criteria

Stability criteria by temporal sampling is set via the Von Neumann condition by analyzing the truncation error relative to the exact analytic solution. For the second order finite difference, the time-step criteria for the numerical scheme to be stable is given by

$$\Delta t \leqslant \frac{min(\Delta x, \Delta z)}{\sqrt{2} \times c_{max}} \tag{2.51}$$

or

$$\Delta t_{min} = \frac{min(\Delta x, \Delta z)}{\sqrt{2} \times c_{max}}. \tag{2.52}$$

The minimum time-step that grant stability is $0.707\Delta x/c_{max}$. The higher is the order of finite difference approximation for the same grid size and maximum velocity, the minimum time-step is larger than $0.707\Delta x/c_{max}$. Note that in the frequency domain the time-step is translated to sampling frequency.

Dispersion Analysis

Numerical dispersion is an artifact of using a finite mesh in wave simulation. It occurs when there is a variation of phase velocity with respect to frequency. Therefore, the criteria to prevent grid dispersion is analyzed by finding the relationship between the phase velocity and frequency. In other words, grid space is chosen such that the phase velocity does not show variation with frequency. The dispersion depends on how many grid points available per minimum wavelength of the source signal. Mathematically,

$$\Delta x \leqslant \frac{c_{min}}{f_{max} \times n_{grid}} \tag{2.53}$$

or

$$\Delta x_{min} = \frac{c_{min}}{f_{max} \times n_{grid}}. \tag{2.54}$$

where c_{min} is the minimum velocity, f_{max} is the maximum frequency of the source signal and n_{grid} is the number of grid points. The grid spacing for Δz follows the same analysis.

For an accurate numerical simulation, first, the grid spacing Δx is chosen by fixing the number of grid points. At least ten grid points per unit of the minimum wavelength is required to avoid dispersion in second order finite difference scheme. Then, using the grid spacing, the time step is determined using Equation 2.54.

2.3.3 System of Linear Equations

The constant density acoustic wave equation in the frequency domain given in Equation 2.23 can be rewritten as,

$$\Big(\nabla^2 + \omega^2 m(x,z)^2\Big) u(\omega, X, X_{j_s}) = -s(X,\omega)\delta(X - X_{j_s}), \tag{2.55}$$

where the spatial coordinates are given by $X = (x, z)$, $X_{j_s} = X(x_{j_s}, z_{j_s})$. Similarly, ω is the angular frequency, $m(x, z) = 1/c(x, z)$ is the reciprocal of P-wave velocity $c(x, z)$. The right hand side term $s(X, \omega)\delta(X_{j_s} - X)$ indicates the seismic source at position X_{j_s}, $j_s = 1, 2, 3, \ldots, n_s$ where n_s is the number of sources. The seismic wave-field is indicated by $u(\omega, X, X_{j_s})$. Employing a second order accurate finite difference approximation in space with PML boundary conditions (Berenger, 1994), the discrete form of the wave equation can be written as follows (Pratt et al., 1998),

$$\mathbf{A}(\omega)\mathbf{u}_{j_s}(\omega) = \mathbf{s}_{j_s}(\omega), \tag{2.56}$$

where $\mathbf{A}$ is the sparse matrix obtained after discretizing the Helmholtz operator. The vector $\mathbf{u}_{j_s}(\omega) \in C^{(N_x \times N_z)\times 1}$ symbolizes the wave field $u(\omega, X, X_{j_s})$ for one particular frequency in a discretized subsurface grid of size $N_x \times N_z$. Similarly, $\mathbf{s}_{j_s}(\omega) \in C^{(N_x \times N_z)\times 1}$ indicates the source j_s for monochromatic frequency in vector form.

The non-zero elements of the matrix depend on finite difference coefficients, the frequency (ω) and the P-wave velocity (c). Inverting this matrix is the most computationally intensive part of wave field forward modeling and inversion. We take advantage of the sparsity of the matrix and adopt the MUltifrontal Massively Parallel sparse direct Solver (MUMPS) (Amestoy et al., 2001) to estimate $\mathbf{u}_{j_s}(\omega) = \mathbf{A}(\omega)^{-1}\mathbf{s}_{j_s}(\omega)$. For a given source, $\mathbf{s}_{j_s}$, the solution of the wave equation results in a monochromatic wave field, $\mathbf{u}_{j_s}$, in the computational grid of size $(N_x \times N_z) \times 1$. The monochromatic data at a given set of receivers is extracted from $\mathbf{u}_{j_s}$ via the following expression $\mathbf{d}^{cal}_{j_s} = \mathbf{R}\mathbf{u}_{j_s}$. The matrix $\mathbf{R}$ of size $n_r \times (N_x \times N_z)$ represents the sampling of the data at the receivers location. In this case, we are assuming that the wave-field has been recorded by n_r receivers. Note that each element of the data,

$d^{cal}(\omega, X_{j_r}, X_{j_s})$, and the wave field, $u(\omega, X, X_{j_s})$, depend on the P-wave velocity of the media, the frequency, and source-receiver configurations. In the next sections, we write the data, the wave field, and other variables that appear in the derivations as a function of the model parameter $\mathbf{m}$ by dropping frequency and source indices to avoid notational clutter.

For all grid points in a two-dimensional space, one can set up a similar system of equations to form the matrix $\mathbf{A}$. It leads to the following equation,

$$\mathbf{A}(\omega)\mathbf{u}(\omega) = -\mathbf{f}(\omega). \tag{2.57}$$

The problem comes down to inverting the matrix $\mathbf{A}(\omega)$ to find the pressure wavefield $\mathbf{u}$. However, the wave field that we get with the direct use of this system of equations includes unwanted reflections from artificial boundaries.

The matrix $\mathbf{A}$ can be symmetric or non-symmetric real or complex sparse and highly sparse depending on the choice of the discretization procedure used. If the ABC boundary condition is used, the matrix $\mathbf{A}$ becomes a complex nonsymmetric matrix (Pratt et al., 1998; Hustedt et al., 2004). If the PML boundary condition is used, the matrix $\mathbf{A}$ becomes complex and symmetric.

For the second order finite difference approximation, each row of the matrix $\mathbf{A}$ has at most five non-zero values. All the rows from the main computation region are real. But the rows from the PML region have complex values. It implies that the matrix $\mathbf{A}$ should be treated as complex. The number of rows corresponds the number of points in the whole computational grid. Thus, the matrix $\mathbf{A}$ is a highly sparse complex matrix.

2.3.4 Gaussian Elimination, LU Decomposition, and MUMPS

One of the advantages of the frequency domain modeling is that at a single frequency, the operator $\mathbf{A}$ remains the same for different forcing term $\mathbf{f}$. If one has a means to deal with the matrix $\mathbf{A}$, it is possible to minimize the computational cost. In the next section, the Gaussian elimination and LU decomposition are compared to show the computational benefit of manipulating the matrix $\mathbf{A}$.

Gaussian Elimination

The most common method to find a solution of a linear system of equations is Gaussian elimination. The Gaussian elimination uses three elementary row operations on a given matrix: swapping row positions, multiplying rows by a non-zero scalar and adding two rows where one of them multiplied by a scalar. These operations turn matrix $\mathbf{A}$ into a triangular

matrix. Then, a back-substitution is used to obtain the required solution. For $N \times N$ were $N = N_x \times N_z$, the complete row operations and back-substitution require $N \times (N-1)/2$ divisions, $(2N^3 + 3N^2 - 5N)/6$ multiplications and $(2N^3 + 3N^2 - 5N)/6$ subtractions. The total approximate operations become $2N^3/3$, which is a complexity of $O(N^3)$. For n_s number of sources in our problem of interest, the complexity becomes $O(n_s N^3)$.

LU Decomposition

LU decomposition has two main phases to solve a linear system of equations: the factorizing phase and the solving phase. The Factoring phase is a process to factorize the matrix $\mathbf{A}$ into upper ($\mathbf{U}_A$) and lower triangular ($\mathbf{L}_A$) matrices such that $\mathbf{A} = \mathbf{L}_A \times \mathbf{U}_A$. This process requires a complexity of $O(N^3)$. For the solving phase, we use the two triangular matrices to find the solution that requires $O(N^2)$ for single forcing term. Although the factorizing phase is the computationally expensive as compared to the solving phase, it is done once at a single frequency. These factorized matrices are used over and over again for n_s number of sources. Therefore, LU decomposition has the complexity of $O(N^3 + n_s N^2)$. It is important to note that for a single frequency and a single source, both methods have the same complexity. For large size models and a significant number of sources, the LU decomposition is computationally more efficient than the Gaussian elimination.

MUMPS Solver

MUMPS is a package for solving a linear system of equations that implement the LU decomposition. It takes advantage of the sparsity and symmetry of the matrix $\mathbf{A}$ to facilitate the efficiency of LU based algorithms. It has a sequential and parallel version. The parallel version can be used as a serial version by setting the number of processors to one. It solves the system of equations in three phases

- **Analysis Phase**: if the matrix is sparse, it accepts the nonzero elements and their indices. The matrix A is reprocessed to estimate the number of operations and memory for the next phases. Similarly, the right-hand side of the system of the equation also analyzed if it is a dense or sparse to take advantage of sparsity for efficient computation.
- **Factorization**: If the parallel version for more than one processor is used, the matrix A is distributed to the various processors according to the information from the analysis phase. It also implements pivoting strategies accordingly. Then, the factorizing process is done by elimination, and the matrices remain in their respective processors for the solving phase. The right-hand side of the equation is also broadcasted to the processors.

- **Solving phase**: the solution is computed using the factored matrices and the right-hand side of the equation through forward and backward substitution.

2.4 Numerical Examples

2.4.1 Comparison of boundary condition without and with PML

To test the forward seismic modeling outlined above, a one layer (Figure 2.2(**a**)) and a two-layer model (Figure 2.2(**b**)) are considered. Each model has node size of 480×480. A Ricker wavelet with a central frequency of 20 Hz is used as source excitation. A sampling time interval of $dt = 0.001$ s, and node spacing of $dx = 5$ m and $dz = 5$ m are used by taking into account the dispersion and stability of the finite difference scheme. This choice of parameters gives a minimum of 10 grid points per wavelength. Direct use of the frequency

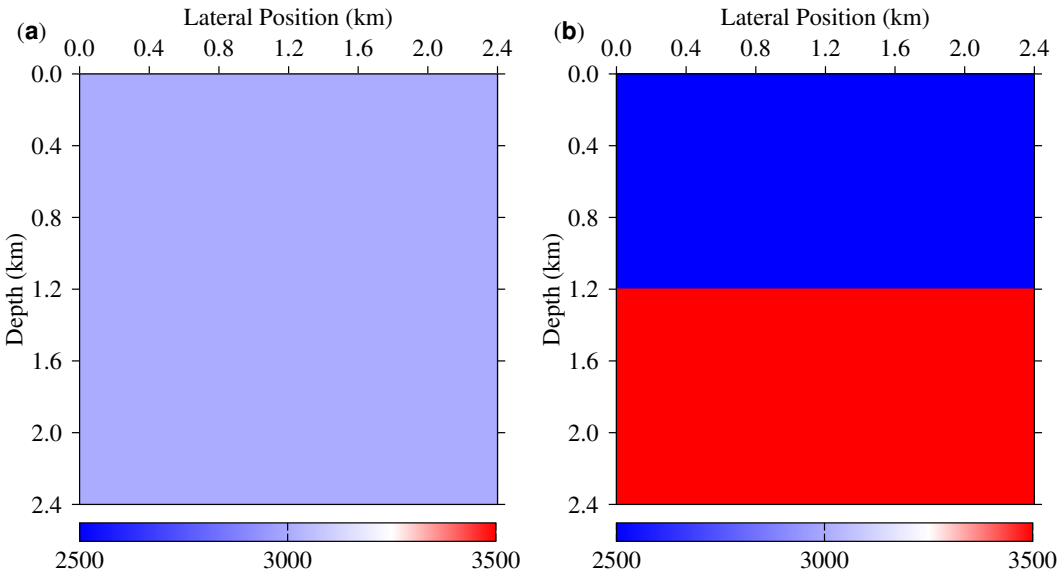


Figure 2.2: Simple velocity models for testing the finite difference approximation (**a**) a constant model and (**b**) a two layered model.

domain modeling gives us a 3D wave-field data (2D in space and frequency). The first modeling example is the comparison of the boundary conditions with and without PML by taking four frequency slices from the 3D frequency domain data. Figure 2.3 (**a**), Figure 2.3 (**b**), Figure 2.3 (**c**), and Figure 2.3 (**d**) are frequency domain wave-field at $f = 20.4545$

Hz, $f = 28.6364$ Hz, $f = 36.8182$ Hz, and $f = 42.2727$ Hz without incorporating the PML boundary condition, respectively. Similarly, Figure 2.4 (**a**), Figure 2.4 (**b**), Figure 2.4 (**c**), and Figure 2.4 (**d**) are frequency domain wave-field at $f = 20.4545$ Hz, $f = 28.6364$ Hz, $f = 36.8182$ Hz, and $f = 42.2727$ Hz by incorporating PML boundary condition with 0.5 km in each side of the computational grid.

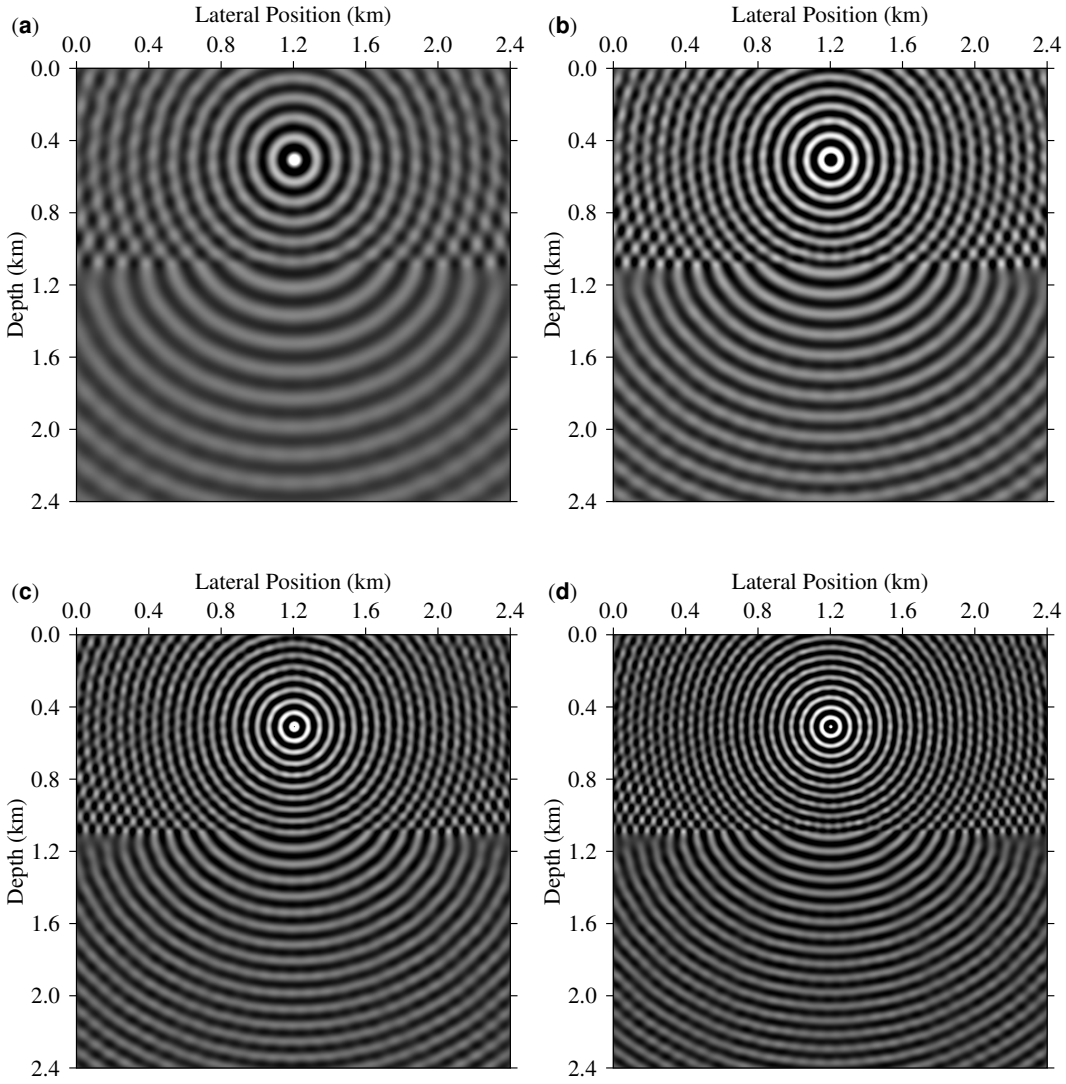


Figure 2.3: A frequency domain wave field for the two layered velocity model source located at the surface without PML boundary condition (**a**) $f = 20.4545$ Hz, (**b**) $f = 28.6364$ Hz, (**c**) $f = 36.8182$ Hz and (**d**) $f = 42.2727$ Hz.

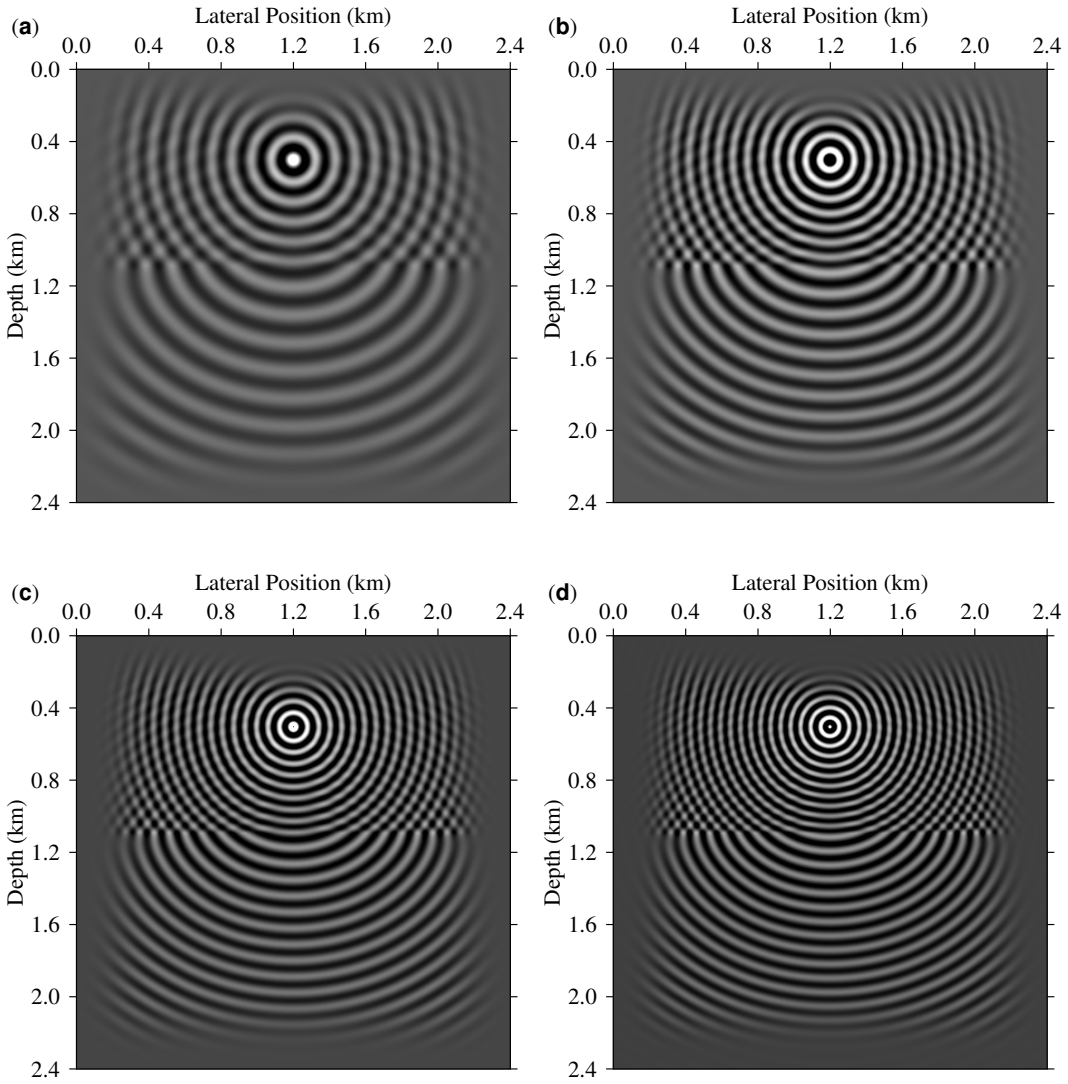


Figure 2.4: A frequency domain wave field for the two layered velocity model source located at the surface with PML boundary condition (**a**) $f = 20.4545$ Hz, (**b**) $f = 28.6364$ Hz, (**c**) $f = 36.8182$ Hz and (**d**) $f = 42.2727$ Hz.

By taking the inverse Fourier transform of a 3D (2D in space and frequency) frequency domain wave-field data, we obtain a time domain version of the wave-field. Figure 2.5 (**a**), Figure 2.5 (**b**), Figure 2.5 (**c**), and Figure 2.5 (**d**) are time domain wave-field at $t = 0.20$ s, $t = 0.30$ s, $t = 0.60$ s, and $t = 0.85$ s respectively for single layer model where the source is located at the centre of computational gird without incorporating the PML boundary condition. Similarly, Figure 2.6 (**a**), Figure 2.6 (**b**), Figure 2.6 (**c**), and Figure 2.6 (**d**) are

time domain wave-field at $t = 0.20$ s, $t = 0.30$ s, $t = 0.60$ s, and $t = 0.85$ s respectively for single layer model where the source is located at the centre of computational gird with the PML boundary condition.

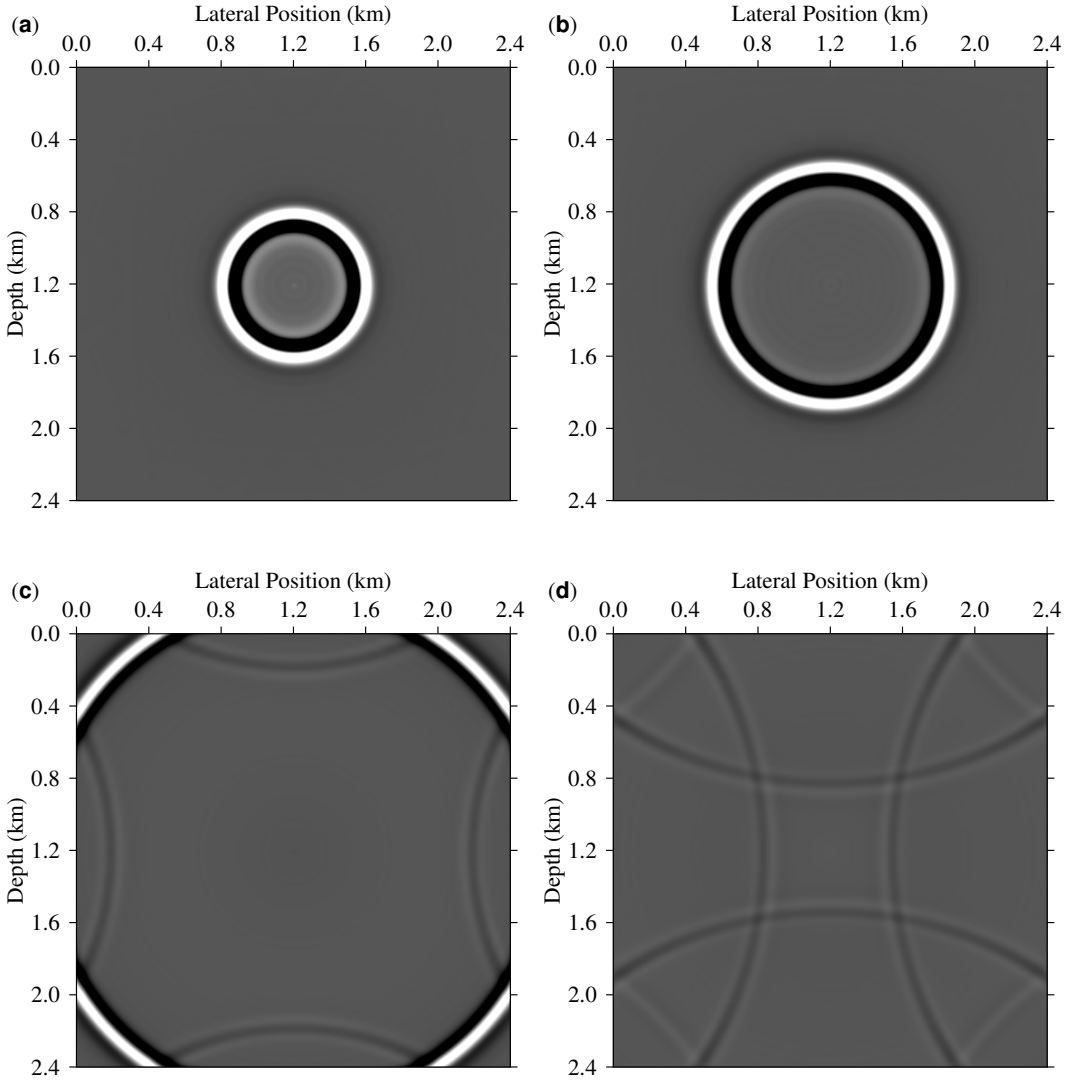


Figure 2.5: A time domain wave field snap shots for the constant velocity model source located at the centre without PML boundary condition (**a**) $t = 0.20$ s, (**b**) $t = 0.30$ s, (**c**) $t = 0.60$ s, and (**d**) $t = 0.85$ s.

Figure 2.7 (**a**),Figure 2.7 (**b**),Figure 2.7 (**c**), and Figure 2.7 (**d**) are time domain wave-field at $f = 0.20$ s, $t = 0.30$ s, $t = 0.60$ s, and $f = 0.85$ s respectively for two layer model where the source is located at the surface without incorporating the PML boundary condition.

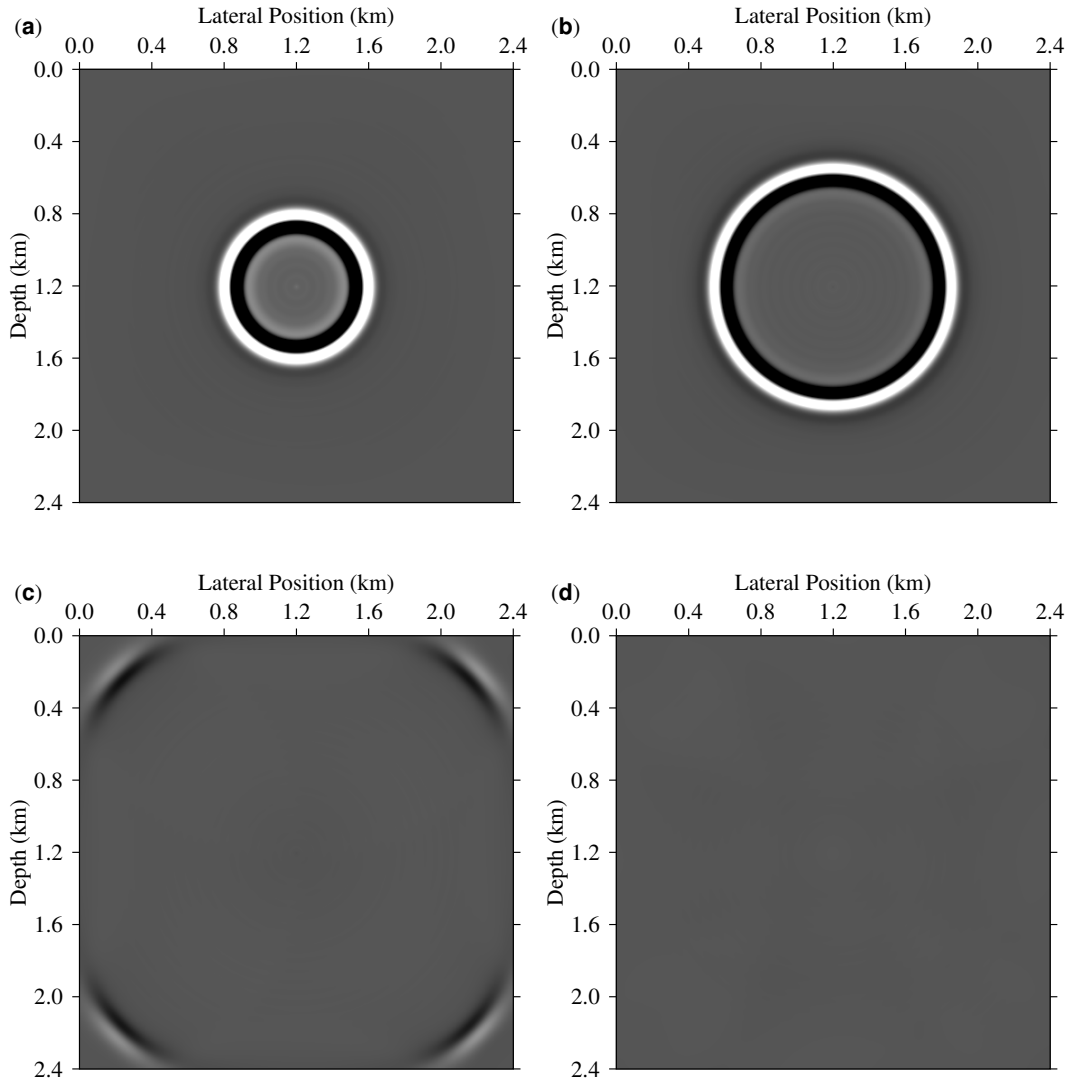


Figure 2.6: A time domain wave field snap shots for the constant velocity model source located at the centre with PML boundary condition (**a**) $t = 0.20$ s, (**b**) $t = 0.30$ s, (**c**) $t = 0.60$ s, and (**d**) $t = 0.85$ s.

Similarly, Figure 2.8 (**a**), Figure 2.8 (**b**), Figure 2.8 (**c**), and Figure 2.8 (**d**) are time domain wave-field at $f = 0.20$ s, $t = 0.30$ s, $t = 0.60$ s, and $f = 0.85$ s respectively for two layer model where the source is located at the surface with the PML boundary condition.

A visualization of the effect of boundary condition can be observed by taking the time domain seismic traces for some number of receivers. Figure 2.9 and Figure 2.10 show seismic

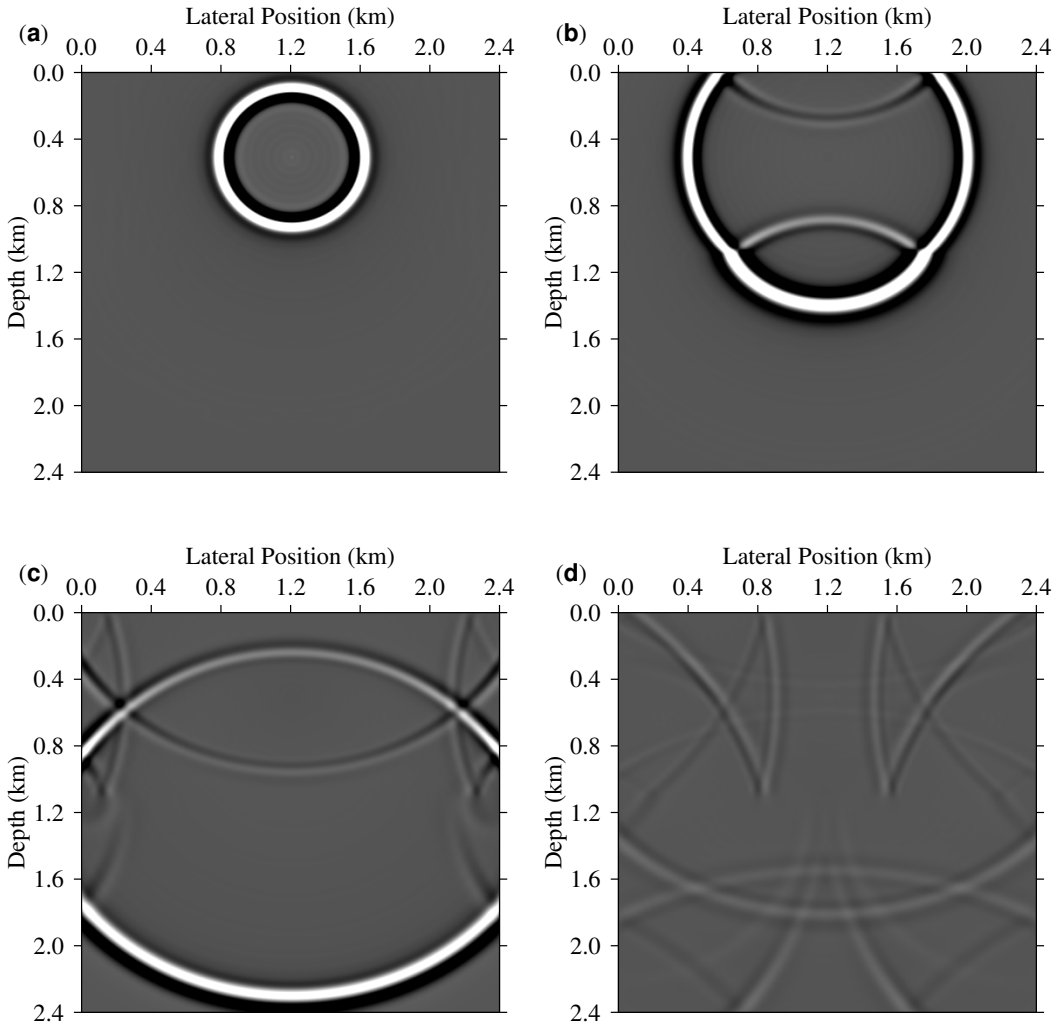


Figure 2.7: A time domain wave field snap shots for the two layered velocity model source located at the surface without PML boundary condition (**a**) $t = 0.20$ s, (**b**) $t = 0.35$ s, (**c**) $t = 0.60$ s, and (**d**) $t = 0.85$ s.

data at different source locations each source separated by 135 m without and with PML, respectively. Each shot has 280 receivers and receiver separation of 5 m. Figure 2.9 (**a**), Figure 2.9 (**b**), Figure 2.9 (**c**), Figure 2.9 (**d**), Figure 2.9 (**e**), and Figure 2.9 (**f**) are data generated without PML boundary condition. Figure 2.10 (**a**), Figure 2.10 (**b**), Figure 2.10 (**c**), Figure 2.10 (**d**), Figure 2.10 (**e**), and Figure 2.10 (**f**) are data generated with PML boundary condition.

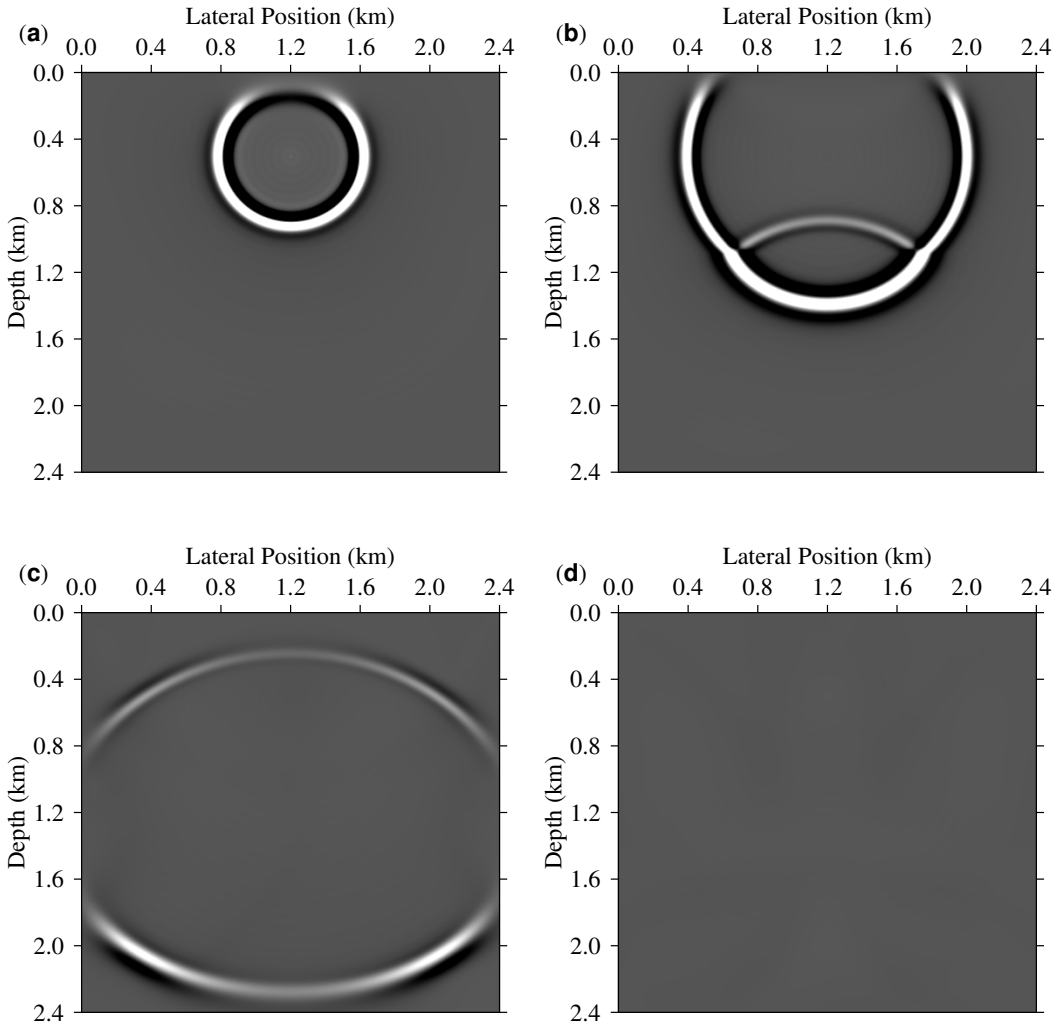


Figure 2.8: A time domain wave field snap shots for the two layered velocity model source located at the surface with PML boundary condition (**a**) $t = 0.20$ s, (**b**) $t = 0.35$ s, (**c**) $t = 0.60$ s, and (**d**) $t = 0.85$ s.

2.4.2 Test of Computational Time via MUMPS solver

In this numerical example, three scenarios are considered. The first scenario is a computational time against the increasing number of nodes by varying the number of sources using a wave-field direct computation method. The second scenario is computational time against the number of nodes by varying the number of sources using an efficient wave-field computation. The third scenario is computational time against the number of sources for fixed number of

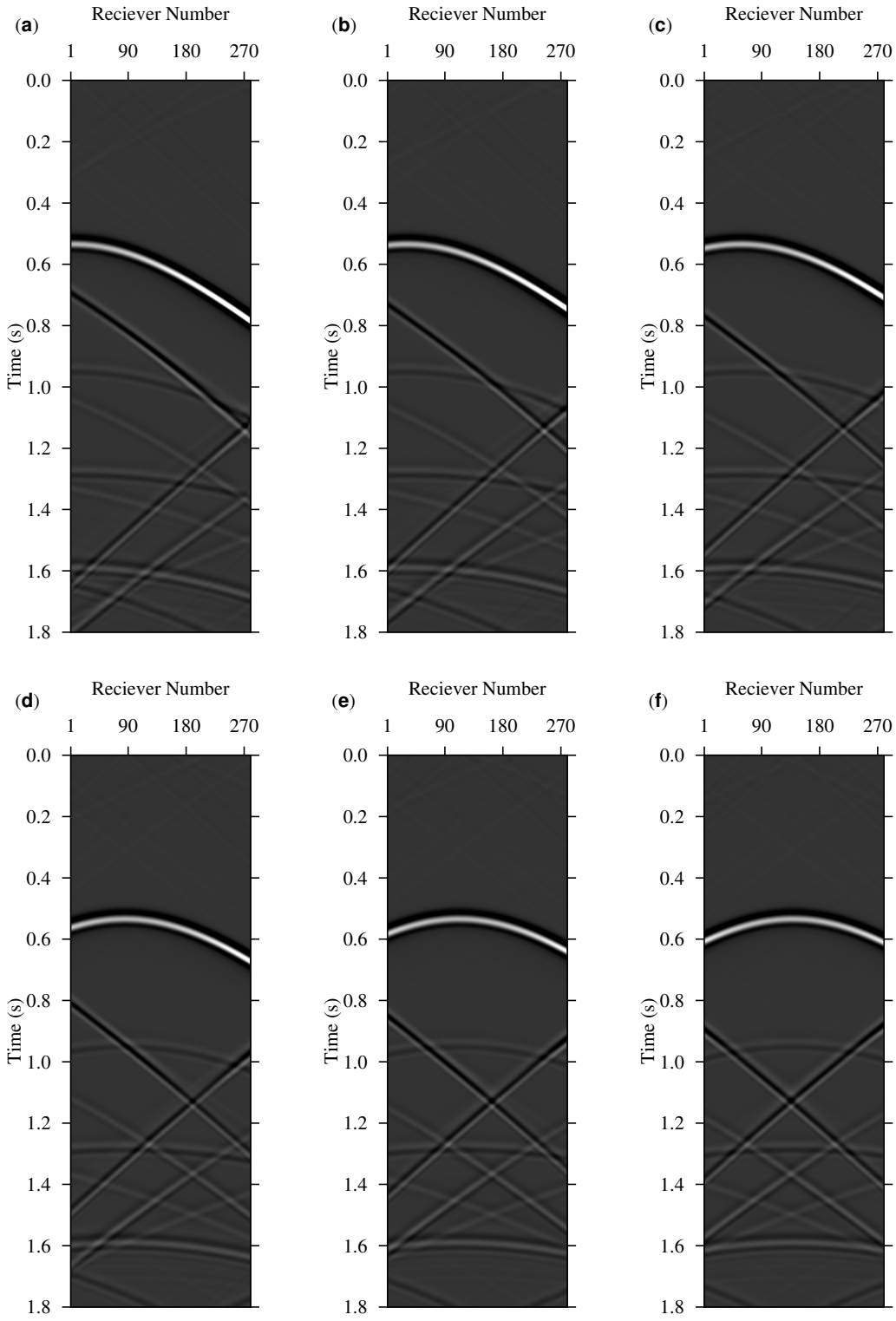


Figure 2.9: Sample time domain data at six source locations without PML boundary condition.

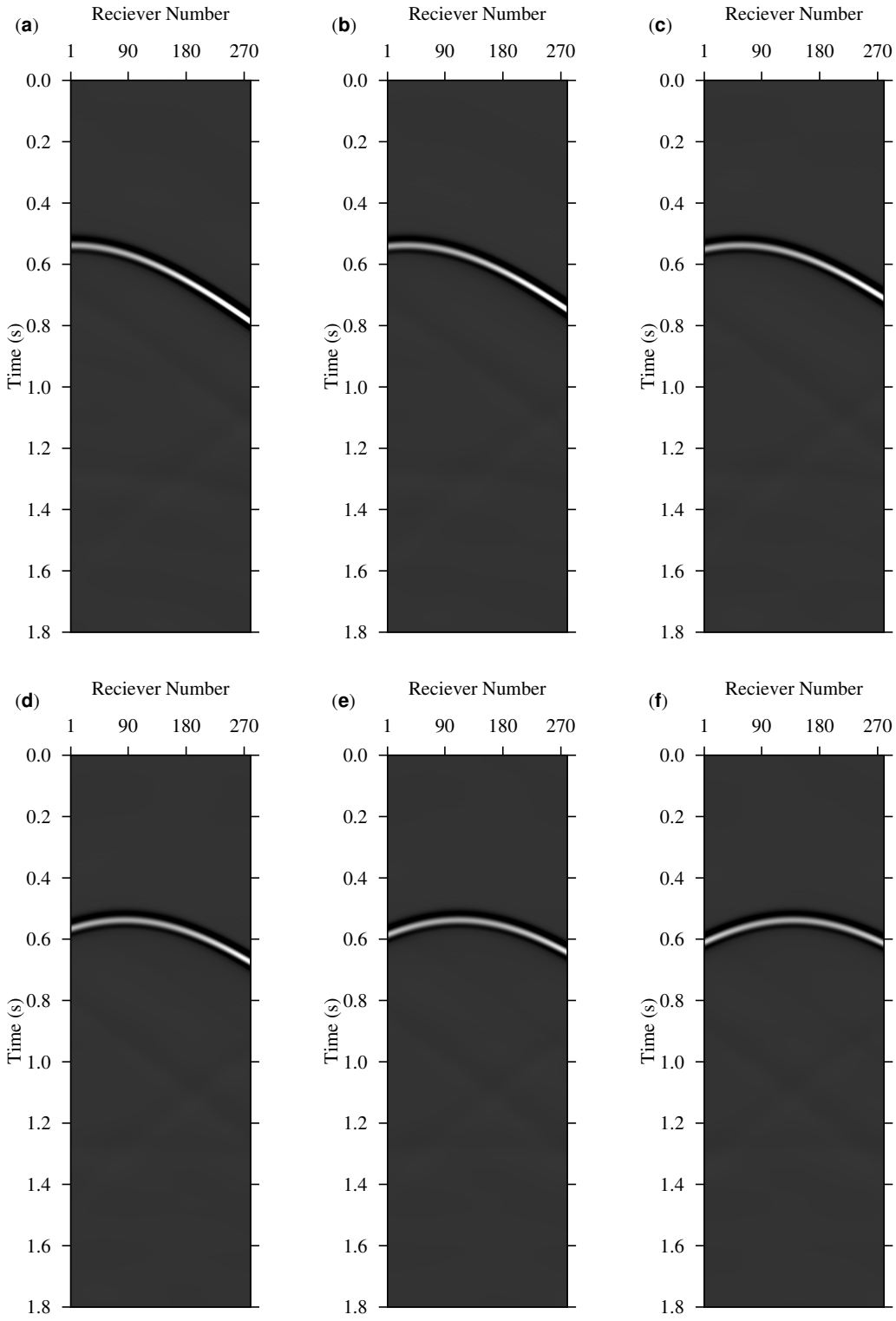


Figure 2.10: Sample time domain data at six source locations with PML boundary condition.

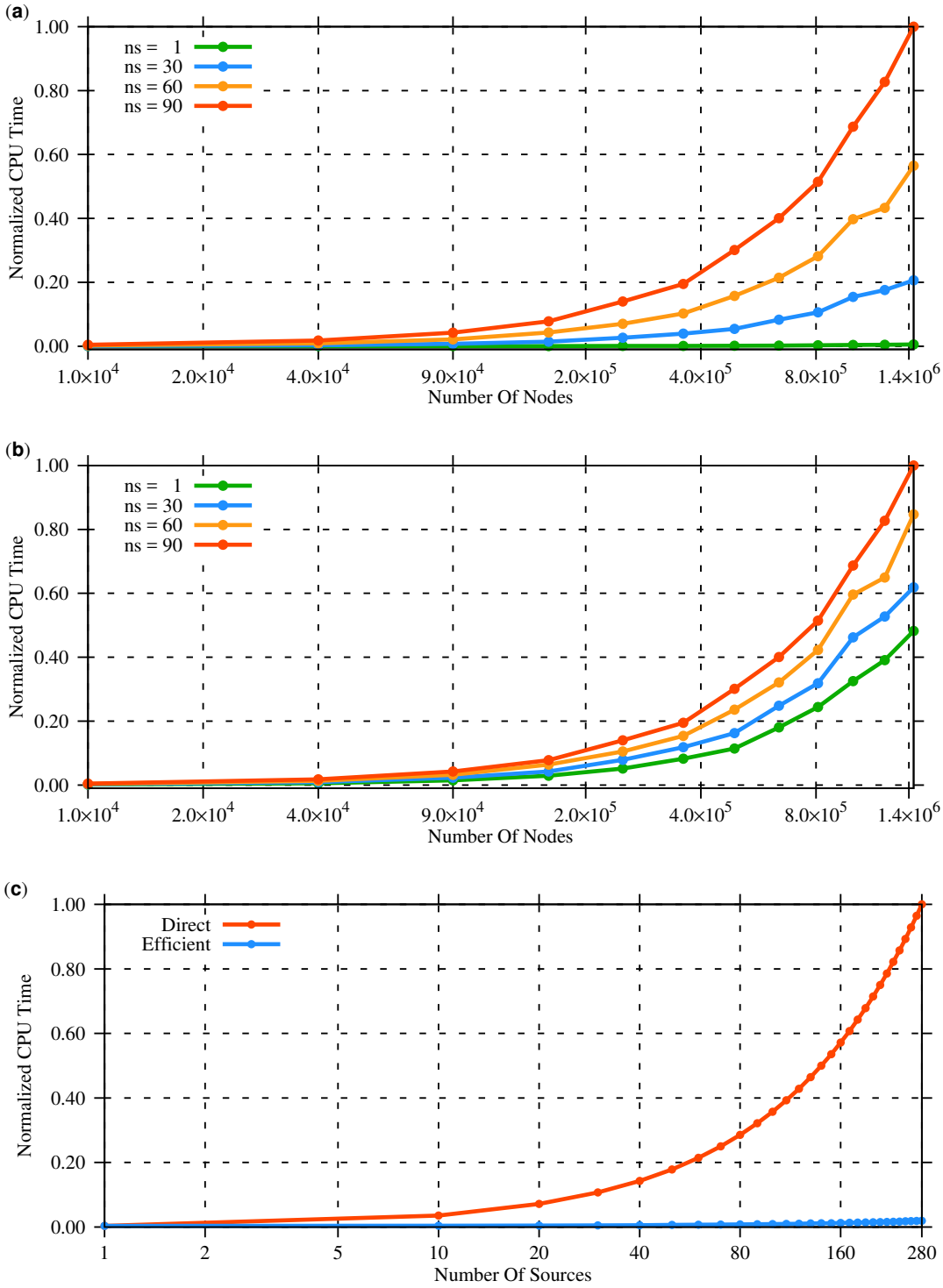


Figure 2.11: A computational time comparison (**a**) with direct method for four different sources, (**b**) with efficient method for four different number of sources and (**c**) direct and efficient methods comparison for a 280×280 grid size for different number of sources.

nodes using both direct and efficient wave-field computation methods.

Figure 2.11 (**a**) is the first scenario where the number of nodes are increased 100×100 to 1200×1200 i.e dx and dz are increased by 100. Each curves depicts for varying number of sources ($ns = 1, 30, 60, 90$) using the direct method. Similarly, Figure 2.11 (**b**) is the second scenario where the number of nodes are increased 100×100 to 1200×1200 i.e dx and dz are increased by 100 at each step. Each curves depicts for varying number of sources ($ns = 1, 30, 60, 90$) using efficient method. Figure 2.11 (**c**) is the third scenario where the number of nodes are fixed to be 280×280 but the number of sources are increased from 1 to 280 for both direct and efficient methods.

2.4.3 Discussion and Summary

The derivation of the acoustic wave equation and modeling of seismic wave in the frequency domain with second order finite difference approximation were discussed in detail. The effectiveness of PML boundary condition relative to ABC boundary condition alone was demonstrated with examples. In addition, the advantage of frequency domain modeling by direct and efficient implementation of wave-field computation using the MUMPS solver was shown with examples.

The generated wave-field in Figures 2.3 and 2.4 in frequency domain and Figures 2.5, 2.6, 2.7, and 2.8 in time domain show drastic decrease of the wave-field the PML region. Similarly, the generated data in time-domain shown in Figures 2.9 and 2.10 show the reflected waves at the boundary without PML and with PML, respectively. Figure 2.8 indicates that the PML boundary condition effectively eliminated the reflection. It is also important to note that PML boundary condition is not perfect; there is always a non-zero reflection. Although the amplitude of the boundary reflection is negligible as compared to the signal of interest, the non-zero boundary reflection can be observed as shown in Figure 2.10. The imperfection of the PML boundary condition could partly be a result of not assigning a large enough PML region and a proper choice of the decay function and parameters involved.

The gain in computational efficiency by adopting the efficient method relative to the direct method is undeniable. It is the result of using a frequency domain modeling and LU decomposition. The frequency domain implementation helps to set up the wave-field computation as a system of linear equations. The LU decomposition allows factorizing the resulting forward operator matrix. The factorized matrix for the same frequency and velocity can repeatedly be used for a number of sources. This advantage was demonstrated for three scenarios. As shown in Figure 2.11 (**a**), the direct method for four different sources shows huge variations in computational time as the number of sources increases from 1 to 90. It happens due to a considerable amount of computational time in factorizing the matrix for each source as

well as the solving phase. In Figure 2.11 (**b**), a small variation in computational time is observed as the number of sources increased from 1 to 90 in the case of the efficient method. The small variation is a result of different computational time for each source in the solving phase. It is important to note that the factorizing phase is is more computational expensive than the solving phase. Figure 2.11 (**c**) also clearly show the comparison of the direct and efficient methods in one plot for fixed number of nodes and a various number of sources. Note also that the y-axes in Figure 2.11 are normalized computational time in log-scale. The normalizing factor in Figure 2.11 (**a**) and Figure 2.11 (**b**) are not the same; it is bigger in (**a**) than (**b**). The true difference in the normalizing factor is clearly shown in Figure 2.11 (**c**).

The computational advantage even becomes more important in the implementation of the forward modeling in Full Waveform Inversion algorithm where many forward operations are needed to compute wave-fields, the gradient, and the Hessian at a single iteration. The forward modeling developed and discussed in this chapter is used in the next four chapters.

CHAPTER 3

Formulation of Full Waveform Inversion

In this chapter, the general mathematical formulation of the frequency domain acoustic full waveform inversion is studied. In particular, efficient ways for computing the gradient and the Hessian for gradient-based optimization are derived. Furthermore, derivation of reverse time migration in the framework of full waveform inversion is discussed. Finally, numerical examples are given to demonstrate how the framework of full waveform inversion can be used for reverse time migration. In short, this chapter provides necessary tools for the full waveform inversion optimization methods that will be covered in the next chapter.

3.1 Introduction to Full Waveform Inversion

Inverse problems arise when data measurements are obtained, and one wants to estimate the unobservable property. In other words, it is a method of inferring about model parameters that characterize a system under study via observations. Inverse problems are common in observational sciences such as geophysical exploration, medical tomography, remote sensing, and astronomy.

Full waveform inversion is an advanced geophysical subsurface parameter estimation method that uses pre-stack seismic data. It can be considered as the extension of least-squares migration. In least-squares migration, one estimates the reflectivity of the subsurface. Least-squares migration provides structural images of the subsurface. Full waveform inversion permits to estimate actual subsurface parameters such as P-wave and S-wave velocities. It

gives high-resolution information of the subsurface. Therefore, full waveform inversion has a huge benefit in characterizing subsurface target reservoirs quantitatively. Quantitative interpretation of seismic data refines our knowledge and decision making ability in subsurface exploration.

An integral part of full waveform inversion is forward modeling. It is the forward modeling that we derived in chapter two that provides the mathematical framework for the FWI algorithm. If the forward model is treated in the time domain, FWI becomes a time-domain algorithm. If it is treated in the frequency domain, the FWI algorithm is also in the frequency domain. FWI is a nonlinear inversion similar to many geophysical inverse problems. The computational cost that involves the computation of several forward problems is also a challenge. As I pointed out in chapter two, the advantage of casting the forward problem in the frequency domain is that it allows us to compute wave-fields for multiple sources or large scale aperture experiments at each frequency efficiently. Successful model recovery is also possible from few frequencies by sequentially inverting the data from low to high-frequency (Pratt et al., 1998; Pratt, 1999).

One challenge of FWI arises from the absence of low-frequencies in the data (Pratt, 1999). The lack of low-frequency data components is a result of several factors including acquisition and processing. This particular case of FWI experiences the phenomena called cycle-skipping that leads to phase ambiguity (Virieux and Operto, 2009; Shah et al., 2012; Li and Demanet, 2016; Métivier et al., 2016; Pan et al., 2016). Consequently, the inversion may tend to be trapped in local minimum. A starting velocity model close to the desired result is required for successful recovery of model parameters.

3.2 Mathematical Formulations of FWI

In general, FWI is derived by defining a cost function (Tarantola and Valette, 1982) that measures the proximity of the observed data to the calculated data. First, we derive a general theoretical inversion framework for any form of cost function denoted by $J(\mathbf{m})$. For a small perturbation, $\mathbf{\Delta m}$, from a known reference model $\mathbf{m}^0$, the cost function $J(\mathbf{m})$ can be expanded using Taylor series as follows,

$$J(\mathbf{m}^0 + \Delta\mathbf{m}) = J(\mathbf{m}^0) + \sum_{j=1}^{M} \frac{\partial J(\mathbf{m}^0)}{\partial m_j}\Delta m_j + \frac{1}{2}\sum_{j=1}^{M}\sum_{l=1}^{M} \frac{\partial^2 J(\mathbf{m}^0)}{\partial m_j \partial m_l}\Delta m_j \Delta m_l + ... \quad (3.1)$$

where M is the size of the model parameters. Assuming that cost function is locally quadratic, higher order terms ($O(\Delta\mathbf{m}^3)$) can be ignored (Bertsekas, 1982; Tarantola, 1984b). Our goal is to find $\Delta\mathbf{m}$ that minimizes the cost function $J(\mathbf{m}^0 + \Delta\mathbf{m})$. Thus, differentiating the cost

function with respect to m_q for $q = 1, 2, 3, \ldots, M$ and again ignoring the resulting higher order term, one can show that

$$\mathbf{p}^0 \quad = \quad -\mathbf{H}^{-1}(\mathbf{m}^0)\mathbf{g}(\mathbf{m}^0), \tag{3.2}$$

where $\mathbf{g}$ is a vector with elements given by

$$g_q = \frac{\partial J(\mathbf{m}^0)}{\partial m_q}, \tag{3.3}$$

that represents the gradient and $\mathbf{H}$ is the Hessian matrix with elements,

$$H_{qj} = \frac{\partial^2 J(\mathbf{m}^0)}{\partial m_q \partial m_j}. \tag{3.4}$$

Equation 3.2 is a general expression of search direction ($\mathbf{p}$) for updating the starting velocity model $\mathbf{m}^0$ at the first iteration,

$$\mathbf{m}^1 = \mathbf{m}^0 + \gamma^0 \mathbf{p}^0. \tag{3.5}$$

The scalar γ^0 is a step length to be determined. Then, the quadratic assumption is compensated by repeating the same procedure until convergence to a global minimum.

Computing the search direction $\mathbf{p}$ from the gradient and inverse of the Hessian is the most computationally expensive part of FWI. In this chapter, an efficient gradient and approximate diagonal computation are presented. I start by defining a cost function. Assuming Gaussian noise, the cost function can be written as (Tarantola and Valette, 1982),

$$J_d(\mathbf{m}) = \frac{1}{2} \sum_{j_{\omega_g}=1}^{n_{\omega_g}} \sum_{j_s=1}^{n_s} \Delta\mathbf{d}(\omega_{j_{\omega_g}}, X_{j_s})^\dagger \Delta\mathbf{d}(\omega_{j_{\omega_g}}, X_{j_s}), \tag{3.6}$$

where n_{ω_g} is the number of frequencies in a group of frequencies, n_s is the number of sources, $\omega_{j_{\omega_g}}$ is the $j_{\omega_g}^{th}$ frequency in a group, X_{j_s} is the j_s^{th} source position, $\Delta\mathbf{d}(\omega_{j_{\omega_g}}, X_{j_s}) = \mathbf{d}^{cal} - \mathbf{d}^{obs}$, $\mathbf{d}^{cal} = \mathbf{d}^{cal}(\omega_{j_{\omega_g}}, X_{j_s})$ is the calculated data, and $\mathbf{d}^{obs} = \mathbf{d}^{obs}(\omega_{j_{\omega_g}}, X_{j_s})$ is the observed data. The symbol † represents the conjugate transpose. It is understood that $\mathbf{d}^{cal} = \mathbf{d}^{cal}(\omega_{j_{\omega_g}}, X_{j_s}, \mathbf{m})$. In other words, the calculated data is a function of the model $\mathbf{m}$.

The goal is to find a model $\mathbf{m}$ that minimizes the cost function by honoring the input data. Thus, we are required to compute the gradient and Hessian of the cost function.

3.2.1 Gradient Computation

The calculation of the gradient ($\mathbf{g}$) of the cost function is not straightforward. We implement an efficient gradient computation method following the techniques described by Pratt et al. (1998) and Shin et al. (2001). The gradient has the form,

$$\mathbf{g} = \sum_{j_{\omega_g}=1}^{n_{\omega_g}} \sum_{j_s=1}^{n_s} \mathbf{g}(\omega_{j_{\omega_g}}, X_{j_s}), \tag{3.7}$$

where $\mathbf{g}(\omega_{j_{\omega_g}}, X_{j_s})$ is the gradient for a single source and a single frequency.

We derive a gradient for single source and single frequency. This gradient can be generalized for n_{ω_g} frequencies and n_s sources to get the gradient for the cost function given in Equation 3.6. To this end, expanding the term $\frac{1}{2}\Delta\mathbf{d}^\dagger\Delta\mathbf{d}$ that appear in this equation, we get

$$J^{t_i}_{j_{\omega_g},j_s} = \frac{1}{2}\sum_{j_r=1}^{n_r}\{(d^{cal}_{j_r} - d^{obs}_{j_r})^*(d^{cal}_{j_r} - d^{obs}_{j_r})\}, \tag{3.8}$$

where j_r is the receiver index and n_r is the number of receivers. Note that the frequency indices (j_{ω_g}) and source indices (j_s) in the right hand side are dropped throughout the derivation for the sake of simplicity. By employing chain rule, the first order derivative of Equation 3.8 with respect to the model parameter m_j,

$$\begin{aligned}
\frac{\partial J^{t_i}_{j_{\omega_g},j_s}}{\partial m_j} &= \frac{1}{2}\sum_{j_r=1}^{n_r}\left\{\frac{\partial (d^{cal}_{j_r})^*}{\partial m_j}\Delta d_{j_r} + \Delta d^*_{j_r}\frac{\partial d^{cal}_{j_r}}{\partial m_j}\right\} \\
&= \frac{1}{2}\sum_{j_r=1}^{n_r}\left\{\left(\frac{\partial d^{cal}_{j_r}}{\partial m_j}\right)^*\Delta d_{j_r} + \Delta d^*_{j_r}\frac{\partial d^{cal}_{j_r}}{\partial m_j}\right\} \\
&= \frac{1}{2}\sum_{j_r=1}^{n_r}\left\{\Delta d_{j_r}\left(\frac{\partial d^{cal}_{j_r}}{\partial m_j}\right)^* + \frac{\partial d^{cal}_{j_r}}{\partial m_j}\Delta d^*_{j_r}\right\} \\
&= \frac{1}{2}\sum_{j_r=1}^{n_r}\left\{\left(\frac{\partial d^{cal}_{j_r}}{\partial m_j}\Delta d^*_{j_r}\right)^* + \frac{\partial d^{cal}_{j_r}}{\partial m_j}\Delta d^*_{j_r}\right\}.
\end{aligned} \tag{3.9}$$

Using the fact that the sum of a complex number and its conjugate is two times the real part, the expression in the curly bracket in Equation 3.9 can be rewritten as,

$$\left(\frac{\partial d^{cal}_{j_r}}{\partial m_j}\Delta d^*_{j_r}\right)^* + \frac{\partial d^{cal}_{j_r}}{\partial m_j}\Delta d^*_{j_r} = 2\Re\left\{\frac{\partial d^{cal}_{j_r}}{\partial m_j}\Delta d^*_{j_r}\right\}, \tag{3.10}$$

where $\Re$ represents real part in the curly bracket. Using Equation 3.10 in Equation 3.9, we

get

$$\frac{\partial J^{t_i}_{j_{\omega_g},j_s}}{\partial m_j} = \sum_{j_r=1}^{n_r} \Re\left\{ \frac{\partial d^{cal}_{j_r}}{\partial m_j} \Delta d^*_{j_r} \right\}. \tag{3.11}$$

Equation 3.11 can also be rewritten as,

$$\frac{\partial J^{t_i}_{j_{\omega_g},j_s}}{\partial m_j} = \Re\left\{ \left(\frac{\partial \mathbf{d}^{cal}}{\partial m_j} \right)^T \Delta \mathbf{d}^* \right\}. \tag{3.12}$$

By taking the partial derivative of the cost function with respect to the remaining $(N_x \times N_z)-1$ model parameters and using $\mathbf{d}^{cal} = \mathbf{Ru}$, Equation 3.12 can be generalized as follows,

$$\frac{\partial J^{t_i}_{j_{\omega_g},j_s}}{\partial \mathbf{m}} = \Re\left\{ \left(\frac{\partial \mathbf{d}^{cal}}{\partial \mathbf{m}} \right)^T (\mathbf{d}^{cal} - \mathbf{d}^{obs})^* \right\} \tag{3.13a}$$

$$= \Re\left\{ \left(\frac{\partial \mathbf{Ru}}{\partial \mathbf{m}} \right)^T (\mathbf{Ru} - \mathbf{d}^{obs})^* \right\} \tag{3.13b}$$

$$= \Re\left\{ \left(\mathbf{R}\frac{\partial \mathbf{u}}{\partial \mathbf{m}} \right)^T (\mathbf{Ru} - \mathbf{d}^{obs})^* \right\} \tag{3.13c}$$

$$= \Re\left\{ \left(\frac{\partial \mathbf{u}}{\partial \mathbf{m}} \right)^T \mathbf{R}^T (\mathbf{Ru} - \mathbf{d}^{obs})^* \right\} \tag{3.13d}$$

$$= \Re\{ \mathbf{F}^T \mathbf{R}^T (\mathbf{Ru} - \mathbf{d}^{obs})^* \}, \tag{3.13e}$$

where $\mathbf{F}$ is a *Jacobian* matrix. The *Jacobian* matrix is a result of the *Fréchet* derivative of the wave field with respect to the model parameters (McGillivray and Oldenburg, 1990; Simon and Blume, 1994).

Next, we derive an expression for the *Jacobian* matrix. To this end, by taking the derivative of both side of Equation 2.56 with respect to m_j via chain rule, we get

$$\frac{\partial \mathbf{A}}{\partial m_j}\mathbf{u} + \mathbf{A}\frac{\partial \mathbf{u}}{\partial m_j} = \mathbf{0}. \tag{3.14}$$

Rearranging the terms in Equation 3.14, one can show that

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial m_j} &= -\mathbf{A}^{-1}\left[\frac{\partial \mathbf{A}}{\partial m_j}\mathbf{u} \right], \\ &= \mathbf{A}^{-1}\mathbf{f}_j, \end{aligned} \tag{3.15}$$

where $\mathbf{f}_j$ is called a virtual source given by

$$\mathbf{f}_j = -\left[\frac{\partial \mathbf{A}}{\partial m_j}\mathbf{u} \right] \tag{3.16}$$

and has a size of $(N_x \times N_z) \times 1$. The result of Equation 3.15 is the j^{th} column of the *Jacobian* matrix. By repeating Equation 3.14 and 3.15 for all m_j, the full *Jacobian* matrix becomes

$$\begin{aligned}\mathbf{F} &= \left[\frac{\partial \mathbf{u}}{\partial m_1} \; \frac{\partial \mathbf{u}}{\partial m_2} \; \frac{\partial \mathbf{u}}{\partial m_3} \; \cdots \; \frac{\partial \mathbf{u}}{\partial m_j} \; \cdots \; \frac{\partial \mathbf{u}}{\partial m_{(N_x \times N_z)}}\right] \\ &= \mathbf{A}^{-1}[\mathbf{f}_1 \; \mathbf{f}_2 \; \mathbf{f}_3 \; \cdots \; \mathbf{f}_j \; \cdots \; \mathbf{f}_{(N_x \times N_z)}] \\ &= \mathbf{A}^{-1}\mathbf{F}^v, \end{aligned} \tag{3.17}$$

where $\mathbf{F}^v$ is a matrix of size $(N_x \times N_z) \times (N_x \times N_z)$ that contains the individual virtual sources in its column. Taking the transpose of Equation 3.17 and inserting back into Equation 3.13, the final expression of the gradient takes the form

$$\mathbf{g}(\omega_{j_{\omega g}}, X_{j_s}) = \Re\{[\mathbf{F}^v]^T[\mathbf{A}^{-1}]^T\mathbf{R}^T(\mathbf{R}\mathbf{u} - \mathbf{d}^{obs})^*\}. \tag{3.18}$$

Equation 3.18 tells us that we need two forward computations to get the gradient: computing the source wave field $\mathbf{u}$ and back propagating the conjugate of the data residual $(\mathbf{R}\mathbf{u} - \mathbf{d}^{obs})^*$. One can also see the derivation given in Pratt et al. (1998) and Shin et al. (2001).

3.2.2 Approximate Hessian Computation (Gauss-Newton)

The Gauss-Newton method is an approximation to the full Newton method using a non-diagonal approximate Hessian. In other words, the diagonal Hessian method is a simplified approximation version of the Gauss-Newton method in which the off-diagonal elements are zeros. The main challenge using the Gauss-Newton method is that it is impractical to build the Hessian explicitly for large problems. Further, the computational cost is also enormous as compared to the gradient and diagonal Hessian methods. Unlike the two methods, the Gauss-Newton method needs at least two additional forward computations to build the Hessian. The advantage of Gauss-Newton over the gradient and approximate Hessian methods is that it gives a more appropriate search direction, and convergence to the minimum can be achieved with relatively large search direction steps even though each step is computationally expensive.

Direct or Non-Constrained approach

To compute the approximate Hessian ($\mathbf{H}_a$) in non-constrained approach, we follow the same procedure as the computation of the gradient until Equation 3.13d. Then, taking the

derivative of Equation 3.13d with respect to $\mathbf{m}$, one can show that

$$\begin{aligned}\mathbf{H} &= \Re\left\{\mathbf{J}_{jac}^T\mathbf{J}_{jac}^*\right\} \\ &+ \Re\left\{\frac{\partial\mathbf{J}_{jac}}{\partial m_1}\Delta\mathbf{d}^*\frac{\partial\mathbf{J}_{jac}}{\partial m_2}\Delta\mathbf{d}^*\frac{\partial\mathbf{J}_{jac}}{\partial m_3}\Delta\mathbf{d}^*\cdots\frac{\partial\mathbf{J}_{jac}}{\partial m_j}\Delta\mathbf{d}^*\cdots\frac{\partial\mathbf{J}_{jac}}{\partial m_{N_{xz}}}\Delta\mathbf{d}^*\right\}, \quad (3.19)\end{aligned}$$

where

$$\mathbf{J}_{jac} = \mathbf{R}\left[\frac{\partial\mathbf{u}}{\partial\mathbf{m}}\right] = \mathbf{R}\left[\frac{\partial\mathbf{u}}{\partial m_1}\frac{\partial\mathbf{u}}{\partial m_2}\frac{\partial\mathbf{u}}{\partial m_3}\cdots\frac{\partial\mathbf{u}}{\partial m_i}\cdots\frac{\partial\mathbf{u}}{\partial m_{N_xN_z}}\right]. \quad (3.20)$$

Taking only the first term in Equation 3.20 by discarding the second term that accounts higher order scattering, we have

$$\mathbf{H}_a = \Re\left\{\mathbf{J}_{jac}^T\mathbf{J}_{jac}^*\right\}. \quad (3.21)$$

Using Equation 3.17, we can write

$$\mathbf{H}_a = \Re\{(\mathbf{F}^v)^T(\mathbf{A}^{-1})^T\mathbf{R}^T\mathbf{R}\mathbf{A}^{-1}(\mathbf{F}^v)^*\}. \quad (3.22)$$

One can efficiently compute the approximate Hessian and then the search direction by (Pratt et al., 1998; Shin et al., 2001),

$$\mathbf{p}_{GN} = -\mathbf{H}_a^{-1}\nabla J, \quad (3.23)$$

where $\mathbf{H}_a$ is the diagonal Hessian matrix given by,

$$\mathbf{H}_a = \Re\{(\mathbf{F}^v)^T(\mathbf{A}^{-1})^T\mathbf{R}^T\mathbf{R}\mathbf{A}^{-1}(\mathbf{F}^v)^*\} \quad (3.24)$$

and

$$\mathbf{h}_i = \mathbf{u}^T\left(\frac{\partial\mathbf{A}}{\partial m_i}\right)^T\left(\left(\frac{\partial\mathbf{A}}{\partial m_i}\right)\mathbf{u}\right)^*. \quad (3.25)$$

Note that the inverse approximate Hessian $\mathbf{H}_a$ cannot be computed for large size problems. The commonly used approach is to adopt the CG method (Hestenes and Stiefel, 1952). It is in the heart of the Gauss-Newton method or full Newton method if the problem is large in which the explicit construction of Hessian is impractical. Therefore, to implement the conjugate gradient for computing the search direction, we only need to know how to perform the product of the Hessian with a vector instead of explicit computation of the inverse of the Hessian, that is

$$\mathbf{H}_a\mathbf{v} = \Re\{(\mathbf{F}^v)^T(\mathbf{A}^{-1})^T\mathbf{R}^T\mathbf{R}\mathbf{A}^{-1}(\mathbf{F}^v)^*\}\mathbf{v}. \quad (3.26)$$

For certain number of iterations, the end result of the conjugate gradient approach is the search direction $\mathbf{p}_{GN}$, which is the product of the inverse of the Hessian and the gradient given in Equation 4.4.

Alternative constrained approaches for the computation of gradient and Hessian are also given in Appendix A with the adjoint state method.

3.2.3 Approximate Diagonal Hessian

Further approximation can be made by assuming that $\mathbf{H}_a$ is diagonally dominant and $\frac{\partial \mathbf{u}}{\partial m_i}$ is perfectly correlated with itself but out-correlated with $\frac{\partial \mathbf{u}}{\partial m_j}$ for $i \neq j$ (Shin et al., 2001). This approximation means that $(\mathbf{A}^{-1})^T\mathbf{R}^T\mathbf{R}\mathbf{A}^{-1}$ is diagonally dominant. For sake of computational cost, we take a crude radical approximation by assuming that the diagonal elements are approximately the same and can be replaced by a constant, say c_A. Thus, Equation 3.22 can be rewritten as

$$\mathbf{H}_{diag} \approx C_A \Re\{(\mathbf{F}^v)^T(\mathbf{F}^v)^*\}. \tag{3.27}$$

Using Equation 3.16, Equation 3.27 can be rewritten as

$$\mathbf{H}_{diag} \approx C_A \Re\{\mathbf{h}_1\ \mathbf{h}_2\ \mathbf{h}_3\ \cdots\ \mathbf{h}_i\ \cdots\ \mathbf{h}_{N_{xz}}\}, \tag{3.28}$$

where

$$\mathbf{h}_i = \mathbf{u}^T\left(\frac{\partial \mathbf{A}}{\partial m_i}\right)^T\left(\left(\frac{\partial \mathbf{A}}{\partial m_i}\right)\mathbf{u}\right)^*, \tag{3.29}$$

which is a vector all zero except at index i. Equation 3.27 implies that the elements of the diagonal matrix are predominantly the auto-correlation of the source wave-field. The advantage of this approximation is that it does not need extra computational cost as the source wave-field are already available when we compute the gradient. This matter will also be discussed in chapter four when we compare various optimization methods for FWI. Our focus in this chapter is to use the FWI mathematical formulation for Reverse time migration and see the connection between the two.

3.3 Migration via the Born Approximation

The Born approximation is a perturbation method that is, generally, used in the scattering theory (Born, 1923). In the seismic data modelling, it is used for approximating the

total seismic wave-field by expressing it as the sum of an incident wave-field and a small perturbation (Hudson and Heritage, 1981; Plessix and Mulder, 2004),

$$u(X,\omega) = u_0(X,\omega) + u_1(X,\omega), \tag{3.30}$$

The small perturbation of the wave-field is as a result of small perturbation in the model parameter (slowness $s = 1/c$),

$$s(X) = s_0(X) + \sigma(X), \tag{3.31}$$

where c is the P-wave velocity, $X = X(x, y, z)$, and σ is a perturbation in the slowness. The slowness is denoted by s to signify the Born approximation and to differentiate it from the model parameter in FWI formulation.

Taking square of both side of Equation 3.31 and discarding the non-linear term with respect to the perturbation σ, we have

$$\begin{aligned} s(X)^2 &= s_0(X)^2 + 2s_0(X)\sigma(X) + \sigma(X)^2 \\ &\approx s_0(X)^2 + r(X), \end{aligned} \tag{3.32}$$

where $r(X) = 2s_0(X)\sigma(X)$ is the model perturbation. By plugging Equation 3.30 and 3.32 into the constant density acoustic wave equation, we get

$$\big(\nabla^2 + \omega^2\big(s_0(X)^2 + r(X)\big)\big)\big(u_0(X,\omega) + u_1(X,\omega)\big) = -f_s(\omega)\delta(X - X_s), \tag{3.33}$$

From Equation 3.33, we can deduce that

$$\big(\nabla^2 + \omega^2 s_0^2(X)\big)u_1(X_s, X, \omega) = -\omega^2 r(X)\big(u_0(X_s, X, \omega) + u_1(X_s, X, \omega)\big), \tag{3.34}$$

The solution to Equation 3.34, perturbation wave-field $u_1(X_s, X, \omega)$, via the Green's function (G_0) has the form

$$u_1(X_s, X, \omega) = \sum_Y G_0(X, \omega; Y)(\omega^2 \Delta s(X)(u_0(X_s, Y, \omega) + u_1(X_s, Y, \omega))). \tag{3.35}$$

The Green's function is simply the solution of the wave equation for an impulsive forcing term,

$$\big(\nabla^2 + \omega^2 s_0^2(X)\big)G_0(X_s, X, \omega) = -\delta(X - X_s), \tag{3.36}$$

which results in

$$u_0(X_s, X, \omega) = f(X_s, \omega)G_0(X_s, X, \omega). \tag{3.37}$$

Using Equations 3.36 and 3.37 in Equation 3.35, and keeping only the first order scattering, we have

$$\begin{aligned} u_1(X_s, X, \omega) &\approx \omega^2 \sum_Y G_0(X, Y, \omega) r(X) u_0(X_s, Y, \omega) \\ &= \omega^2 \sum_Y G_0(X, Y, \omega) r(X) f(X_s, \omega) G_0(X_s, Y, \omega) \end{aligned} \tag{3.38}$$

and the corresponding scattered data $d_b(X_s, X_r, \omega)$ recorded at X_r becomes

$$d_b(X_s, X_r, \omega)^{cal} \approx \omega^2 \sum_Y G_0(X_r, Y, \omega) r(X) f(X_s, \omega) G_0(X_s, Y, \omega). \tag{3.39}$$

Note that I used a subscript b to signify that the calculated data is via Born approximation. Equation 3.39 is a linear with respect to $r(X)$ equation that can be rewritten as,

$$d_b^{cal} = Lr, \tag{3.40}$$

where L is the forward linear operator. The cost function associated to the linear approximation similar to Equation 5.1 along with 3.8 is defined as,

$$J_b(\mathbf{r}) = \frac{1}{2} \sum_\omega \sum_s \sum_r \Delta d_b(X_s, X_r, \omega)^* \Delta d_b(X_s, X_r, \omega), \tag{3.41}$$

where $\Delta d_b(X_s, X_r, \omega) = d_b^{cal}(X_s, X_r, \mathbf{r}, \omega) - d^{obs}(X_s, X_r, \omega)$. The next step is to find $r(X)$ that minimizes the cost function. We learned from previous section that for a given objective function, the perturbation in the model parameter $r(X)$ with a known s_0 can be obtained by with the following expression (Plessix and Mulder, 2004),

$$\mathbf{I} = \triangle \mathbf{r} = -\mathbf{H}_b^{-1} \nabla_{\mathbf{r}} J_b. \tag{3.42}$$

Since the calculated data is linear with respect to $r(X)$, the gradient of Equation 3.41 with respect to $r(X)$ is straightforward. That is,

$$\nabla_{\mathbf{r}} J_b(X) = \Re \Big\{ \sum_\omega \omega^2 \sum_s \sum_r G_0(X_s, X, \omega) f_s(\omega) G_0(X, X_r, \omega) \Delta d_b^*(X_s, X_r, \omega) \Big\}. \tag{3.43}$$

Equation 3.43 implies that the gradient is the cross correlation of the source wave-field and back-propagated wave-field. By the same argument for approximate Hessian calculation in section 3.2.2, the approximate diagonal Hessian associated with the cost function J_b is (Plessix and Mulder, 2004),

$$H_b(X, X) \approx \sum_\omega \omega^4 \sum_s |f_s(\omega)|^2 |G_0(X_s, X, \omega)|^2. \tag{3.44}$$

The approximate diagonal Hessian implies the auto-correlation of the source wave-field. On account of Equations 3.43 and 3.44, Equation 3.42 becomes,

$$\mathbf{r}(X) \approx \frac{\Re\Big\{ \sum_{\omega} \omega^2 \sum_{s} \sum_{r} G_0(X_s, X, \omega) f_s(\omega) G_0(X, X_r, \omega) \Delta d_b^*(X_s, X_r, \omega) \Big\}}{\sum_{\omega} \omega^4 \sum_{s} |f_s(\omega)|^2 |G_0(X_s, X, \omega)|^2} \tag{3.45}$$

The migrated image, therefore, is the cross correlation of the source wave-field with the back-propagated wave-field normalized by the auto correlation of the source wave-field i.e The imaging condition. Imaging condition based on Claerbout (1971) is that the down-going wave and the up-going wave coincides in time and space, which gives us the images of reflectors.

The migrated image can be put in the form of velocity 'reflectivity' as,

$$\sigma = \frac{1}{c} - \frac{1}{c_0} = \frac{c_0 - c}{c \times c_0} = -\frac{1}{c_0}\frac{\Delta c}{c}. \tag{3.46}$$

Then, the velocity 'reflectivity' becomes,

$$\frac{\Delta c}{c} = -c_0 \times \sigma. \tag{3.47}$$

Since $r = 2 \times s_0 \times \sigma$ and $s_0 = 1/c_0$, we obtain

$$\sigma = \frac{1}{2} \times c_0 \times r. \tag{3.48}$$

Inserting the expression of σ into Equation 3.47, we have

$$\frac{\Delta c}{c} = -\frac{1}{2} \times c_0^2 \times r. \tag{3.49}$$

3.4 Reverse Time-Migration

Reverse time-migration is also based on the cross correlation imaging condition. In time-domain, the source wave-fields (u_s) are computed by propagating the source forward in time from the source location and the receiver wave-fields (u_r) are computed by propagating the data backward in time from the receivers (Baysal et al., 1983). The resulting image from the cross-correlation of the two wave-fields gives the migrated image, (Claerbout, 1971; Kaelin and Guitton, 2006; Sava and Fomel, 2006; Sava and Vasconcelos, 2011),

$$I_{rtm}(X, \tau) = \sum_{s} \sum_{t} u_s(t - \tau, X) u_r(t + \tau, X). \tag{3.50}$$

where τ is the time-lag. Equation 3.50 in frequency domain can be written as

$$I_{rtm}(X,\tau) = \sum_s \sum_\omega u_s^*(\omega, X) u_r(\omega, X) e^{2i\omega\tau}. \tag{3.51}$$

For zero-lag cross correlation ($\tau = 0.0$), we have

$$I_{rtm}(X) = \sum_s \sum_\omega u_s^*(\omega, X) u_r(\omega, X). \tag{3.52}$$

By normalizing the resulting image by the source illumination (auto-correlation of the source wave-field) (Claerbout, 1971; Kaelin and Guitton, 2006; Guitton et al., 2006), we obtain

$$I_{rtm}(X) = \frac{\sum_s \sum_\omega u_s^*(\omega, X) u_r(\omega, X)}{\sum_s \sum_\omega u_s^*(\omega, X) u_s(\omega, X)}. \tag{3.53}$$

Equation 3.53 is similar to Equation 3.45 obtained via the Born approximation except that in reverse time migration the source and receiver wave-field are not restricted to first order scattering as in the Born approximation. The migrated image in reverse time-migration take into account higher order scatterings. Therefore, we can use the FWI formulation for computing the gradient and approximate diagonal Hessian. On account of Equation 3.18 and 3.28 the migrated image in reverse time-migration becomes,

$$\mathbf{I}_{rtm} = -(C_A \mathbf{H}_{diag})^{-1} \Re\{[\mathbf{F}^v]^T [\mathbf{A}^{-1}]^T \mathbf{R}^T (\mathbf{R}\mathbf{u} - \mathbf{d}^{obs})^*\}. \tag{3.54}$$

Since the gradient and the approximate Hessian are computed from the objective function with respect to $m = 1/c$, the resulting image from Equation 3.54 is ,

$$\begin{aligned} I_{rtm}(X) &= m - m_0 \\ &= \frac{1}{c} - \frac{1}{c_0} \\ &= \frac{c_0 - c}{c \times c_0} \\ &= -\frac{1}{c_0}\frac{\Delta c}{c}. \end{aligned} \tag{3.55}$$

This implies that the migrated image related to the velocity 'reflectivity' is given by,

$$\frac{\Delta c}{c} = -c_0 \times I_{rtm}(X). \tag{3.56}$$

Equation 3.49 and Equation 3.56 show the meaning of migrated image obtained by Born approximation and RTM, respectively. Both methods give similar qualitative information.

Thus, the sum of search directions of the contribution from all sources and a certain number of frequencies at the first iteration of the FWI algorithm gives the migrated image, i.e., RTM image. The main advantage of reverse time migration is that it provides a higher resolution image as compared to the Born approximation approach because of the full wave.

3.5 Numerical Examples

In this section, the framework of full waveform inversion is used for reverse time migration. We compare two approaches of computing the search directions: using diagonal Hessian and Gauss-Newton. The search direction using Gauss-Newton method is done via the conjugate gradient method (Shewchuk, 1994). To grant and speed up convergence, the approximate Hessian is modified such that $\mathbf{H}_a + \mu\mathbf{I}$ is used (Marquardt, 1963). To compute the search direction, we can write the equation in a linear system of equation as

$$(\mathbf{H}_a + \mu\mathbf{I})\mathbf{p} = \mathbf{b}, \tag{3.57}$$

where $\mathbf{b} = -\mathbf{g}$ and μ is a known as Levenberg's parameter. The conjugate gradient algorithm with this modification is given in Algorithm 2.

3.5.1 Velocity Model One

Seismic data are generated with a Ricker wavelet of a central frequency 20 Hz and a true two-layer velocity model shown in Figure 3.1(**a**) with the 2D acoustic wave equation and second order finite difference approximation with PML boundary condition. A total of 96 shots separated by 25 m and each shot has 480 number of receivers with separation 5 m. The grid spacing, $dx = dz$, is 5 m. Figure 3.1(**b**) is a constant velocity model used as input to the migration algorithm. Figure 3.2(**a**) is the migrated image obtained using the diagonal Hessian scaling and Figure 3.2(**b**) is a migrated image obtained by the Gauss-Newton approach.

3.5.2 Velocity Model Two

Seismic data are generated with a Ricker wavelet of central frequency 20 Hz and a true two-layer velocity model shown in Figure 3.3(**a**) with the 2D acoustic wave equation and second order finite difference approximation with PML boundary condition. A total of 96 separated by 25 m and each shot has 480 number of receivers with separation 5 m. Figure 3.3(**b**) is a constant velocity model used as input to the migration algorithm. Figure 3.4(**a**) is the migrated image obtained using the diagonal Hessian scaling and Figure 3.4(**b**) is a migrated image obtained by the Gauss-Newton approach.

Algorithm 1 : Pseudocode for reverse time migration via the Conjugate Gradient algorithm.

Input: $\mathbf{c}_0$, n_ω, $i\omega = 0$, $\mathbf{p} = \mathbf{0}$
while *($i\omega \leqslant n_\omega$)* **do**
  $\mathbf{g}_{i\omega} \leftarrow$ *use FWI engine*
  $\mathbf{p}_{i\omega} \leftarrow$ *use CG algorithm*
  $\mathbf{p} = \mathbf{p} + \mathbf{p}_{i\omega}$
  $i\omega = i\omega + 1$
end
Output: $\mathbf{p}$
return *Migrated Image* $= \mathbf{p}$

Algorithm 2 : Conjugate Gradient algorithm to compute search direction $\mathbf{p}_{i\omega}$ at a given frequency ω.

Input: $i = 1$, $\mathbf{p}_0 = \mathbf{0}$,
$\mathbf{r}_0 = \mathbf{b} - (\mathbf{H}_a + \mu\mathbf{I})\mathbf{p}_0 \leftarrow$ *use FWI engine* ,
$\mathbf{h}_i = \mathbf{p}_0$, $\mathbf{r}_i = \mathbf{r}_0$,
$\sigma_{new} = \mathbf{r}_i^T \mathbf{r}_i$,
$\sigma_0 = \sigma_{new}$.
while *($i < Iter_{max}$) and ($\sigma_{new} < \epsilon\sigma_0$)* **do**
  $\mathbf{q}_i = (\mathbf{H}_a + \mu\mathbf{I})\mathbf{p}_i \leftarrow$ *use FWI engine*
  $\alpha_i = \frac{\sigma_{new}}{\mathbf{p}_i^T \mathbf{q}_i}$
  $\mathbf{p}_{i+1} = \mathbf{p}_i + \alpha_i \mathbf{h}_i$
  $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{q}_i$
  $\sigma_{old} = \sigma_{new}$
  $\sigma_{new} = \mathbf{r}_{i+1}^T \mathbf{r}_{i+1}$
  $\beta_{i+1} = \frac{\sigma_{new}}{\sigma_{old}}$
  $\mathbf{h}_{i+1} = \mathbf{p}_i + \beta_{i+1} \mathbf{h}_i$
  $i = i + 1$
end
Output: $\mathbf{p}_{i\omega} \leftarrow \mathbf{p}_i$

3.5.3 Velocity Model Three

Seismic data are generated with a Ricker wavelet of central frequency 20 Hz and a true three-layer velocity model shown in Figure 3.5(**a**) with the 2D acoustic wave equation and second order finite difference approximation with PML boundary condition. A total of 96 shots separated by 25 m and each shot has a 480 number of receivers with separation 5 m.

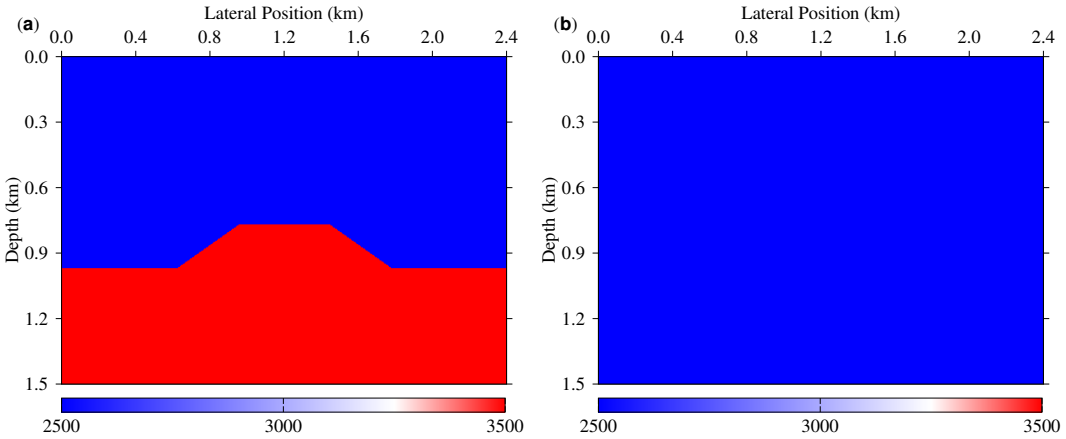


Figure 3.1: Velocity models for testing the framework of FWI to reverse time migration (**a**) a two layered model with rectangular shape and **b**) input constant velocity model.

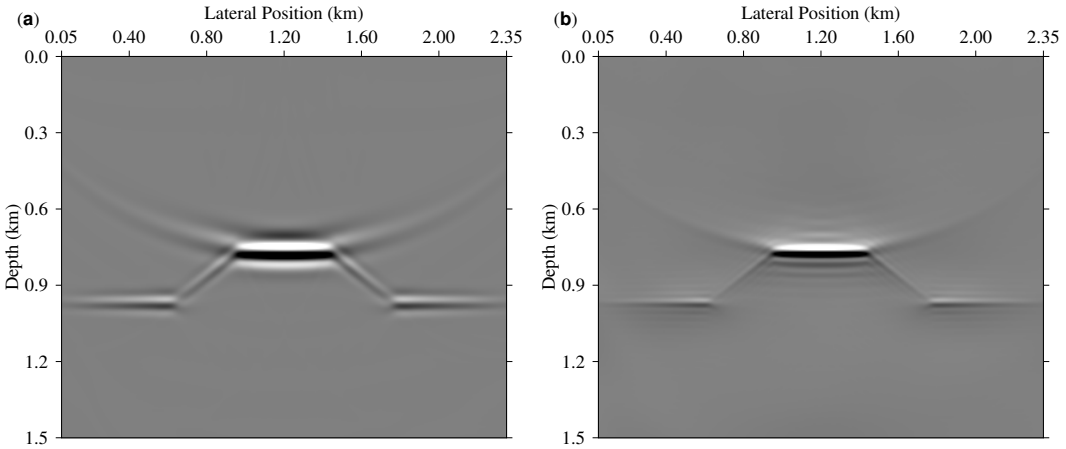


Figure 3.2: Reverse time migration results (**a**) by diagonal hessian scaling and **b**) by Gauss-Newton - conjugate gradient approach.

Figure 3.5(**b**) is a constant velocity model used as input to the migration algorithm. Figure 3.6(**a**) is the migrated image obtained using the diagonal Hessian scaling and Figure 3.6(**b**) is a migrated image obtained by the Gauss-Newton approach.

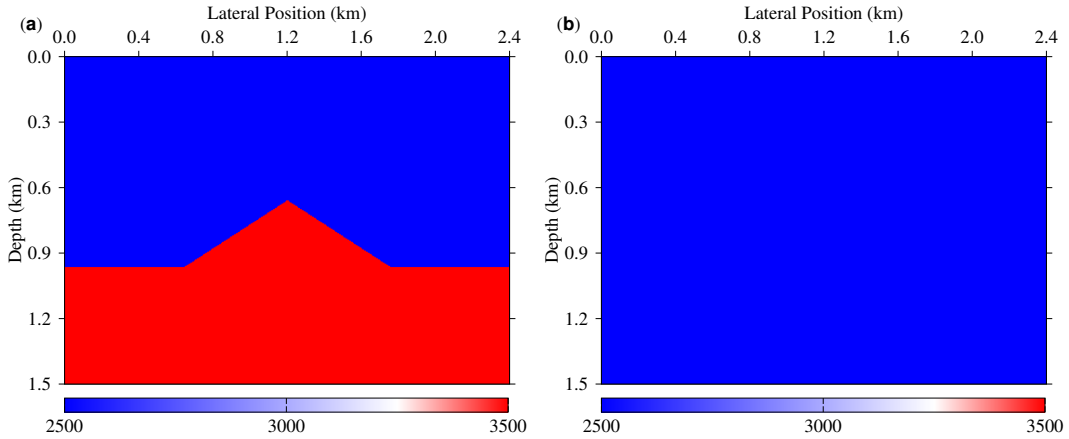


Figure 3.3: Velocity models for testing the framework of FWI to reverse time migration (**a**) a two layered model with a triangular shape and **b**) input constant velocity model.

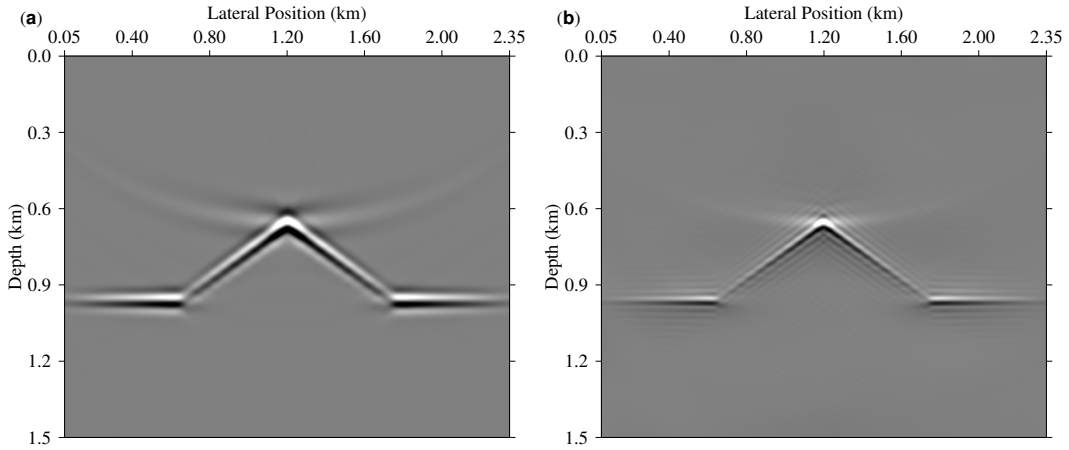


Figure 3.4: Reverse time migration results (**a**) by diagonal hessian scaling and **b**) by Gauss-Newton - conjugate gradient approach.

3.6 Discussion and Summary

In this chapter, the mathematical formulation of FWI in the least square sense is derived. The Born approximation relative to RTM is discussed. The relationship between RTM and FWI in frequency domain is established. The framework of FWI is used to test reverse

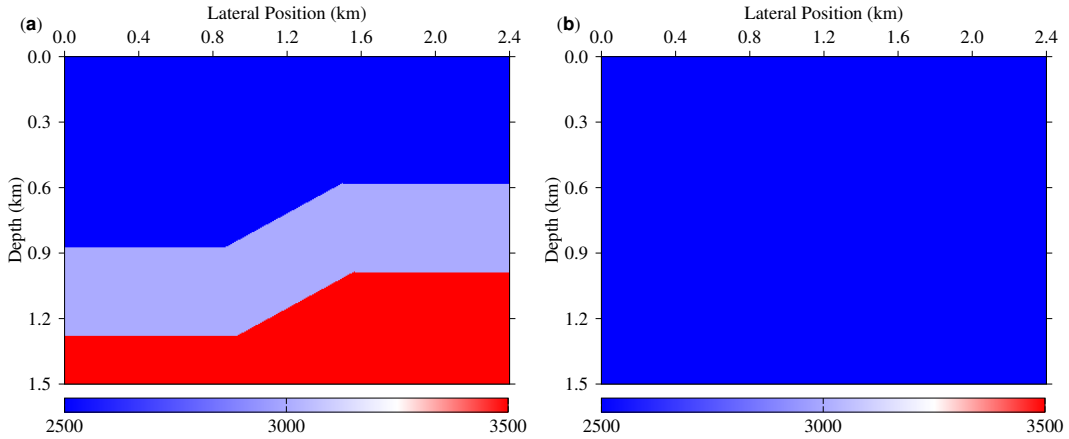


Figure 3.5: Velocity models for testing the framework of FWI to reverse time migration (**a**) a three layered model and **b**) input constant velocity model.

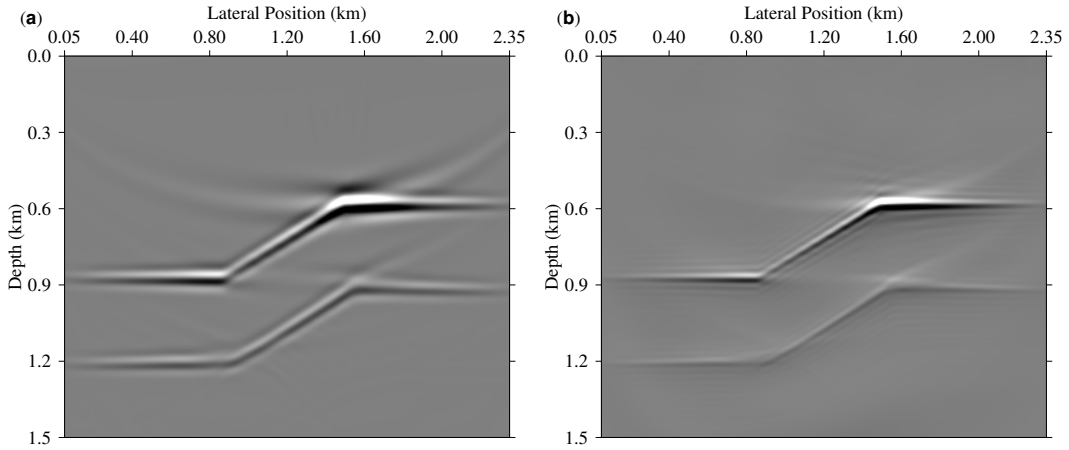


Figure 3.6: Reverse time migration results (**a**) by diagonal Hessian scaling and **b**) by Gauss-Newton - conjugate gradient approach.

time-migration with numerical examples. In particular, two scaling namely diagonal Hessian and Gauss-Newton are tested for producing high resolution RTM images.

Three simple models were examined each with the two scalings. It can be clearly see from Figure 3.2, Figure 3.4 and Figure 3.6 that the Gauss-Newton approach gives a more sharper reflector as compared to the diagonal scaling. It is not surprising as the Gauss-Newton appropriately scales the gradient to resolve the reflector better. The Gauss-Newton tends

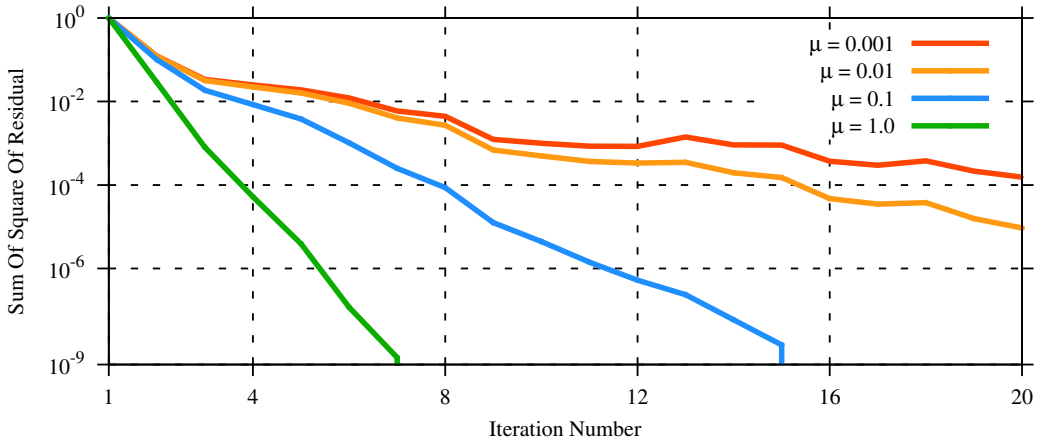


Figure 3.7: The convergence of the conjugate gradient algorithm at the first frequency f = 2.5 Hz for computing the search direction using the Gauss-Newton approach. In this figure, the four curves show normalized sum of square of residual versus iteration numbers.

to eliminate the wavelet, but it also produces artifacts near the main reflector amplitude. It is also evident that the diagonal Hessian scaling is computationally inexpensive relative to the Gauss-Newton case. The diagonal scaling does not need extra computational time as the diagonal Hessian can be computed along with the gradient. On the other hand, the Gauss-Newton approach uses the conjugate gradient algorithm to compute the search direction that needs two extra forward computation at each iteration. The conjugate gradient algorithm does not coverage rapidly and hence takes a longer time to converge. With the aid of the pre-conditioner, it is possible to speed up the convergence at the cost of its ability to give sharper reflectors.

Figure 3.7 shows the convergence curve of the square of the residual versus iteration number of the conjugate gradient for four different values of μ = 0.001, 0.01, 0.1, 0.1 at a single frequency. It is easy to see that the smaller values of μ show nonsmooth curve that need a large number of iterations. As the values of μ increase, the square of the residual decreases linearly with fewer iterations.

The most important point to note is that the computation of the search direction at first iteration in the framework of FWI gives a migrated image. In RTM, the velocity is not updated instead the sum of search directions from various shots and frequencies provides the final migrated image. The FWI algorithms in the next subsequent chapters use FWI framework derived in this chapter with an optimal number of internal iteration along with step length estimation to update the velocity.

CHAPTER 4

Optimization in Full Waveform Inversion

In this chapter, four gradient-based optimization methods in FWI are presented: gradient descent, diagonal Hessian, Gauss-Newton, and Quasi-Newton. In particular, the Quasi-Newton is discussed in depth. The FWI algorithm is extended with regularization techniques via the L-BFGS: 2D smoothing and total variation regularizations. Finally, numerical examples with three velocity models that have a different degree of complexity are given to compare the optimization methods. This chapter also provides a framework for next two chapters, where time-lapse inversion strategies are covered.

4.1 Introduction to Numerical Optimization in FWI

A numerical optimization is an important tool for finding a solution to a problem of a given physical system. This is done by defining an objective function that portrays a quantitative measure of the underlying system. The aim of numerical optimization is, therefore, to find a solution by maximizing or minimizing the objective function subject to constraints. The three essential things that one should take into account in any optimization problem are the convexity of the problem, the performance and the convergence of the algorithms that are used to minimize or maximize the objective function. A successful optimization technique is the one that converges to the global minimum faster. See a simple one-dimensional polynomial function that has both local and global minimum in Figure 4.1.

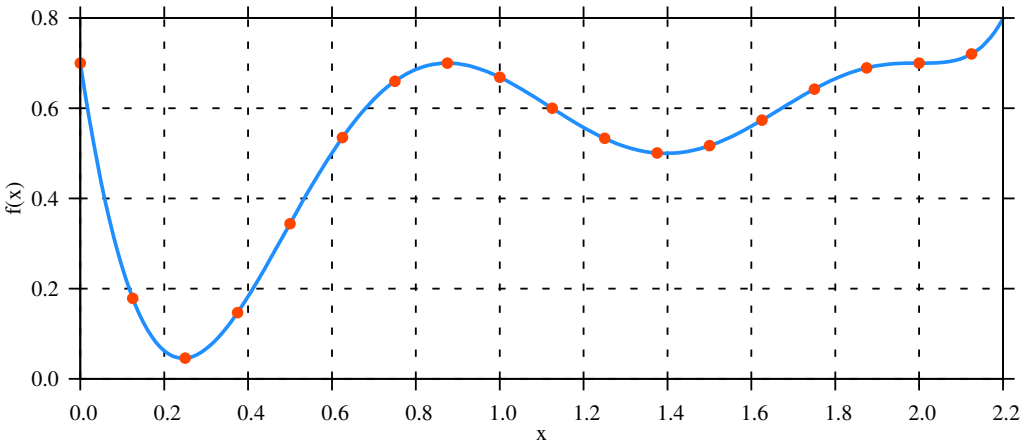


Figure 4.1: A one-dimensional polynomial curve with local and global minimum.

There are several categories of optimization methods; however, they can broadly be grouped into non-gradient and gradient based. The non-gradient based methods do not seek gradient and Hessian of the objective function. Therefore, they can handle problems in which the objective function do not have smooth first or second order derivative. Non-gradient based methods include genetic algorithms, grid searches, stochastic, and nonlinear simplex, etc. In the gradient based methods, at least smooth first order derivative of the objective function must exist. These kinds of optimization techniques include gradient descent, quasi-Newton, and Newton. The Newton methods require second order derivative of the objective function, the Hessian. In some optimization problems, a hybrid of the non-gradient and gradient-based techniques can also be used.

In full waveform inversion, the gradient-based methods are the most common ones. The efficient computations of gradient and Hessian are derived in Chapter 3. In this section, the implementation of the gradient and Hessian of different FWI optimization techniques are covered.

4.2 Gradient Descent (GD)

One of the simplest line search methods is the steepest descent algorithm. It is a generic algorithm to minimize objective functions and works well as long as functions are well behaved. The search direction in the gradient descent method is simply the negative of the gradient,

$$\mathbf{p}_{GD} = -\nabla J, \tag{4.1}$$

where ∇J is the gradient

$$\mathbf{g}(\omega_{j_{\omega_g}}, X_{j_s}) = \Re\{[\mathbf{F}^v]^T[\mathbf{A}^{-1}]^T\mathbf{R}^T(\mathbf{Ru} - \mathbf{d}^{obs})^*\}. \tag{4.2}$$

The solution update from k^{th} iteration to $(k+1)^{th}$ iteration take the form

$$\mathbf{m}^{k+1} = \mathbf{m}^k + \gamma \mathbf{p}_{GD}, \tag{4.3}$$

where γ is the step length. Although, the computational time for each cycle is relatively small, this method is inefficient. The linear rate of convergence (convergence rate = 1) (Bertsekas, 1999) requires many iterations of the algorithm to reach the minimum and it is susceptible to being trapped on local minimum within the space. Therefore, the optimal choice of the step length via line search methods is very crucial.

4.3 Gauss-Newton (GN)

The Gauss-Newton method is an approximation to the full Newton method using non-diagonal approximate Hessian. In other words, the diagonal Hessian method is a simplified approximation version of the Gauss-Newton method in which the off-diagonal elements are zeros. Unlike the gradient descent method, the pure Newton as well as the Gauss-Newton methods can have super-linear rate of convergence (convergence rate > 1) (Bertsekas, 1999). Under certain conditions, both methods can achieve quadratic rate of convergence (convergence rate = 2) (Schaback, 1985; Jiang, 1999). The main challenge using the Gauss-Newton method is that it is impractical to build the Hessian explicitly for large problems. Moreover, the computational cost is enormous as compared to the gradient and diagonal Hessian methods.

Unlike the two methods, the Gauss-Newton method needs at least two additional forward computations to build the Hessian. The advantage of Gauss-Newton over the gradient and approximate Hessian methods is that it gives a better search direction. Although it demands high computational time at each solution update, the convergence to the minimum can be achieved with relatively large search direction steps. Gauss-Newton can be thought of as the first order scattering that contributes towards the full Hessian.

The direct or Non-constrained approach to establish the Gauss-Newton method is already derived in Chapter 3. Further, the constrained approach is given in Appendix A. Thus, the Gauss-Newton search direction is computed by the following expression,

$$\mathbf{p}_{GN} = -\mathbf{H}_a^{-1}\nabla J, \tag{4.4}$$

where $\mathbf{H}_a$ is an approximate Hessian matrix given by,

$$\mathbf{H}_a = \Re\{(\mathbf{F}^v)^T(\mathbf{A}^{-1})^T\mathbf{R}^T\mathbf{R}\mathbf{A}^{-1}(\mathbf{F}^v)^*\}. \tag{4.5}$$

For large size problems, determining the inverse of the approximate Hessian $\mathbf{H}_a$ can be intensive computationally. The commonly used approach to overcome this issue is the conjugate gradient method. It is in the heart of the Gauss-Newton and full Newton methods in which explicit construction of Hessian is impractical for large problems. The conjugate gradient method needs certain number of product of the Hessian with vector operations instead of explicit computation of the inverse of the Hessian, that is

$$\mathbf{H}_a\mathbf{v} \quad = \Re\{(\mathbf{F}^v)^T(\mathbf{A}^{-1})^T\mathbf{R}^T\mathbf{R}\mathbf{A}^{-1}(\mathbf{F}^v)^*\}\mathbf{v}. \tag{4.6}$$

For a specified number of CG iterations, the end result of the algorithm is the search direction $\mathbf{p}_{GN}$, which is the product of the inverse of the Hessian and the gradient.

Besides, it is susceptible to failure when optimization commences from distant positions in the space as the Hessian may not be positive definite and hence not convex. It is important to remember that positive definiteness is a condition required for the search direction both to point downhill and to be reasonably oriented. To guarantee a positive definite Hessian, a modified version, known as Levenberg's approach (Marquardt, 1963), is often used. The idea is simply to replace $\mathbf{H}_a$ by $\mathbf{H}_a + \lambda\mathbf{I}$. Note that if λ is large, this method is equivalent to a simple gradient descent. Taking into account Levenberg's modification, Equation 4.7 can be expressed as,

$$(\mathbf{H}_a + \lambda\mathbf{I})\mathbf{v} = -\nabla J. \tag{4.7}$$

The solution update from k^{th} iteration to $(k+1)^{th}$ iteration take the form

$$\mathbf{m}^{k+1} = \mathbf{m}^k + \gamma\mathbf{p}_{GN}. \tag{4.8}$$

4.4 Diagonal Hessian (DH)

As derived in chapter three, the diagonal Hessian approach is one of the possible choices for optimizing the FWI problem. The efficient gradient calculation allows us to compute approximate diagonal without additional computational cost (Shin et al., 2001). Unlike the gradient descent, the diagonal Hessian gives an appropriate search direction with less number of steps to reach the minimum and hence faster convergence.

$$\mathbf{p}_{DH} = -\mathbf{H}_{daig}^{-1}\nabla J, \tag{4.9}$$

where $\mathbf{H}_{diag}$ is the diagonal Hessian matrix given by,

$$\mathbf{H}_{diag} \approx C_A \Re\{\mathbf{h}_1\ \mathbf{h}_2\ \mathbf{h}_3\ \cdots\ \mathbf{h}_i\ \cdots\ \mathbf{h}_{N_{xz}}\} \tag{4.10}$$

and

$$\mathbf{h}_i = \mathbf{u}^T \left(\frac{\partial \mathbf{A}}{\partial m_i}\right)^T \left(\left(\frac{\partial \mathbf{A}}{\partial m_i}\right)\mathbf{u}\right)^*. \tag{4.11}$$

The solution update from k^{th} iteration to $(k+1)^{th}$ iteration takes the form

$$\mathbf{m}^{k+1} = \mathbf{m}^k + \gamma \mathbf{p}_{DH}. \tag{4.12}$$

4.5 Quasi-Newton

BFGS is an alternative Quasi-Newton numerical optimization technique to compute the approximate Hessian (or inverse of the approximate Hessian). There are two varieties of BFGS methods: the Direct BFGS and the Limited Memory BFGS (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970; Nocedal, 1980).

4.5.1 Direct BFGS

The direct use of BFGS begins with the a known diagonal matrix (usually identity matrix). At each iteration the Hessian (or inverse of Hessian) is updated and saved for later use for the next iteration. In BFGS, the Hessian is chosen to satisfy a secant equation, which is the Taylor series of the gradient function,

$$\nabla J(\mathbf{m}_k + \Delta \mathbf{m}_k) = \nabla J(\mathbf{m}_k) + \mathbf{H}\Delta \mathbf{m}. \tag{4.13}$$

The Quasi-Newton condition for updating the Hessian can be written as

$$\mathbf{H}_{k+1}(\mathbf{m}_{k+1} - \mathbf{m}_k) = \nabla J(\mathbf{m}_{k+1}) - \nabla J(\mathbf{m}_k). \tag{4.14}$$

Equation 4.15 can be rewritten in the most common form as

$$\mathbf{H}_{k+1}\mathbf{s}_k = \mathbf{y}_k, \tag{4.15}$$

where $\mathbf{s}_k = \mathbf{m}_{k+1} - \mathbf{m}_k$ and $\mathbf{y}_k = \nabla J(\mathbf{m}_{k+1}) - \nabla J(\mathbf{m}_k)$.

Davidon Fletcher and Powell (DFP)

Let $(\mathbf{H}_{k+1}^{DFP})^{-1}$ be $\mathbf{B}_{k+1}$. Equation 4.15 can be rewritten as,

$$\mathbf{B}_{k+1}\mathbf{y}_k = \mathbf{s}_k. \tag{4.16}$$

Equation 4.16 is the secant equation for the DFP method (Davidon, 1991). The approximate inverse is updated by adding two matrices, that is

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \mathbf{V_1}_k + \mathbf{V_2}_k. \tag{4.17}$$

The matrices $\mathbf{V_1}_k$ and $\mathbf{V_2}_k$ are symmetric rank one matrices to be chosen such that the Hessian remain symmetric and positive definite, that is

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \alpha\mathbf{v_1}_k\mathbf{v_1}_k^T + \beta\mathbf{v_2}_k\mathbf{v_2}_k^T, \tag{4.18}$$

where $\mathbf{v_1}_k = \mathbf{s}_k$ and $\mathbf{v_2}_k = \mathbf{B}_k\mathbf{y}_k$. Multiplying both side of Equation 4.18 by $\mathbf{y}_k$ on the right side, we have

$$\mathbf{B}_{k+1}\mathbf{y}_k = \mathbf{B}_k\mathbf{y}_k + \alpha\mathbf{v_1}_k\mathbf{v_1}_k^T\mathbf{y}_k + \beta\mathbf{v_2}_k\mathbf{v_2}_k^T\mathbf{y}_k, \tag{4.19}$$

Employing the secant condition $\mathbf{s}_k = \mathbf{B}_{k+1}\mathbf{y}_k$ in Equation 4.19 leads to

$$\mathbf{s}_k = \mathbf{B}_k\mathbf{y}_k + \alpha\mathbf{v_1}_k\mathbf{v_1}_k^T\mathbf{y}_k + \beta\mathbf{v_2}_k\mathbf{v_2}_k^T\mathbf{y}_k, \tag{4.20}$$

By replacing $\mathbf{v_1}_k = \mathbf{s}_k$ and $\mathbf{v_2}_k = \mathbf{B}_k\mathbf{y}_k$ in Equation 4.20 and rearranging terms, we have

$$\mathbf{s}_k - \mathbf{B}_k\mathbf{y}_k = \alpha\mathbf{s}_k(\mathbf{s}_k^T\mathbf{y}_k) + \beta(\mathbf{B}_k\mathbf{y}_k)(\mathbf{y}_k^T\mathbf{B}_k\mathbf{y}_k). \tag{4.21}$$

Since $(\mathbf{s}_k^T\mathbf{y}_k)$ and $(\mathbf{y}_k^T\mathbf{B}_k\mathbf{y}_k)$ scalar, Equation 4.21 can be rewritten as

$$\mathbf{s}_k - \mathbf{B}_k\mathbf{y}_k = \alpha(\mathbf{s}_k^T\mathbf{y}_k)\mathbf{s}_k + \beta(\mathbf{y}_k^T\mathbf{B}_k\mathbf{y}_k)(\mathbf{B}_k\mathbf{y}_k). \tag{4.22}$$

Note that $\mathbf{B}$ is symmetric. Comparing both side of Equation 4.21, α and β are found to be

$$\alpha = \frac{1}{\mathbf{s}_k^T\mathbf{y}_k} \tag{4.23}$$

and

$$\beta = -\frac{1}{\mathbf{y}_k^T\mathbf{B}_k\mathbf{y}_k}. \tag{4.24}$$

Substituting Equations 4.23 and 4.24 into Equation 4.18, we have

$$(\mathbf{H}_{k+1}^{DFP})^{-1} = \mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{s}_k^T\mathbf{y}_k} - \frac{\mathbf{B}_k\mathbf{y}_k\mathbf{y}_k^T\mathbf{B}_k}{\mathbf{y}_k^T\mathbf{B}_k\mathbf{y}_k}. \tag{4.25}$$

Equation 4.25 is enough to calculate the search direction. If the Hessian is required and since the matrices in Equation 4.25 are special kind of matrices, the Hessian can be calculated efficiently by applying the Sherman-Morrison formula.

For an invertible square matrix **A** and **v1** and **v2** are column vectors, the Sherman-Morrison formula (Sherman and Morrison, 1950) states that

$$(\mathbf{A} + \mathbf{v_1}\mathbf{v_2}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{v_1}\mathbf{v_2}^T\mathbf{A}^{-1}}{1 + \mathbf{v_2}^T\mathbf{A}^{-1}\mathbf{v_1}} \tag{4.26}$$

for $1 + \mathbf{v_2}^T\mathbf{A}^{-1}\mathbf{v_1} \neq 0$. On account of Equation 4.26, the DFP Hessian becomes,

$$\mathbf{H}_{k+1}^{DFP} = \mathbf{H}_k + \frac{(\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k + \mathbf{y}_k^{\mathrm{T}}\mathbf{H}_k\mathbf{y}_k)(\mathbf{s}_k\mathbf{s}_k^{\mathrm{T}})}{(\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k)^2} - \frac{\mathbf{H}_k\mathbf{y}_k\mathbf{s}_k^{\mathrm{T}} + \mathbf{s}_k\mathbf{y}_k^{\mathrm{T}}\mathbf{H}_k}{\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k}. \tag{4.27}$$

The Hessian $\mathbf{H}_0$ initialized to be a diagonal or an identity matrix with the straightforward inverse. For the remaining subsequent iterations k, the inverse of the Hessian at $k-1$ should be available in memory. Thus, for large size problems, this approach is again impractical.

The search direction can be found by applying the inverse of the Hessian on the gradient,

$$\mathbf{p}_{BFGS_{k+1}} = -\mathbf{B}_{k+1}\nabla J_{k+1} = -(\mathbf{H}_{k+1}^{DFP})^{-1}\nabla J_{k+1}. \tag{4.28}$$

Broyden-Fletcher-Goldfarb-Shanno (BFGS)

The BFGS update for the inverse of the Hessian can be derived in the same steps as the DFP method. The DFP and BFGS are complementary equations of one another. The secant equation for BFGS is Equation 4.15. Thus, **B** replaced by **H**, **s** replaced by **y**, **y** replaced by **s** that appeared in DFP to get the BFGS Hessian update. By doing so, the Hessian becomes

$$(\mathbf{H}_{k+1}^{BFGS}) = \mathbf{H}_k + \frac{\mathbf{y}_k\mathbf{y}_k^T}{\mathbf{y}_k^T\mathbf{s}_k} - \frac{\mathbf{H}_k\mathbf{s}_k\mathbf{s}_k^T\mathbf{H}_k}{\mathbf{s}_k^T\mathbf{H}_k\mathbf{s}_k}. \tag{4.29}$$

Since the matrices in Equation 4.29 are special kind, the inverse of $(\mathbf{H}_{k+1}^{BFGS})$ can be derived via the Sherman-Morrison formula, that is

$$(\mathbf{H}_{k+1}^{BFGS})^{-1} = \mathbf{H}_k^{-1} - \frac{\mathbf{H}_k\mathbf{y}_k\mathbf{s}_k^{\mathrm{T}} + \mathbf{s}_k\mathbf{y}_k^{\mathrm{T}}\mathbf{H}_k}{\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k} + \frac{(\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k + \mathbf{y}_k^{\mathrm{T}}\mathbf{H}_k\mathbf{y}_k)(\mathbf{s}_k\mathbf{s}_k^{\mathrm{T}})}{(\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k)^2} \tag{4.30}$$

or equivalently

$$(\mathbf{H}_{k+1}^{BFGS})^{-1} = \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}\right)\mathbf{H}_k^{-1}\left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}. \tag{4.31}$$

From Equations 4.25 and 4.31 we can see that both DFP and BFGS methods require to keep the inverse of the Hessian at j^{th} iteration in a memory for calculating the inverse of the Hessian at $(j+1)^{th}$ iteration.

4.5.2 Limited Memory BFGS (L-BFGS)

The Limited memory BFGS approach is an answer for large size problems in which the direct DFP or BFGS method is nonviable. In L-BFGS, the history of models and gradients are kept and used to implicitly build the action of the inverse of the Hessian on the gradient to calculate the search direction (Nocedal, 1980). The L-BFGS and the Gauss-Newton via conjugate gradient are similar in a sense that explicit calculation of the inverse of the Hessian is not required. It is also important to note that the L-BFGS method also requires some memory to save the history of model parameters and gradients that are vectors. The vectors take up limited memory as compared to the Hessian in direct use of BFGS method, which is one of the advantages of the L-BFGS method. Besides, it requires fewer operations in the calculation of the search direction because only vectors are involved.

In BFGS method, the inverse of the Hessian at k^{th} iteration is constructed from the $k-1$ updates that are implicitly embedded in the inverse of the Hessian at the $(k+1)^{th}$ step. In other words, there is no restriction on the number of updates to be used in the computation of the inverse of the Hessian. The L-BFGS, unlike the direct BFGS, uses only a specified optimal number of history of model parameters and gradients at each iteration. The number of updates, N_m, is

$$N_m = \begin{cases} k, & \text{if } k < m \\ N_m, & \text{if } k \geqslant N_m \end{cases}. \tag{4.32}$$

The inverse of the Hessian updates at the k^{th} iteration come from $k - N_m$ to $k-1$ most recent history of $\mathbf{s}$ and $\mathbf{y}$ i.e.
$\{\mathbf{s}_{k-N_m}, \mathbf{s}_{k-(N_m-1)}, \mathbf{s}_{k-(N_m-2)} \cdots \mathbf{s}_{k-1}\}$ and $\{\mathbf{y}_{k-N_m}, \mathbf{y}_{k-(N_m-1)}, \mathbf{y}_{k-(N_m-2)} \cdots \mathbf{y}_{k-1}\}$. For optimal number of history of $\mathbf{s}$ and $\mathbf{y}$, the BFGS and L-BFGS give approximately same search direction.

The inverse of the Hessian in L-BFGS method is calculated implicitly by the recursive method over the specified number of updates. Thus, the inverse of the Hessian expression obtained

in the direct BFGS method is rewritten recursively. The recursive version of Equation 4.31 for N_m updates is given by,

$$\begin{aligned}(\mathbf{H}_{k+1}^{LBFGS})^{-1} &= (\mathbf{V}_k^T \cdots \mathbf{V}_0^T)\mathbf{H}_0^{-1}(\mathbf{V}_0 \cdots \mathbf{V}_k) \\ &+ \sum_{j=k-N_m}^{k} \frac{1}{\mathbf{y}_j^T \mathbf{s}_j}(\mathbf{V}_k^T \cdots \mathbf{V}_{j+1}^T)\mathbf{s}_j\mathbf{s}_j^T(\mathbf{V}_{j+1} \cdots \mathbf{V}_k) \\ &+ \frac{\mathbf{s}_k\mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}. \end{aligned} \tag{4.33}$$

The matrix $\mathbf{H}_0$ is initialized to be either a diagonal or an identity matrix. The corresponding search direction for the L-BFGS method can be obtained by the expression,

$$\mathbf{p}_k^{LBFGS} = -(\mathbf{H}_{k+1}^{LBFGS})^{-1}\nabla J_k. \tag{4.34}$$

On account of Equation 4.33, Equation 4.34 can be easily computed efficiently in two steps. For a step length γ, the solution update is given by,

$$\mathbf{m}_{k+1}^{LBFGS} = \mathbf{m}_k + \gamma \mathbf{p}_k^{LBFGS}. \tag{4.35}$$

4.6 Step Length γ

Once the search direction is computed by the employed algorithm, how far to go in that direction to eventually reach the minimum is determined by the step length, γ. Exact or inexact methods can be used to estimate a step length. Exact methods look for a value of a step length that minimize $J(\mathbf{m}^k + \gamma\mathbf{p}^k)$, i.e. finding a value of γ which makes derivative of J with respect to γ vanish: $J'(\mathbf{m}^k + \gamma\mathbf{p}^k)$=0.

Sufficient decrease condition (Armijo's rule)	Curvature condition
$J(\mathbf{m}^k + \gamma\mathbf{p}^k) < J(\mathbf{m}^k) + c_1(\mathbf{g}^k)^T\mathbf{p}^k$	$\mathbf{g}(\mathbf{m}^k + \gamma\mathbf{p}^k)^T\mathbf{p}^k \geqslant c_2(\mathbf{g}^k)^T\mathbf{p}^k$

Table 4.1: A general two-step Wolfe conditions for estimating a step length, where $c_1 = 10^{-4}$ and $c_2 = 0.9$ are constants.

Unlike the exact method, an inexact method does not need an advanced minimization

technique. It uses intelligent approaches for estimating γ such as backtracking, quadratic or cubic interpolation. If the estimated step length satisfies the two step Wolfe conditions (Armijo, 1966; Wolfe, 1969), it is an optimal step length. The Wolfe conditions shown in Table 4.1 take into account the Armijo's inequality and curvature condition of the cost function $J(\mathbf{m}^{k+1})$ that ensure a sufficient decrease of the cost function and its gradient with respect to the step length, respectively.

4.7 Regularization in FWI

The standard step in formulating inverse problems is to write down a system of equations (linear or nonlinear) that fully describe the relationship between the observed data, the physics model, and the unknown model parameters. Depending on the number of equations and the unknown model parameters, the system of equations can be classified as an under-determined, even-determined and over-determined system of equations. For certain defined number of equations and unknowns, the problem is said to be overdetermined if the system of equations outnumbers the number of unknown parameters. In this case, the systems of equations contain too much information which may lead to contradictions. In even-determined problems, the number of equations is equal to the number of unknowns. It implies that there is enough information to determine the model parameters. But this does not mean that the system of equation can estimate the parameters accurately. It still depends on the number of independent equations in the system. Under-determined problems arise when the number of equations is fewer than the number of unknowns, i.e., there is less information to retrieve the parameters of interest.

In general, an inverse problem could be well-posed or an ill-posed problem. A problem is called well-posed if the solution exists, the solution is unique, and the solution continuously depends on the data. If it is not well-posed, it is an ill-posed problem, i.e., the solution may not exist, or the solution is not unique, or a small perturbation in the data brings large change in the solution.

In many geophysical problems, most often, the solutions are non-unique, and so the problems turn out to be ill-posed. Such kinds of problems need prior information incorporated into the inverse problem through regularization techniques to get the desired results. Regularization penalizes models in favor of a prior information without losing the necessary information from the input data.

In the next two section, two regularization techniques namely 2D smoothing and total variation regularizations are incorporated into the full waveform inversion formulation.

4.7.1 2D Smoothing Regularization

The 2D smoothing regularization as the name indicates it helps to smooth out outliers in the model during the inversion process. Thus, a regularization term with 2D smoothing differencing operators is added to the principal objective function (Tikhonov, 1963; Hansen, 1992; Engl et al., 1996; Calvetti et al., 2000).

Let the main objective function term be $J^d(\mathbf{m})$. Equation 3.6 can be modified by adding the regularization function term $\mathbf{J}^{SM}(\mathbf{m})$ as

$$J(\mathbf{m}) = J^d(\mathbf{m}) + \mu J^{SM}(\mathbf{m}), \tag{4.36}$$

where the regularization function J^{SM} is given by,

$$J^{SM}(\mathbf{m}) = ||\mathbf{D}_x\mathbf{m}||_2^2 + ||\mathbf{D}_z\mathbf{m}||_2^2, \tag{4.37}$$

and μ is the regularization parameter. The matrices $\mathbf{D}_x$ & $\mathbf{D}_z$ are lateral and vertical first order difference operators, respectively. The goal is to find a model $\mathbf{m}$ that minimizes the cost function by honoring the input data and by reducing roughness.

Gradient

The total gradient that corresponds to Equation 4.36 becomes,

$$\mathbf{g} = \mathbf{g}^d + \mu\mathbf{g}^{SM}. \tag{4.38}$$

The gradient ($\mathbf{g}^d$) of the cost function without the regularization term are derived in Chapter 3. The derivative of the regularization term with respect to the model $\mathbf{m}$ is found to be,

$$\begin{aligned} \mathbf{g}^{SM} &= \mathbf{D}_x^T\mathbf{D}_x\mathbf{m} + \mathbf{D}_z^T\mathbf{D}_z\mathbf{m}, \\ &= (\mathbf{D}_x^T\mathbf{D}_x + \mathbf{D}_z^T\mathbf{D}_z)\mathbf{m}. \end{aligned} \tag{4.39}$$

Hessian

The contribution of the Hessian from the main cost function $J^d(\mathbf{m})$ is already derived in Chapter 3. The total Hessian from both J^d and J^{SM} has the form

$$\mathbf{H} = \mathbf{H}^d + \mu\mathbf{H}^{SM}, \tag{4.40}$$

where $\mathbf{H}^d = \mathbf{H}_{daig}$ in the case of diagonal Hessian or $\mathbf{H}^d = \mathbf{H}_a$ in the case of Gauss-Newton. The contribution of the regularization function to wards the total Hessian is given by,

$$\mathbf{H}^{SM} = \mathbf{D}_x^T\mathbf{D}_x + \mathbf{D}_z^T\mathbf{D}_z. \tag{4.41}$$

4.7.2 2D Total Variation Regularization

Total variation regularization is one of the few regularization methods often used in Mathematics and Engineering particularly in digital image processing (Rudin et al., 1992; Vogel and Oman, 1996; Strong and Chan, 2003). Images or signals with excessive details (noise) have high total variation. The total variation regularization minimizes excess variation in the image or signal; thereby removing unwanted noise. The main advantage of total variation regularization is its ability in reducing noise by preserving important features (edge) in an image.

To incorporate a TV regularization, Equation 3.6 can be modified by adding the regularization function term $\mathbf{J}^{TV}(\mathbf{m})$ as

$$J(\mathbf{m}) = J^d(\mathbf{m}) + \mu J^{TV}(\mathbf{m}), \tag{4.42}$$

where the regularization function J^{TV} is given by,

$$J^{TV}(\mathbf{m}) = \sum_i^{N_x}\sum_j^{N_z}\sqrt{(\mathbf{D}_{i,j}^x\mathbf{m})^2 + (\mathbf{D}_{i,j}^z\mathbf{m})^2 + \beta}. \tag{4.43}$$

A constant β is added in the expression to keep the function convex.

Gradient

The total gradient that corresponds to Equation 4.42 becomes,

$$\mathbf{g} = \mathbf{g}^d + \mu\mathbf{g}^{TV}. \tag{4.44}$$

The gradient ($\mathbf{g}^d$) of the cost function without the regularization term are derived in Chapter 3. The derivative of the regularization term with respect to the model $\mathbf{m}$ is found to be,

$$\begin{aligned}\mathbf{g}^{TV} &= \mathbf{D}_x^T\mathbf{Q}\mathbf{D}_x\mathbf{m} + \mathbf{D}_z^T\mathbf{Q}\mathbf{D}_z\mathbf{m},\\ &= (\mathbf{D}_x^T\mathbf{Q}\mathbf{D}_x + \mathbf{D}_z^T\mathbf{Q}\mathbf{D}_z)\mathbf{m},\end{aligned} \tag{4.45}$$

where $\mathbf{Q}$ is a diagonal matrix whose elements are given by,

$$Q_{i,i} = \frac{1}{\sum_{i}^{N_x}\sum_{j}^{N_z}\sqrt{(\mathbf{D}^x_{i,j}\mathbf{m})^2 + (\mathbf{D}^z_{i,j}\mathbf{m})^2 + \beta}}. \tag{4.46}$$

Hessian

The total Hessian from both J^d and J^{TV} has the form,

$$\mathbf{H} = \mathbf{H}^d + \mu\mathbf{H}^{TV}, \tag{4.47}$$

where $\mathbf{H}^d = \mathbf{H}_{daig}$ in the case of diagonal Hessian or $\mathbf{H}^d = \mathbf{H}_a$ in the case of Gauss-Newton. The contribution of the TV regularization function to wards the total Hessian is given by,

$$\mathbf{H}^{TV} \approx \mathbf{D}_x^T\mathbf{Q}\mathbf{D}_x + \mathbf{D}_z^T\mathbf{Q}\mathbf{D}_z. \tag{4.48}$$

Note that another term from the $\mathbf{H}^{TV}$ is ignored to keep the Hessian positive definite. See Appendix B for detail derivations of gradient and Hessian of the total variation regularization.

4.8 Numerical Examples

To demonstrate the performance of the four optimization methods, three velocity models are considered: 'Mac' model, Amoco model and Marmousi model. These models have different degree of complexity.

4.8.1 'Mac' Velocity Model

The 'Mac' model is a simple layered model with an anomaly. Its size is $N_x \times N_z = 360 \times 140$ with grid spacings $dx = 8$ m and $dz = 8$ m. The true 'Mac' velocity model is given in Figure 4.2 (**a**). Data are generated using the true velocity model and a Ricker wavelet whose central frequency is 20 Hz. A total of 72 shots each separated by 40 m. Each shot has 178 receivers with receiver separation of 16 m. Figures 4.3 (**a**), (**b**), and (**c**) are sample data at three different shot locations. Figure 4.2 (**b**) is an initial velocity model used for inversion.

The first inversion example is without noise and without proper regularization. But the solutions are bounded (constrained) between a minimum and maximum to avoid instability in the inversion. Figure 4.2 (**c**) is velocity inversion result by gradient descent (GD).

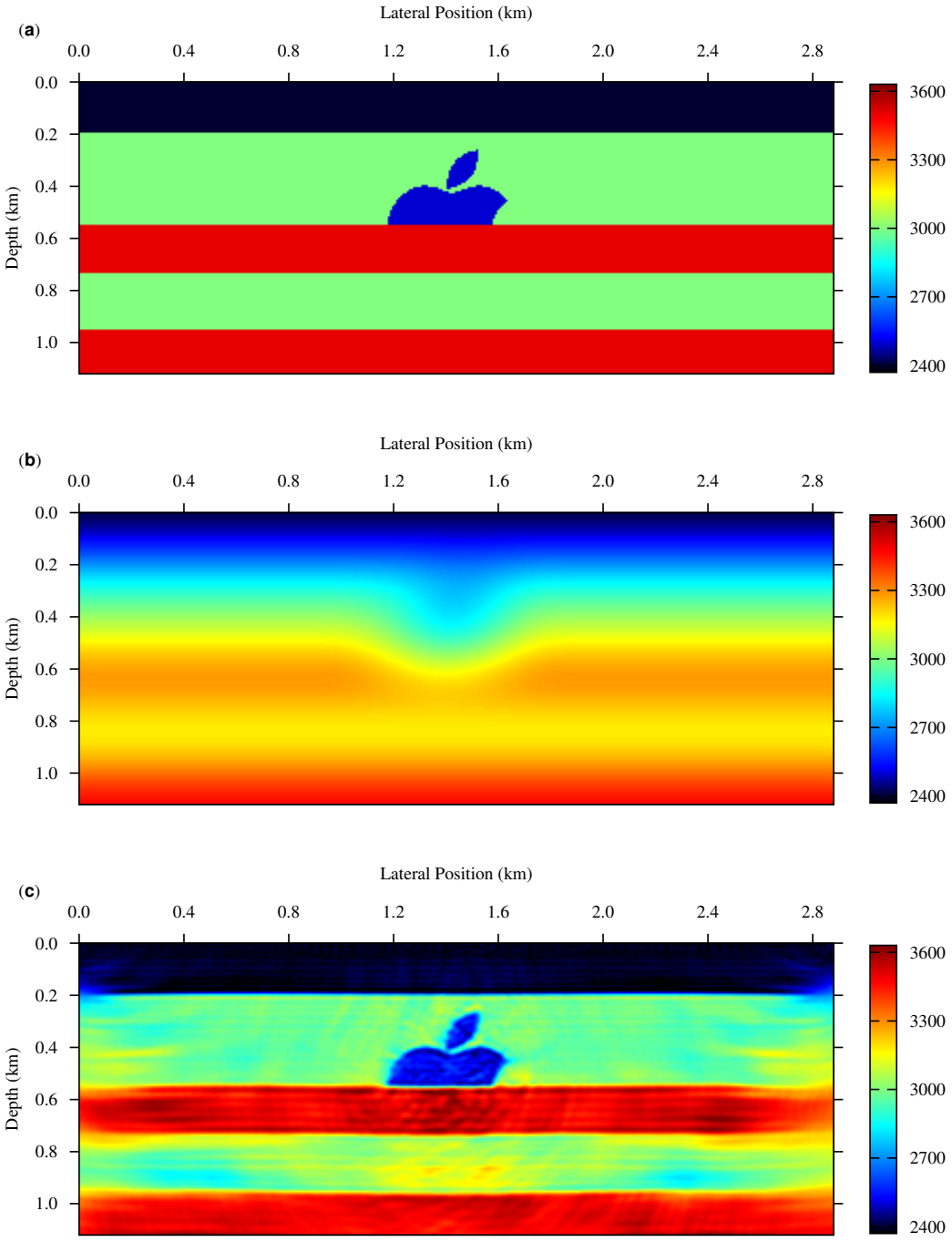


Figure 4.2: (**a**) A true 'Mac' model, **b**) initial velocity model and **c**) inversion result by gradient descent.

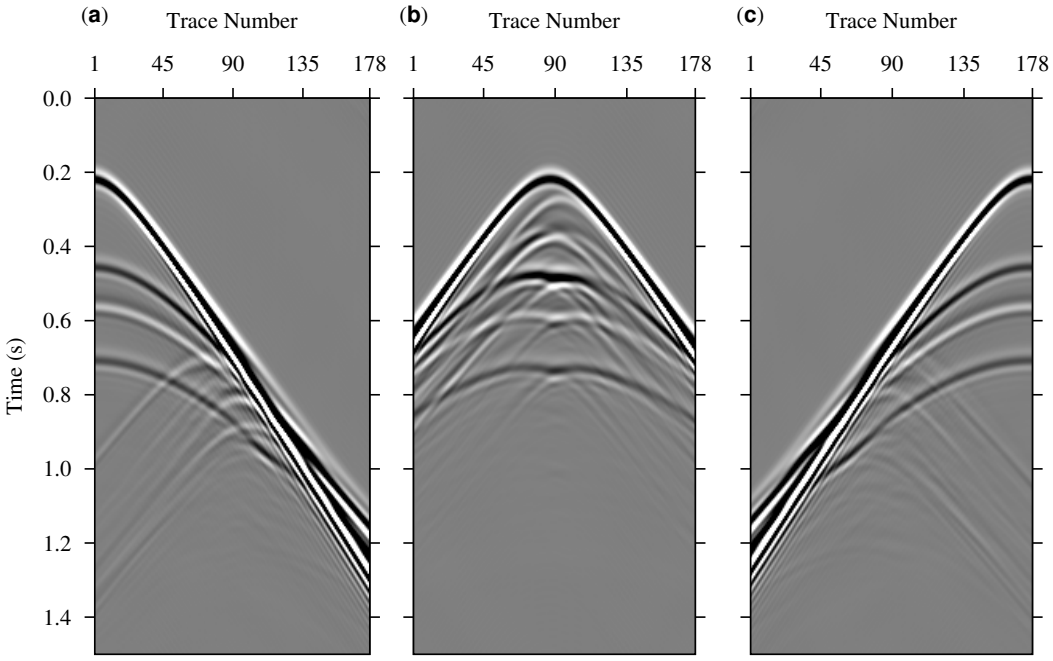


Figure 4.3: A sample generated pressure seismic data of the 'Mac' model at three different source locations.

Figure 4.4 (**a**), Figure 4.4 (**b**), and Figure 4.4 (**c**) are velocity inversion results by diagonal Hessian (DH), Gauss-Newton (GN), and Limited Memory BFGS (L-BFGS), respectively. Figure 4.5 is a comparison of average vertical absolute errors of gradient descent, diagonal Hessian, Gauss-Newton, and L-BFGS. Figure 4.6 also shows a comparison of average vertical absolute errors but in this case only for L-BFGS by varying the number of update parameter Nm = 5,15,30. Refer Appendix D to see how the errors are calculated.

The behavior of the cost function as a function of normalized computational CPU time for the four inversion methods are also compared. Figure 4.7 (**a**) is the normalized CPU time against iteration number (maximum iteration 82). Figure 4.7 (**b**) shows the normalized cost function against iteration number. Figure 4.7 (**c**) also shows the normalized cost function against normalized CPU time. Similarly, the effect of the number of updates Nm = 5,15,30 in L-BFGS on the behavior of normalized cost function against normalized computational time is analyzed. Figure 4.8 (**a**) is a plot of normalized CPU time versus iteration number. Figure 4.8 (**b**) and Figure 4.8 (**c**) show the normalized cost function against iteration number and normalized CPU time, respectively. The y-axis is plotted in log-scale.

The second example for 'Mac' model is by adding noise (SNR = 30) with Total variation (TV) regularization using the L-BFGS method. The noise was added using the Seismic Unix package (Stockwell, 2009), *suaddnoise*, that uses the following formulas, $\mathbf{d_n} = \mathbf{d} + scale \times \mathbf{n}$,

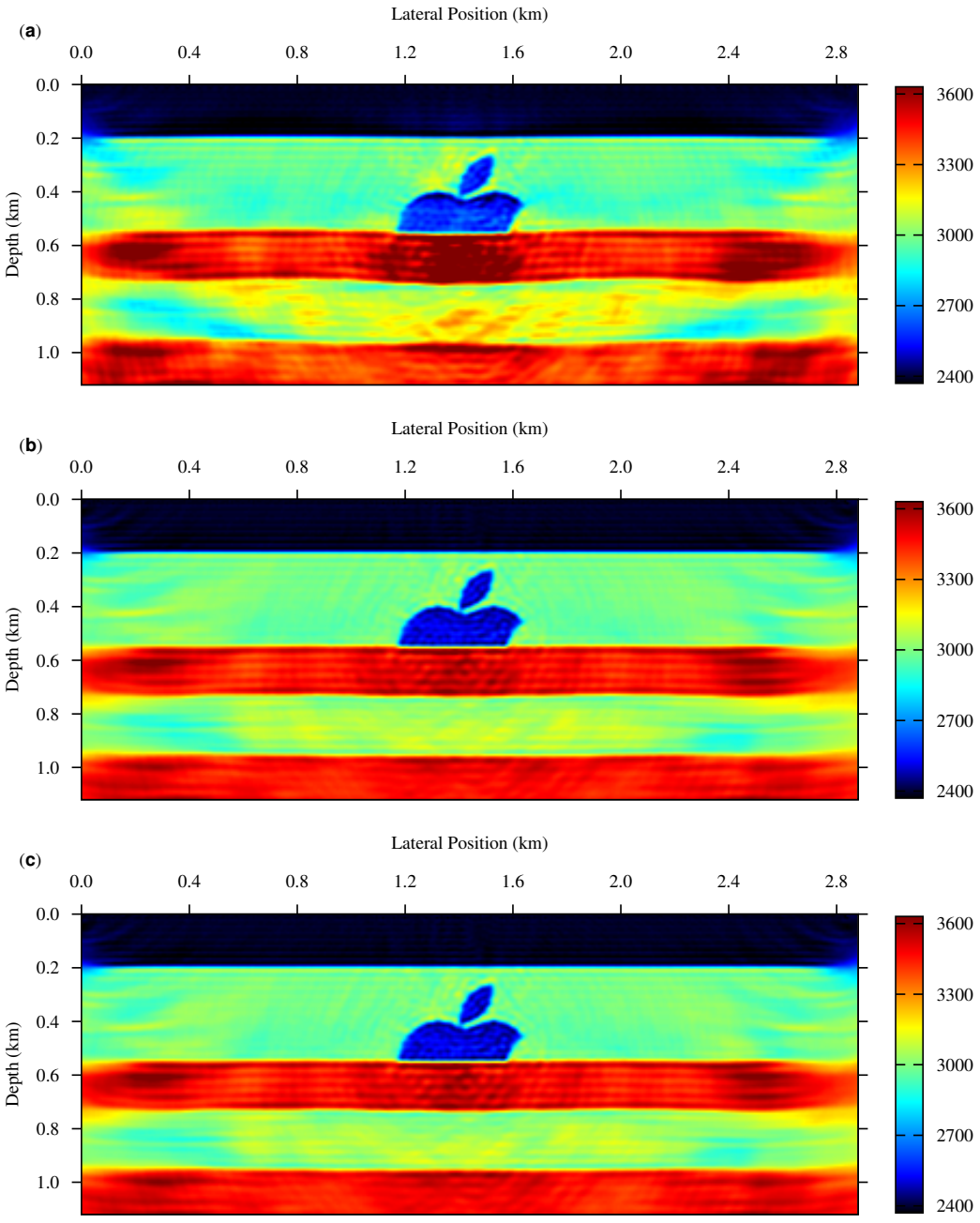


Figure 4.4: Inversion results by (**a**) diagonal hessian, **b**) Gauss-Newton and **c**) L-BFGS methods

$scale = \frac{1}{SNR}\frac{d_{max}}{\sqrt{2}}\frac{1}{n_{rms}}$, and $n_{rms} = (\frac{1}{N}\sum_j n_j^2)^{1/2}$, where $\mathbf{d}$ is the original data, n is the Gaussian noise, d_{max} is the maximum value of $\mathbf{d}$, and N is the number of elements, and $\mathbf{d}_n$ is the data with noise. In this scenario, the performances of the regularization for different

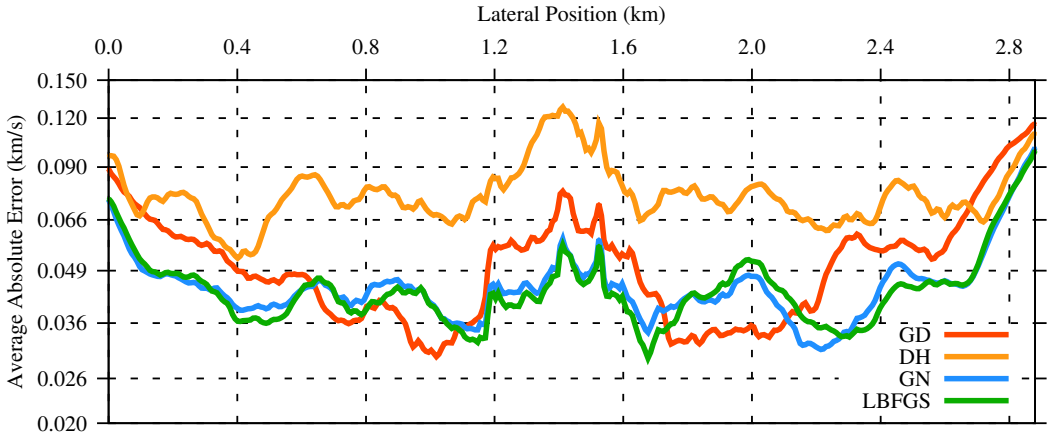


Figure 4.5: A comparison of vertical profile average absolute error of the four inversion methods: GD (Gradient Descent), DH (Diagonal Hessian, GN (Gauss-Newton) and LBFGS (L-BFGS).

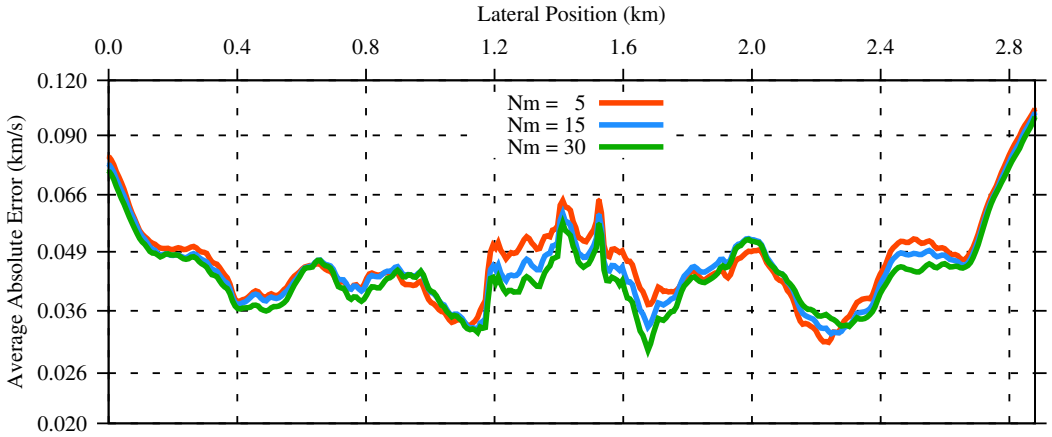


Figure 4.6: A comparison of vertical profile average absolute error using L-BFGS algorithm for three different values of Nm (maximum number of history of gradients): Nm = 5,15,30.

selected μ values are compared. Figure 4.9 (**a**), Figure 4.9 (**b**) and Figure 4.9 (**c**) are velocity inversion results using TV regularization with $\mu = 0.25$, $\mu = 1.0$, and $\mu = 3.0$. The regularization parameters are selected from various trail values. In a smaller sized inverse problems, best parameters can be selected using the formal Chi-squared test. It is also important to note in our examples that the regularization parameters refers to the ratio of regularization function (J^{TV}) to the main cost function (J^{d}). Although these parameters are constant throughout the inversion, they are dynamic at each iteration and for a different frequency, internally. Figure 4.10 (**a**) is the comparison of average vertical absolute error for selected values of regularization parameters (μ = 0.25,0.5,1.0,3.0).

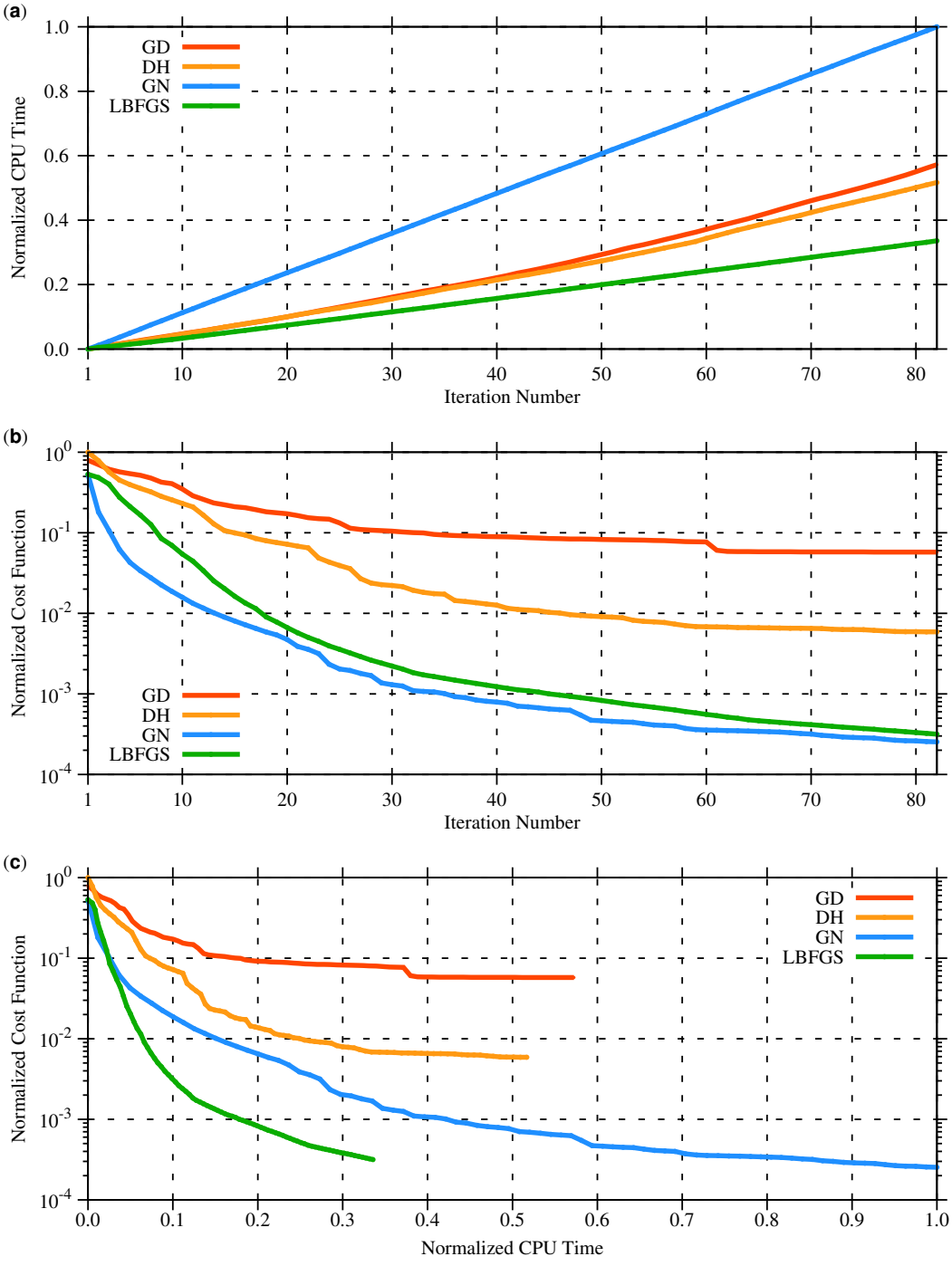


Figure 4.7: These figures depict the behavior of the cost function against iteration number and computational time for the four inversion methods: (**a**) normalized CPU time against iteration number, (**b**) normalized cost function (in log scale) against iteration number and (**c**) normalized cost function (in log scale) against normalized CPU time.

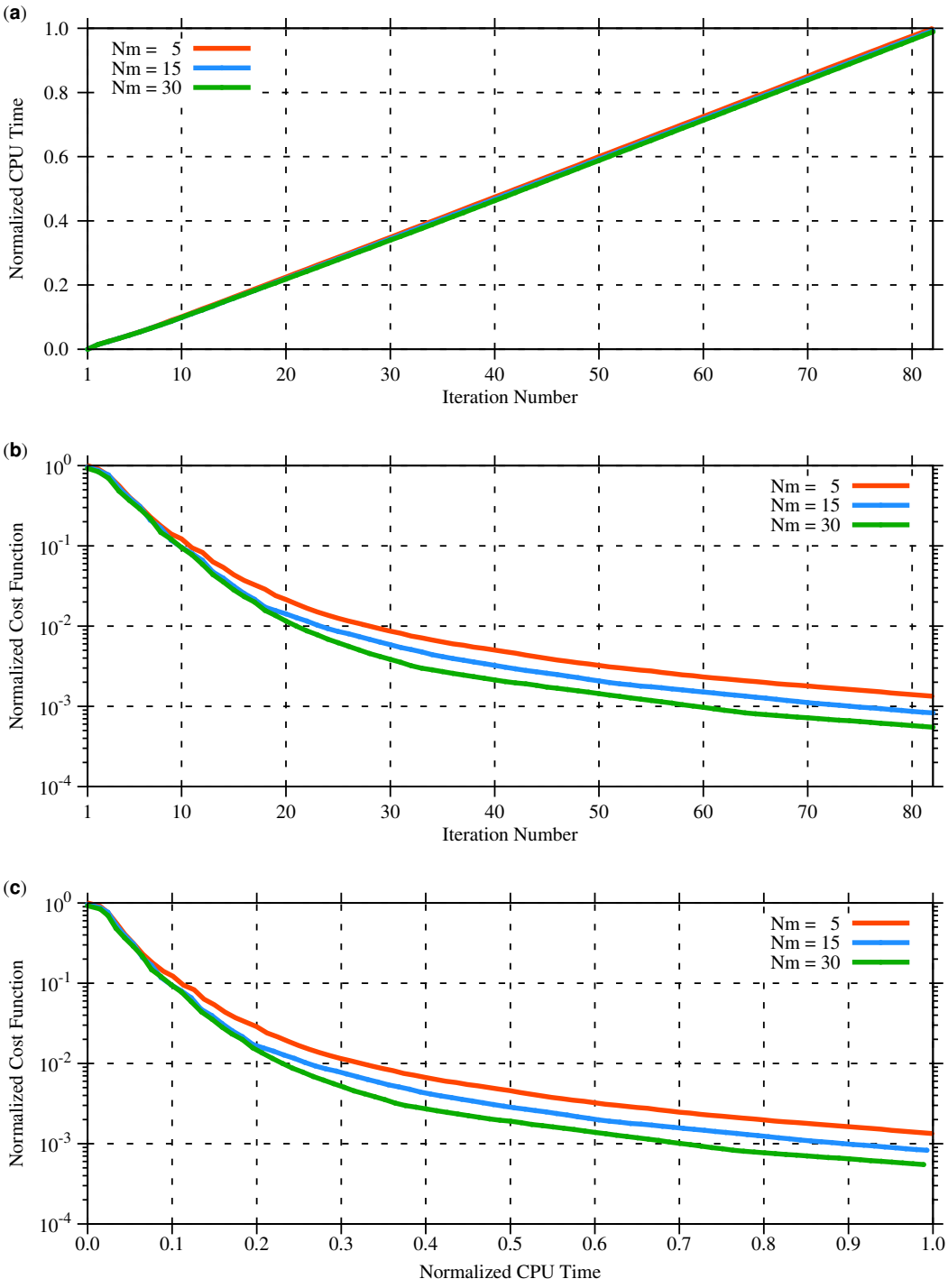


Figure 4.8: These figures depict the behavior of the cost function against iteration number and computational time using the L-BFGS method for Nm = 5,15,30: (**a**) normalized CPU time against iteration number, (**b**) normalized cost function (in log scale) against iteration number and (**c**) normalized cost function (in log scale) against normalized CPU time.

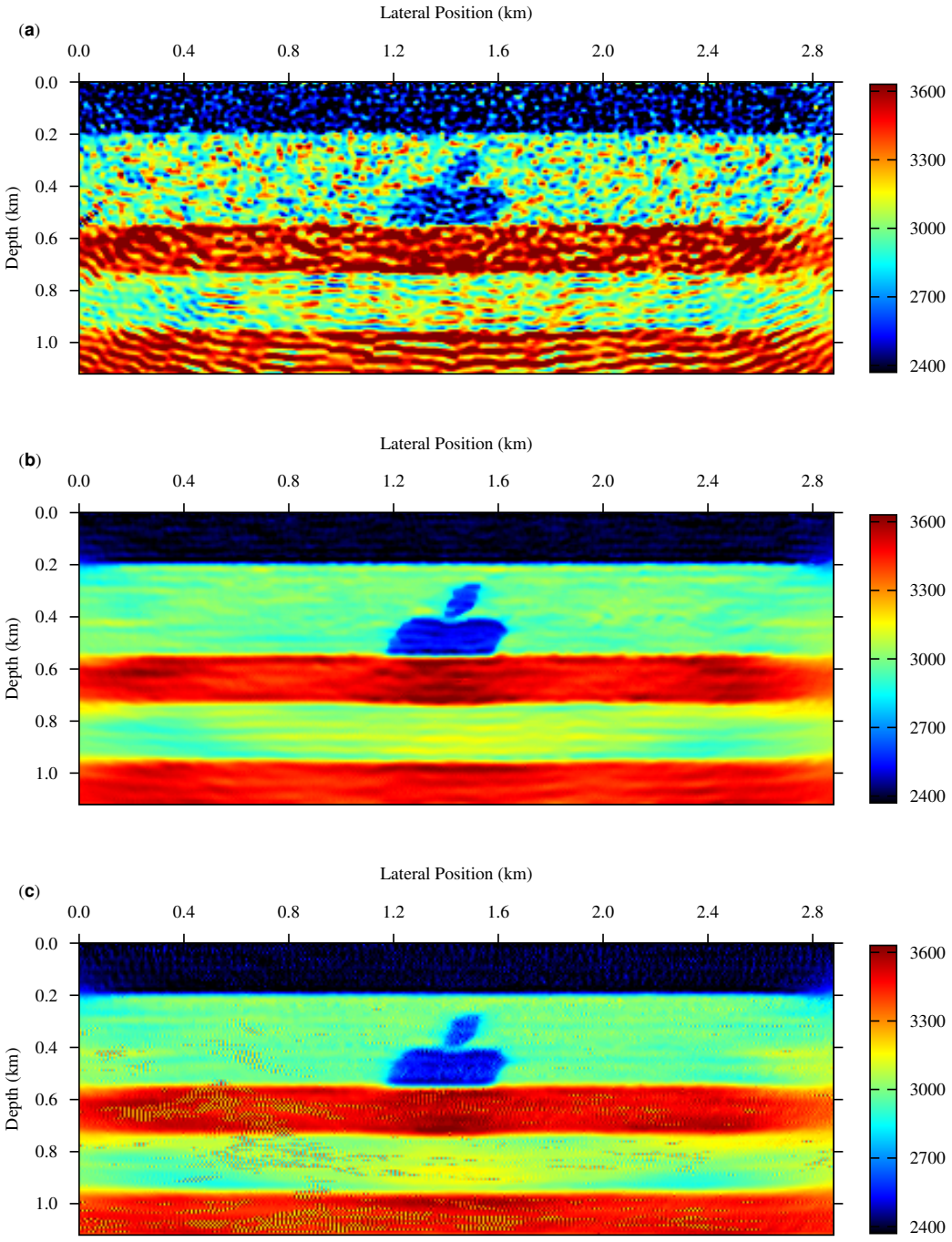


Figure 4.9: Inversion results with noise (SNR = 30) using the L-BFGS method and a TV-regularization with regularization factors: (**a**) $\mu = 0.25$, **b**) $\mu = 1.0$ and **c**) $\mu = 3.0$.

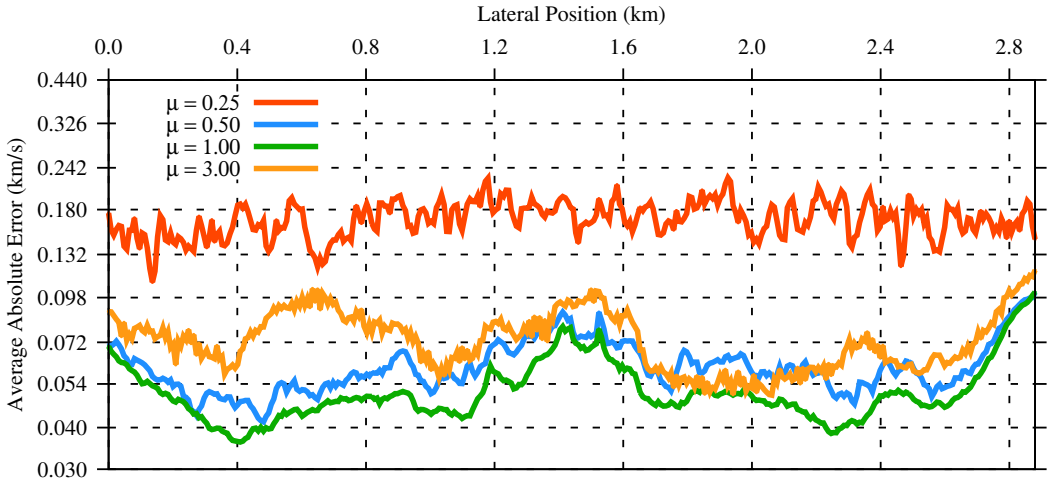


Figure 4.10: A comparison of vertical profile average absolute error with noise (SNR = 30) using the L-BFGS method and a TV-regularization with regularization factor: μ = 0.25,0.5,1.0,3.0.

The normalized cost function and normalized CPU time are also compared for these selected parameters. Figure 4.11 (**a**) is a plot of normalized CPU time versus iteration number. Figure 4.11 (**b**) and Figure 4.11 (**c**) are normalized cost function against iteration number and normalized cost functions, respectively. Each curve was normalized by the largest values of the cost function among the four methods.

The third example for 'Mac' model is data with noise (SNR = 30) with 2D smoothing (SM) regularization using the L-BFGS method. In this scenario, the performance of the regularization for different values of selected regularization parameters are compared. Figure 4.12 (**a**), Figure 4.12 (**b**) and Figure 4.12 (**c**) are velocity inversion results using smoothing regularization with $\mu = 0.1$, $\mu = 0.25$, and $\mu = 1.0$. The regularization parameters are selected from various trail values. Figure 4.13 (**a**) is the comparison of average vertical absolute errors for μ = 0.1,0.25,1.0,3.0.

The vertical average absolute error for smoothing (SM) and total variation (TV) regularization are also compared. Figure 4.14 shows vertical average absolute errors : SM $\mu = 0.10$ and TV $\mu = 0.25$. This figure represents a case where the weight to the regularizations are not optimal. Figure 4.15 shows vertical average absolute errors: SM $\mu = 0.25$ and TV $\mu = 1.0$. This figure represents a case where the weight to the regularizations are optimal. With this optimal regularization parameters, Figure 4.16 (**a**) is the normalized CPU time versus iteration number. Figure 4.16 (**b**) and 4.16 (**c**) are normalized cost function against iteration number and normalized CPU time, respectively. Since the regularizations are different each curves are normalized by its own largest value for the sake of comparison.

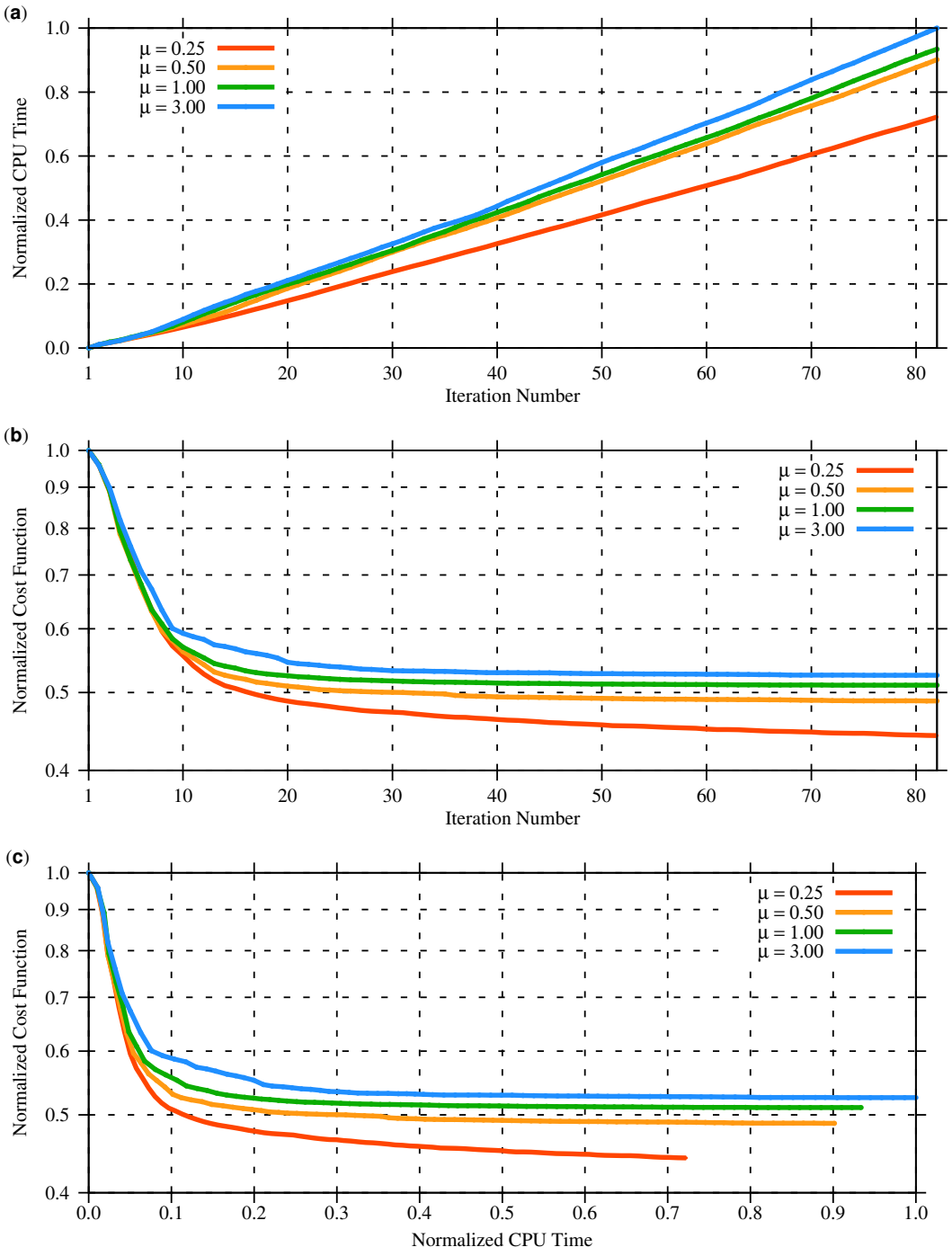


Figure 4.11: These figures depict the behavior of the cost function (data with noise SNR = 30) against iteration number and computational time using the L-BFGS method and TV-regularization for four different regularization factors (μ = 0.25,0.5,1.0,3.0.): (**a**) normalized CPU time against iteration number, (**b**) normalized cost function (in log scale) against iteration number and (**c**) normalized cost function (in log scale) against normalized CPU time.

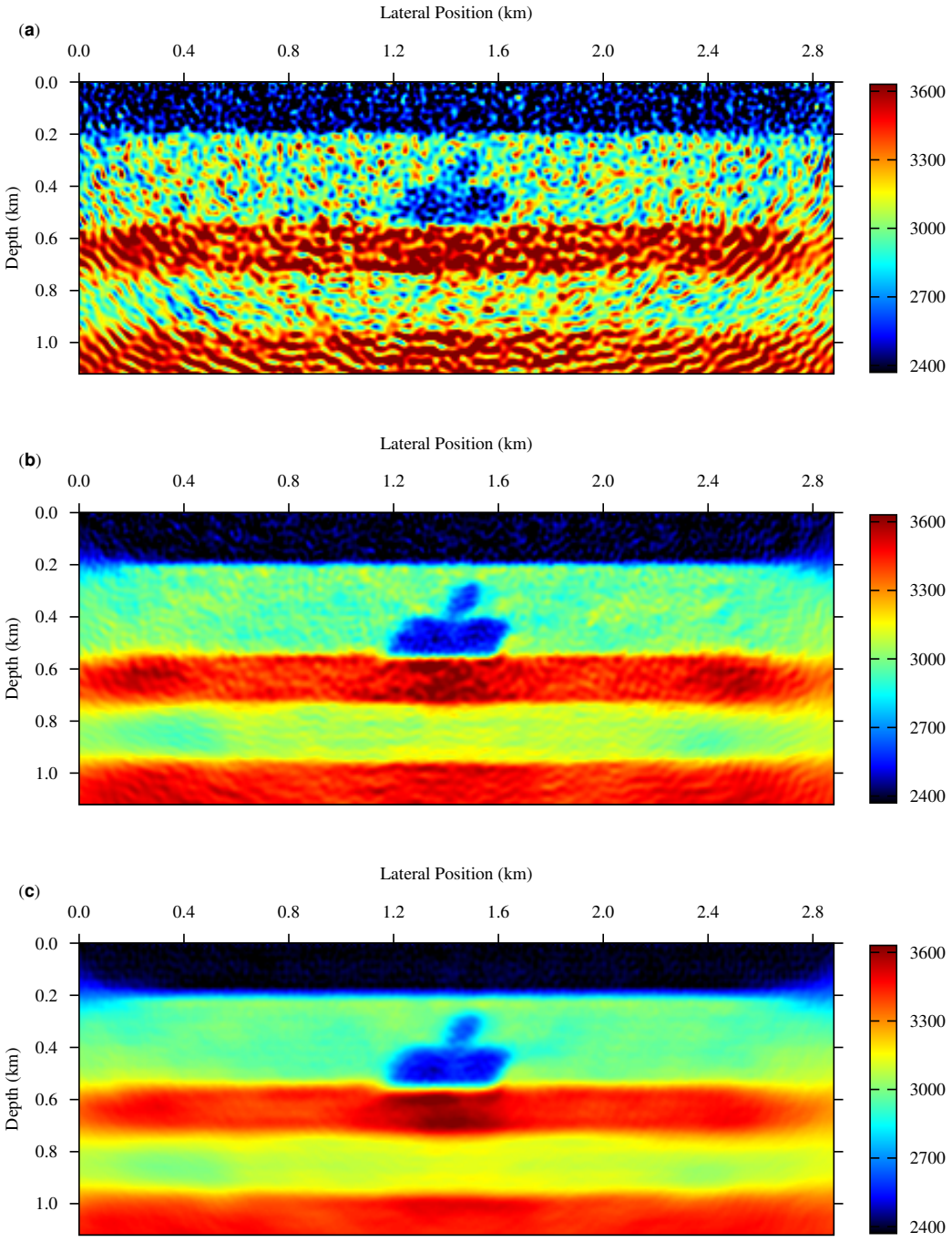


Figure 4.12: Inversion results with noise (SNR = 30) using the L-BFGS method and a 2D Smoothing regularization with regularization factors: (**a**) $\mu = 0.1$, **b**) $\mu = 0.25$ and **c**) $\mu = 1.0$.

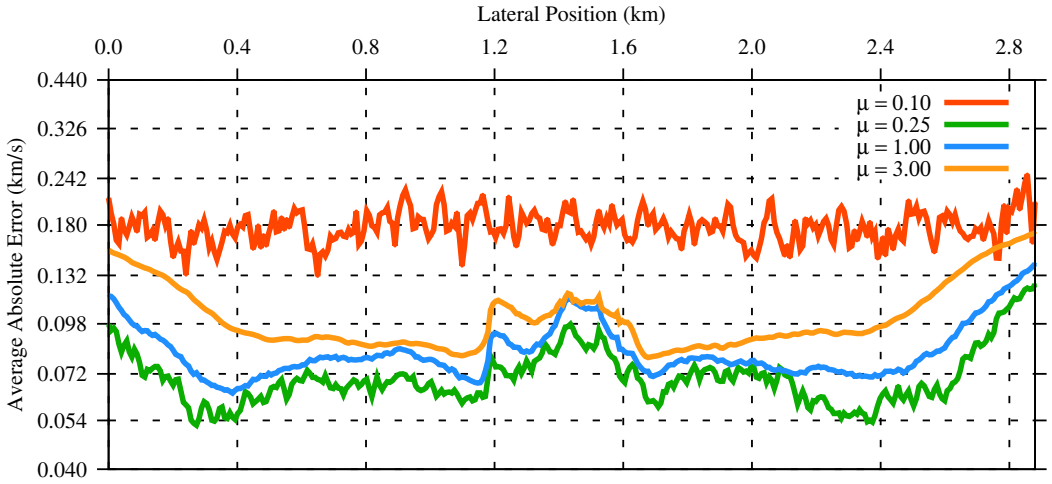


Figure 4.13: A comparison of vertical profile average absolute error data with noise (SNR = 30) using the L-BFGS method and a 2D smoothing regularization with four different regularization factors: $\mu = 0.1,0.25,1.0,3.0$.

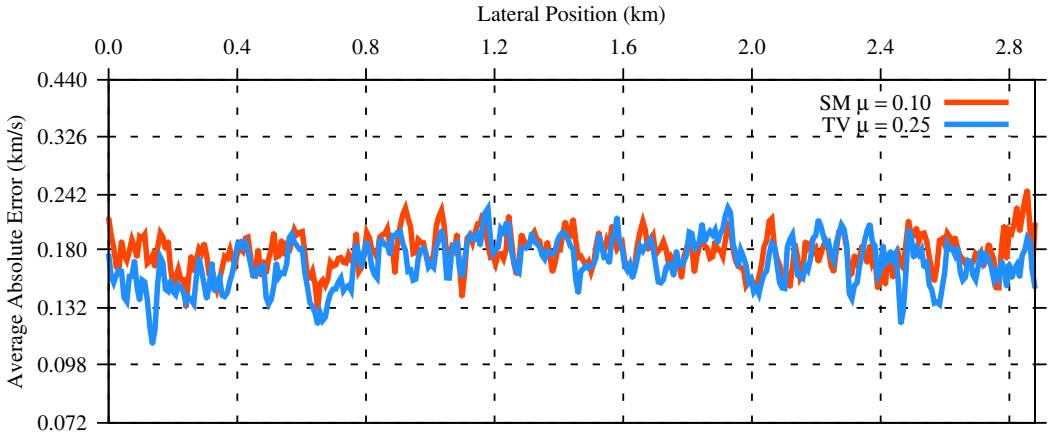


Figure 4.14: A comparison of vertical profile average absolute error data with noise (SNR = 30) of TV and 2D smoothing regularizations with their corresponding regularization factors: SM (smoothing) with $\mu = 0.1$ and TV (Total Variation) with $\mu = 0.25$.

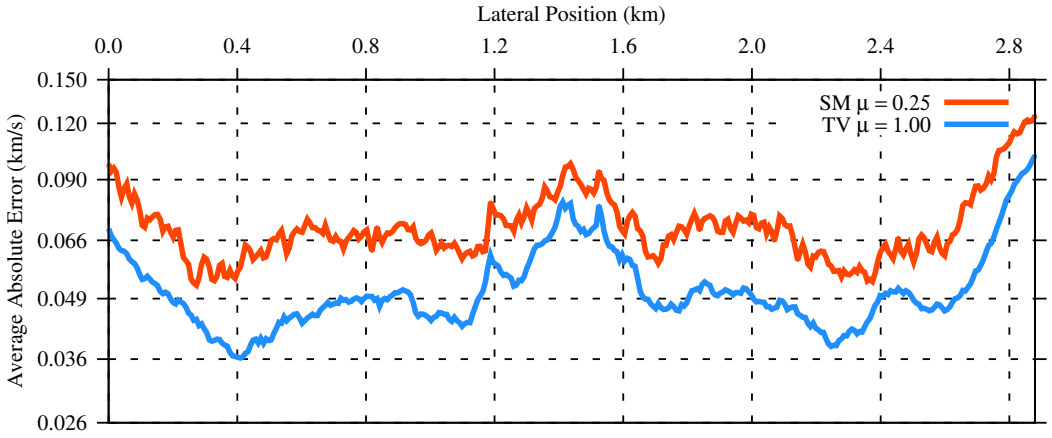


Figure 4.15: A comparison of vertical profile average absolute error data with noise (SNR = 30) of TV and 2D smoothing regularizations with their corresponding optimal regularization factors: SM (Smoothing) with $\mu = 0.25$ and TV (Total Variation) with $\mu = 1.0$.

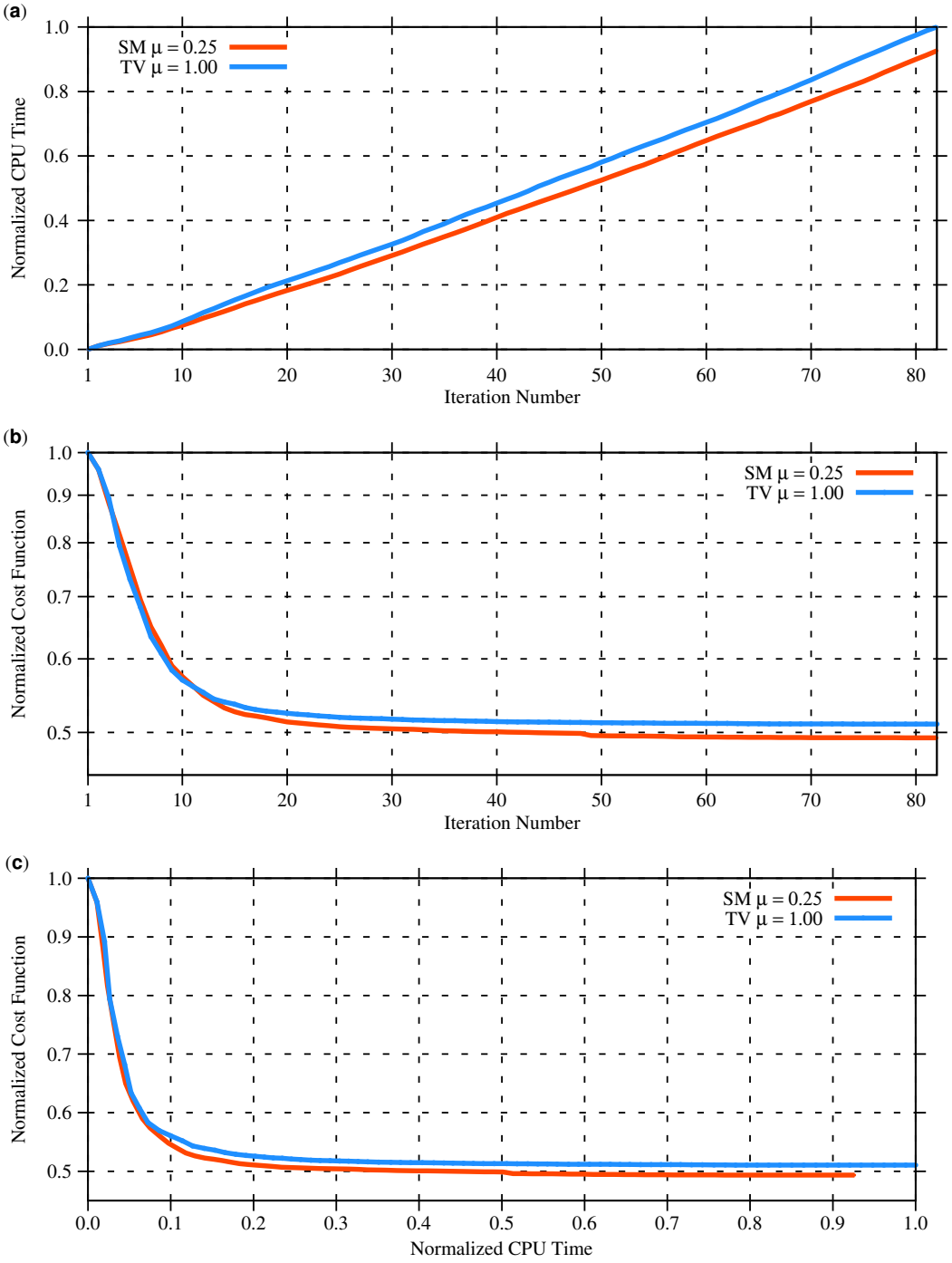


Figure 4.16: These figures depict the comparison of behavior of the cost functions (data with noise SNR = 30) against iteration number and computational time of Smoothing and TV regularizations with their optimal regularization parameters $\mu = 0.25$ and 1.0: (**a**)normalized CPU time against iteration number, (**b**) normalized cost function (in log scale) against iteration number and (**c**) normalized cost function (in log scale) against normalized CPU time.

4.8.2 Amoco Velocity Model

The Amoco model is obtained from the Madagascar repository. It was created in 1997 and presented by John Etgen and Carl Regone at the 1998 SEG convention; it is described as Carpatheans thrusting over the North Sea and was developed to illustrate the limitation of Kirchhoff migration. The original 2D Amoco model from the Madagascar repository is very large. Only the certain portion of the original model with size $N_x \times N_z = 800 \times 321$ with grid spacings $dx = 12.5$ m and $dz = 12.5$ m is used for the examples in this thesis. The true Amoco velocity model is displayed in Figure 4.17 (**a**). Data are generated using the true velocity model and a Ricker wavelet whose central frequency is 10 Hz. A total of 80 shots each separated by 125 m. Each shot has 200 receivers with receiver separation of 50 m.

Figure 4.17 (**b**) is an initial velocity model used for inversion. The first inversion example is without noise and proper regularization. But the solutions are bounded (constrained) between a minimum and maximum to avoid instability in the inversion. Figure 4.17 (**c**) is velocity inversion result by gradient descent (GD). Figure 4.18 (**a**), Figure 4.18 (**b**), and Figure 4.18 (**c**) are velocity inversion results by diagonal Hessian (DH), Gauss-Newton (GN) and Limited Memory BFGS (L-BFGS). Figure 4.19 shows a comparison of average vertical absolute errors of gradient descent, diagonal Hessian, Gauss-Newton and L-BFGS. Figure 4.20 also shows a comparison of average vertical absolute errors but in this case only for L-BFGS by varying the number of update parameter Nm = 3,15,25.

The behavior of the cost function as a function of normalized computational CPU time for the four inversion methods are also compared. Figure 4.21 (**a**) is the normalized CPU time against iteration number (maximum iteration 82). Figure 4.21 (**b**) shows the normalized cost function against iteration number. Figure 4.21 (**c**) shows the normalized cost function against the normalized CPU time. Similarly, the effect of the number of updates Nm = 3,15,25 in L-BFGS on the behavior of normalized cost function against normalized computational time is analyzed. Figure 4.22 (**a**) is a plot of normalized CPU time versus iteration number. Figure 4.22 (**b**) and Figure 4.22 (**b**) show the normalized cost function against iteration number and normalized CPU time, respectively. Note that the y-axis is plotted in log-scale.

The second example for the Amoco model is by adding noise (SNR = 30) with Total variation (TV) regularization using the L-BFGS optimization method. In this scenario, the performance of the regularization for different values of selected regularization parameters are compared. Figure 4.23 (**a**), Figure 4.23 (**b**) and Figure 4.23 (**c**) are velocity inversion results using TV regularization with regularization parameters $\mu = 0.25$, $\mu = 1.0$, and $\mu = 1.5$. The parameters are selected from various trail values. Figure 4.24 (**a**) is the comparison of average vertical absolute error for μ = 0.25,1.0,1.5. The normalized cost function and normalized CPU time are also compared for these selected parameters. Figure 4.25 (**a**) is

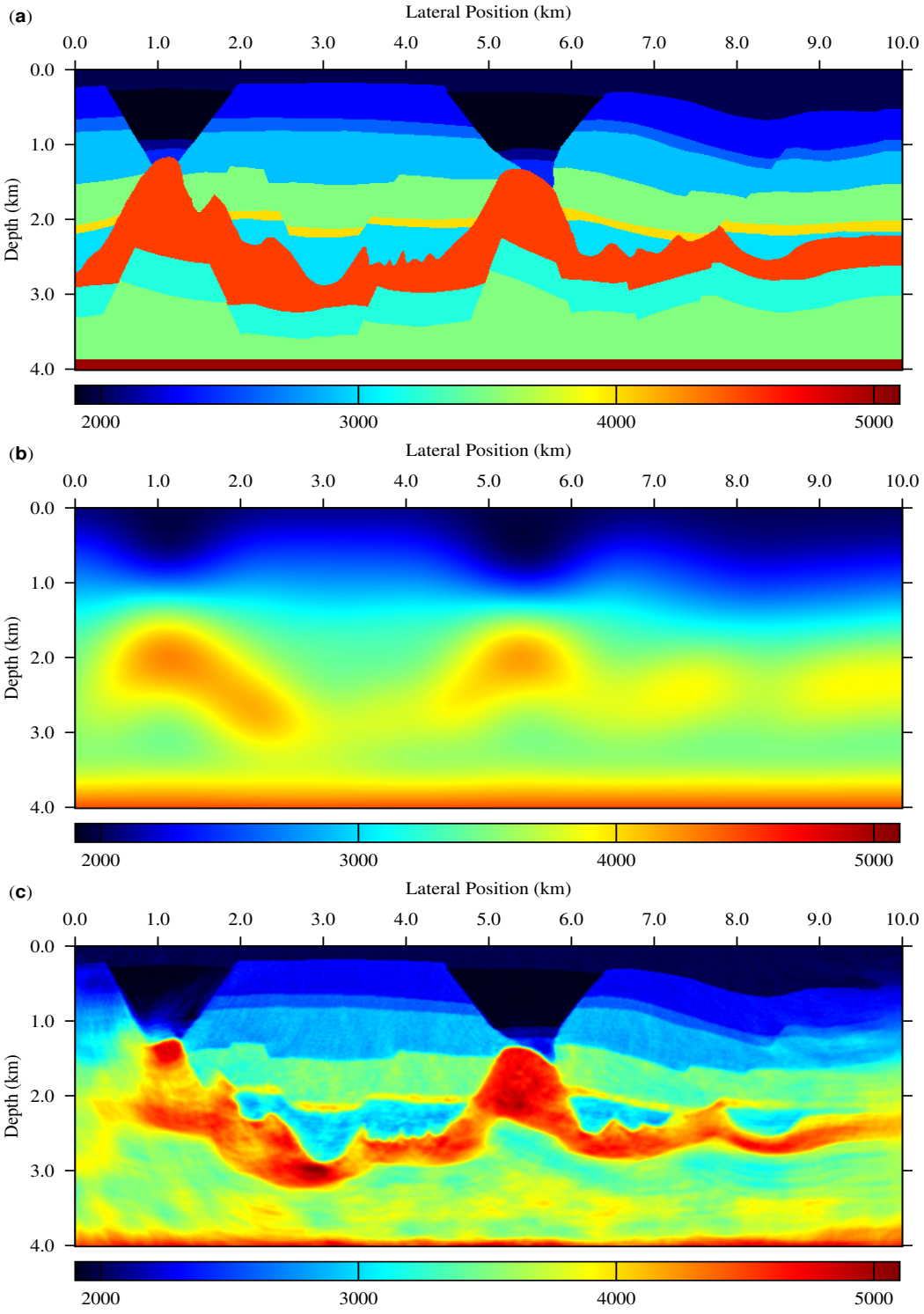


Figure 4.17: (**a**) A true Amoco model, **b**) initial velocity model and **c**) inversion result by gradient descent.

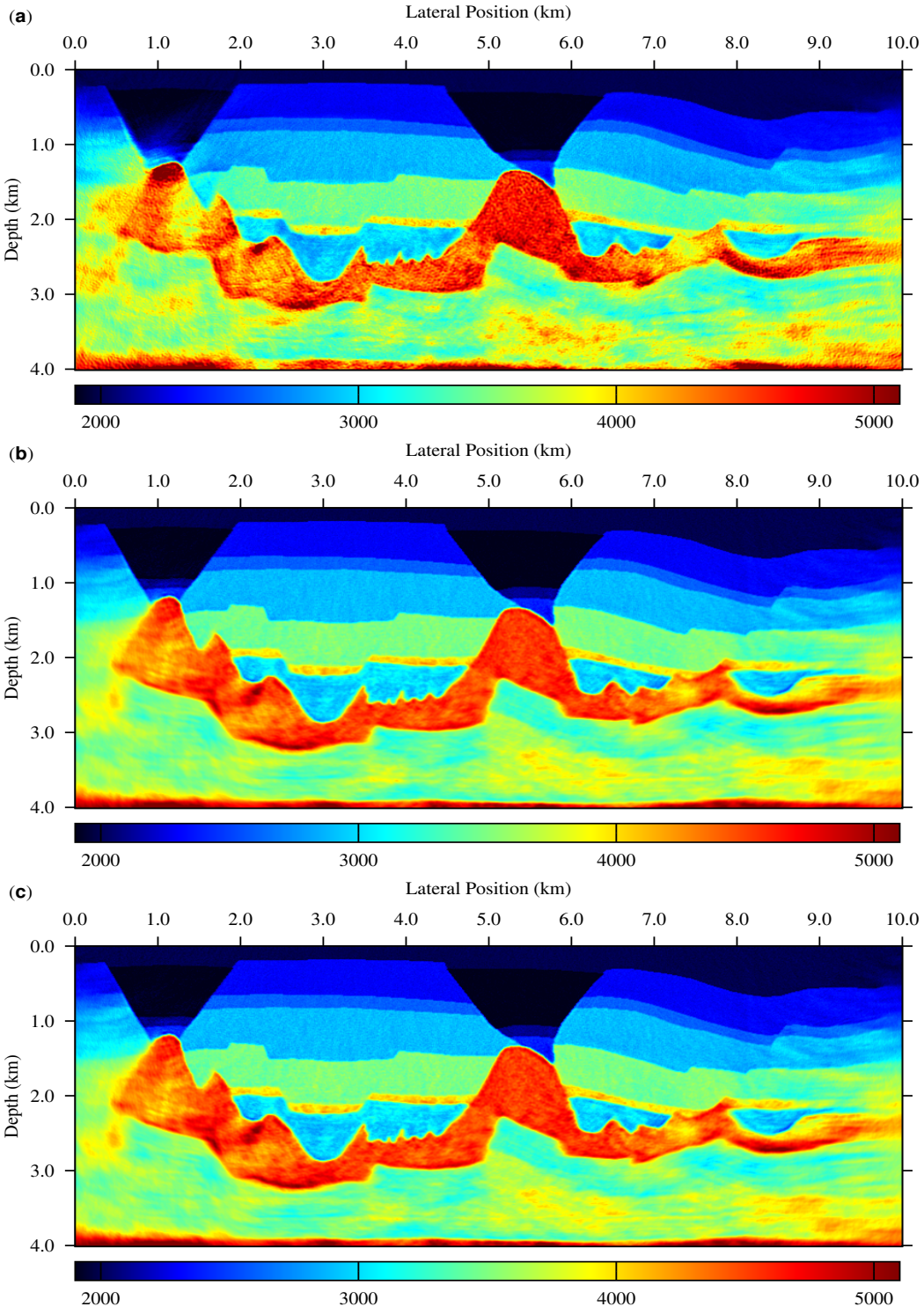


Figure 4.18: Inversion results by (**a**) diagonal hessian, **b**) Gauss-Newton and **c**) L-BFGS methods.

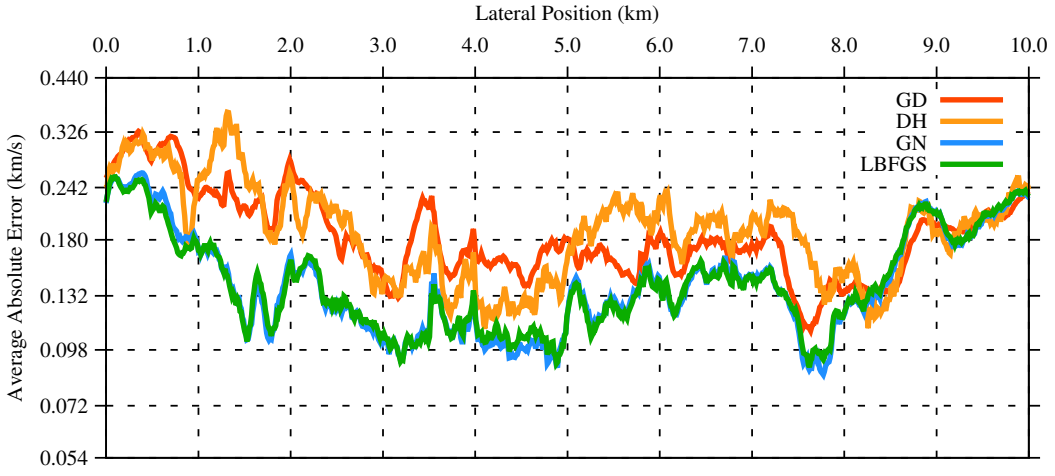


Figure 4.19: Comparison of Amoco model inversion results vertical profile average absolute error of the four inversion methods: GD (Gradient Descent), DH (Diagonal Hessian, GN (Gauss-Newton) and LBFGS (L-BFGS).

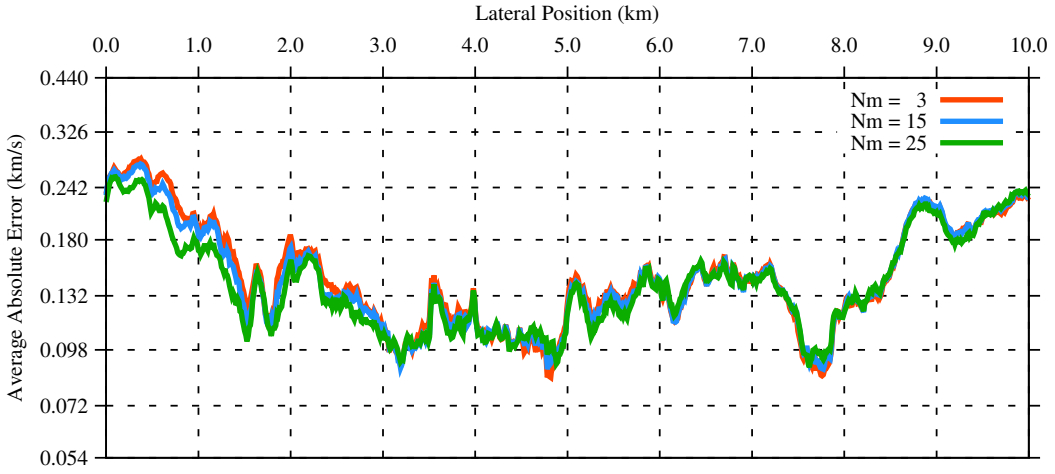


Figure 4.20: Amoco inversion results: a comparison of vertical profile average absolute error using L-BFGS algorithm for three different values of Nm (maximum number of history of gradients): Nm = 3,15,25.

a plot of the normalized CPU time versus iteration number. Figure 4.25 (**b**) and Figure 4.25 (**c**) are the normalized cost function against iteration number and the normalized cost functions, respectively. Each curve is normalized by the largest values of the cost function among the four methods.

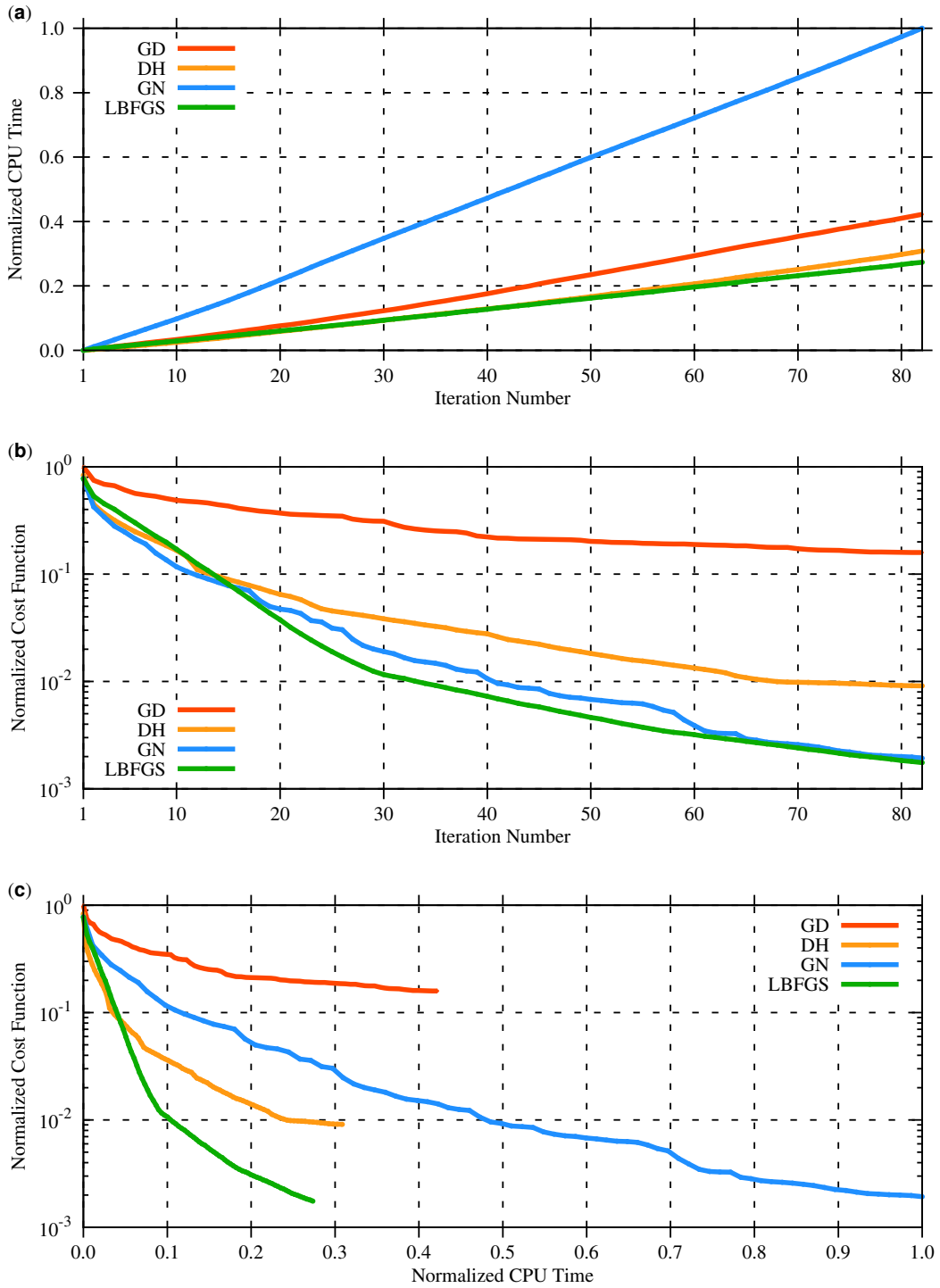


Figure 4.21: Amoco model inversion results: these figures depict the behavior of the cost function against iteration number and computational time for the four inversion methods: (**a**) normalized CPU time against iteration number, (**b**) normalized cost function (in log scale) against iteration number and (**c**) normalized cost function (in log scale) against normalized CPU time.

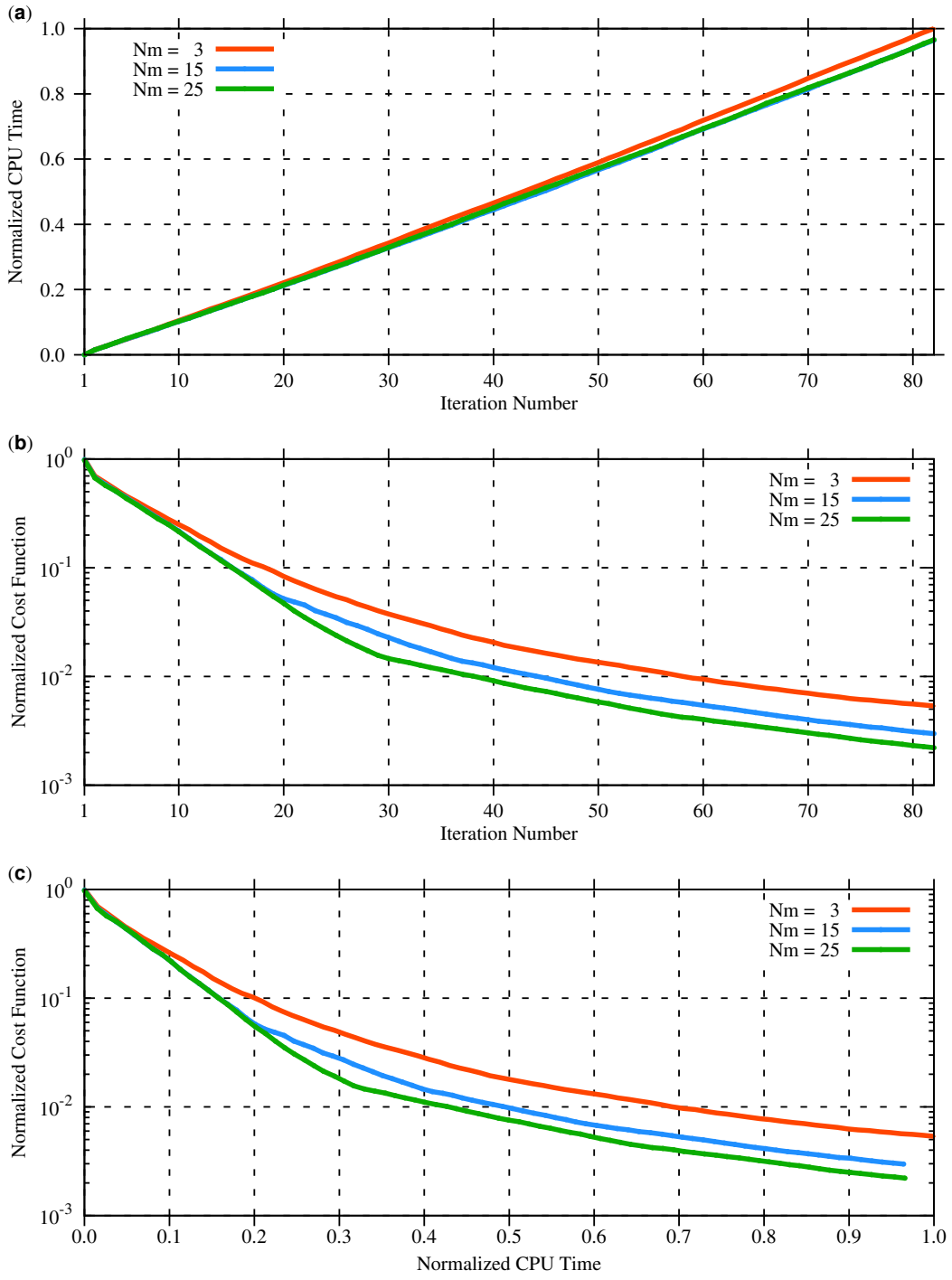


Figure 4.22: Amoco inversion results with L-BFGS algorithm for different values of Nm = 3,15,25: the behavior of the cost function against iteration number and computational time for the four inversion methods: (**a**) normalized CPU time against iteration number, (**b**) normalized cost function (in log scale) against iteration number and (**c**) normalized cost function (in log scale) against normalized CPU time.

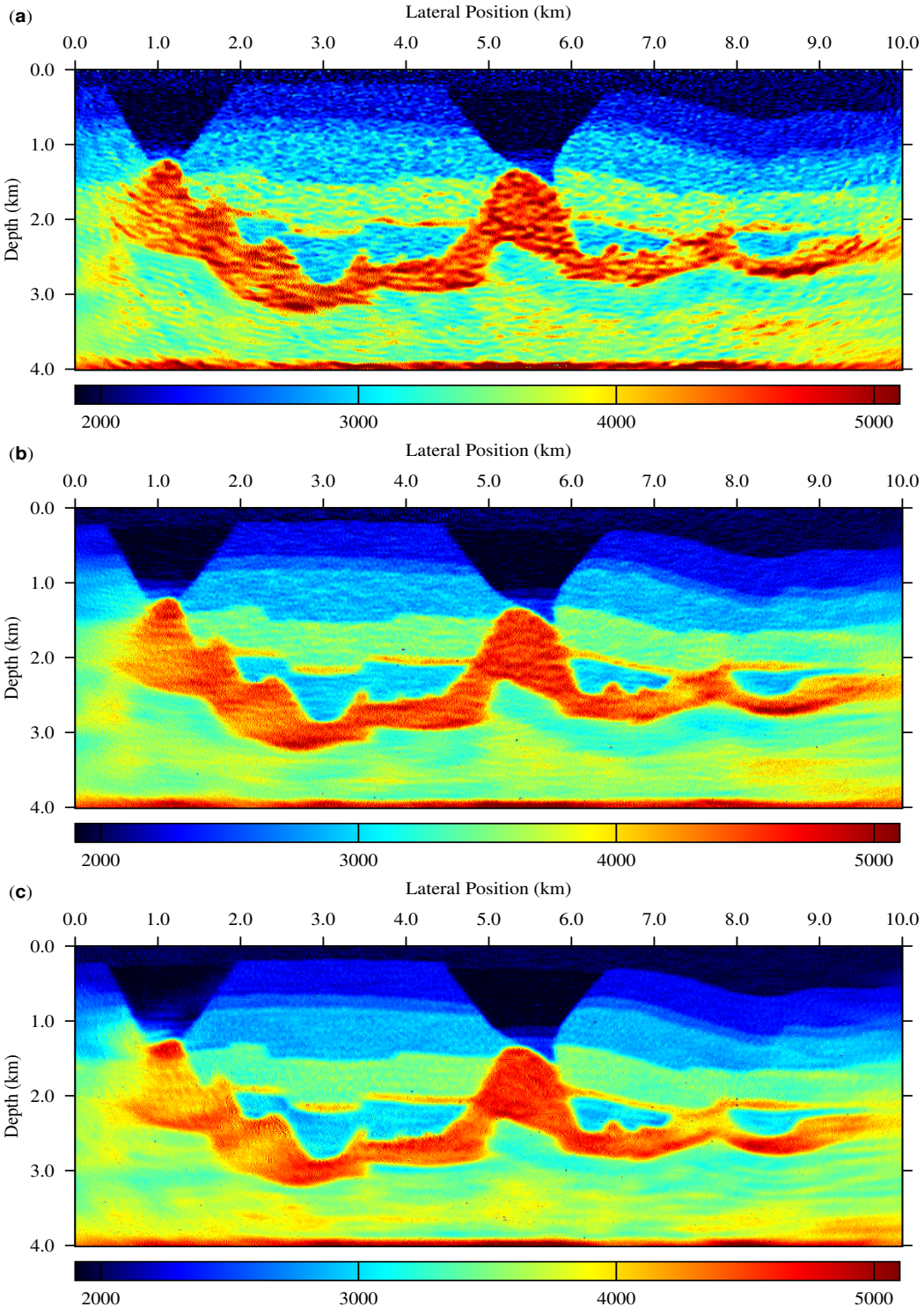


Figure 4.23: Amoco inversion results with noise (SNR = 30) via L-BFGS and TV regularization for different regularization factors: (**a**) $\mu = 0.25$, **b**) $\mu = 1.0$ and **c**) $\mu = 1.5$.

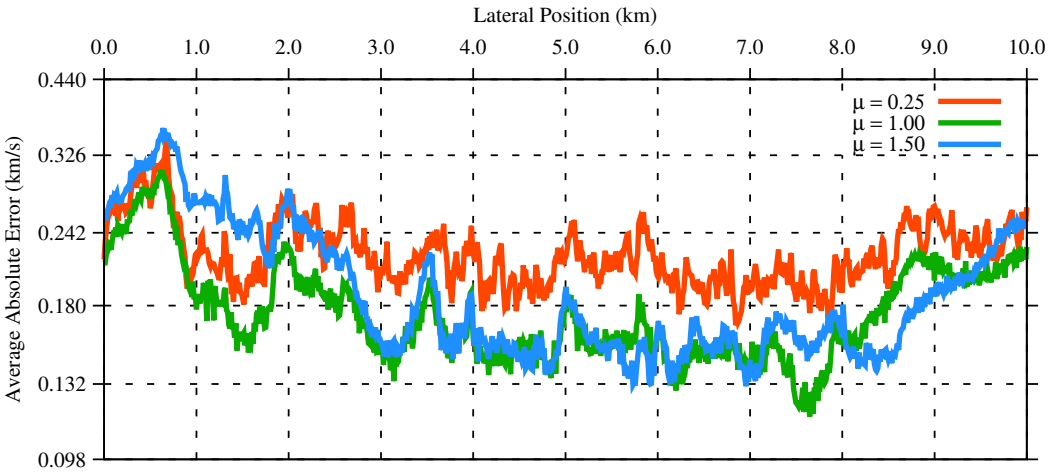


Figure 4.24: Amoco inversion results: comparison of vertical profile average absolute error data with noise (SNR = 30) using the L-BFGS method TV regularization with three different regularization factors: $\mu = 0.25,1.0,1.5$.

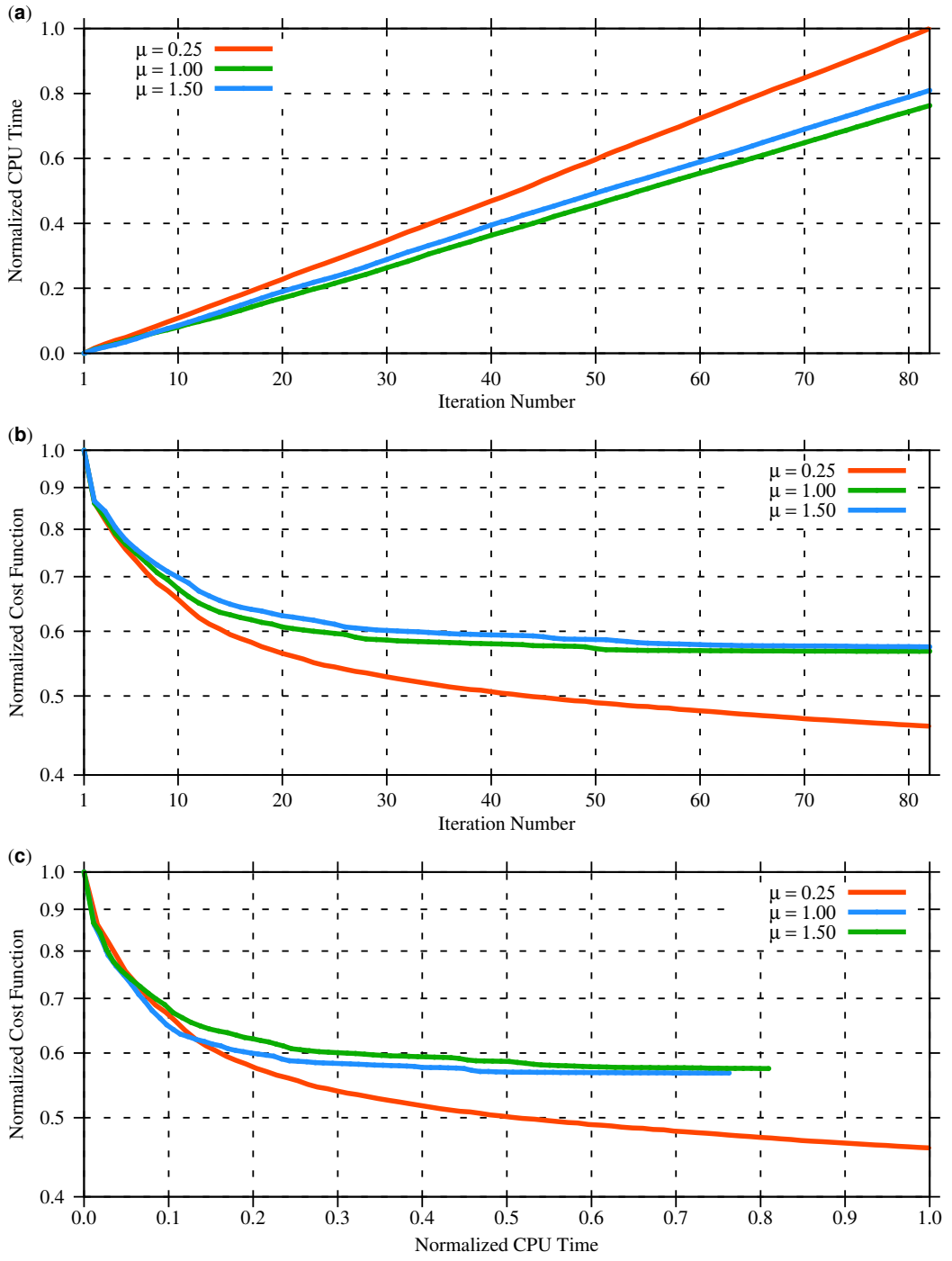


Figure 4.25: Amoco inversion results with noise (SNR = 30) via L-BFGS algorithm and TV regularization for different regularization factors (μ = 0.1,1.0,1.5): the behavior of the cost function against iteration number and computational time for the four inversion methods: (**a**) normalized CPU time against iteration number, (**b**) normalized cost function (in log scale) against iteration number and (**c**) normalized cost function (in log scale) against normalized CPU time.

4.8.3 Marmousi Velocity Model

Marmousi model is obtained from the Madagascar repository (Versteeg, 1994). The original model is modified such that the size is reduced to $N_x \times N_z = 767 \times 251$ with grid spacings $dx = 12.0$ m and $dz = 12.0$ m. The true Marmousi velocity model is displayed in Figure 4.26 (**a**). Data are generated using the true velocity model and a Ricker wavelet whose central frequency is 10 Hz. A total of 98 shots each separated by 96 m. Each shot has 267 receivers with reciever separation of 36 m.

Figure 4.26 (**b**) is an initial velocity model used for inversion. Similar to the Amoco model, the first inversion example is without noise and proper regularization. But the solutions are bounded (constrained) between a minimum and maximum to avoid instability in the inversion. Figure 4.26 (**c**) is velocity inversion result by gradient descent (GD). Figure 4.27 (**a**), Figure 4.27 (**b**), and Figure 4.27 (**c**) are velocity inversion results by diagonal hessian (DH), Gauss-Newton (GN) and Limited Memory BFGS (L-BFGS). Figure 4.28 shows comparison of average vertical absolute errors of gradient descent, diagonal Hessian, Gauss-Newton and L-BFGS. Further, Figure 4.29 shows comparison of average vertical absolute errors for L-BFGS by varying the number of update parameter Nm = 5,15,25.

The behavior of the cost function as a function of normalized computational CPU time for the four inversion methods are also compared. Figure 4.30 (**a**) shows the normalized CPU time against iteration number (maximum iteration 82). Figure 4.30 (**b**) shows the normalized cost function against iteration number. Figure 4.30 (**c**) also shows the normalized cost function against normalized CPU time. Similarly, the effect of the number of updates Nm = 5,15,25 in L-BFGS on the behavior of normalized cost function against normalized computational time is analyzed. Figure 4.31 (**a**) is a plot of normalized CPU time versus iteration number. Figure 4.31 (**b**) and Figure 4.31 (**b**) show the normalized cost function against iteration number and normalized CPU time, respectively. Note again that the y-axis is plotted in log-scale.

The second example for Marmousi model is by adding noise (SNR = 30) with Total variation (TV) regularization using the L-BFGS method. In this scenario, the performance of the regularization for different values of selected regularization parameters are compared. Figure 4.32 (**a**), Figure 4.32 (**b**) and Figure 4.32 (**c**) are velocity inversion results using TV regularization with $\mu = 0.1$, $\mu = 1.0$, and $\mu = 1.5$. The parameters are selected from various trail values. Figure 4.33 (**a**) depicts the comparison of average vertical absolute errors for selected values of regularization parameters (μ = 0.1,1.0,1.5).

The normalized cost function and normalized CPU time are also compared for these selected parameters. Figure 4.34 (**a**) shows a plot of normalized CPU time versus iteration number. Figure 4.34 (**b**) and Figure 4.34 (**c**) are normalized cost function against iteration number

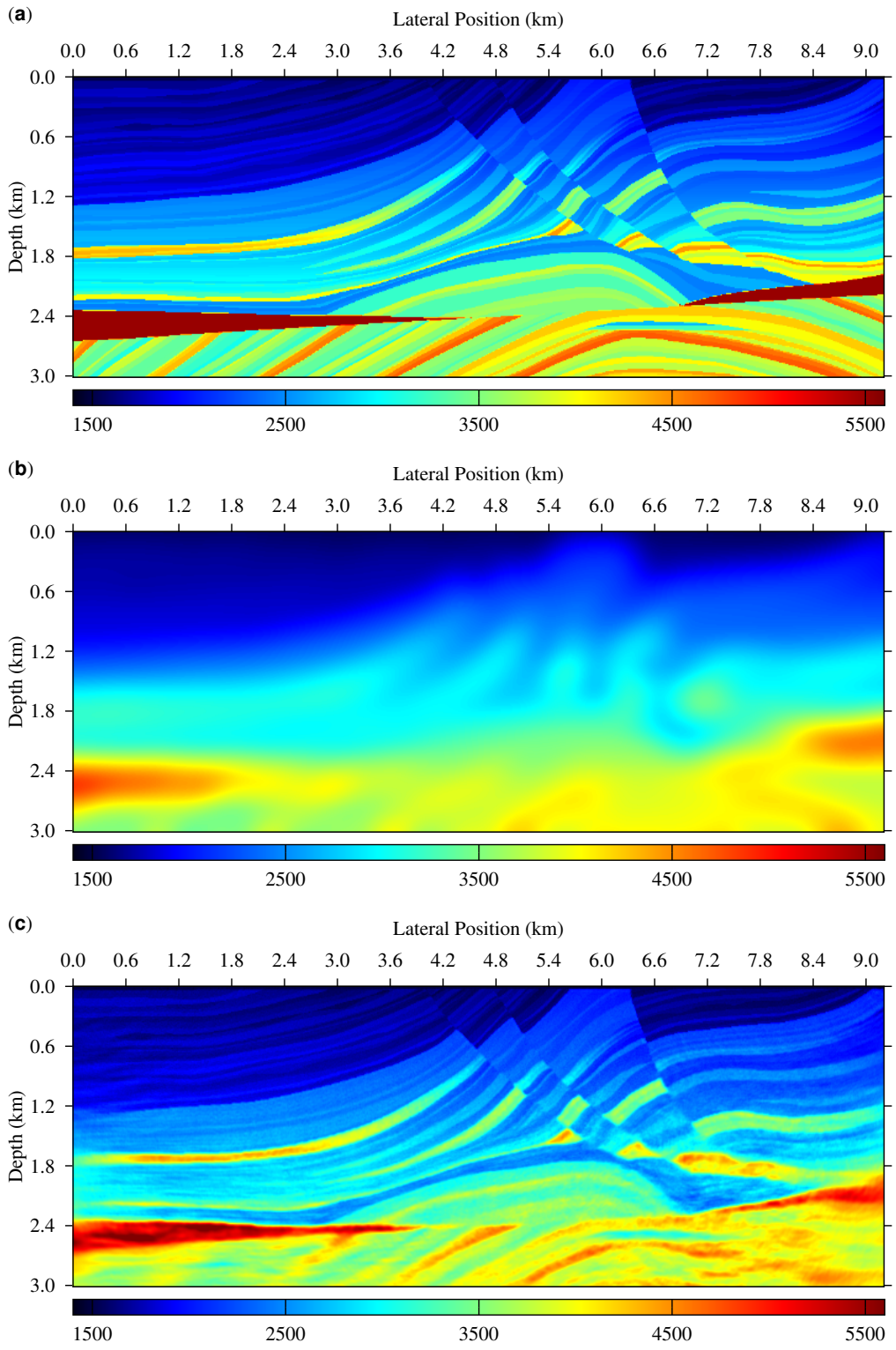


Figure 4.26: (**a**) A true Marmousi velocity model, **b**) initial velocity model and **c**) inversion result by gradient descent.

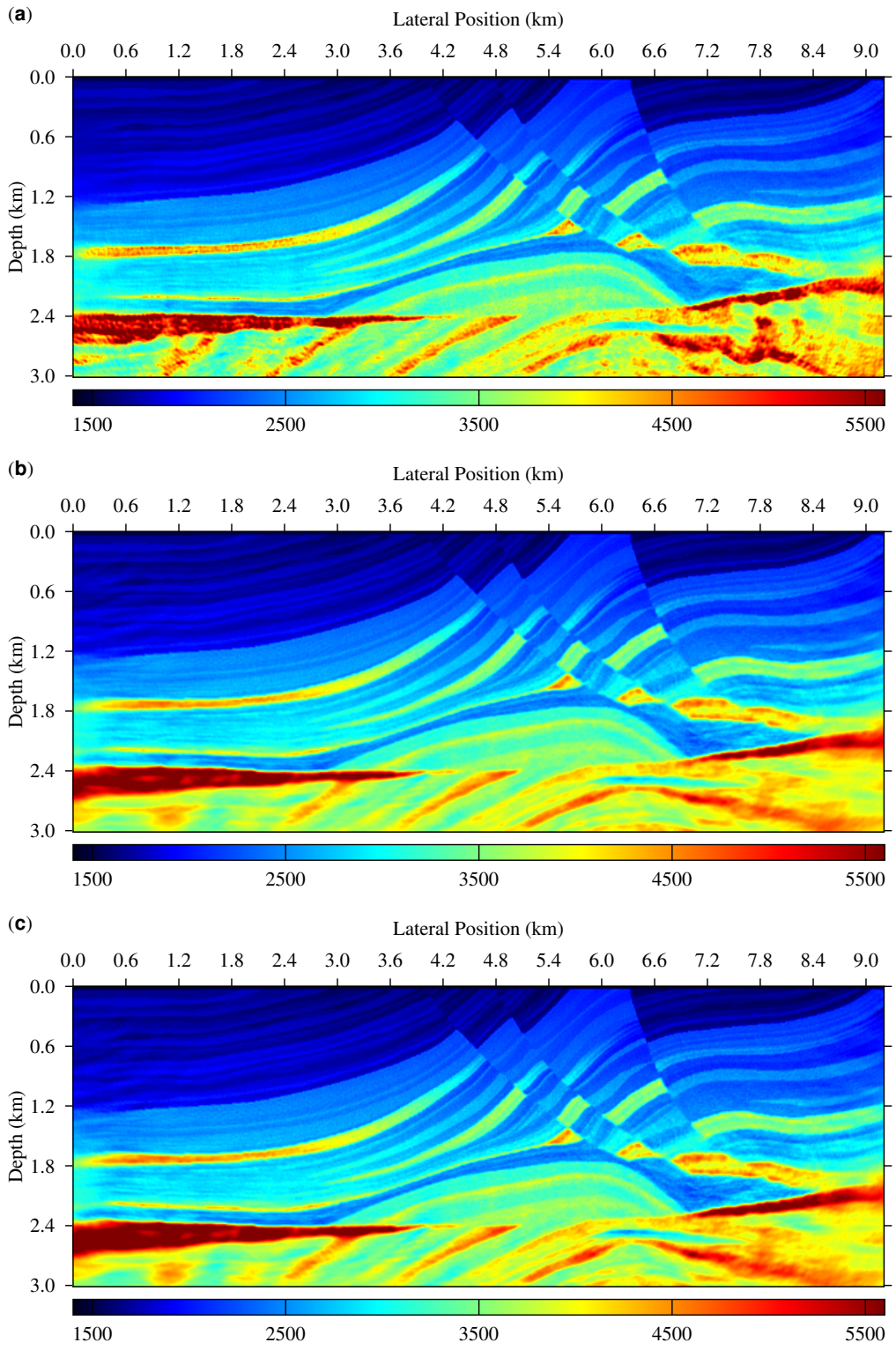


Figure 4.27: Marmousi velocity inversion results: (**a**) diagonal Hessian, **b**) Gauss-Newton and **c**) L-BFGS.

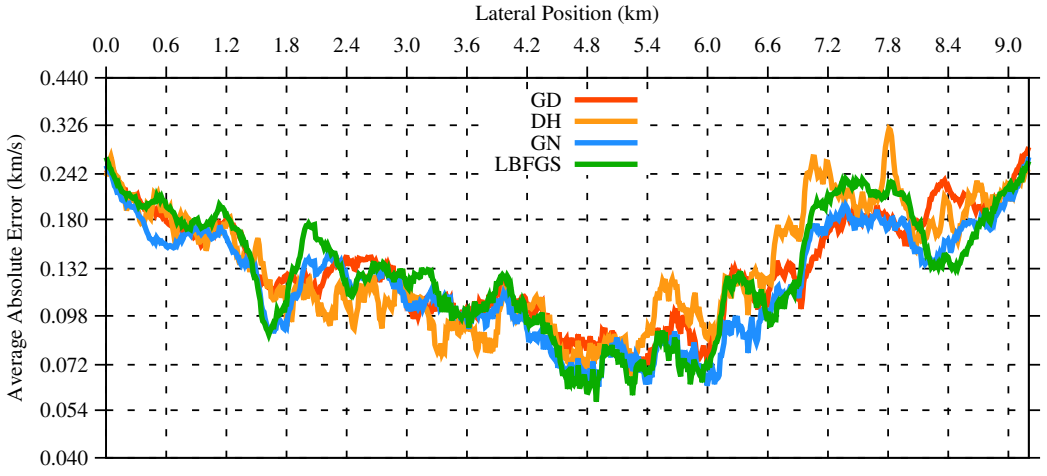


Figure 4.28: Comparison of Marmousi velocity inversion results vertical profile average absolute error of the four inversion methods: GD (Gradient Descent), DH (Diagonal Hessian, GN (Gauss-Newton) and LBFGS (L-BFGS).

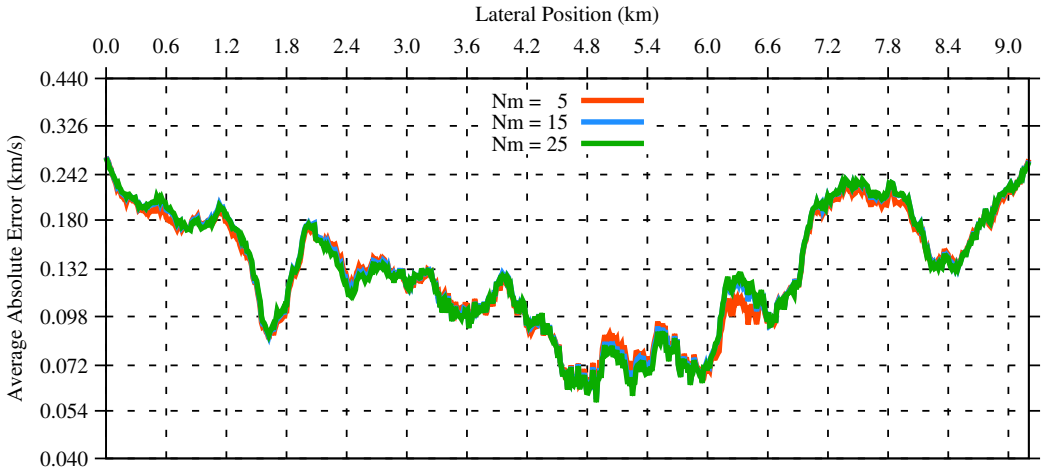


Figure 4.29: Marmousi inversion results: a comparison of vertical profile average absolute error using L-BFGS algorithm for three different values of Nm (maximum number of history of gradients): Nm = 5,15,25.

and normalized cost functions, respectively. Each curve was normalized by the largest values of the cost function among the four methods.

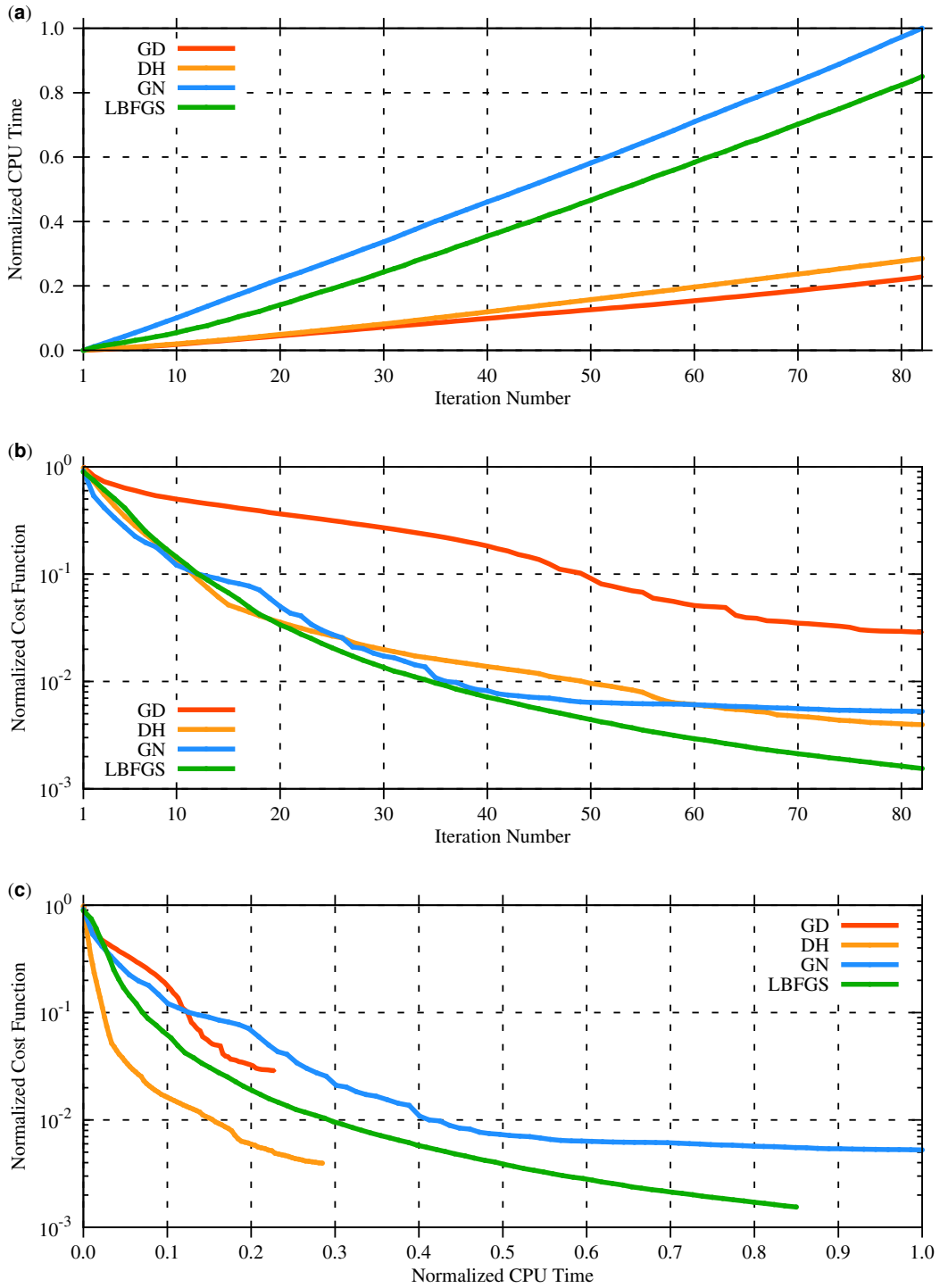


Figure 4.30: Marmousi velocity inversion results: these figures depict the behavior of the cost function against iteration number and computational time for the four inversion methods: (**a**) normalized CPU time against iteration number, (**b**) normalized cost function (in log scale) against iteration number and (**c**) normalized cost function (in log scale) against normalized CPU time.

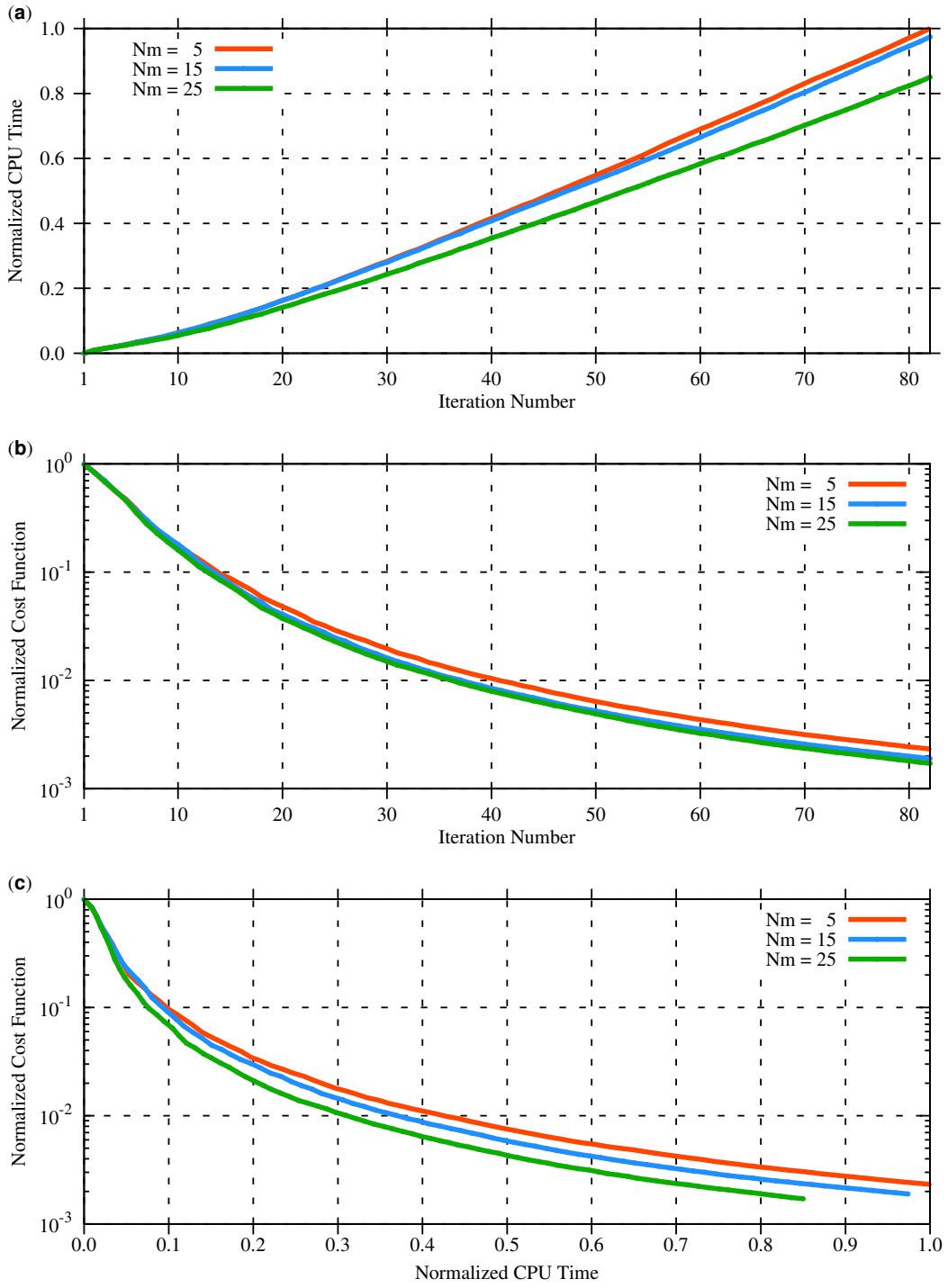


Figure 4.31: Marmousi velocity inversion results with L-BFGS algorithm for different values of $Nm = 5,15,25$: the behavior of the cost function against iteration number and computational time for the four inversion methods: (**a**) normalized CPU time against iteration number, (**b**) normalized cost function (in log scale) against iteration number and (**c**) normalized cost function (in log scale) against normalized CPU time.

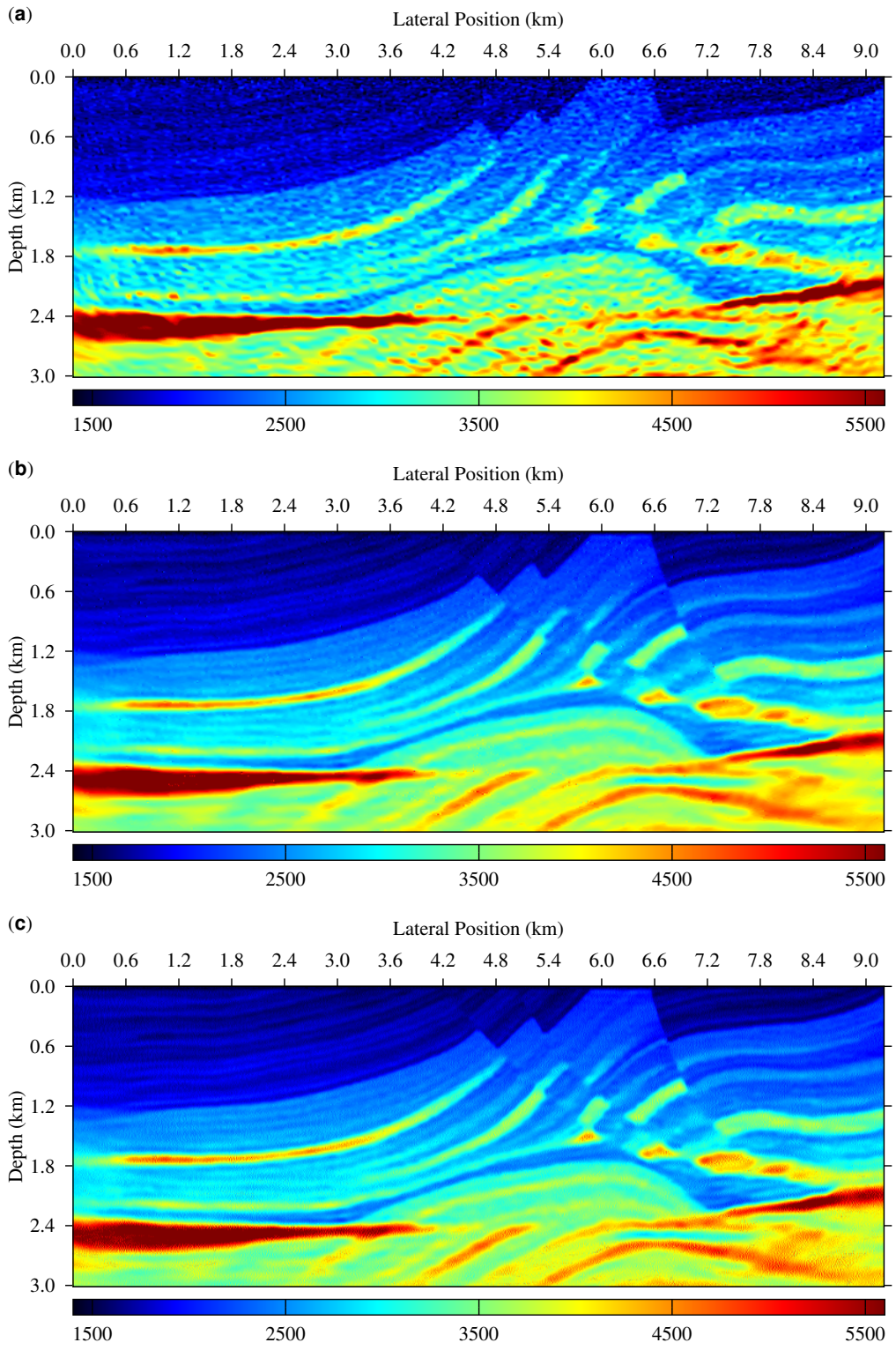


Figure 4.32: Marmousi velocity inversion results with noise (SNR = 30) via L-BFGS and TV regularization: (**a**) $\mu = 0.1$, **b**) $\mu = 1.0$, and **c**) $\mu = 1.5$.

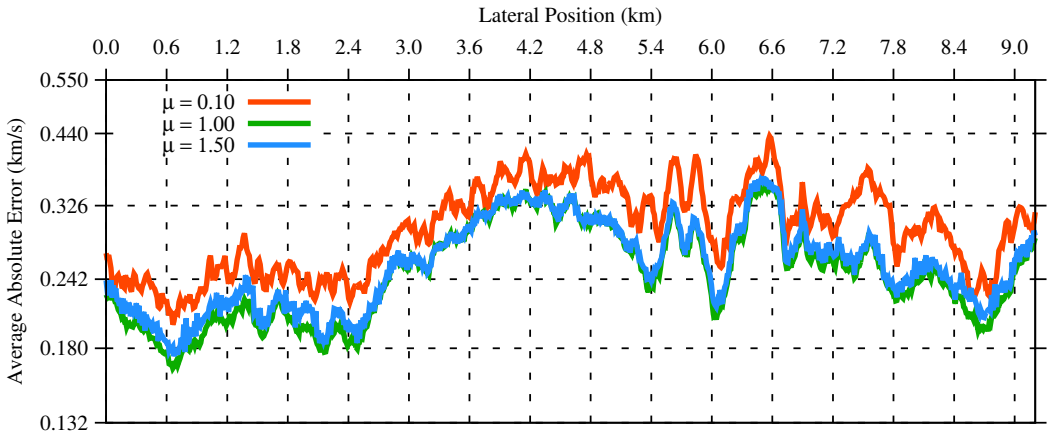


Figure 4.33: Marmousi velocity inversion results: comparison of vertical profile average absolute error data with noise (SNR = 30) using the L-BFGS method and TV regularization with three different regularization factors: μ = 0.1,1.0,1.5.

4.9 Discussion and Summary

Since the diagonal Hessian and Gauss-Newton ways of computing search direction were discussed in Chapter 3. In this chapter, one more additional way of computing the search direction namely the Quasi-Newton method was derived and discussed in detail. Four full waveform inversion optimization methods were discussed and demonstrated with numerical examples. The inversion results with error analysis and computational time efficiencies of each optimization methods were analyzed with the behavior of cost function against iteration number and time.

To demonstrate the performance of the optimization methods, three velocity models with a different degree of complexity were considered. The first model is the 'Mac' model, which is a simple layered model with an anomaly. The second model is the Amoco model, which is more complex than the 'Mac' model with thrust structure. The third model is the Marmousi model which has a higher degree of complexity as compared to the first two models.

4.9.1 Inversion Velocity Results and Error Analysis

In the 'Mac' model, it can be visual be inspected in Figure 4.2 (**c**) and Figure 4.3 (**a**) that the diagonal Hessian approach produces sharper velocity contrasts but high artifacts in high-velocity regions as compared to the gradient descent. The error analysis in Figure 4.5

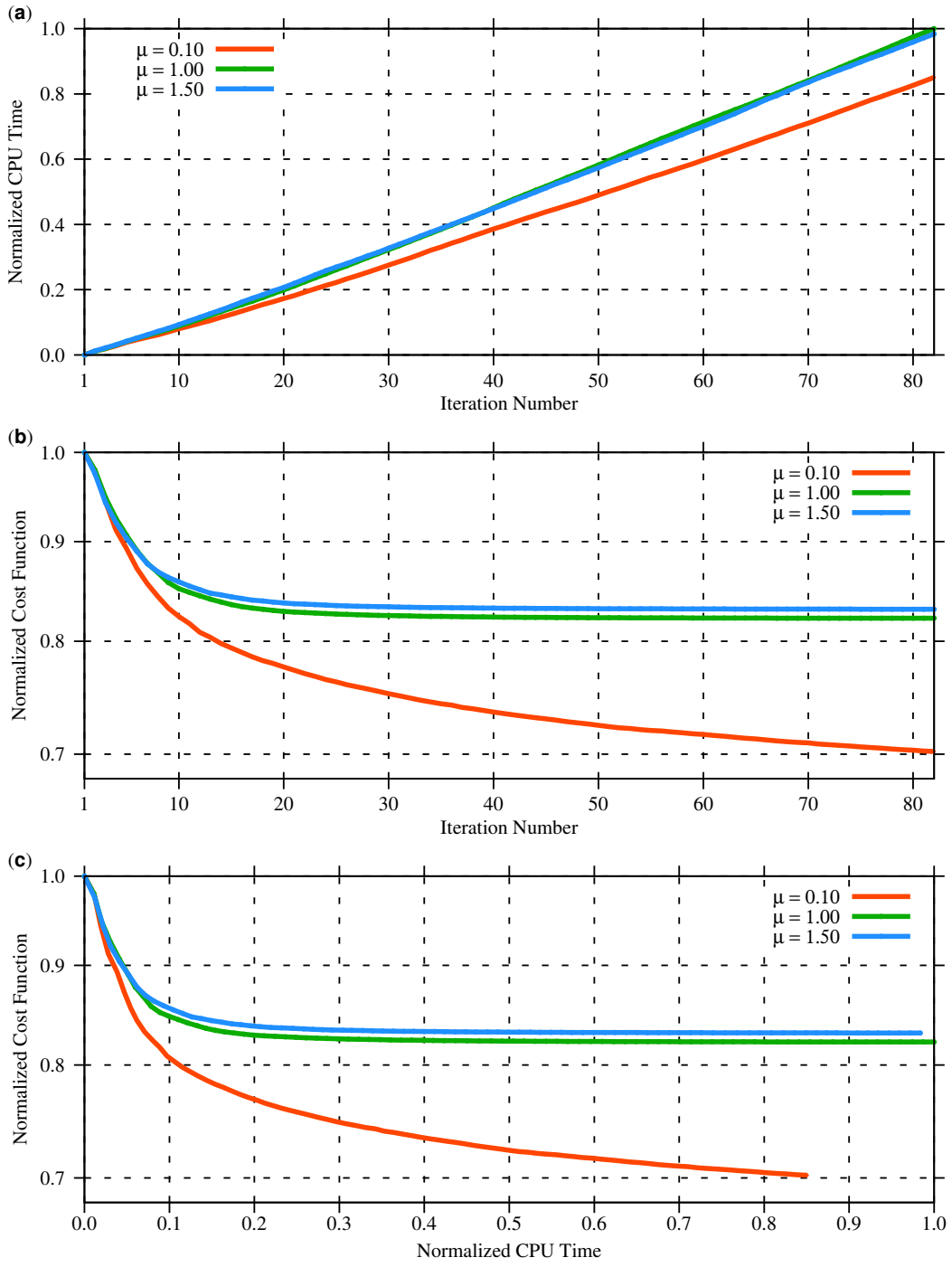


Figure 4.34: Marmousi velocity inversion results with noise (SNR = 30) via L-BFGS algorithm and TV regularization for different regularization factors (μ = 0.1,1.0,1.5): the behavior of the cost function against iteration number and computational time: (**a**) normalized CPU time against iteration number, (**b**) normalized cost function (in log scale) against iteration number and (**c**) normalized cost function (in log scale) against normalized CPU time.

also shows this fact. Although the diagonal Hessian gives a sharper result, the vertical profile error tends to be higher because of the contribution of the significant artifact at higher velocity regions. On the other hand, the Gauss-Newton and the L-BFGS algorithms give better and competitive results as shown in the error analysis in Figure 4.5. Note that the data for the first 'Mac' model example is without noise and the inversion is not properly regularized except that the solution is bounded between a minimum and maximum values. In addition, the L-BFGS algorithm for three different values of Nm = 5,15,30 is compared by calculating the average vertical profile errors as shown in Figure 4.6. It shows the vertical profiles with less error with increasing the number of maximum updates. Given a relatively large difference between the maximum updates used, the improvements are not significant.

The second example using the 'Mac' model is data with noise (SNR = 30). In this example, only the L-BFGS method with smoothing and TV regularization was considered. First, the TV regularization for different values of regularization parameter μ = 0.25,1.0,3.0 were compared as shown in Figure 4.9. It can be easily observed that with $\mu = 0.25$ the noise is dominant. For a large value of $\mu = 3.0$, although the edges are preserved, it produces unwanted regularization artifacts. For an optimal value of $\mu = 1.0$, the noise is removed and no regularization artifacts present. The average vertical profile errors shown in Figure 4.10 are plotted for four regularization parameters $\mu = 0.25, 0.50, 1.0, 3.0$. It shows less error with the optimal regularization parameter $\mu = 1.0$ but large errors for the smaller and larger parameter values. The smoothing regularization with regularization parameters $\mu = 0.1, 0.25, 3.0$ in Figure 4.12 also show similar results. However, with a large value of the regularization parameter, the velocity model becomes smoother, unlike the TV regularization that preserves the edges. The average vertical profile errors shown in Figure 4.13 are plotted for four regularization parameters $\mu = 0.1, 0.25, 1.0, 3.0$. It shows less error with the optimal regularization parameter $\mu = 0.25$ but large errors for the smaller and larger parameter values. Comparisons of average vertical profiles for smaller regularization parameters $\mu = 0.1$ and $\mu = 0.25$ for Smoothing and TV regularization, respectively, are shown in Figure 4.14. Comparisons of average vertical profiles for optimal regularization parameters $\mu = 0.25$ and $\mu = 1.0$ for Smoothing and TV regularization, respectively, are shown in Figure 4.14. Overall, both regularization methods reduce noise in the velocity image. With corresponding optimal values of regularization parameters, TV regularization does perform better than the smoothing regularization in reducing error without compromising edges.

The Amoco model also shows similar outcomes in the comparison of the four optimization methods as described in the 'Mac' model. Figure 4.17 (**c**) and Figure 4.18 (**a**) show that the diagonal Hessian approach produces sharper velocity contrasts but high artifacts in high velocity regions as compared to the gradient descent. The Gauss-Newton and the L-BFGS algorithms give better and competitive results as shown in Figure 4.18 (**b**), Figure 4.18 (**c**) and in the error analysis shown in Figure 4.19. The effect of a maximum number of

updates Nm for three different values is also shown in Figure 4.20. Given large variation among the number of updates, the effect on the velocity error is insignificant except marginal improvement from lateral position 0km to 3.0km.

Inversion of the Amoco model data with noise (SNR = 30) with L-BFGS and TV regularization are also performed. Figure 4.21 (**a**), (**b**) and (**c**) show the velocity inversion results for three selected regularization parameters $\mu = 0.25, 1.0, 1.5$. Overall, the optimal value $\mu = 1.0$ gives less error as compared to the other two regularization parameters as shown in Figure 4.22. For small values of μ, the noise is dominant. For larger values μ, it suppresses the noise but introduces unwanted regularization artifacts.

Similar inversion examples for the Marmousi model also performed. For the 'Mac' model and Amoco model, the diagonal Hessian (DH) shows sharper images comparable to the images obtained via Gauss-Newton (GN) and L-BFGS. But it produces larger artifacts at high-velocity regions as depicted in the error analysis. This characteristic is also observed in the Marmousi model. One thing that stands out in Marmousi model is that the large velocity regions situated at a depth below 2km. One of the main difficulties in seismic imaging is sensing deeper subsurface structures as in the Marmousi model. This is partly a consequence of limitations of the seismic data. On the other hand, the imaging algorithms also play roles to the outcomes of the image. Although the images lack continuity and do enhance large velocities, it is visible that in Figure 4.27 (**a**) the diagonal Hessian approach gives a better image in the high-velocity region when compared to the other three methods. In error analysis given for 'Mac' and Amoco model, the diagonal Hessian gives higher average vertical profile error due to the enhancement in the high-velocity regions. However, the performance of the four optimization methods given in the error analysis in Figure 4.28 are indistinguishable. In short, the diagonal Hessian can image large velocity variations in deeper regions. The effect of a maximum number of updates in the L-BFGS algorithm for three values $Nm = 5, 15, 25$ depicted in Figure 4.29 is indistinguishable although the variations among the number of updates are relatively large.

4.9.2 Overall Convergence and Computational Time

The velocity images that can be obtained via the four optimizations are reliable under the condition that they do honor the data and the prior information which minimizes the defined objective function. Depending on the complexity of the model and data, the performance of each optimization methods differs. Given a several value problem, in some cases, the quality of the velocity image obtained by one can be superior to the other. On the other hand, all the four optimization methods could also give similar velocity images as long as each minimization leads to a global minimum with their corresponding different search directions.

One other important factor that hugely distinguishes one optimization method to the other is computational time.

From Figure 4.6, Figure 4.20, and Figure 4.31 of the 'Mac', Amoco and Marmousi model respectively, distinct performance of the four optimization methods can be clearly seen in terms of computational time. For the three models, the Gauss-Newton method takes extremely long time to reach a reasonable minimum of the objective function as compared to gradient descent, diagonal Hessian, and L-BFGS. This result is not surprising since the Gauss-Newton method uses the conjugate gradient method that needs at least two forward computations at each conjugate gradient iteration in addition to the efficient forward computation required to compute source wave-field and gradient. Note that the conjugate gradient loop is within the main iteration at a given frequency. In addition, finding the optimal step length also puts some more computational time burden.

Unlike the Gauss-Newton method, the gradient descent, diagonal Hessian, and L-BFGS only need to use the efficient computation method to obtain source wave field and gradient as well as finding the optimal step length. Thus, the computational time burden in those three methods is minimal. The primary factor that separates the three methods is determining the optimal step length which needs one or more forward computation at each of the main iteration if the first step length fails to satisfy the Wolfe conditions. Depending on the accuracy of the Hessian scaling, the step length determines the length of the search direction to direct cost function to the global minimum correctly. Thus, finding the optimal step length for the gradient descent and diagonal Hessian which do not have a proper Hessian scaling like that of the Gauss-Newton and L-BFGS methods takes more time. From the normalized cost function versus normalized computational time curves in Figure 4.6, Figure 4.20, and Figure 4.31, one can observe that the L-BFGS algorithm reaches the best minimum with relatively shorter time than the three methods. It is apparently at the cost of memory requirement that depends on the size of the model and number of maximum updates Nm. The effect of the maximum update Nm on the behavior of the cost function as a function of time is also plotted for three different values as shown in Figure 4.8, Figure 4.21, and Figure 4.31. Increasing the number of maximum updates means increasing more information to the Hessian from the gradients of the previous iterations. Thus, a larger value of maximum updates corresponds to relatively minimum cost function at a relatively small computational time as compared to the smaller values. This fact can be observed from those figures although the differences are not significant.

Figure 4.11, Figure 4.25, and Figure 4.34 also show the behavior of cost function against computational time for the 'Mac', Amoco and Marmousi models, respectively, for noisy data using L-BFGS algorithm along with TV regularization. They depict the effect of regularization parameter μ. The velocity image obtained with a small value of μ results in

over-fitting. In other words, the noise signature is visible in the velocity image. In this case, the convergence is faster and gives relatively lower minimum cost function as compared to velocity images obtained by higher μ values. This finding is expected because the objective of adding regularization is to honor both the input data and the prior information. A prior information imposed in this case is minimizing the total variation in the velocity image. Therefore, an optimal value of μ should give the desired result; even though its convergence takes a longer time and the cost function is higher than with smaller μ. All in all, the L-BFGS algorithm, if not best, is a very reasonable alternative optimization method for FWI in comparison to the Gauss-Newton method.

CHAPTER 5

Time-Lapse Full Waveform Inversion

In this chapter, traditional time-lapse FWI strategies are presented with the mathematical tools derived in the previous two chapters: independent inversion and double difference inversion. Furthermore, a new time-lapse inversion strategy, Joint Reparametrized Time-Lapse Full Waveform Inversion[1]*, is proposed. Four time-lapse inversion strategies (independent, double difference, joint, and joint reparameterized are explored with synthetic numerical examples by adopting a modified Marmousi velocity model.*

5.1 Introduction

In the time-lapse seismic method, often referred to as 4D seismic, a series of seismic data sets are acquired at different times to study the temporal variation of a target subsurface reservoir. This technique is being used as a subsurface monitoring tool in the oil and gas industry for fluid driven production enhancement as well as for environmental monitoring purpose in CO_2 sequestration (Lazaratos and Marion, 1997; Burkhart et al., 2000; Benson and Doughty, 2006; Lumley, 2010; Vedanti and Sen, 2009; Lumley, 2001).

Assuming that the geological structures are time-invariant, the expected time-lapse changes are due to time dependent modifications in the reservoir. These changes, for instance,

[1] A paper entitled 'Joint Reparametrized Time-Lapse Full Waveform Inversion' is under review in *Geophysics*.

occur when fluids with different physical properties replace one another. However, the time-lapse signature that can be obtained from time-lapse seismic data does not always reflect lithological changes. Time-lapse data are also liable to contamination by acquisition and processing artifacts. A major contributor to artifacts in the time-lapse signature is the non-repeatability of the seismic experiment (Kragh and Christie, 2002).

A qualitative interpretation that can be made by analyzing amplitude changes and time shifts on post stack time-lapse seismic data (Watts et al., 1996; Burkhart et al., 2000; Vedanti and Sen, 2009; Isaac and Lawton, 2006) are often not sufficient for a complete understanding of detailed reservoir conditions. Thus, to quantitatively characterize the dynamics of a reservoir, one should adopt pre-stack imaging and inversion algorithms that yield high-resolution images of physical parameters (Pratt et al., 1998; Pratt, 1999; Hu et al., 2009; Brossier, 2011; Ma and Hale, 2012; Virieux and Operto, 2009). Traditionally, time-lapse seismic inversion is carried out by processing baseline and monitor data sets independently. Then, the difference between baseline and monitor surveys are computed.

Artifacts that accumulate as a result of acquisition and data processing errors may often have a strength that is comparable to the true time-lapse signal of interest. Consequently, artifacts can have a negative impact on the interpretation of time-lapse signals. Therefore, developing seismic inversion methods that are capable of suppressing the aforementioned artifacts is a problem of practical interest. Time-lapse seismic inversion for monitoring has been extensively studied by different authors. For instance, time-lapse AVO inversion has been addressed by Buland and El Ouair (2006), Kato and Stewart (2010) and Saeed (2014). Similarly, linearized time-lapse inversion was presented by Ayeni and Biondi (2011). Double difference tomography (Zhang and Thurber, 2003; Watanabe et al., 2004; Denli and Huang, 2009; Tao et al., 2013) and joint and double difference full waveform inversion (FWI) (Denli and Huang, 2009; Queißer and Singh, 2010; Zheng et al., 2011; Zhang and Huang, 2013; Asnaashari et al., 2013; Yang et al., 2013; Alemie, 2014; Maharramov et al., 2015) are also viable ways of tackling time-lapse inversion.

In this chapter, I present time-lapse inversion strategies that are based on Full Waveform Inversion (FWI). In particular, I propose a Joint Reparametrized Time-Lapse FWI algorithm (JRTL FWI), which is capable of attenuating artifacts introduced into the time-lapse signal of interest. The seismic data are modeled via a 2D frequency domain acoustic wave equation with Perfectly Matched Layer (PML) boundary conditions. The inversion algorithm is derived using a least squares data misfit that is minimized via the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization method (Nash and Nocedal, 1991; Ma and Hale, 2012). Geophyiscal inverse problems, in general, are ill-posed problems; the solution to the inverse problem may not be unique, or it may not depend continuously on the data. Regularization techniques are often used to estimate a stable solution. The latter entails

adopting regularization constraints that minimize the complexity of the solution (Tarantola and Valette, 1982; Sun and Shuster, 1992; Zhdanov, 1993). For instance, one can adopt a quadratic regularization term that penalizes roughness by minimizing the Euclidian norm of the first or second derivative of the model. I use the quadratic regularization into the algorithm that penalizes the Euclidian norm of the first order derivative of model parameters in the lateral and vertical directions. The algorithm is also assisted by a two-step Wolfe's condition for the estimation of optimal step length (Wolfe, 1969).

The proposed JRTL FWI algorithm is compared with three time-lapse inversion strategies: independent inversion, double difference inversion, and joint inversion. I study the four inversion strategies with synthetic data examples. These examples are produced by adopting a modified Marmousi velocity model (Versteeg, 1994). I alter the model in a target area by incorporating 15% decrease in velocity to mimic a time-lapse change in a reservoir. I also explore the characteristics of the algorithm in conditions where one cannot guarantee the repeatability of the seismic experiment.

5.2 Time-Lapse Inversion Methods

In the previous chapter, a general mathematical formulation of FWI is discussed. Four optimization methods are also covered in detail with numerical examples. In this section, the ordinary FWI is extended specifically to time-lapse full waveform inversion via the L-BFGS method. Four time-lapse inversion strategies with the L-BFGS method are explored: independent inversion, double difference inversion, joint inversion and joint reparameterized inversion. Each of the time-lapse inversion strategies outlined below uses the general theoretical scheme with their corresponding modification.

It is also important to note that the time-lapse seismic in general is a 4D problem, i.e., 3D in space and time-lapse time. In our treatment, it is a 3D problem, i.e., 2D in space and time-lapse time. Each time-lapse time is represented by time-frame $t_0, t_1, t_3, ...$ For the sake of simplicity, only two time-frames are considered throughout the thesis: t_0 is a baseline frame, and t_1 is a monitor frame.

5.2.1 Independent Inversion

In independent time-lapse FWI, each time-frame data (t_0 and t_1) are independently inverted. Then, the difference between the baseline and monitor velocity images is computed to estimate the time-lapse signature. Assuming Gaussian noise, the cost function for each

time-frame t_i can be written as (Tarantola and Valette, 1982),

$$J_{t_i}(\mathbf{m}_{t_i}) = \frac{1}{2}\sum_{j_{\omega_g}=1}^{n_{\omega_g}}\sum_{j_s=1}^{n_s}\Delta\mathbf{d}_{t_i}(\omega_{j_{\omega_g}}, X_{j_s})^{\dagger}\Delta\mathbf{d}_{t_i}(\omega_{j_{\omega_g}}, X_{j_s}) + \mu_{t_i}J_{t_i}^{SM}(\mathbf{m}_{t_i}), \tag{5.1}$$

where n_{ω_g} is the number of frequencies in a group of frequencies, $\Delta\mathbf{d}_{t_i}(\omega_{j_{\omega_g}}, X_{j_s}) = \mathbf{d}_{t_i}^{cal} - \mathbf{d}_{t_i}^{obs}$, $\mathbf{d}_{t_i}^{cal} = \mathbf{d}_{t_i}^{cal}(\omega_{j_{\omega_g}}, X_{j_s})$ is the calculated data and $\mathbf{d}_{t_i}^{obs} = \mathbf{d}_{t_i}^{obs}(\omega_{j_{\omega_g}}, X_{j_s})$ is observed data. The symbol † represents conjugate transpose. The extra term $J_{t_i}^{SM}$ in Equation 5.1 is a 2D smoothing regularization function given by,

$$J_{t_i}^{SM}(\mathbf{m}_{t_i}) = ||\mathbf{D}_x\mathbf{m}_{t_i}||_2^2 + ||\mathbf{D}_z\mathbf{m}_{t_i}||_2^2, \tag{5.2}$$

where μ_{t_i} is the regularization parameter, $\mathbf{m}_{t_i}$ indicates the P-wave velocity whose j^{th} element is given by $m_{t_{ij}} = \frac{1}{c_{t_{ij}}}$, and $\mathbf{D}_x$ and $\mathbf{D}_z$ are lateral and vertical first order difference operators, respectively.

Using the efficient gradient computation, the gradient has the form,

$$\mathbf{g}_{t_i} = \mathbf{g}_{t_i}^{d} + \mu_{t_i}\mathbf{g}_{t_i}^{SM}, \tag{5.3}$$

where $\mathbf{g}_{t_i}(\omega_{j_{\omega_g}}, X_{j_s})$ is the gradient for a single source and a single frequency. A detailed derivation of the gradient for single source and single frequency is given in Chapter 3 and Chapter 4. The term $\mathbf{g}_{t_i}^{SM}$ is the contribution of the regularization towards the gradient. It is the derivative of the regularization term with respect to the model $\mathbf{m}_{t_i}$, which is found to be,

$$\begin{aligned}\mathbf{g}_{t_i}^{SM} &= \mathbf{D}_x^T\mathbf{D}_x\mathbf{m}_{t_i} + \mathbf{D}_z^T\mathbf{D}_z\mathbf{m}_{t_i},\\ &= (\mathbf{D}_x^T\mathbf{D}_x + \mathbf{D}_z^T\mathbf{D}_z)\mathbf{m}_{t_i}.\end{aligned} \tag{5.4}$$

The approximate Hessian is computed by employing the L-BFGS algorithm. The pseudocode is displayed in Algorithm 3. The output of the L-BFGS algorithm is, therefore, the search direction denoted by $\mathbf{p}_{t_i}$. Then, the solution is updated with the expression

$$\mathbf{m}_{t_i}^{k+1} = \mathbf{m}_{t_i}^{k} + \gamma_{t_i}^{k}\mathbf{p}_{t_i}^{k}. \tag{5.5}$$

5.2.2 Double Difference Inversion

The baseline (t_0) inversion in double difference FWI follows exactly the same procedure as the independent inversion. First, the baseline velocity ($\mathbf{m}_{t_0}$) is inverted and the result is

Algorithm 3 L-BFGS algorithm to compute search direction $\mathbf{p}_{t_i}$: k indicates current iteration, N_g is the number of history of gradients in the previous iterations, and t_i is a label to time-frames, where $i = 0, 1, 2, \ldots$. In our case, for two time-frames, $i = 0, 1$.

Input: $l = k - 1, \Delta\mathbf{m}^l = \mathbf{m}_{t_i}^{l+1} - \mathbf{m}_{t_i}^l, \Delta\mathbf{g}^l = \mathbf{g}_{t_i}^{l+1} - \mathbf{g}_{t_i}^l$,
$\rho^l = \frac{1}{(\Delta\mathbf{m}^l)^T \Delta\mathbf{g}^l}, \mathbf{p} \leftarrow \mathbf{g}_{t_i}^k, \varphi = \frac{(\Delta\mathbf{g}^l)^T \Delta\mathbf{m}^l}{(\Delta\mathbf{m}^l)^T \Delta\mathbf{m}^l}$
while $l \geqslant k - N_g$ **do**
  $\alpha^l \leftarrow \rho^l (\Delta\mathbf{g}^l)^T \mathbf{p}$
  $\mathbf{p} = \mathbf{p} - \alpha^l \Delta\mathbf{m}^l$
  $l = l - 1$
end
$\mathbf{p} \leftarrow \varphi\mathbf{p}$, $l = k - N_g$
while $l \leqslant k - 1$ **do**
  $\beta = \rho^l (\Delta\mathbf{m}^l)^T \mathbf{p}$
  $\mathbf{p} = \mathbf{p} + (\alpha^l - \beta)\Delta\mathbf{g}^l$
  $l = l + 1$
end
Output: search direction
return $\mathbf{p}_{t_i} \leftarrow -\mathbf{p}$

used for the formulation of the double difference method to invert the monitor model ($\mathbf{m}_{t_1}$). To this end, the monitor data residual is expressed in the following equation (Zhang and Thurber, 2003),

$$\Delta\mathbf{d}_{t_1}^{dd} = (\mathbf{R}_{t_1}\mathbf{u}_{t_1} - \mathbf{d}_{t_1}^{obs}) - (\mathbf{R}_{t_0}\mathbf{u}_{t_0} - \mathbf{d}_{t_0}^{obs}), \tag{5.6}$$

where $\mathbf{R}_{t_0}$ and $\mathbf{R}_{t_1}$ are the baseline and monitor data sampling operators, respectively. The corresponding cost function is given by,

$$J^{t_1}(\mathbf{m}_{t_1}) = \frac{1}{2}\sum_{j_{\omega_g}=1}^{n_{\omega_g}}\sum_{j_s=1}^{n_s} \Delta\mathbf{d}_{t_1}^{dd}(\omega_{j_{\omega_g}}, X_{j_s})^\dagger \Delta\mathbf{d}_{t_1}^{dd}(\omega_{j_{\omega_g}}, X_{j_s}) + \mu_{t_1} J_{t_1}^{SM}(\mathbf{m}_{t_1}). \tag{5.7}$$

Next, the derivation of the gradient for this particular cost function, $\mathbf{g}_{t_1}^{dd}(\omega_{j_{\omega_g}}, X_{j_s})$, for single source and single frequency is derived.

Gradient in Double Difference

The gradient in the case of double difference for monitor time-frame (t_1) inversion has different expression than the independent inversion due to the unique definition of the cost function. The derivation of the gradient in this particular case follows exactly the same procedure in Chapter 3 except that the $\Delta\mathbf{d}$ must be replaced by the $\Delta\mathbf{d}^{dd}$ given in Equation

5.6. To this end, we start from Equation 3.12 by replacing $\Delta\mathbf{d}$ by $\Delta\mathbf{d}^{dd}$. Note also that only $\mathbf{d}_{t_1}^{cal}$ depends on $(m_j)_{t_1}$. Thus, by taking the partial derivative of J_{t_1} with respect to $(m_j)_{t_1}$ following similar steps in Equation 3.9 and using the fact in Equation 3.10, we get

$$\left(\frac{\partial J_{j_{\omega_g},j_s}^{t_1}}{\partial (m_j)_{t_1}}\right) = \Re\left\{\left(\frac{\partial \mathbf{d}_{t_1}^{cal}}{\partial (m_j)_{t_1}}\right)^T (\Delta\mathbf{d}_{t_1}^{dd})^*\right\}. \tag{5.8}$$

The generalized first order derivative of the cost function can be put in a more compact vector-matrix form as follows,

$$\begin{aligned}\mathbf{g}_{t_1}^{dd}(\omega_{j_{\omega_g}}, X_{j_s}) &= \Re\left\{\left(\frac{\partial \mathbf{d}_{t_1}^{cal}}{\partial \mathbf{m}_{t_1}}\right)^T (\Delta\mathbf{d}_{t_1}^{dd})^*\right\} \\ &= \Re\{\mathbf{F}_{t_1}^T \mathbf{R}_{t_1}^T (\Delta\mathbf{d}_{t_1}^{dd})^*\},\end{aligned} \tag{5.9}$$

Following similar steps from Equation 3.14 to 3.17, the gradient of the cost function in double difference monitor inversion becomes

$$\mathbf{g}_{t_1}^{dd}(\omega_{j_{\omega_g}}, X_{j_s}) = \Re\{[\mathbf{F}_{t_1}^v]^T [\mathbf{A}_{t_1}^{-1}]^T \mathbf{R}_{t_1}^T (\Delta\mathbf{d}_{t_1}^{dd})^*\}. \tag{5.10}$$

Equation 5.10 is the same expression as Equation 3.18 except that in the double difference monitor inversion discards the data left unfitted in the baseline inversion. Hence, only the monitor data that matter are back-propagated.

The generalized gradient for n_{ω_g} frequencies and n_s sources is given by

$$\mathbf{g}_{t_1} = \sum_{j_{\omega_g}=1}^{n_{\omega_g}} \sum_{j_s=1}^{n_s} \mathbf{g}_{t_1}^{dd}(\omega_{j_{\omega_g}}, X_{j_s}) + \mu_{t_1} \mathbf{g}_{t_1}^{SM}. \tag{5.11}$$

It is the definition of the data residual given by Equation 5.6 that makes the inversion in the double difference method different from the independent inversion. In other words, Equation 5.3 for $i = 1$ and Equation 5.11 differ only in the back propagation of the data residual. The rest of the algorithm shares the same procedure as the independent inversion.

5.2.3 Joint Inversion

In joint inversion method, I define the cost function as the sum of the individual cost functions of each time-frame

$$J(\mathbf{m}) = \sum_{t_i} \{J_{t_i}^d(\mathbf{m}_{t_i}) + \mu_{t_i} J_{t_i}^{SM}(\mathbf{m}_{t_i})\}, \tag{5.12}$$

where

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}_{t_0} \\ \mathbf{m}_{t_1} \end{bmatrix}. \tag{5.13}$$

The derivative of the cost function 5.12 with respect to each time-frame $\mathbf{m}_{t_i}$ yields $\mathbf{g}_{t_i}$, which is identical to Equation 5.3. The search direction has the following form,

$$\begin{bmatrix} \mathbf{p}_{t_0} \\ \mathbf{p}_{t_1} \end{bmatrix}^k = - \begin{bmatrix} \mathbf{H}_{t_0}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{t_1}^{-1} \end{bmatrix}^k \begin{bmatrix} \mathbf{g}_{t_0} \\ \mathbf{g}_{t_1} \end{bmatrix}^k. \tag{5.14}$$

Note that in Equation 5.14, $\mathbf{H}_{t_0}^{-1}$ and $\mathbf{H}_{t_1}^{-1}$ are not computed explicitly. They are displayed just for the purpose of clarity. Instead, the L-BFGS algorithm is used to compute the search directions $\mathbf{p}_{t_0}$ and $\mathbf{p}_{t_1}$. Then, the concatenated gradients and search directions are used to estimate a single step length, γ^k, at each iteration. Then, the solution is updated as follows,

$$\mathbf{m}^{k+1} = \begin{bmatrix} \mathbf{m}_{t_0} \\ \mathbf{m}_{t_1} \end{bmatrix}^{k+1} = \begin{bmatrix} \mathbf{m}_{t_0} \\ \mathbf{m}_{t_1} \end{bmatrix}^k + \gamma^k \begin{bmatrix} \mathbf{p}_{t_0} \\ \mathbf{p}_{t_1} \end{bmatrix}^k. \tag{5.15}$$

The motivation behind the joint inversion is the existence of a strong correlation between data sets in time-lapse seismic surveys. In this case, it is the estimation of a single step length that makes this method a joint inversion. It creates a link between the two time-frames.

The joint inversion is expected to produce results that are similar to those obtain via the independent inversion. In fact, the only link between the models at different time frames is the step length that is simultaneously computed to minimize the cost function of the joint inversion. As one can imagine, this link only marginally assists the cooperative inversion of baseline and monitor models. As a result, the improvement that can be achieved by the joint inversion method in comparison to the independent inversion is minimal.

5.2.4 Joint Reparametrized Inversion

The main objective of the joint reparametrized inversion is to incorporate a priori information systematically through the inversion process (Alemie and Sacchi, 2016). In this regard, it can be considered as an indirect regularization technique for time-lapse inversion. The main difference to the previously presented methods is that the joint reparametrized inversion incorporates strong a priori information into the inversion. This is possible by emphasizing the fact that there exists correlation between baseline and monitor data sets.

The definition of the cost function as well as the baseline inversion have similarity with that

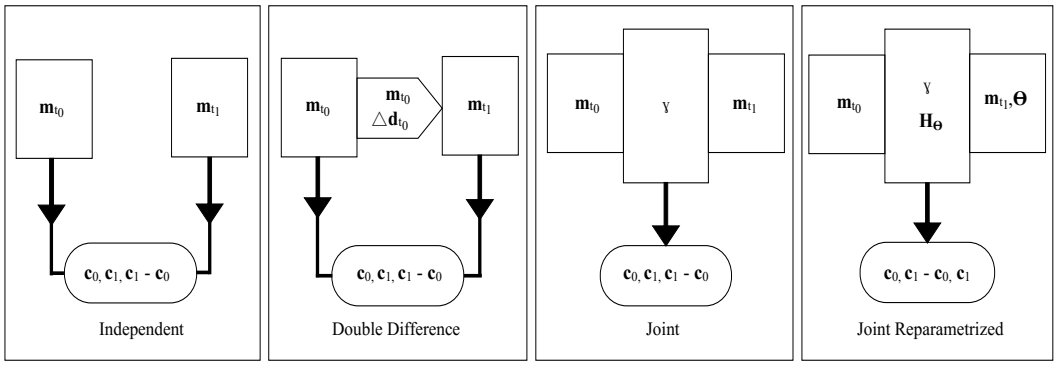


Figure 5.1: A simplified diagram that illustrates the four inversion strategies.

of the joint inversion strategy shown in Equation 5.12 in section 5.2.3. Next, I focus on deriving the reparametrization method proposed in this paper. To this end, the difference between baseline and monitor model parameters at the k^{th} iteration is

$$\mathbf{\Theta}^k = \mathbf{m}_{t_1}^k - \mathbf{m}_{t_0}^k. \tag{5.16}$$

This parametrization implies that the j^{th} element of the velocity of the monitor data (t_1) at the k^{th} iteration can be expressed as

$$(c_j^k)_{t_1} = \frac{(c_j^k)_{t_0}}{1 + \Theta_j^k (c_j^k)_{t_0}}. \tag{5.17}$$

The difference parameter, $\mathbf{\Theta}$, is calculated by assuming that the gradient of the joint cost function with respect to this parameter is equal to the difference of the gradients of the cost function with respect to the individual parameters, $\mathbf{m}_{t_0}$ and $\mathbf{m}_{t_1}$. To clearly understand $\mathbf{\Theta}$, I write the solution update, Equation 5.15, in the gradient descent method,

$$\begin{bmatrix} \mathbf{m}_{t_0} \\ \mathbf{m}_{t_1} \end{bmatrix}^{k+1} = \begin{bmatrix} \mathbf{m}_{t_0} \\ \mathbf{m}_{t_1} \end{bmatrix}^{k} + \gamma^k \begin{bmatrix} \mathbf{g}_{t_0} \\ \mathbf{g}_{t_1} \end{bmatrix}^{k}. \tag{5.18}$$

Note in the gradient descent method that only the gradient information is used to update the solution. I can also calculate the difference between baseline and monitoring model parameters at $(k+1)^{th}$ iteration that corresponds to Equation 5.18 as follows,

$$\begin{aligned} \mathbf{\Theta}^{k+1} &= [\mathbf{m}_{t_1} - \mathbf{m}_{t_0}]^k + \gamma^k [\mathbf{g}_{t_1} - \mathbf{g}_{t_0}]^k, \\ &= \mathbf{\Theta}^k + \gamma^k \mathbf{g}_{\Theta}^k. \end{aligned} \tag{5.19}$$

Equation 5.19 is an expression similar to the model update given by Equation 5.18 except

that I have expressed the model difference at the $(k+1)^{th}$ iteration in terms of the difference $\Delta\mathbf{m}$ and the difference of gradients ($\mathbf{g}_\Theta$) obtained at the $(k)^{th}$ iteration. Since the solution update (Equation 5.18) for each time-frame are linear with respect to the corresponding $\mathbf{g}_{t_i}$, the difference also results in a linear equation. The linearity is the consequence of using gradient descent. However, it suggests that the gradient of the cost function with respect to $\boldsymbol{\Theta}$ can be approximated by the difference of the gradients of the individual model parameters,

$$\nabla_\Theta J_\Theta \approx \mathbf{g}_\Theta^k = \mathbf{g}_{t_1}^k - \mathbf{g}_{t_0}^k. \tag{5.20}$$

In general, the gradient descent method is not a suitable optimization procedure as the step length alone can not properly scale the gradient to find a proper search direction. Taking advantage of the L-BFGS method, I can compute the Hessian associated to the difference of the gradients $\mathbf{g}_\Theta$, implicitly. That is to say, I calculate a search direction for a new parameter whose gradient is $\mathbf{g}_\Theta$ and Hessian $\mathbf{H}_\Theta$,

$$\begin{aligned} \mathbf{p}_\Theta &= \mathbf{H}_\Theta^{-1}(\mathbf{g}_{t_1} - \mathbf{g}_{t_0}), \\ &= \mathbf{H}_\Theta^{-1}\mathbf{g}_\Theta. \end{aligned} \tag{5.21}$$

The Hessian, $\mathbf{H}_\Theta$, can be thought of as an intermediate Hessian for the two time-frames. It strengthens the strong correlation assumption between time-lapse data sets of time-lapse seismic surveys. Thus, the search direction for the joint reparametrized inversion can be written as

$$\begin{bmatrix} \mathbf{p}_{t_0} \\ \mathbf{p}_\Theta \end{bmatrix}^k = - \begin{bmatrix} \mathbf{H}_{t_0}^{-1} & \mathbf{0} \\ \mathbf{0} & {\mathbf{H}_\Theta}^{-1} \end{bmatrix}^k \begin{bmatrix} \mathbf{g}_{t_0} \\ \mathbf{g}_\Theta \end{bmatrix}^k, \tag{5.22}$$

and the solution is updated as follows,

$$\begin{bmatrix} \mathbf{m}_{t_0} \\ \boldsymbol{\Theta}_{t_1} \end{bmatrix}^{k+1} = \begin{bmatrix} \mathbf{m}_{t_0} \\ \boldsymbol{\Theta}_{t_1} \end{bmatrix}^k + \gamma^k \begin{bmatrix} \mathbf{p}_{t_0} \\ \mathbf{p}_{\Theta_{t_1}} \end{bmatrix}^k. \tag{5.23}$$

Equation 5.23 suggests that the joint reparametrized inversion can be considered as a joint inversion of the baseline velocity and the time-lapse velocity difference. Refer Appendix C to see more on the interpretation of the reparametrization method. The computation of the search direction $\mathbf{p}_{t_0}$ uses Algorithm 3 and the computation of search direction $\mathbf{p}_{\boldsymbol{\Theta}_{t_1}}$ uses the modified Algorithm 4.

Note that the input to Algorithm 3 are the history of the inverted models $\mathbf{m}$ and gradients $\mathbf{g}$. The input to Algorithm 4 are the history of $\boldsymbol{\Theta}$ and the difference of gradients $\mathbf{g}_\Theta$. A summary of the four inversion methods is illustrated in a simplified diagram in Fig. 5.1.

Algorithm 4 : L-BFGS algorithm to compute search direction $\mathbf{p}_{\Theta t_i}$: k indicates current iteration, N_g is the number of history of gradients in the previous iterations, and t_i is a label to time-frames for $i > 0$. For only two time-frames, $i = 1$. A very small constant number ϵ is used to prevent division by zero.

Input: $l = k - 1$,
$\Delta\boldsymbol{\Theta}^l = \boldsymbol{\Theta}^{l+1}_{t_i} - \boldsymbol{\Theta}^l_{t_i}, \Delta\mathbf{g}^l_\Theta = \mathbf{g}^{l+1}_{\Theta t_i} - \mathbf{g}^l_{\Theta t_i}$,
$\rho^l = \frac{1}{((\Delta\boldsymbol{\Theta}^l)^T \Delta\mathbf{g}^l_\Theta + \epsilon)}, \mathbf{p}_\Theta \leftarrow \mathbf{g}^k_{\Theta t_i}, \varphi = \frac{(\Delta\mathbf{g}^l_\Theta)^T \Delta\boldsymbol{\Theta}^l}{((\Delta\boldsymbol{\Theta}^l)^T \Delta\boldsymbol{\Theta}^l + \epsilon)}$
while $l \geqslant k - N_g$ **do**
  $\alpha^l \leftarrow \rho^l (\Delta\mathbf{g}^l_\Theta)^T \mathbf{p}_\Theta$
  $\mathbf{p}_\Theta = \mathbf{p}_\Theta - \alpha^l \Delta\boldsymbol{\Theta}^l$
  $l = l - 1$
end

$\mathbf{p}_\Theta \leftarrow \varphi \mathbf{p}_\Theta$, $l = k - N_g$
while $l \leqslant k - 1$ **do**
  $\beta = \rho^l (\Delta\boldsymbol{\Theta}^l)^T \mathbf{p}_\Theta$
  $\mathbf{p}_\Theta = \mathbf{p}_\Theta + (\alpha^l - \beta) \Delta\mathbf{g}^l_\Theta$
  $l = l + 1$
end
Output: search direction
return $\mathbf{p}_{\Theta t_i} \leftarrow -\mathbf{p}_\Theta$

5.3 Synthetic Examples

5.3.1 Model and Data Acquisition Specifications

In this section, the four time-lapse inversion methods are investigated with synthetic examples by adopting the modified Marmousi velocity model. The model is modified so that our example mimics a time-lapse seismic problem. I have considered two time-frames: Fig. 5.2 (**a**) is the true baseline velocity model and Fig. 5.2 (**b**) is the true monitor velocity model produced by altering the true baseline velocity model with 15% decrease in velocity at a depth approximately 2.1 km. Fig. 5.2 (**c**) is the true monitor velocity difference i.e. the difference between the baseline velocity model and the true monitor velocity model.

To generate the time-lapse data, I used grid sizes $dx = dz = 12$ m, 96 shots with shot separation of 72 m and each shot having 287 receivers with 24 m receiver separation. To simulate a non-repeatable data acquisition, 10 regularly selected sources of the baseline geometry were shifted horizontally by 12 m. In other words, every source-receiver pair of the selected sources had horizontal acquisition difference of 12 m as compared to the corresponding baseline acquisition geometry. A Ricker wavelet with a central frequency of

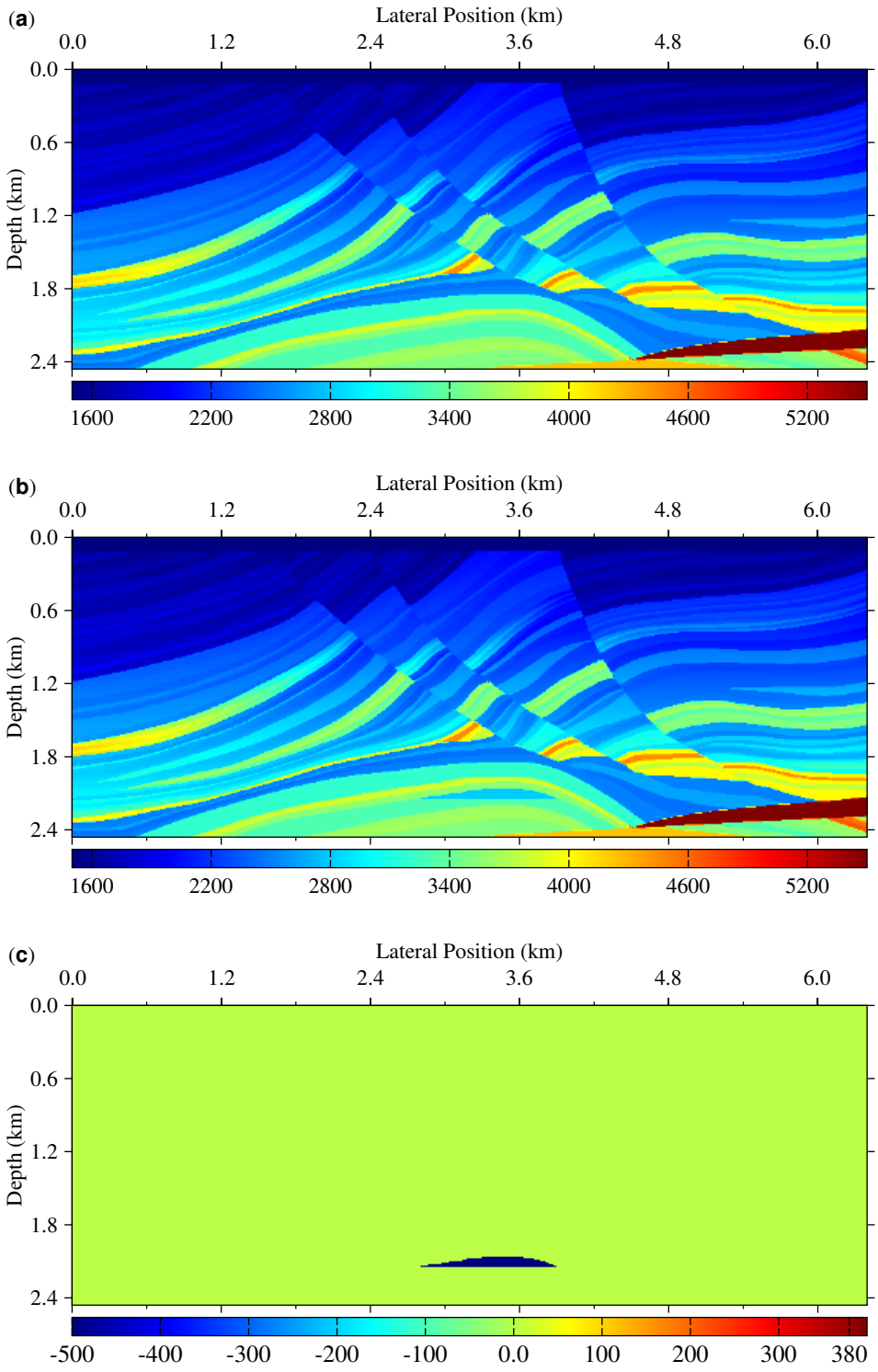


Figure 5.2: (**a**) True baseline velocity model (**b**) true monitor velocity model and (**c**) true time lapse velocity change.

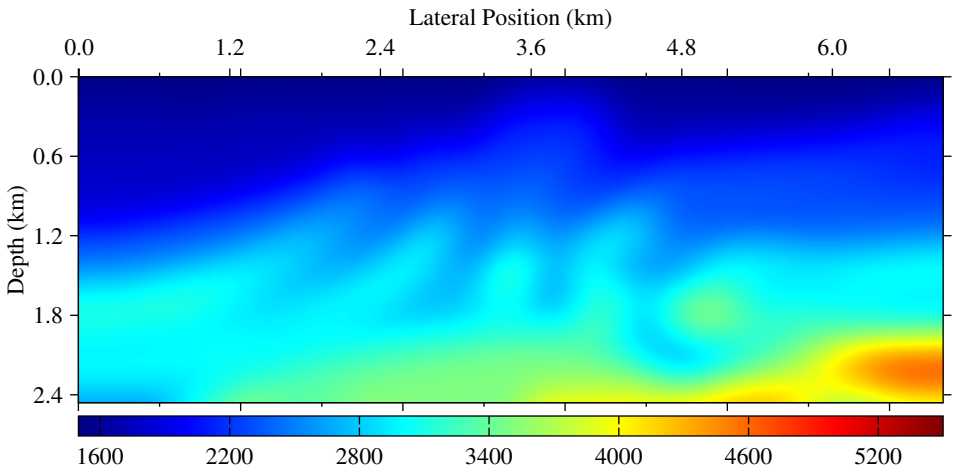


Figure 5.3: Initial velocity model used for time-lapse inversion.

15 Hz was utilized to model the seismic sources.

5.3.2 Results

The initial velocity used for inversion is shown in Figure 5.3. A total of 14 frequencies in a frequency range from 2.68 Hz to 34.42 Hz were selected for the inversion. The inversion at each frequency run until the evaluated cost function was less than 0.001% or the maximum iteration of 101 was reached. Three synthetic scenarios based on the criteria of repeatability are presented below: repeatable, non-repeatable but true acquisition geometries, and non-repeatable.

Case One: Repeatable

In this example, I assumed that the seismic experiments are repeatable. The same acquisition geometry for both baseline and monitor frames were used to generate the synthetic data. In addition, the same geometry was used for the inversion. Figures 5.4 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by independent method, respectively. Figures 5.5 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and their difference by double difference method, respectively. Figures 5.6 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by the joint method, respectively. Figures 5.7 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint reparametrized method, respectively.

Figures 5.8 (**a**), (**b**), (**c**), and (**d**) are vertical profiles of the monitor inversion by independent, double difference, joint and joint reparametrized methods, respectively. Each section of Figure 5.8 depicts a comparison of inverted monitor velocity model against the true and initial velocity models. Figure 5.9 shows a representative mean horizontal profile errors for each method. The mean absolute errors between the true and estimated velocity differences were calculated for all lateral positions by averaging in depth. Refer Appendix D to see how the errors are calculated.

Figures 5.10 (**a**), (**b**) and (**c**) are comparison of the behavior of the cost function and computational time for the four time-lapse inversion methods in repeatable case. The normalized CPU time is plotted against the number of iteration as shown in Figure 5.10 (**a**). Figure 5.10 (**b**) depicts the normalized cost function as a function of iteration. Figure 5.10 (**c**) shows the cost function as a function of normalized CPU time. The cost functions and the CPU time at each iteration are aggregates of 14 frequencies. Note that the evaluation of the cost functions and the computational time in the case of joint and joint reparametrized are straightforward as the cost functions are the sum of the cost function of the baseline and monitor cost functions. For the independent and double-difference methods, I evaluated the cost functions and computational times of the baseline and monitoring survey inversions independently, and I added them up for each iteration.

Case Two: Non-Repeatable but true acquisition geometries

In this scenario, the acquisition geometries for baseline and monitor cases were different. The baseline geometry was similar to the repeatable. In the monitor survey, 10 regularly sampled sources were shifted horizontally each by 12 m. However, it is assumed that the true geometries were known in both cases. I used two different known geometries. Figures 5.11(**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by independent method, respectively. Figures 5.12 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and their difference by double difference method, respectively. Figures 5.13 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint method, respectively. Figures 5.14 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint reparametrized method, respectively.

Figures 5.15 (**a**), (**b**), (**c**), and (**d**) are vertical profiles of the monitor inversion by independent, double difference, joint and joint reparametrized methods, respectively. Each section of Figure 5.15 depicts a comparison of inverted monitor velocity model against the true and initial velocity models. Figure 5.16 shows a representative mean horizontal profile errors for each method. The mean absolute errors between the true and estimated velocity differences were calculated for all lateral positions by averaging in depth.

Figures 5.17 (**a**), (**b**) and (**c**) are comparison of the behavior of the cost function and computational time for the four time-lapse inversion methods in repeatable case. The normalized CPU time is plotted against the number of iteration as shown in Figure 5.17 (**a**). Figure 5.17 (**b**) depicts the normalized cost function as a function of iteration. Figure 5.17 (**c**) shows the cost function as a function of normalized CPU time. The cost functions and the CPU time at each iteration are aggregates over the 14 frequencies.

Case Three: Non-Repeatable

This scenario represents the true non-repeatability experiment. Similar to case two, the true acquisition geometries for baseline and monitoring surveys were different. In the inversion process, however, I assumed that both have the same geometries i.e. the baseline geometry. Figures 5.18 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity, and the difference by independent method, respectively. Figures 5.19 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity, and their difference by double difference method, respectively. Figures 5.20 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity, and the difference by joint method, respectively. Figures 5.21 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity, and the difference by joint reparametrized method, respectively.

Figures 5.22 (**a**), (**b**), (**c**), and (**d**) are vertical profiles of the monitor inversion by independent, double difference, joint and joint reparametrized methods, respectively. Each section of Figure 5.22 depicts a comparison of inverted monitor velocity model against the true and initial velocity models. Figure 5.23 shows a representative mean horizontal profile errors for each method. The mean absolute errors between the true and estimated velocity differences were calculated for all lateral positions by averaging in depth.

Figures 5.24 (**a**), (**b**) and (**c**) are comparison of the behavior of the cost function and computational time for the four time-lapse inversion methods in a non-repeatable case. The normalized CPU time is plotted against the number of iteration as shown in Figure 5.24 (**a**). Figure 5.24 (**b**) depicts the normalized cost function as a function of iteration. Figure 5.24 (**c**) shows the cost function as a function of normalized CPU time. The cost functions and the CPU time at each iteration are aggregates over the 14 frequencies.

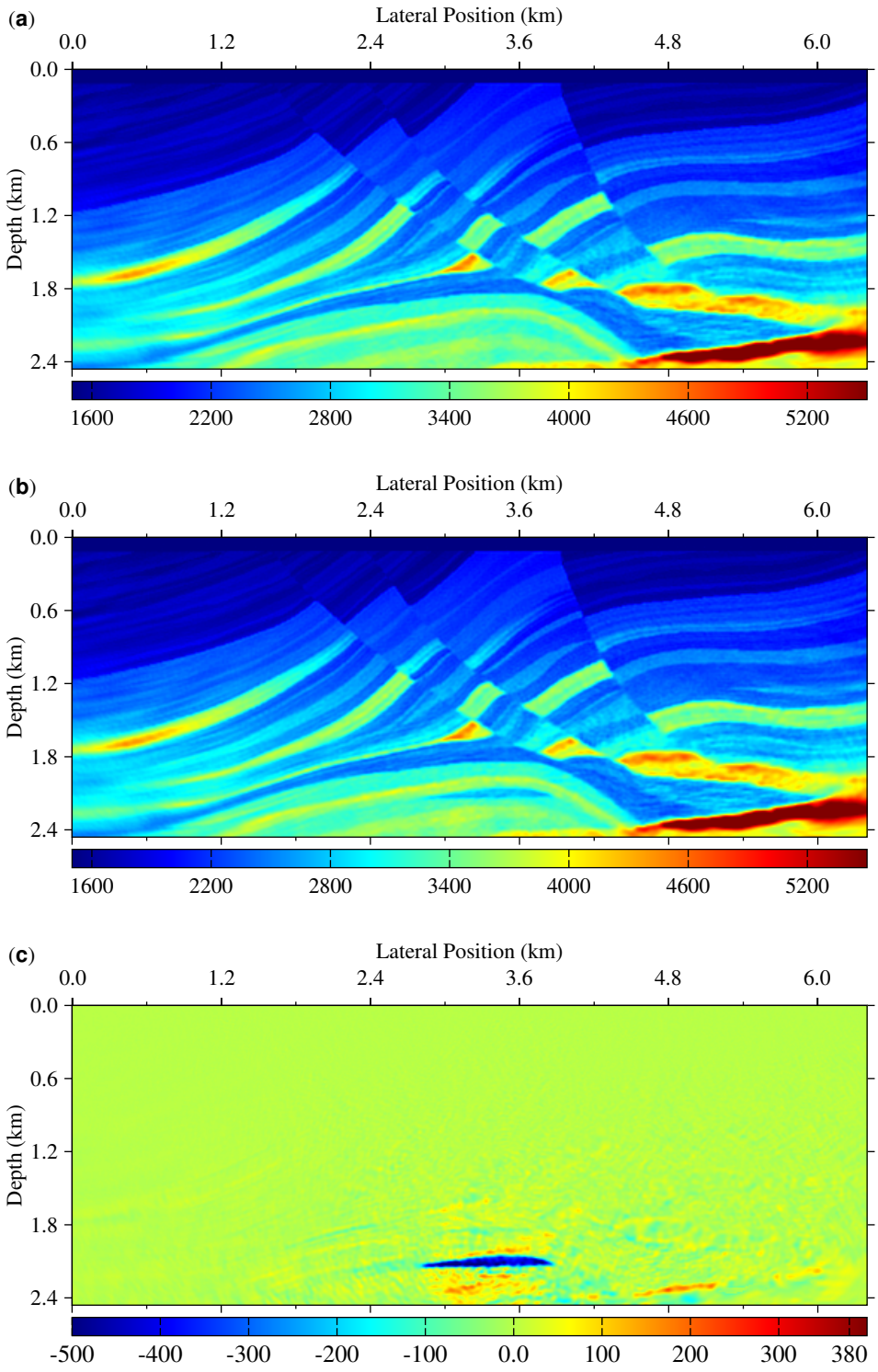


Figure 5.4: Case One: inversion by the independent method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

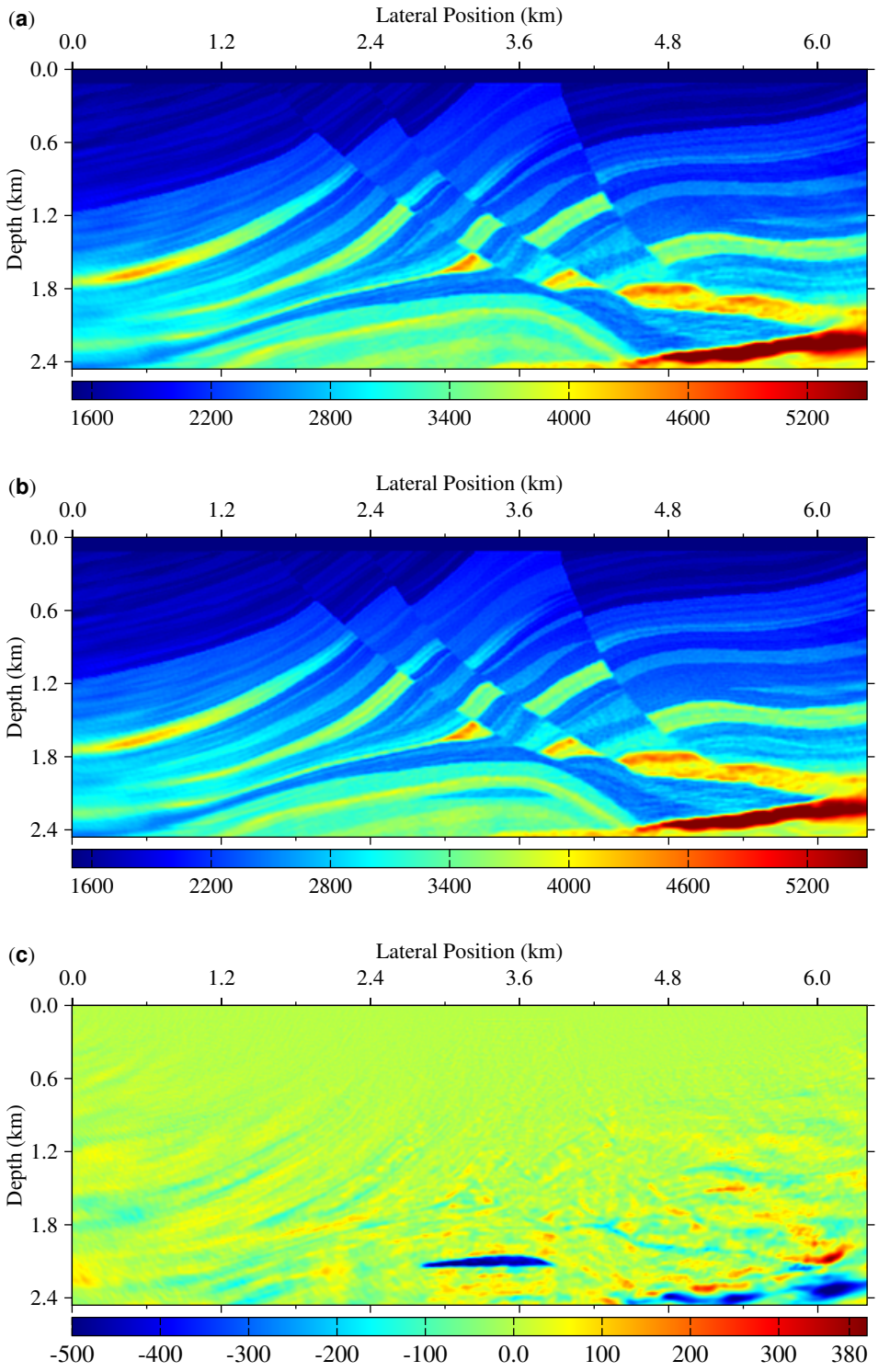


Figure 5.5: Case One: inversion by the double difference method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

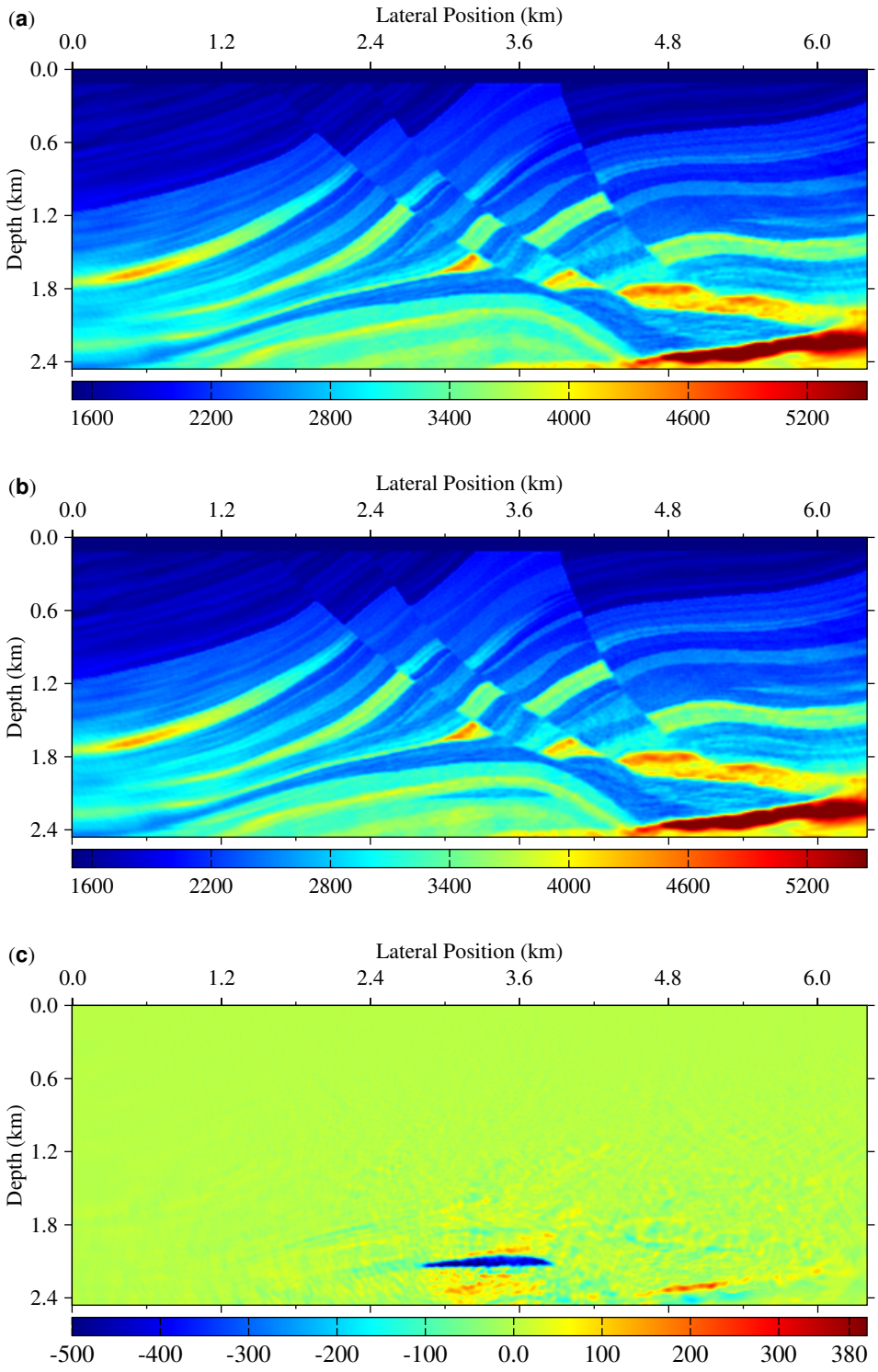


Figure 5.6: Case One: inversion by the joint method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

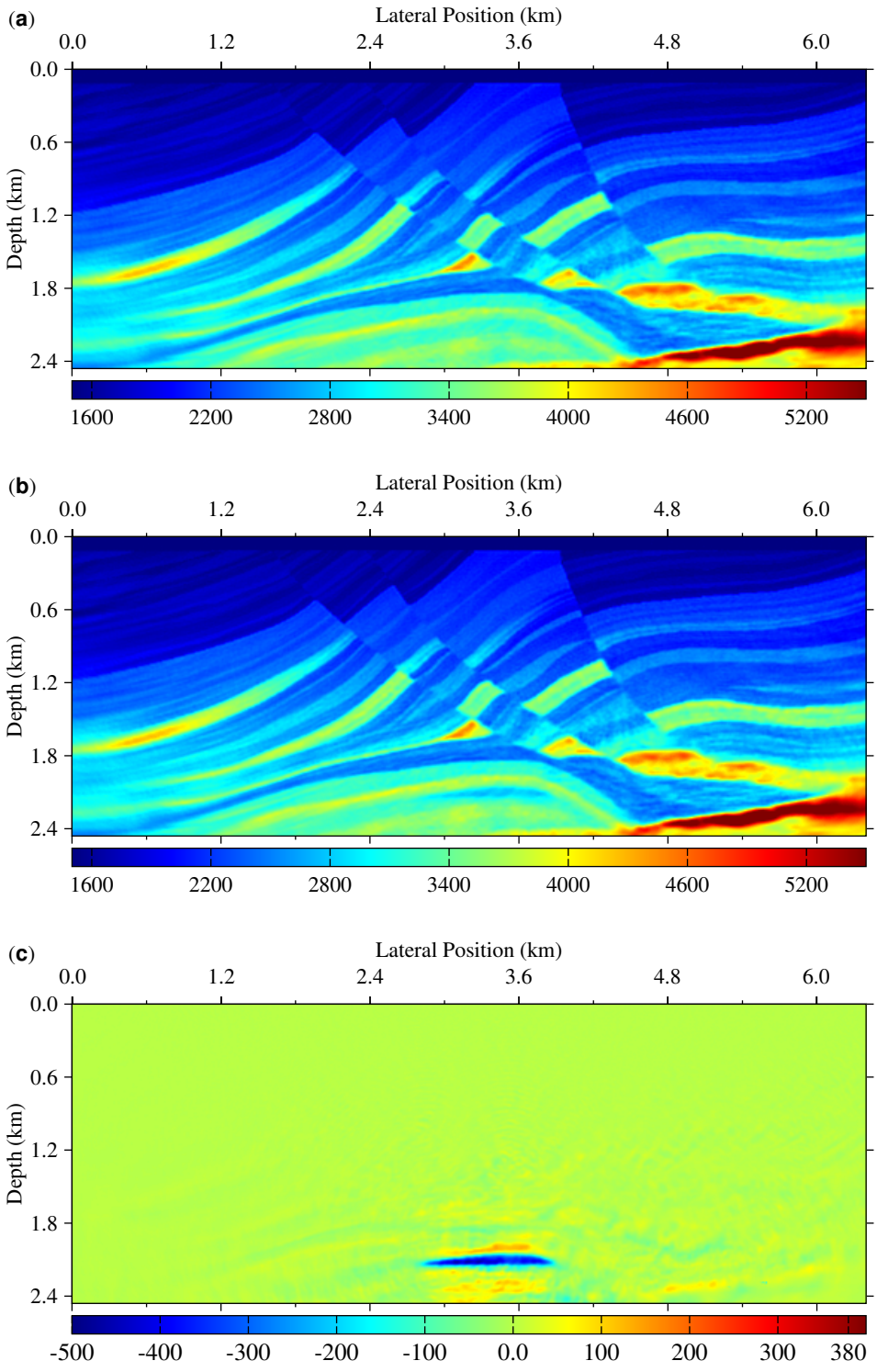


Figure 5.7: Case One: inversion by the joint reparametrized method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

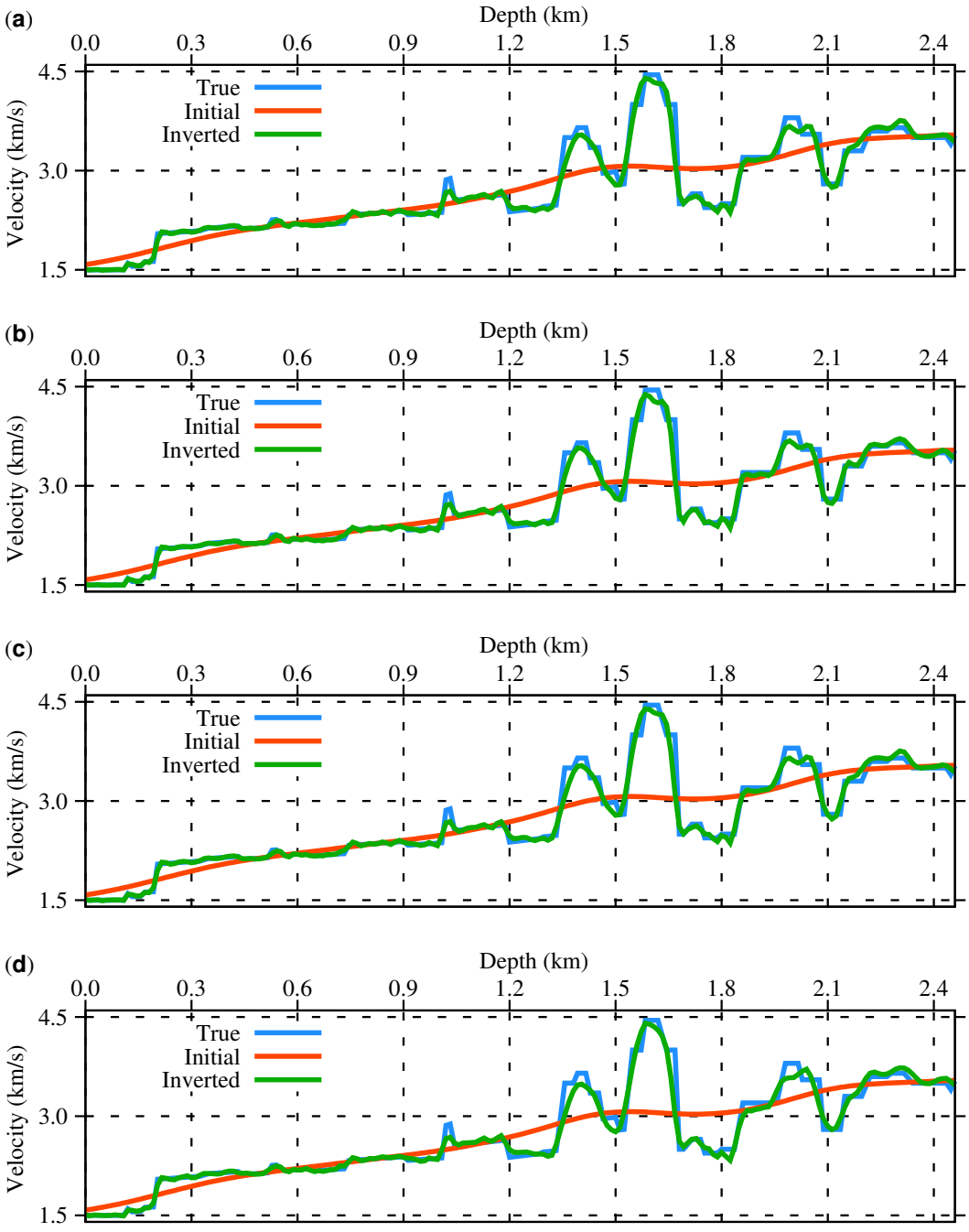


Figure 5.8: Case One: comparison of vertical profiles of inverted monitor velocity with the true and initial velocity models (**a**) independent, (**b**) double difference, (**c**) joint, and (**d**) joint reparametrized.

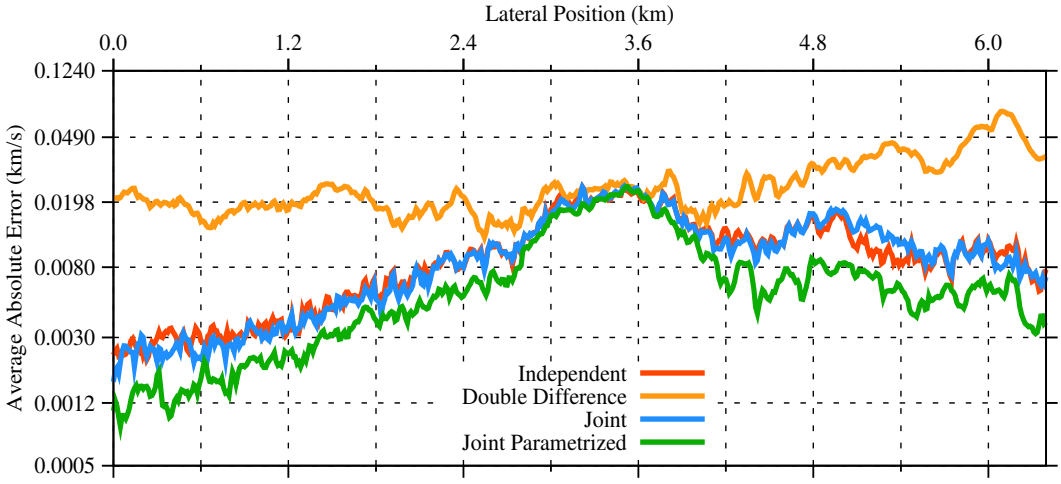


Figure 5.9: Case One: error comparison of the four methods for the repeatable case. The average absolute errors are representatives of horizontal profile errors that were averaged vertically. The vertical axes are plotted in log-scale.

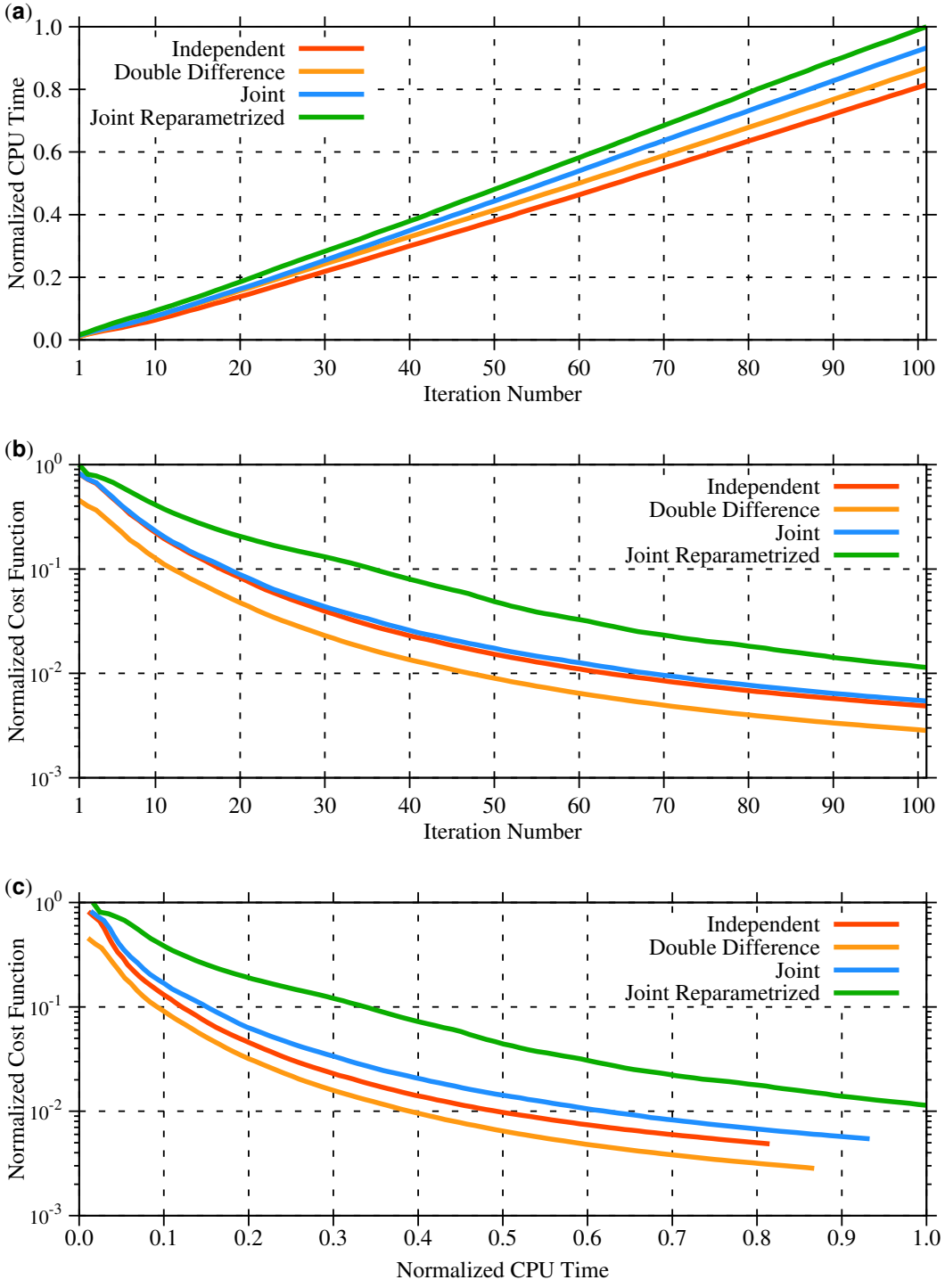


Figure 5.10: Case One: (**a**) normalized computation time versus iteration number (**b**) normalized cost function versus iteration number, (**c**) normalized cost function versus normalized computation time. The vertical axes are plotted in log-scale.

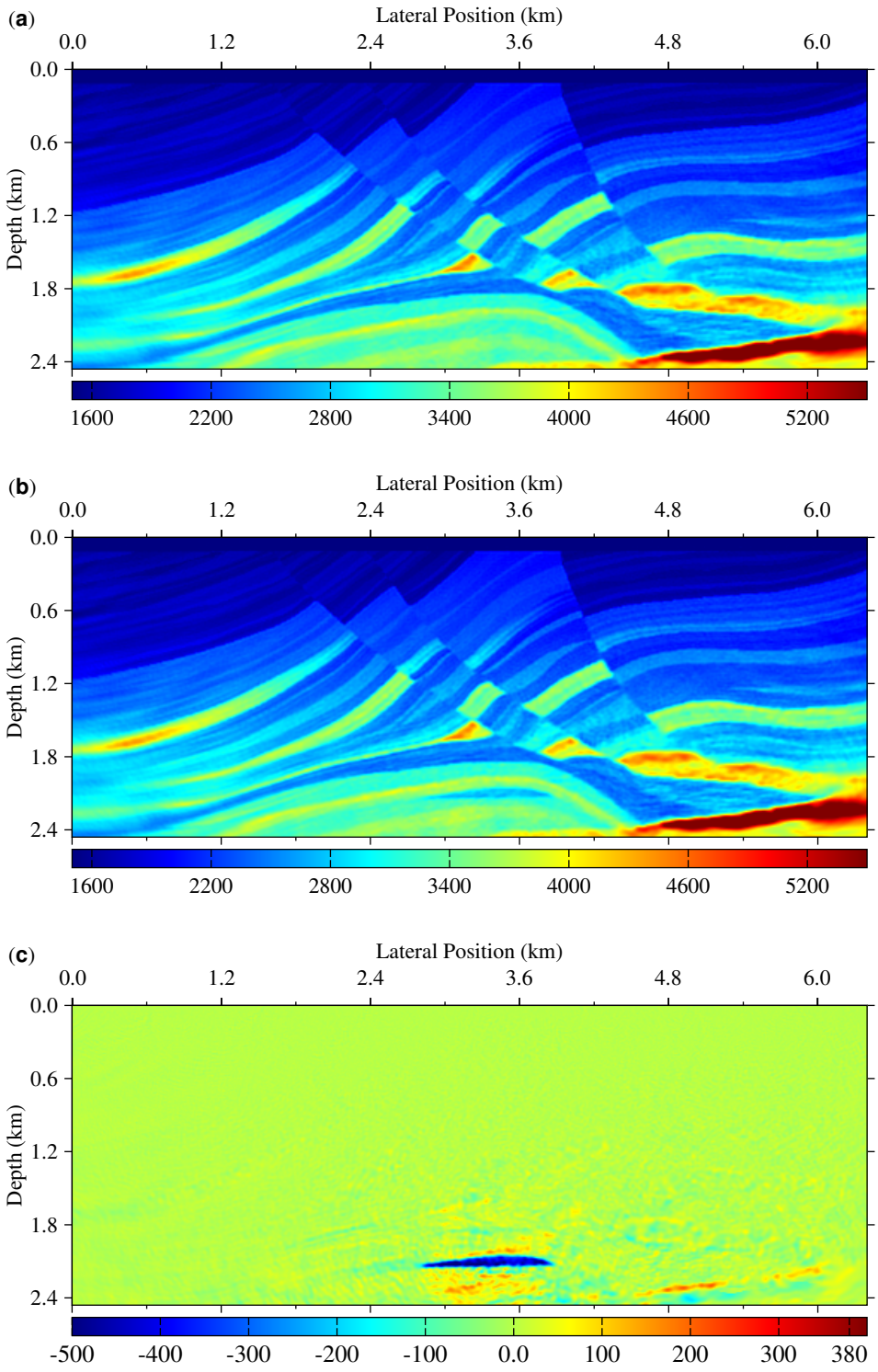


Figure 5.11: Case Two: inversion by the independent method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

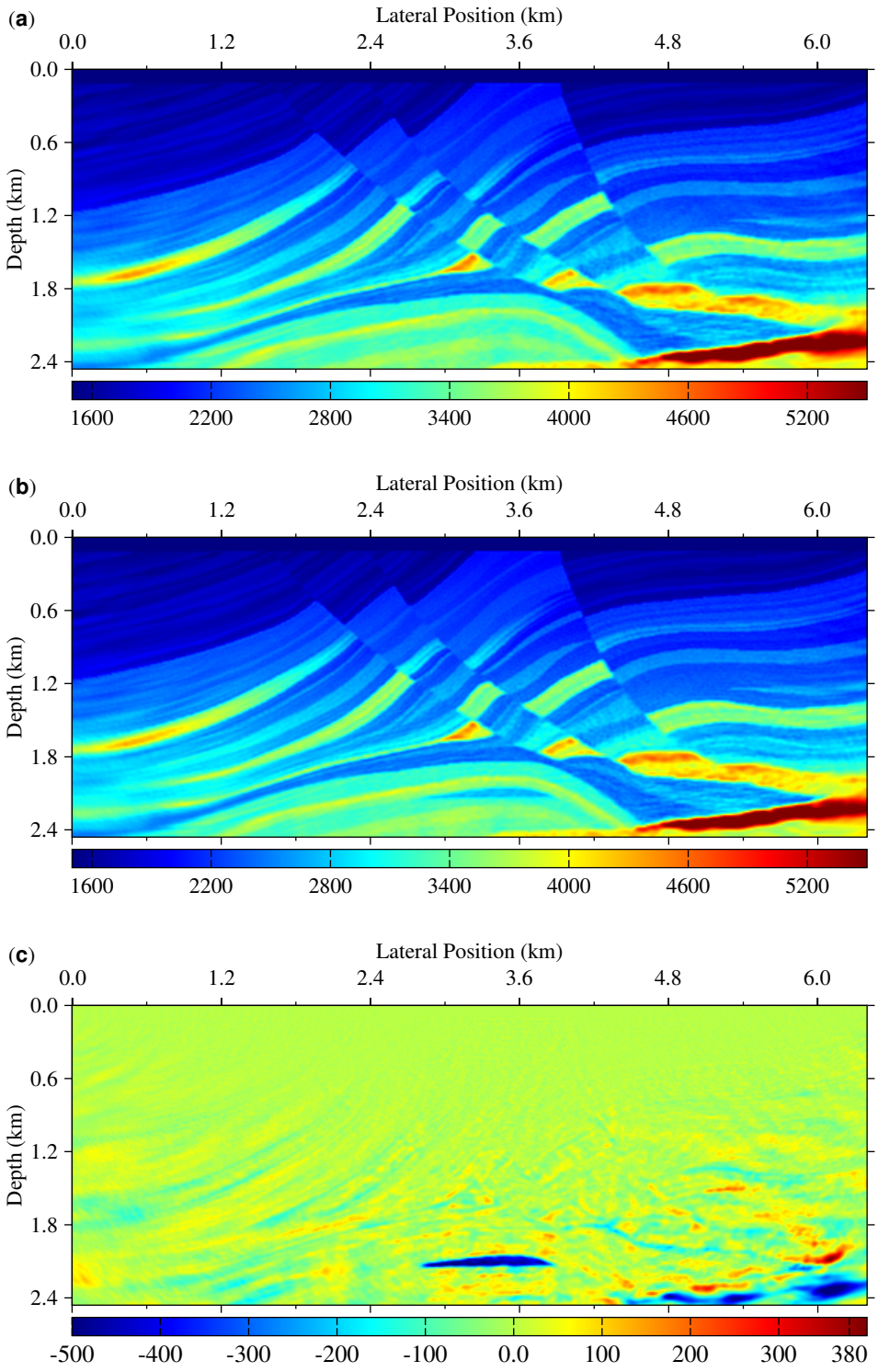


Figure 5.12: Case Two: inversion by the double difference method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

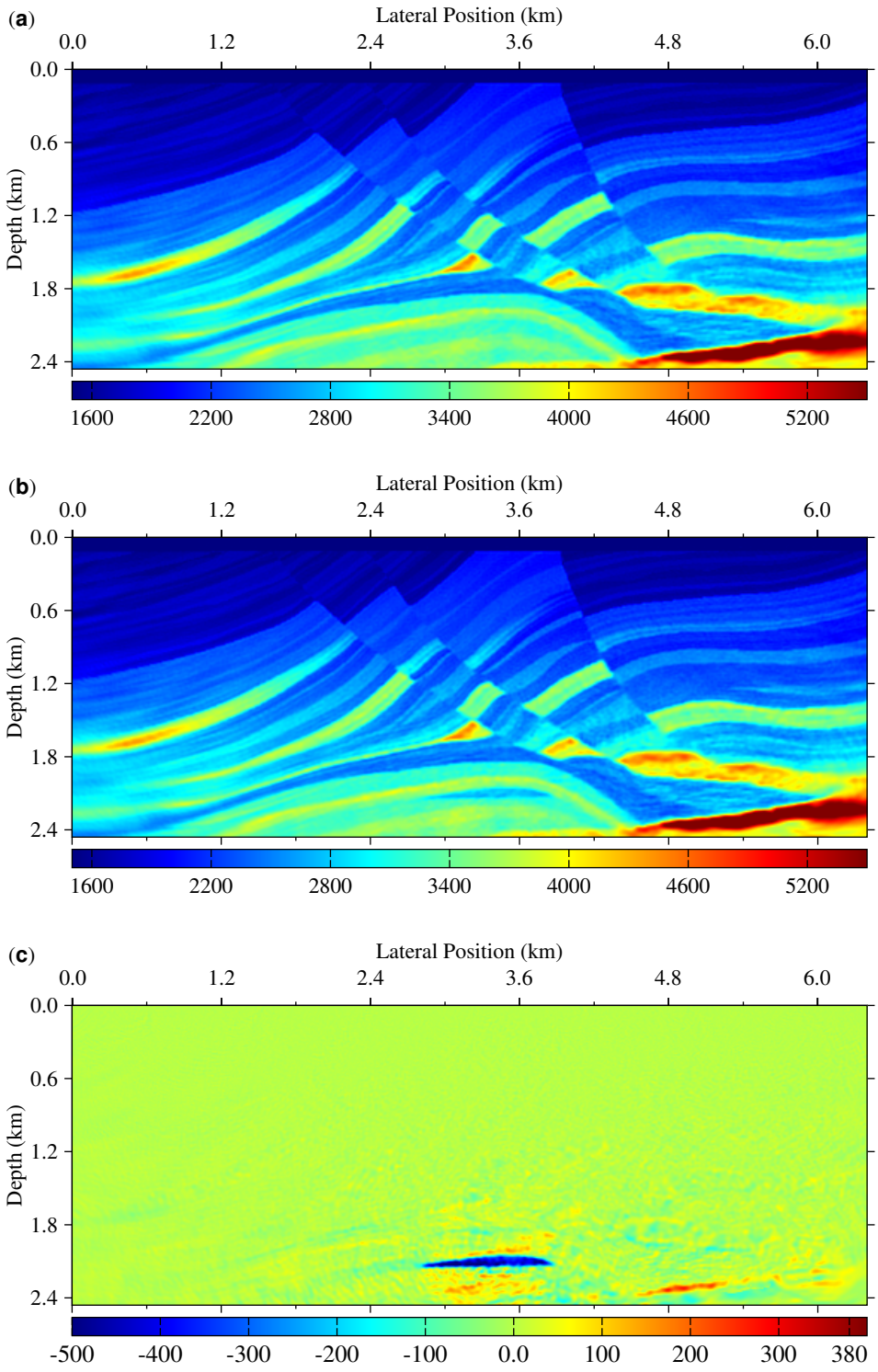


Figure 5.13: Case Two: inversion by the joint method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

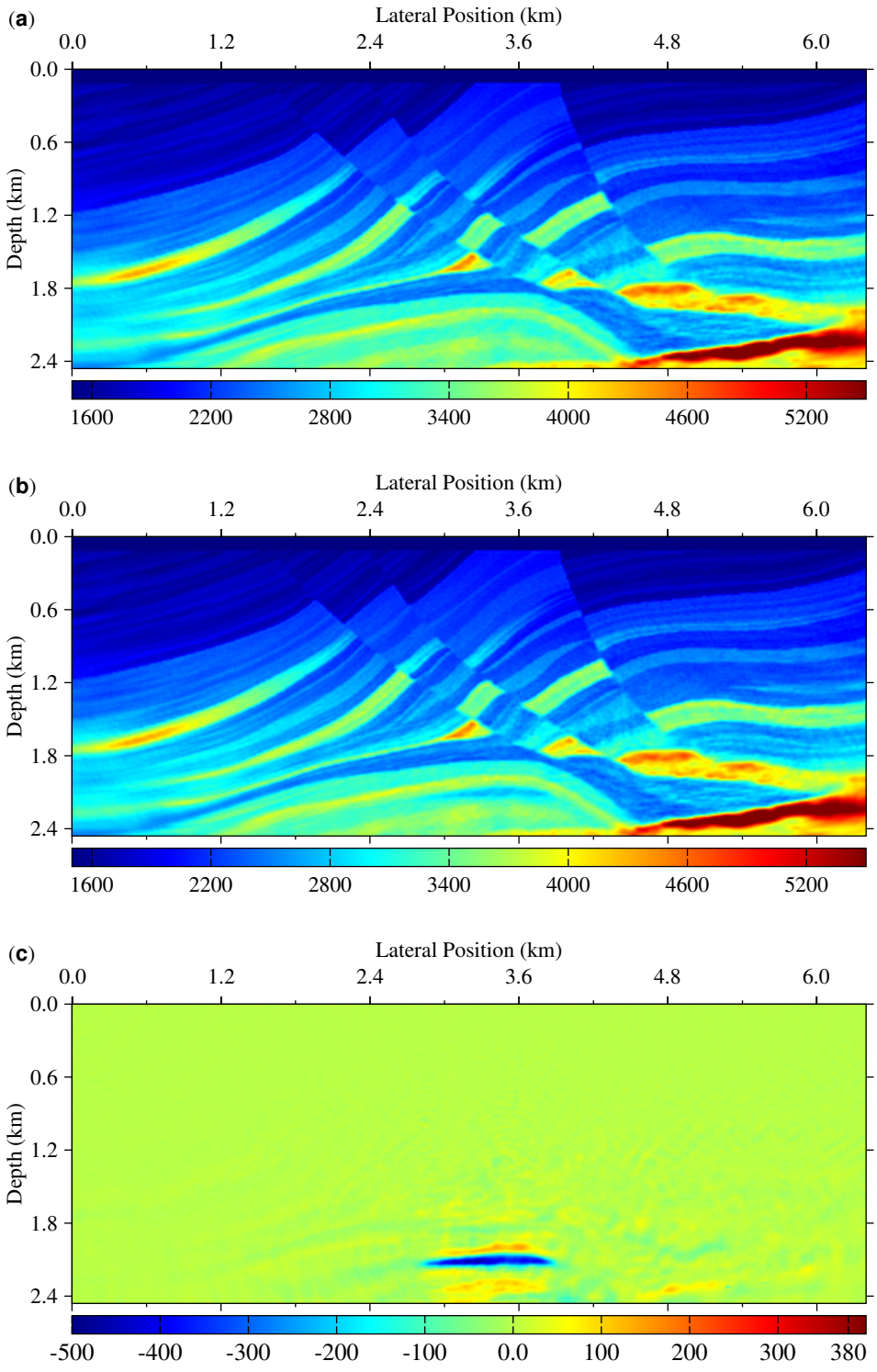


Figure 5.14: Case Two: inversion by the joint reparametrized method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

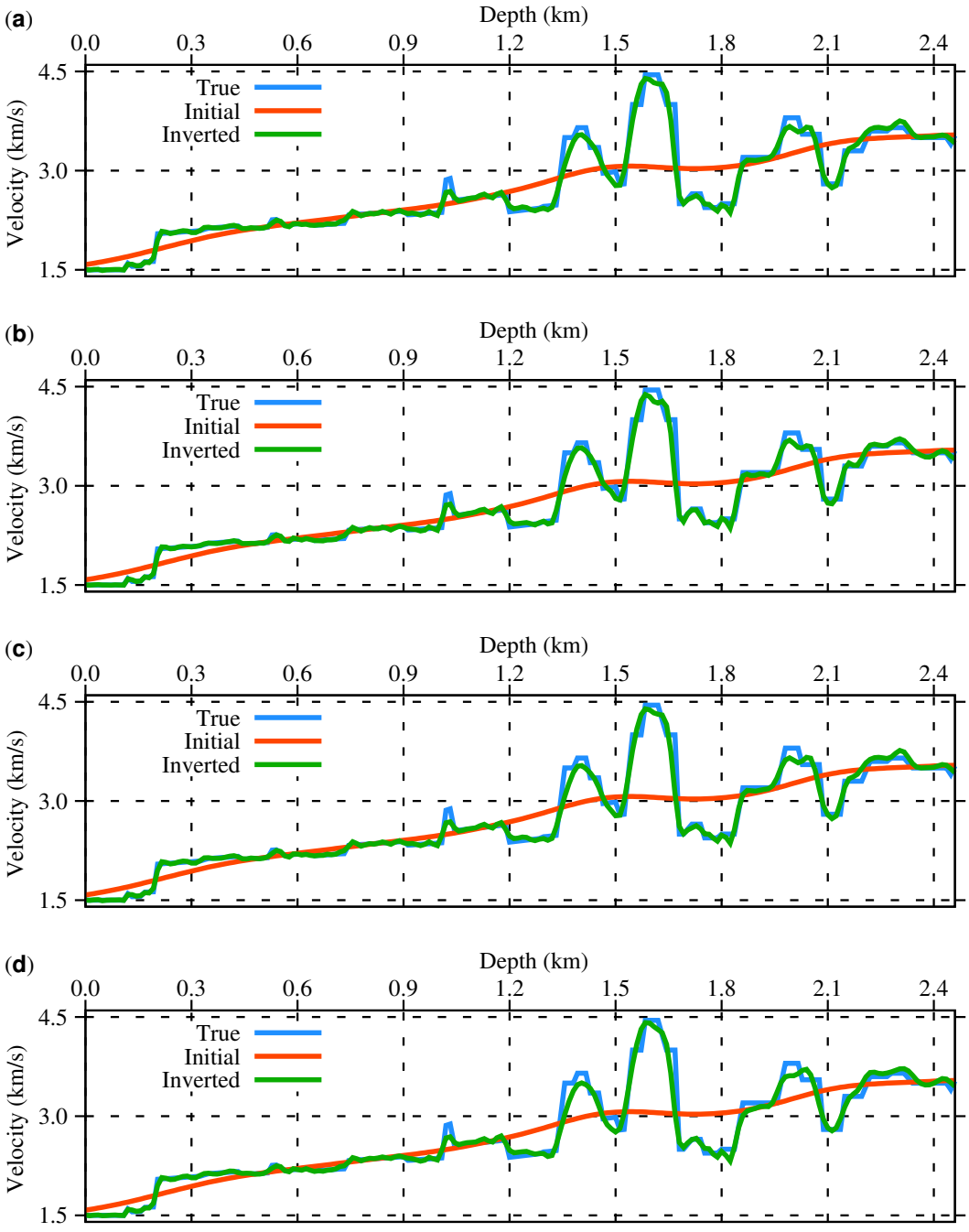


Figure 5.15: Case Two: comparison of vertical profiles of inverted monitor velocity with the true and initial velocity models (**a**) independent, (**b**) double difference, (**c**) joint, and (**d**) joint reparametrized.

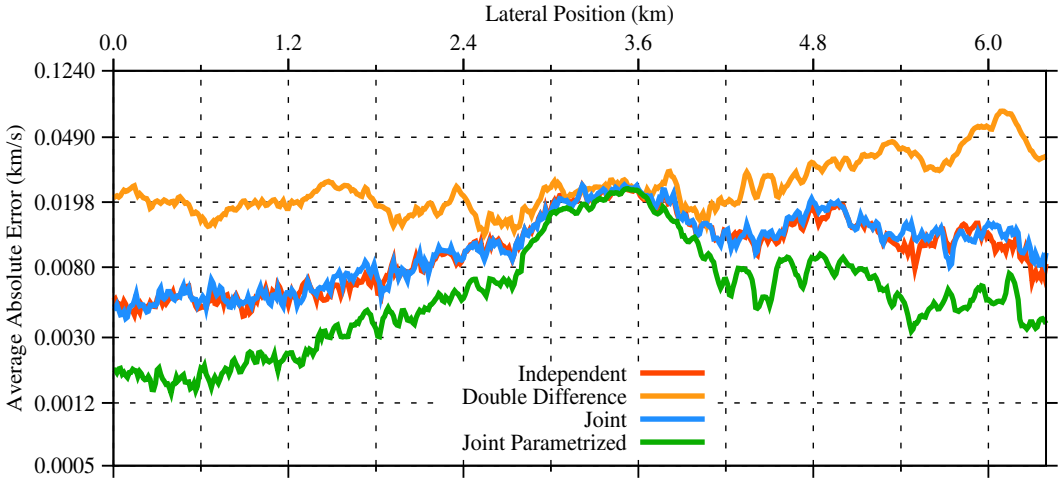


Figure 5.16: Case Two: error comparison for the non-repeatable case. The mean absolute errors are representatives of horizontal profile errors that were averaged vertically. The vertical axes are plotted in log-scale.

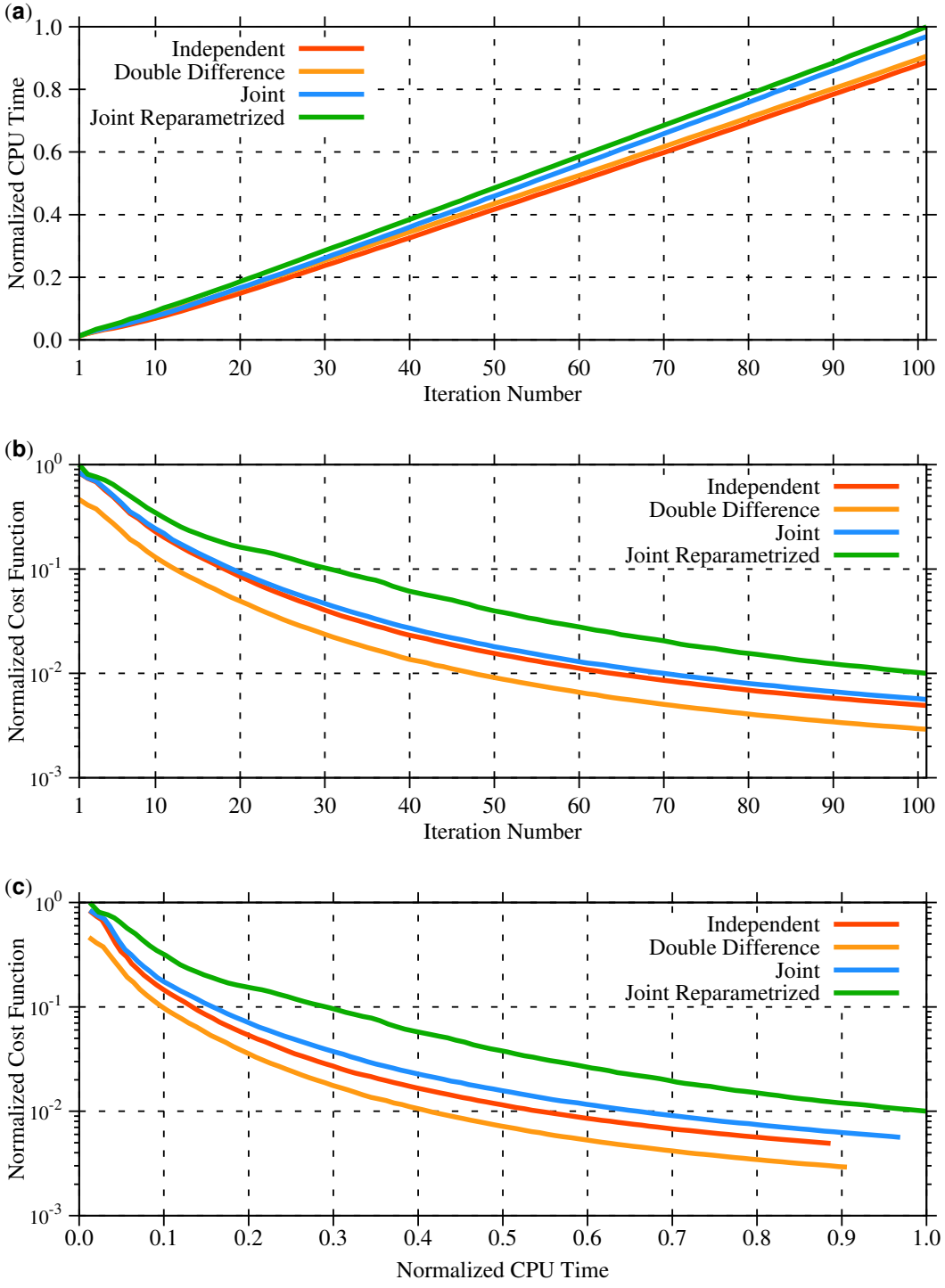


Figure 5.17: Case Two: (**a**) normalized computation time versus iteration number (**b**) normalized cost function versus iteration number, (**c**) normalized cost function versus normalized computation time. The vertical axes are plotted in log-scale.

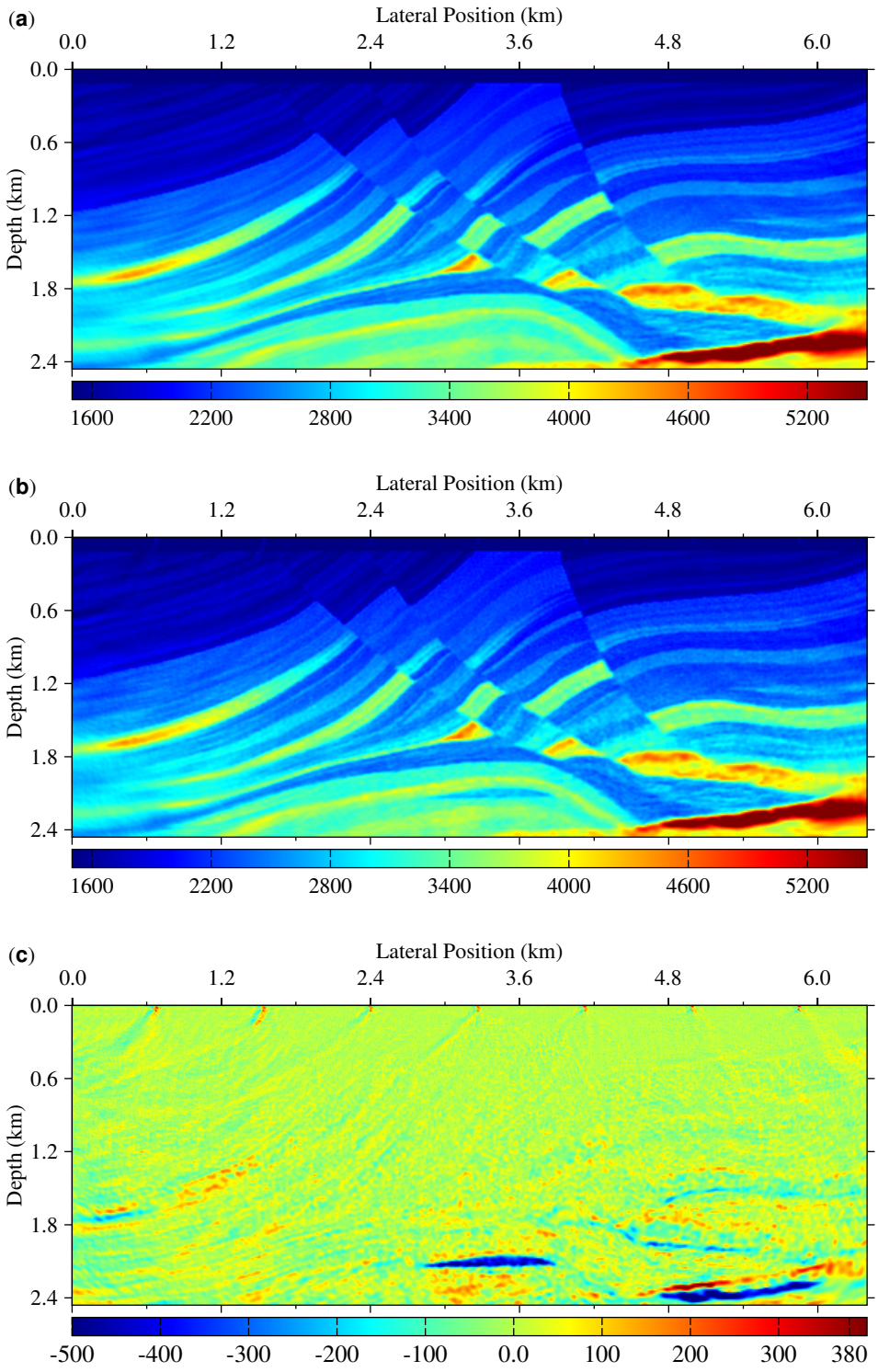


Figure 5.18: Case Three: inversion by the independent method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

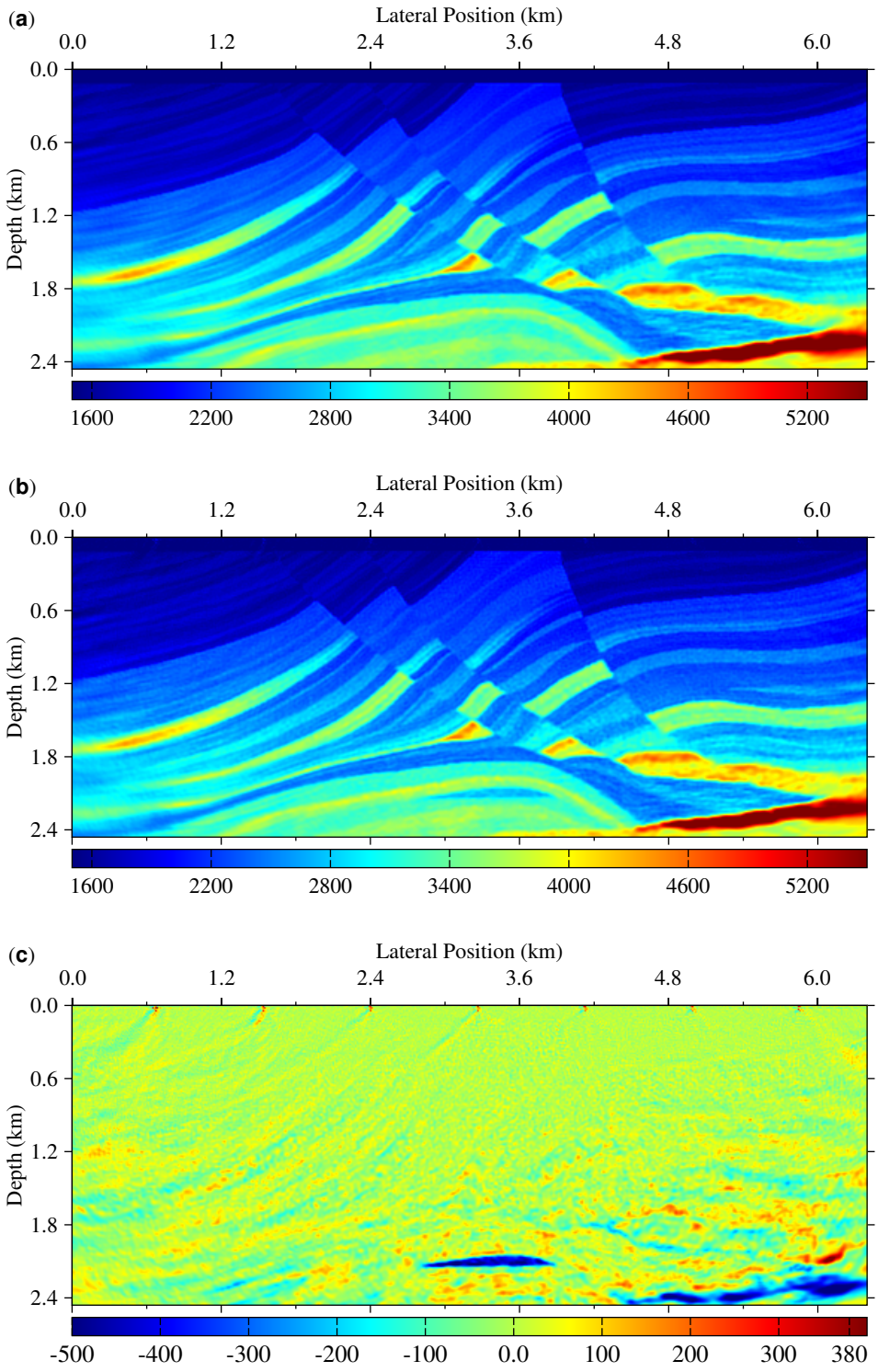


Figure 5.19: Case Three: inversion by the double difference method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

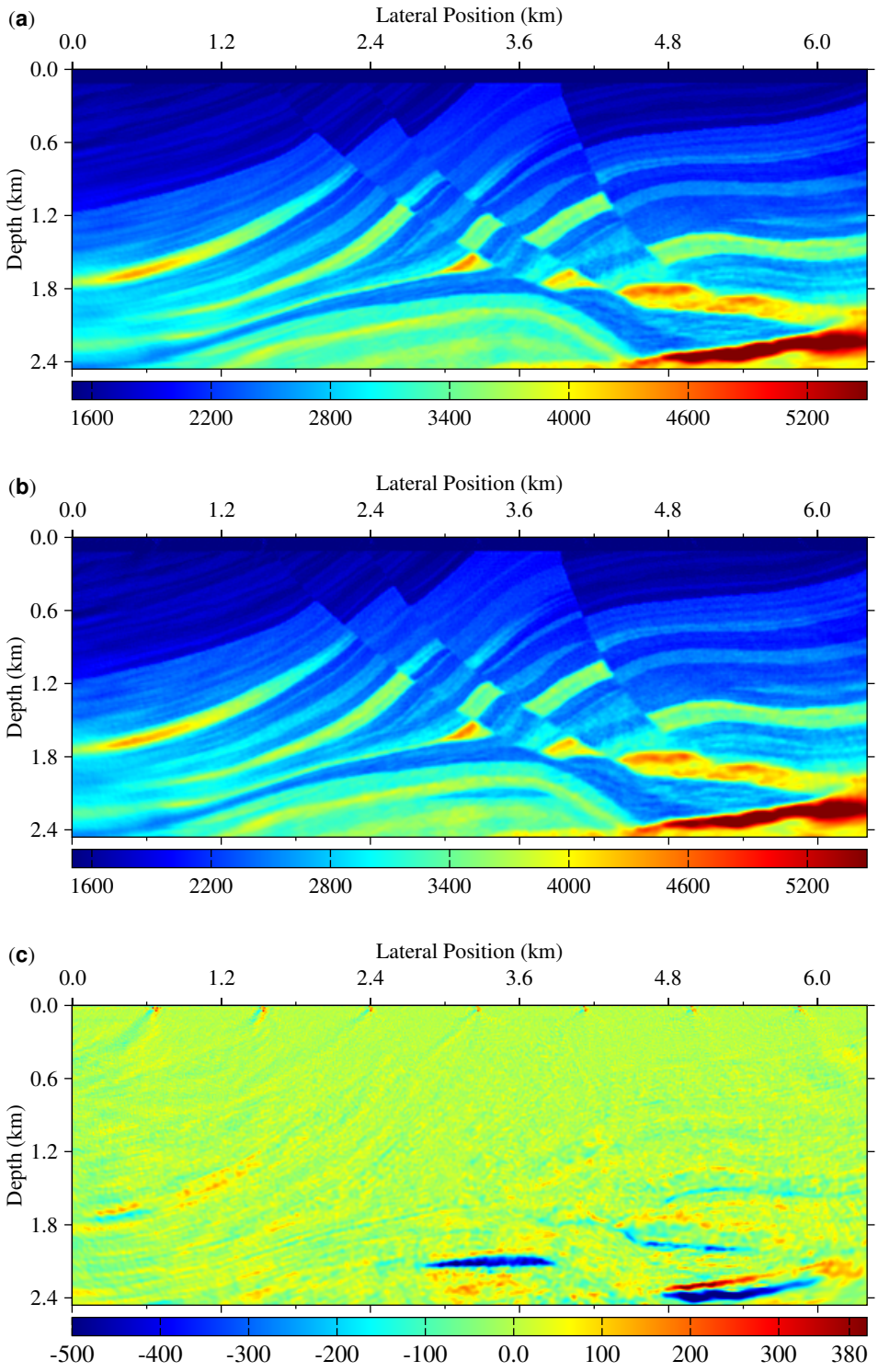


Figure 5.20: Case Three: inversion by the joint method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

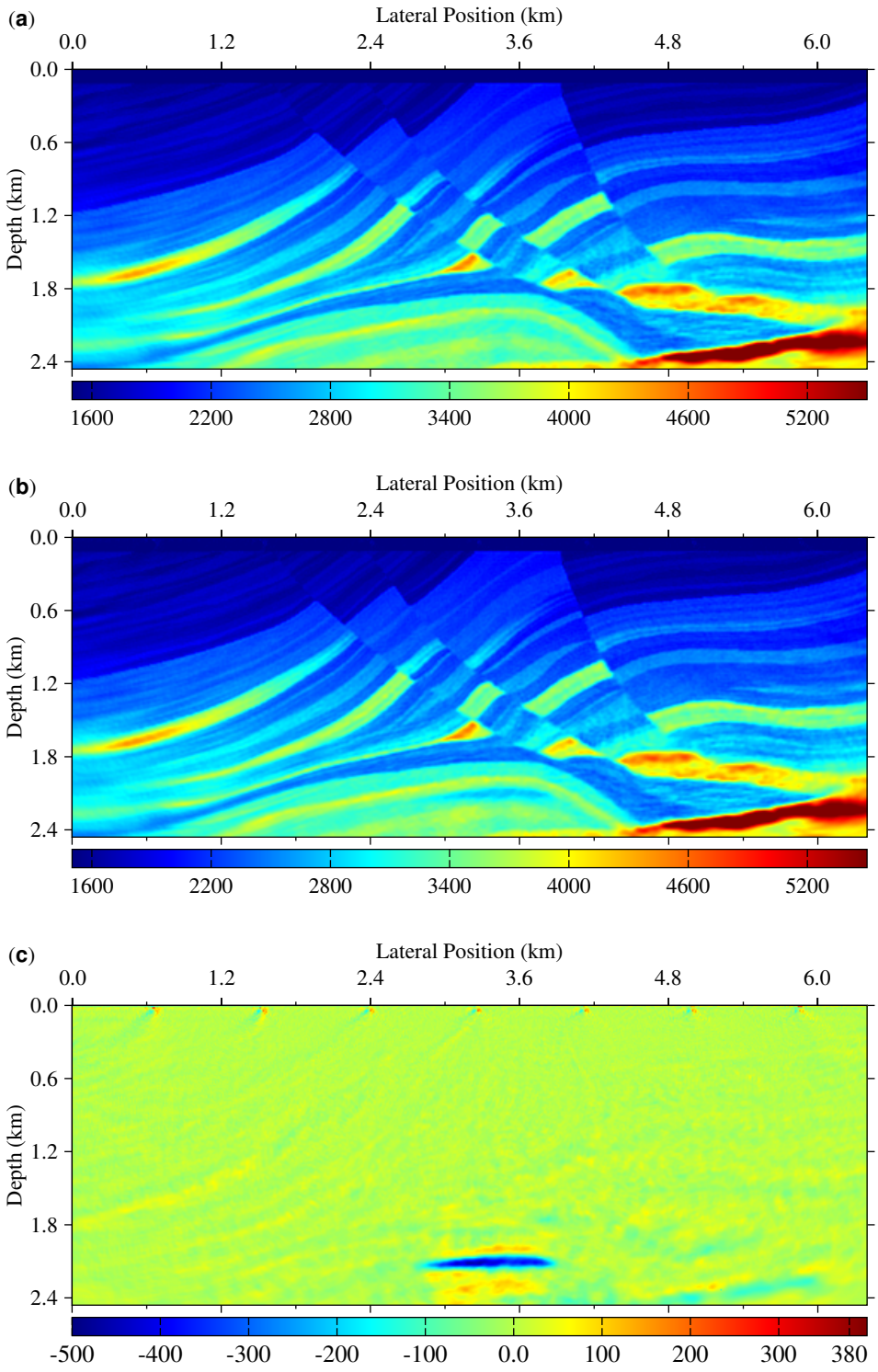


Figure 5.21: Case Three: inversion by the joint reparametrized method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

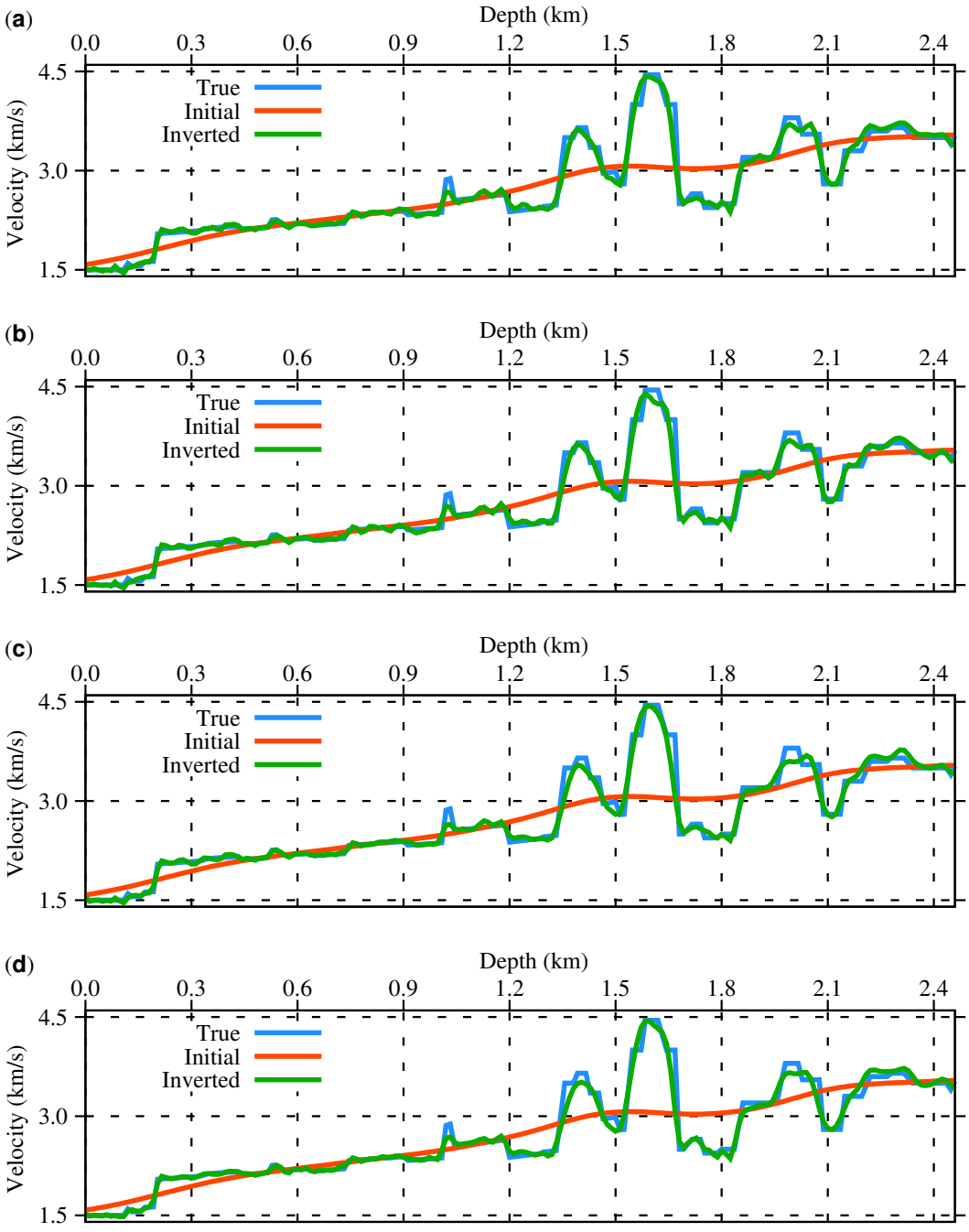


Figure 5.22: Case Three: comparison of vertical profiles of inverted monitor velocity with the true and initial velocity models (**a**) independent, (**b**) double difference, (**c**) joint, and (**d**) joint reparametrized.

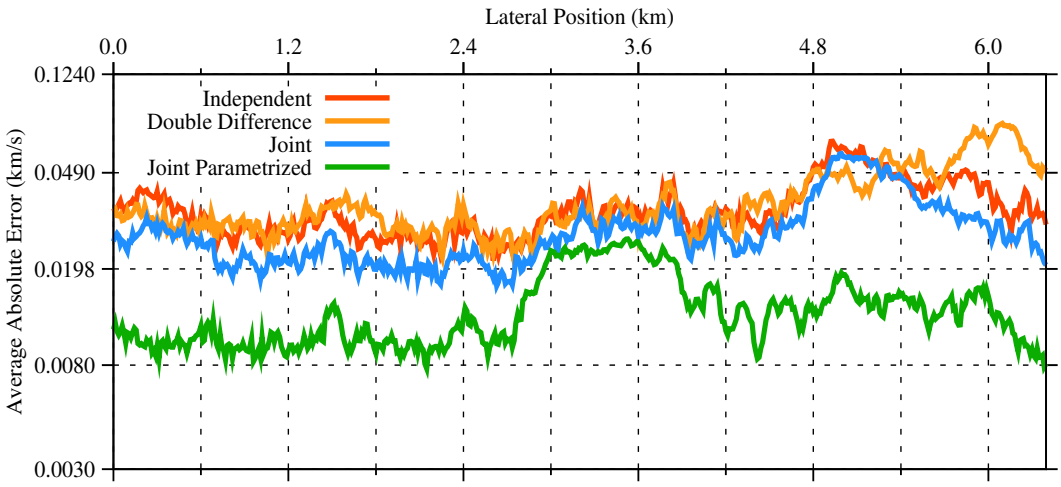


Figure 5.23: Case Three: error comparison for the non-repeatable case. The mean absolute errors are representatives of horizontal profile errors that were averaged vertically. The vertical axes are plotted in log-scale.

5.3.3 Discussions and Summary

The strength of the artifacts on time-lapse signatures that are obtained from the three synthetic examples is consistent with our expectations. In all cases, I expected artifacts attributed to the nonlinear optimization procedure in time-lapse settings. Different level of artifacts likely manifest on the velocity difference as the nature of each algorithm for time-lapse data set is not exactly the same. In other words, a small velocity difference between the baseline and monitor results in differencing artifacts that have strength comparable to the signal of interest. It is on this basis that the idea of jointly inverting baseline and monitor data sets is important. Besides, non-repeatability brings more artifacts as well that are responsible for intensifying differencing artifacts.

The four inversion strategies showed similar relative performance in both case 1 and case 2 synthetic scenarios, although with different level of artifacts as depicted in the error analysis provided in Figures 5.9 and Fig. 5.16. The independent and joint inversions showed a similar degree of performance. It is also clearly shown that the double difference inversion showed the strongest artifacts, whereas the joint reparametrized inversion demonstrated solution with the least energetic artifacts.

In the synthetic case 3, the artifacts in the velocity differences are stronger as compared to those in case 1 and case 2. It is expected for time-lapse data with non-repeatable surveys. Unlike case 1 and 2, the independent and joint inversion methods in case 3 showed results similar to those obtained via double difference inversion. In other words, both methods were

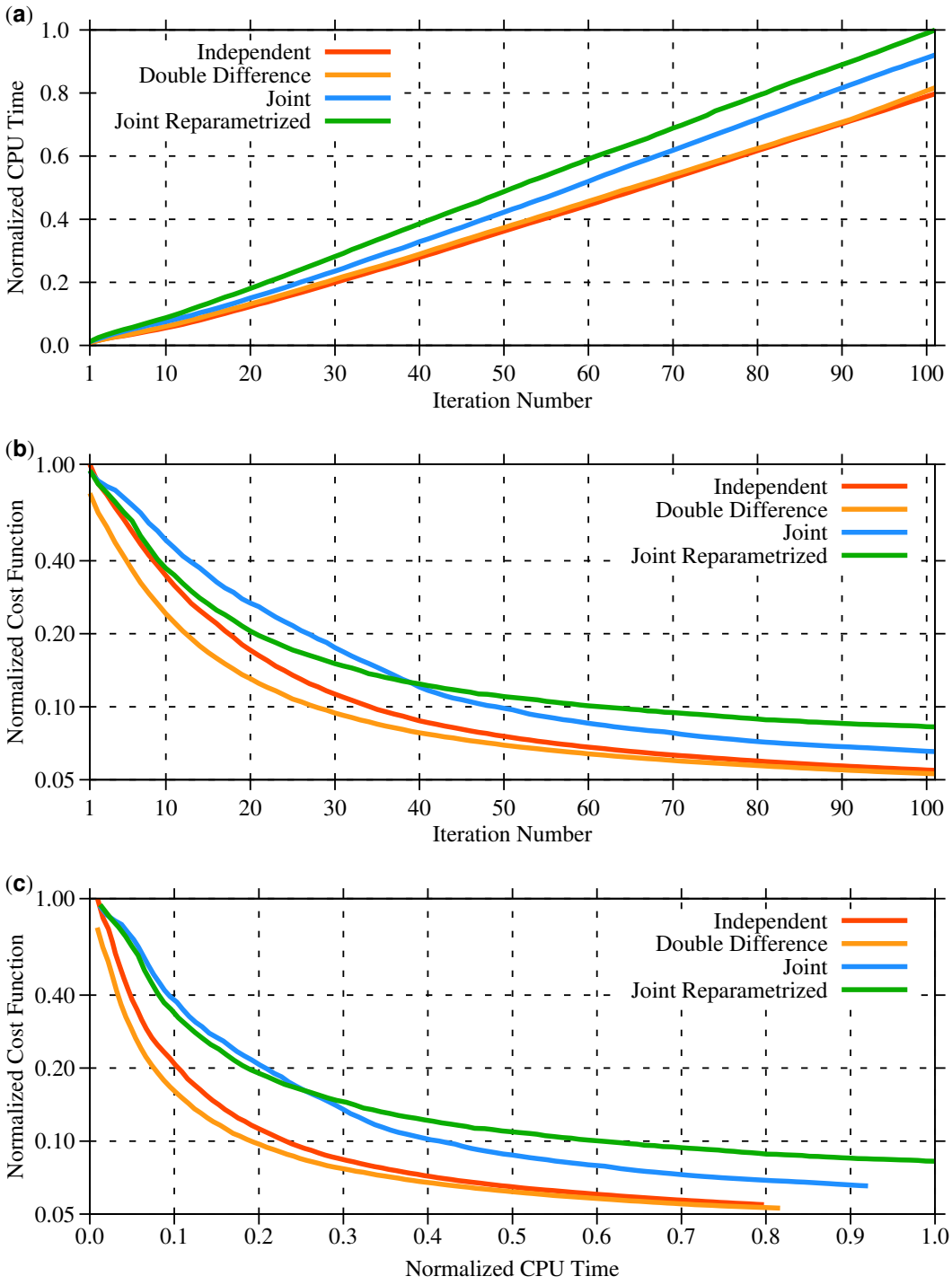


Figure 5.24: Case Three: (**a**) normalized computation time versus iteration number (**b**) normalized cost function versus iteration number, (**c**) normalized cost function versus normalized computation time. The vertical axes are plotted in log-scale.

unable to reduce non-repeatability artifacts as shown in the error analysis provided in Figure 5.23. The joint reparameterized inversion outperformed all three methods as depicted in the error analysis provided by Figure 5.23 and through visual inspection of velocity differences given in Figure 5.18 (**c**), Figure 5.19 (**c**), Figure 5.20 (**c**), and Figure 5.21 (**c**). The improved performance of joint reparameterized inversion in cases 1, 2 and 3 is due to the combined effect of the introduction of the same step length γ for the baseline and monitor parameters as well as the reparametrization technique. In other words, the method assumes that time-lapse seismic data sets are strongly correlated. It is also worth pointing out that the double difference method provided the best estimate of time-lapse signature on the target area. However, it yielded the strongest artifacts elsewhere. Strong artifacts were the consequence of time shifts in the data in conjunction with the use of different initial velocities for the baseline and monitor inversion which biases the monitor results. On the other hand, the joint reparameterized inversion provided a reasonable time-lapse signature at the target area and significantly reduced artifacts outside the target area.

Based on the results in Figure 5.10(**a**), 5.17(**a**) and 5.24(**a**) the computational time versus the iteration number of the four inversion methods show similar behavior. The normalized cost function versus iteration number in Figure 5.10 (**b**), 5.17 (**b**) and 5.24 (**b**) as well as the normalized cost function versus normalized CPU time given in Figure 5.10 (**c**), 5.17 (**c**) and 5.24 (**c**) clearly show how fast the cost function decays both in terms of iteration number and time, respectively. Note also that the vertical axes in those figures are plotted in log-scale which helps to unmagnified differences. The cost function of the double difference inversion decreased faster than the cost function of other three methods. It is expected as the data residual of the baseline inversion was subtracted in the definition of the cost function as shown in Equations 5.6 and 5.7. In the joint inversion, the cost function decreased slower than in the other three methods. The joint reparametrized inversion showed faster convergence than the joint inversion and slower convergence than the independent and double difference inversion methods. This outcome can be explained by the existence of a trade-off between data fitting and the strong correlation assumption that was made to mitigate the artifacts on the velocity differences.

In conclusion, I explored four time-lapse inversion strategies namely independent inversion, double difference inversion, joint inversion, and joint reparametrized inversion with three possible synthetic scenarios. In the given tests, I showed that the Joint Reparametrized Time-Lapse Full Waveform Inversion (JRTL FWI) provided improved time-lapse velocity differences in the three synthetic scenarios than the other three methods. JRTL FWI was able to minimize artifacts due to non-repeatability.

The application of JRTL FWI is not limited to 2D time-lapse inversion; it is extendable to 4D inversion with more than two surveys, and the technique can also be adopted to

elastic time-lapse inversion. It is worth pointing out that the proposed inversion technique takes advantage of the modified L-BFGS algorithm that uses the history of the difference of gradients. Although the idea of joint inversion approach can work in any optimization methods, joint reparametrized inversion may be challenging to implement via methods other than L-BFGS.

In the next chapter, one typical example in time-lapse synthetic model based on CO_2 sequestration monitoring is given. The time-lapse inversion strategies explored in this chapter are implemented to test this model in detail with non-repeatability experiment as well as data decimation that mimic non-repeatability.

CHAPTER 6

Time-Lapse FWI: CO_2 Sequestration Monitoring

In this chapter, a typical CO_2 sequestration synthetic velocity models are generated taking into account realistic reservoir conditions with Gassmann's fluid substitution. These models are also explored with the time-lapse inversion strategies that are covered in Chapter 5. In addition to geometric acquisition difference, two more examples with data decimation are added to mimic non-repeatability.

6.1 Introduction

In the last few years, a concern on climate change, global warming, in particular, has grown. The impacts of global warming are being manifested in increasing global average temperature, rising sea levels, and change in precipitation with consequences for low-lying inhabited areas, agriculture, biodiversity, and human health (IPPC, 2001). Human activity such as combustion of fossil fuels and release of greenhouse gases into the atmosphere is believed to be a major contributor to global warming. Carbon dioxide (CO_2) is the principal component of greenhouse gases. A large amount of CO_2 emission comes from coal and gas-driven power plants and automotive transportation. In addition, development of liquefied natural gas from gas reservoirs requires the management of large amount of CO_2 released as a byproduct in the process (Lumley, 2001).

According to International Energy Agency world outlook 2004 (EIA, 2004), 85% of the world's energy comes from the burning of fossil fuels and the world demand is increasing.

Unless competitive alternative energy sources are developed, the burning of fossil fuels and CO_2 emission will continue. Developing technologies for reducing the greenhouse gas emission is, therefore, the primary solution for such complex problem. Thus, a large number of small to large-scale researches on CO_2 reduction mechanisms are underway around the world both in academia and industry level.

Carbon Capture and Storage (CCS) is one of the emerging set of technologies to address the issue. It involves CO_2 capture, compression, transportation and sequestration or storage. Before the introduction of CCS technology for environmental purposes, the first CO_2 sequestration project was started commercially for enhanced oil and recovery (EOR) in 1972 at Kelly-Snider oils field in West Texas (Langston et al., 1988). Following this success, many CO_2 related projects are growing, and the oil and gas industry is continuously developing various CO_2 capturing, transporting and injecting technologies. It is also being implemented in Enhanced Coal-bed Methane Recovery (ECBM), CO_2 sequestration in depleted oil and gas Fields, and deep saline aquifers.

Technical and economic assessments suggest that over the coming century, CCS may contribute up to 20% of CO_2-emission reductions, equivalent to reductions expected from efficiency improvements and large-scale deployment of renewable energy resources (IPPC, 2005). The two vital processes in CCS include Carbon Capture, and then Carbon sequestration or storage.

6.1.1 CO_2 Capture

With over 60% of worldwide emissions coming from point sources that are potentially amenable to CO_2 capture, the prospects for CCS to significantly reduce CO_2 emissions are great (IPPC, 2005). Power plants and other industrial facilities are the stationary sources which emit a huge amount of flue gas. The flue gas contains mostly nitrogen and only from 5 to 15% of CO_2. Thus, separating the CO_2 is very expensive and needs sophisticated technologies.

The CO_2 can be captured before combustion or after combustion of the fossil fuel. The pre-combustion one uses gasifying chemicals to extract H_2 and releases the CO_2. The post-combustion case the fossil fuel is combusted with oxygen to produce CO_2 and water in which the CO_2 can be easily separated. Although the cost is high, the post-combustion capturing is a demonstrated technology that can be easily be applied. On the other hand, the alternatives have advantages over the post-combustion but not well established and need development (Simbeck, 2004). The captured and compressed CO_2 can be transported by a pipeline or trucks to depending on the geographical location of the sequestration sites, where practical and economic issues are optimal.

6.1.2 CO_2 Storage

For CO_2 to be stored in geological formations, various regulatory and legal requirements will have to be satisfied; the storage must be safe and secure. Therefore, potential sequestration sites must undergo appropriate site characterization to ensure that the site can safely and securely store CO_2. The efficiency of the reservoir depends on the reservoir's storage capacity, stability, and risk of leakage (Benson and Doughty, 2006; Davis et al., 2003; Holloway, 2001; Rochelle et al., 2004; Torp and Gale, 2004)

A number of geological formations can be used to store CO_2 into deep underground into porous rock, such as sandstone, shale, dolomite, basalt, or deep coal seams. Suitable formations for CO_2 sequestration are located under one or more layers of cap-rock, which trap the CO_2 and prevent upward migration. These sites have to be monitored rigorously to ensure that the CO_2 remains permanently underground.

Thus, understanding of the geologic, hydrologic, geomechanical, and geochemical processes controlling the fate and migration of CO_2 in the subsurface is necessary to provide a base for developing methods to characterize storage sites, to select sites with minimal leakage risk, and ensure reservoir integrity. The three principal trapping mechanisms, which determine the short or long term fate of CO_2 in a geological reservoir, are hydrodynamic, solubility and mineral trapping (Hitchom, 1984). It is important to understand the behavior of these processes, as they may be either faster or slower than the time span of injection.

A suitable storage formation should be deeper in such a way that CO_2 density is high enough to allow efficient pore filling and decrease the buoyancy difference, have a thick cap rock to secure seal, and sufficient porosity and permeability to permit a high flow rate. In this regard, large sedimentary basins are best suited, because they have tremendous pore volume and connectivity and they are widely distributed (Bachu et al., 2003). Vast formations of sedimentary rocks with various textures and compositions provide both the volume to sequester the CO_2 and the seals to trap it underground. On the other hand, basaltic rocks are also proposed for storing CO_2. This kind of reservoir has carbonate mineral precipitation property in rock-fluid chemical reaction (McGrail et al., 2006; Rogers et al., 2006; Matter et al., 2007; Oelkers et al., 2008; Matter et al., 2007; Schaef et al., 2010). If this reaction occurs, changes in the elastic properties of basalts will be the combination of fluid substitution of CO_2 with water and carbonate precipitation. However, seismic monitoring methods may not be applicable in a basaltic reservoir because of the strong seismic scattering characteristic of the sharp impedance contrast between a flow of basalts and sedimentary inter-beds (Pujol et al., 1989).

There are about four possible geological settings that CO_2 can be sequestered either for enhanced recovery or permanent storage.

CO_2 Sequestration in Deep Saline Aquifers

Saline aquifers have the lowest storage efficiency of the geological sinks suitable for disposal of CO_2, with estimates as low as 2% of the pore volume(Van der Meer, 1998). It is due to unstable displacement fronts that occurs when the fluid is displaced by much lower viscosity CO_2. Also, the effect of buoyancy results in the CO_2 filling only the uppermost part of the aquifer at significant distances away from the injection well. However, saline aquifers appear to be the most economically attractive option currently and have the largest ultimate storage capacity (Rigg and Bradshaw, 2001). Saline aquifer storage will pose one of the biggest challenges for monitoring due to the lack of pre-existing data for characterization of the reservoir. The petrophysical data required for modeling injection, migration, and trapping of the CO_2 in the subsurface may be lacking if there is limited baseline information initially available from wells logs and geophysical surveys.

CO_2 Sequestration in Enhanced Oil Recovery

Oil is a non-renewable resource; therefore, the life-cycle of an oil field is limited. The production phases of an oil field can be categorized primary, secondary and tertiary recovery (Andrei et al., 2010). CO_2 enhanced oil recovery is a tertiary recovery phase of an oil field. The injection of carbon dioxide (CO_2) into oil reservoirs for EOR has received considerable attention and has been in practice in oil and gas industry in the last four decades.

The storage efficiency of CO_2-EOR is estimated to be between 13-68% (Holt et al., 1995). Where there is no government economic budget to reduce CO_2 emissions, it also provides an economic incentive for storage. As many fields undergoing water-flood approach the limit of recovery, $25 - 50\%$ of the original oil is left behind. Based on experience with CO_2 flooding worldwide, a considerable part of residual and trapped oil can be recovered with miscible CO_2 injection under appropriate conditions(Holt et al., 1995).

Enhanced Coal-bed Methane Recovery

Injection into coal seams that are too deep to mine is another possibility for geological storage, which is particularly attractive if it can be coupled with enhanced coal-bed methane recovery (ECBM). Injected CO_2 improves recovery by directly displacing methane that is adsorbed onto the coal. The CO_2 will remain sequestered as long as the coal is never mined or substantially depressurized (Gale and Freund, 2001). Coal beds provide the greatest storage density of CO_2 to depths of $600m$ and are still competitive at greater depths. A key factor in favor of coal-bed sequestration is that in many cases coal seams are nearby to coal-fired power stations, resulting in low transportation costs from source to sink. Large

scale ECBM operations are now underway in the western United States and Alberta, Canada. One of the difficulties for ECBM is finding coal-seams that are too deep to mine but are still permeable enough to inject. For instance, low production rates and a lack of infrastructure restrict the applicability of ECBM methods in Australia (Rigg and Bradshaw, 2001)

CO_2 Storage in Depleted Oil and Gas Fields

The storage efficiency in depleted oil and gas fields is estimated to be much higher than in saline aquifers, with approximate storage capacity equal to a number of hydrocarbons that have been extracted (Holloway, 1997). Depleted fields have a lot of advantages for sequestration because these areas are usually extensively studied during exploration and production. Since depleted fields are already a naturally proven hydrocarbon storage formation, it can also be a permanent storage for CO_2 with much less concern. Also, the availability of vast amount of baseline geophysical data, knowledge acquired from the interpretation of the data, and the existence of production wells can reduce risk and uncertainty.

Depleted oil and gas reservoirs are estimated to have the capacity to sequester between 675 and 900 billion tons of carbon, saline aquifers between 1000 and 10,000 Gt, and deep unmineable coal beds between 3 and 200 Gt (IPPC, 2005). Sequestration capacity estimates for saline aquifers and coal beds are highly uncertain, although, in the past several years, there has been some progress in developing standard methods for capacity estimation and improving regional estimates (Bachu et al., 2007).

6.2 CO_2 Sequestration Monitoring

CO_2 sequestration at any geological setting has to be monitored to track CO_2-plume migration, manage the injection process, and grant public safety from possible CO_2 leakage. Therefore, over a lifetime of CO_2 sequestration, a combination of various monitoring techniques such as seismic monitoring, electromagnetic monitoring, gravity monitoring, and geochemical monitoring can be applied. Seismic monitoring can detect changes in elastic parameters. Electromagnetic monitoring can detect the decrease in electrical conductivity when CO_2 is present. Gravity measurements are sensitive to bulk-rock density when CO_2 is present.

For example, at Sleipner Field (Norway), a combination of time-lapse 2D and 3D imaging has been used to track the migration of the injected CO_2 in the Utsira brine formation with great success (Arts et al., 2005). Recently, gravity measurements were used to estimate the in situ density of CO_2 at Sleipner (Nooner et al., 2007). At Weyburn (Canada), a comprehensive program that included time-lapse 3D seismic imaging, geochemical sampling, and soil gas

surveys was used as a multifaceted approach to demonstrate effective containment (Wilson and Monea, 2005; Njiekak et al., 2013; Martínez and Schmitt, 2013). The In Salah (Algeria) Project plans to install a permanent 3D seismic monitoring array, sample soil gases and introduce tracers for tracking CO_2 breakthrough into the gas reservoir. In addition to these commercial-scale projects, monitoring methods have been tested on a smaller scale at pilot test sites (Watson et al., 2004; Hovorka et al., 2006; Sakai, 2006). Surface to borehole seismic imaging (VSP), cross-well seismic, cross-well EM, well logs, brine and gas composition sampling and analysis, soil-gas sampling, and groundwater sampling have been used to monitor the fate and migration of CO_2 in the subsurface (Watson et al., 2004; Perkins et al., 2005; Hovorka et al., 2006; Sakai, 2006; Benson and Doughty, 2006).

Regarding the technological advancement and capability of monitoring, seismic methods are the most commonly used CO_2 monitoring techniques, which include time-lapse cross-well imaging (between two or more wells)(Spetzler et al., 2008), time-lapse 3D vertical seismic profiling (O'Brien et al., 2004), coda-wave interferometry analysis of time-lapse VSP data (Zhou et al., 2010), and 4D seismic monitoring. These methods have been used with great success for CO_2 sequestration monitoring. Under the condition that feasibility studies on the application of 4D seismic for a certain CO_2 sequestration site, 4D seismic monitoring is an adequate to image CO_2 plume migration (Fabriol et al., 2011). As mentioned above, Sleipner Field is one of the best-known instances, where 4D seismic monitoring is in practice (Chadwick et al., 2008). To get more insight, it is discussed in detail in the next section.

6.3 Sleipner Field

Sleipner Field is located in the Norwegian sector of the North Sea, about $250km$ from the west coast of Norway. StatoOil and its Sleipner partners have been producing natural gas but with the excess of CO_2 (4 to 9.5%) by the European natural gas distribution pipeline standards. To meet the European gas pipeline specification limit of CO_2 concentration, which is below 2.5%, CO_2 was stripped from the natural gas stream by amine process and injected into the Utsira formation, major saline aquifer of late Cenozoic age.

The Utsira Sand comprises a restricted deposit of Mio-Pliocen age extending for more than 400 km from north to south and 40 km and 100 km from east to west. Internally, it consists of stacked overlapping fan-lobes and separated by thin intra-reservoir mudstone or saly horizon. The top Utsira Sand surface varies quite smoothly in the depth range 550 m - 1500 m, and around 800 m - 900 m near Sleipner. Isopachs of the reservoir sand define two main depocentres, one in the south, around Sleipner, where thickness locally exceed 300 m, and another 200 km to the south with thickness approaching 200 m (Chadwick et al., 2004). The cap-rock succession overlying the Utsira reservoir is several hundred meters thick.

CO_2 was being injected 1012 m deep and about 200 m below reservoir top at a rate approaching 1 million tons per year. This makes the Sleipner field on of the world's first industrial-scale CO_2 storage operation (Zweigel et al., 2001; Chadwick et al., 2004, 2008). To track plume migration, demonstrate containment within the storage reservoir and provide quantitative information to understand the flow proses, a compressive time-lapse surface seismic program has been carried out, with 3D surveys in 1994,1999, 2001, 2004, 2006, and 2008 (Chadwick et al., 2010).

Quantitative interpretation of the time-lapse data with various techniques has been done. Well-logs, a two-way travel time shift, flow simulation and history-matching, AVO analysis, model-based post-stack and pre-stack stratigraphic impedance inversion were exploited to characterize the dynamics of the Utsira sand reservoir. The detailed seismic analysis has been used by Chadwick et al. (2008). The CO_2 is observed to rise vertically from the injection point but appears to be delayed by intermediate shale layers. The seismic results have also been used to verify the amount of CO_2 in storage. Arts et al. (2004b) estimates that a volume of less than 4000 m (2800 t) of CO_2 can be detected using seismic analysis. Well-logs reveal several thin mudstones within the Utsira sand with thickness 5m and above (Chadwick et al., 2010). And, time-lapse data show a progressive attenuation of seismic events beneath the CO_2 plume which increases with time (Clochard et al., 2009). The pre-stack inversion results were compared to those of post-stack inversion (Clochard et al., 2009); pre-stack inversion is found to be better in characterizing a part of the reservoir.

Little geochemical activity has been detected over the duration of the project. About 80-85% of the CO_2 is thought to be hydrodynamically trapped. 15-20% solubility trapped and less than 1% mineral trapped at present (Johnson et al., 2004).

Chadwick et al. (2010) noted that application of Full waveform inversion has a potential to improve the resolution of parameters of interest. Queißer and Singh (2013) showed the application of Full waveform inversion to invert P-wave velocity. The P-wave velocity was related to gas saturation by Gassmann's fluid substitution model. They found that a considerable fraction of injected CO_2 (greater than 24% in dissolved form in 2006) and they suggest that saline aquifers could be a natural reservoir for CO_2 storage.

Sleipner Field Saline Aquifer CO_2 Storage (SACS) is located from 800 to 1000 m deep and has thickness of between 200 m and 300 m (Zweigel et al., 2001). Since this project is very well explored and documented, I chose to use the available information in the literature to generate 2D velocity models making use of the Biot-Gassmann's fluid substitution steps. Usually, the Biot-Gassmann's equation (Gassmann, 1951; Biot, 1956a) along with Batzle-Wang equations (Batzle and Wang, 1992) are used to estimate unknown fluid parameters from known reservoir conditions.

Parameter	Value	Parameter	Value
Porosity (ϕ)	0.37	Density of the solid mineral (ρ_s)	2650 kg/m^3
Pressure (P)	10 Mpa	Bulk modulus of the solid mineral (K_s)	36.9 Gpa
Temperature (T)	37^{o}C	Initial P-wave velocity (V_{Ip})	2050 m/s
		Initial S-wave velocity (V_{Is})	643 m/s
		Initial density (ρ_I)	2072 kg/m^3
		Density of brine (ρ_{brine})	1090 kg/m^3
		Bulk modulus of brine (K_{brine})	2.3 Gpa
		Density of CO_2 (ρ_{co2})	650 kg/m^3
		Bulk modulus of CO_2 (K_{co2})	0.0675 Gpa

Table 6.1: Reservoir parameters input to the fluid substitution model. Note that the temperature and pressure are assumed to be constant and were not utilized in the substitution process.

I adopted the Sleipner field CO_2 reservoir conditions given in Table 6.1 from various authors (Arts et al., 2004a; Zweigel et al., 2001; Chadwick et al., 2004, 2008; Rubino et al., 2011) to build 2D baseline and monitor P-wave velocity models. To do so, the remaining parameters which helps to determine the P-wave velocity after fluid substitution are described in the following steps.

Gassmann's Fluid Substitution

Gassmann's fluid substitution theory is widely used for predicting seismic velocity changes because of different fluid saturations in reservoirs (Batzle and Wang, 1992; Mavko and Mukerji, 1995; Berryman, 1999; Batzle et al., 2001; Smith et al., 2003; Han and Batzle, 2004; Schmitt, 2015). Gassmann (1951) derived the substitution model under the following assumptions: the system is closed (undrained), macroscopically isotropic, and homogeneous with connected pore network. The equations of the Gassmann's theory are given by,

$$\frac{K_{sat}}{K_s - K_{sat}} = \frac{K_{frame}}{K_s - K_{frame}} + \frac{K_f}{\phi(K_s - K_f)}, \tag{6.1}$$

$$\mu_{sat} = \mu_{frame}, \tag{6.2}$$

and

$$\rho_{sat} = \rho_{frame} + \phi\rho_f, \tag{6.3}$$

where K, μ and ρ are bulk modulus, shear modulus, and density, respectively. The subscripts sat, $frame$, s, and f represents saturated, dry rock frame, solid mineral grain, and fluid, respectively. Shear modulus (μ) is the resistance to shear or rigidity; defined as the ratio of extensional stress to extensional strain. Bulk modulus (K) is the resistance to compression or incompressibility; defined as the ratio of hydrostatic stress to volumetric strain.

Bulk modulus is more sensitive to water saturation. The bulk-volume deformation produced by a passing seismic wave results in a pore-volume change and causes a pressure increase in pore fluid (water). This pressure increase stiffens the rock frame and causes an increase in bulk modulus. Shear deformation, however, does not produce a pore-volume change, and consequently, different fluids do not affect shear modulus. Equation (6.2) shows this fact. Therefore, any fluid-saturation effect should correlate mainly to a change in bulk modulus. The following are assumptions in Gassmann's fluid substitution model:

Isotropic and homogeneous rocks

This assumption implies that the rock has only one type of mineral with constant matrix bulk and shear modulus. However, the real reservoir rocks are composed of several mineral types with their modulus values. In addition, the pore space may also be filled with a mixture of fluids.

To relatively relax this assumption, Hill (1952) proposed a simpler method, which takes into account the average matrix bulk modulus of the various minerals present. It can be done by taking the arithmetic average of the lower bound matrix bulk moduli proposed by Reuss (1929) and the upper bound matrix bulk moduli proposed Voigt (1889).

For N number of mineral constituents, the equation for the Reuss's average, which represents the softest rock, is given by

$$K_{Reuss} = \sum_{i=1}^{N} f_i K_i, \tag{6.4}$$

$$\mu_{Reuss} = \sum_{i=1}^{N} f_i \mu_i \tag{6.5}$$

and the Voigt's average, which represents the stiffest rock, is given by

$$\frac{1}{K_{Voigt}} = \sum_{i=1}^{N} f_i K_i, \tag{6.6}$$

$$\frac{1}{\mu_{Voigt}} = \sum_{i=1}^{N} f_i \frac{1}{\mu_i}, \tag{6.7}$$

where K_i and μ_i refers to the bulk and shear moduli of the i^{th} mineral constituent respectively and f_i is the corresponding volume fraction of mineral type. The arithmetic average of the two bounds is given by

$$K_{RVH} = \frac{1}{2}(K_{Reuss} + K_{Voigt}) \tag{6.8}$$

$$\mu_{RVH} = \frac{1}{2}(\mu_{Reuss} + \mu_{Voigt}). \tag{6.9}$$

Another method for calculating the lower bound and the upper bound is derived by Hashin and Shtrikman (1963). For two constituent minerals, these bounds are given by

$$K^{HS\pm} = K_1 + \frac{f_2}{(K_2 - K_1)^{-1} + f_1(K_1 + \frac{4}{3}\mu_1)^{-1}}, \tag{6.10}$$

$$\mu^{HS\pm} = \mu_1 + \frac{f_2}{(\mu_2 - \mu_1)^{-1} + \frac{f_1(K_1+2\mu_1)}{5\mu_1(K_1+\frac{4}{3}\mu_1)}}, \tag{6.11}$$

where the expressions give the upper bound when the stiffest material is subscripted 1 in the expressions above, and the lower bound when the softest material is subscripted 1. For two minerals, the effective matrix modulus is the average of the lower and the upper bounds (Hill, 1952; Mavko et al., 1998) is given by,

$$K_{Hill} = \frac{1}{2}(K^{HS+} + K^{HS-}) \tag{6.12}$$

$$\mu_{Hill} = \frac{1}{2}(\mu^{HS+} + \mu^{HS-}). \tag{6.13}$$

The corresponding Voigt-Reuss and Hashin-Shtrikman bounds are very close if the minerals are elastically similar. If there is a significant deviation in mineral constitutes, the bounds deviate, significantly, which makes the prediction difficult.

Anisotropy is a more realistic representation of a reservoir, which is a measure of fine-scale structure of a reservoir. Brown and Korringa (1975) relate the anisotropic rock effective elastic compliance tensor to the same rock filled with fluid, and for an isotropic and monomineralic rock, their relations reduce to Gassmann's equation. For fluid substitution theory, knowledge of the anisotropic symmetry and pore-space compressibility are required (Adam et al., 2006).

Low-Frequency

The low-frequency assumption is originated from on the assumption that the pore spaces are well connected, and the pore pressure is at equilibrium. As the seismic wave propagates, it loads the rock with incremental compression which induces increments in pore pressure. At low frequency, the pore fluid has enough time to flow within pore space to eliminate any pore

pressure gradient (Smith et al., 2003). This assumption breaks at high frequency because the pore fluid does not have sufficient time to redistribute itself to eliminate the pressure gradient. It is usually assumed that Gassmann's equation works well for seismic applications with a frequency less than 100 Hz (Mavko et al., 1998). However, laboratory data show that rocks may result in acoustic wave dispersion in typical seismic frequency band (Batzle et al., 2001; Han and Batzle, 2004).

Biot (1956b,c) developed a theory based on the model of pores being long, cylindrical tubes, that allows for a macroscopic flow of the fluid phases. Biot's theory reduces to the Gassmann approach at low frequencies but predicts higher P-wave and S-wave velocities than does Gassmann in the high-frequency limit. Although Biot's theory works reasonably well for very porous, high permeability sediments (Stoll, 1998), it does not give accurate results for consolidated rocks. Mavko and Jizba (1991) developed a procedure for estimating the wave speeds over the complete range of frequencies, which required knowledge of the wave speeds in the dry rock at a given value of the effective stress, and in the high-stress limit, when all crack-like pores are closed.

Closed System

In this assumption, the system is assumed closed and no pore fluid movement across the boundaries. This not the case in real reservoir conditions.

No Chemical Interaction

This assumption implies that there is no chemical reaction between the pore-fluid and the solid frame. Classical fluid substitution calculations fail when the injected fluid chemically reacts with the host rock, thus changing the mechanical properties of the rock. If minerals precipitate on and between grains or dissolve in the pore space, the elastic moduli of the rock frame may change significantly.

Therefore, sole use of classical fluid substitution models using non-reactive fluids, such as Gassmann (1951) or Brown and Korringa (1975), are likely to fail. The presence of a reactive fluid, or reactant, can induce changes (decrease or increase) in seismic wave propagation velocities of the host rock beyond that predicted by Gassmann's fluid substitution calculations (Vanorio et al., 2010).

In general, Gassmann's fluid substitution model is applied in 4D monitoring in the clastic reservoirs. However, in the carbonates, this model often produces underestimated velocity changes.

Steps in Fluid substitution

Step 1: Initial bulk modulus and shear modulus

The initial bulk modulus (K_I) and shear modulus (μ_I) can be calculated from the initial bulk density (ρ_I), P-wave velocity (V_{Ip}), and S-wave velocity (V_{Is}), that should be available before fluid substitution. Using the relationship among elastic parameters, the bulk modulus and shear modulus can be calculated with expressions,

$$K_I = \rho_I \left(V_{Ip}^2 - \frac{4}{3} V_{Is}^2 \right) \tag{6.14}$$

and

$$\mu_I = \rho_I V_{Is}^2, \tag{6.15}$$

where $\rho_I = \rho_{brine}\phi + (1-\phi)\rho_s = 2072$ kg/m^3. Using the reservoir parameters given in Table 6.1, the initial bulk modulus and the initial shear modulus are found to be $K_I = 7.5683$ Gpa and $\mu_I = 0.857$ Gpa, respectively. The shear modulus μ_I is insensitive in fluid substitution and remains constant, $\mu_I = \mu_{after}$.

Step 2: Calculate frame bulk modulus

The frame bulk modulus can be calculated from initial conditions by rearranging the Gassmann's equation given by,

$$K_{frame} = \frac{K_I \left(\frac{\phi K_s}{K_{brine}} + 1 - \phi \right) - K_s}{\frac{\phi K_s}{K_{brine}} + \frac{K_I}{K_s} - 1 - \phi}. \tag{6.16}$$

We assume that the frame bulk modulus remains constant during fluid substitution. Using the reservoir parameters given in Table 6.1 along with the result of Equation 6.14 (K_I), the frame bulk modulus is found to be $K_{frame} = 2.6815$ Gpa.

Step 3: Calculate bulk modulus and bulk density of fluid after substitution

Given the saturation level (S_l) of initial fluid, the bulk density (ρ_{Af}) and the bulk modulus of the fluid after substitution (K_{Af}) can be calculated using the following expressions,

$$\rho_{Af} = S_l \rho_{brine} + (1 - S_l)\rho_{co2}. \tag{6.17}$$

and

$$K_{Af} = \left[\frac{S_l}{K_{brine}} + \frac{1 - S_l}{K_{co2}} \right]^{-1}. \tag{6.18}$$

Step 4: Calculate the saturated bulk density and bulk modulus

The saturated bulk density can be calculated from porosity (ϕ), the bulk density of the fluid after substitution (ρ_{Af}), and the density of solid mineral (ρ_s) using the expression,

$$\rho_{sat} = \phi \rho_{Af} + (1 - \phi)\rho_s. \tag{6.19}$$

Using the results of the above steps, the saturated bulk modulus can also be calculated using the Biot-Gassmann's equation from the porosity ϕ, the frame bulk modulus (K_{frame}), the solid mineral bulk modulus (K_s), and the bulk modulus of the fluid after substitution (K_{Af}),

$$K_{sat} = K_{frame} + \frac{\left(1 - \frac{K_{frame}}{K_s}\right)^2}{\frac{\phi}{K_{Af}} + \frac{1-\phi}{K_s} - \frac{K_{frame}}{(K_s)^2}}. \tag{6.20}$$

Step 5: Calculate P-wave velocity and S-wave velocity

Using the well known elastic relationships, the P-wave velocity (V_p^{sat}) and S-wave velocity (V_s^{sat}) can be calculated from saturated bulk density (ρ_{sat}), the saturated bulk modulus (K_{sat}) and the shear bulk modulus ($\mu_I = \mu_{after}$),

$$V_p^{sat} = \sqrt{\frac{K_{sat} + \frac{4}{3}\mu_I}{\rho_{sat}}} \tag{6.21}$$

and

$$V_s^{sat} = \sqrt{\frac{\mu_I}{\rho_{sat}}}. \tag{6.22}$$

Using Table 6.1 along with the fluid substitution steps outlined above, 21 different possible CO_2-Brine saturation levels with 5% step are produced. The P-wave velocity, S-wave velocity, bulk density and bulk modulus against the saturation levels are plotted as shown in Figure 6.1 (**a**), Figure 6.1 (**b**), Figure 6.1 (**c**), and Figure 6.1 (**c**), respectively. The density and the S-wave velocity change linearly with increase in CO_2 saturation. The saturated bulk modulus and the P-wave velocity show a non-linear behavior. Both parameters drops drastically to 40% saturation level and show almost no change beyond that level.

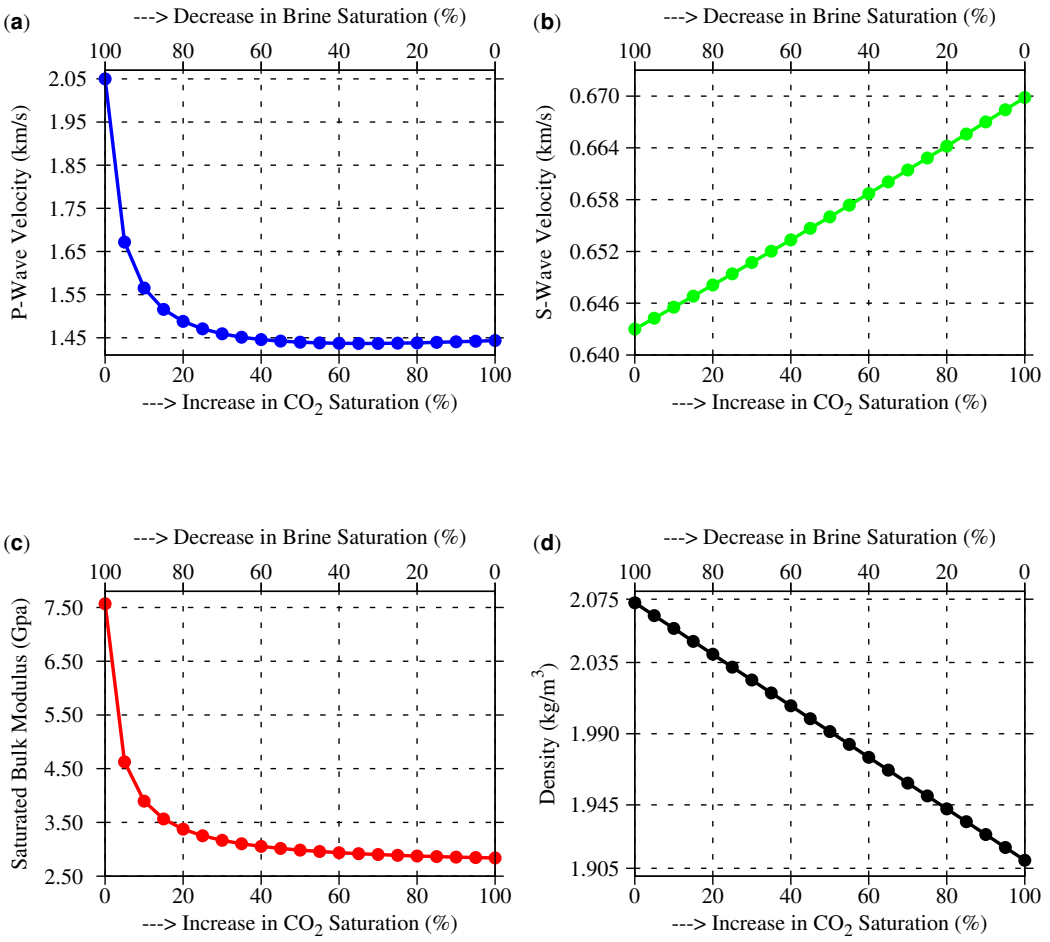


Figure 6.1: These plots are generated using Gassmann's fluid substitution model with the reservoir parameters given in Table 6.1. Brine is replaced by CO_2 for 21 percentage values from 0 to 100: (**a**) P-wave velocity versus CO_2 saturation in % (decrease in brine saturation), (**b**) S-wave velocity versus CO_2 saturation in % (decrease in brine saturation), (**c**) saturated bulk modulus versus CO_2 saturation in % (decrease in brine saturation), and (**d**) bulk density versus CO_2 saturation in % (decrease in brine saturation).

6.4 Numerical Time-Lapse Inversion Examples

To demonstrate the performance of the four time-lapse inversion methods in the CO_2 sequestration setting, synthetic baseline and monitor P-wave velocity models are generated. The true baseline 2D velocity model is shown in Figure 6.2 (**a**). The true monitor velocity

model which represents time-lapse velocity after CO_2 'injection' is shown in Figure 6.2 (**b**). The true velocity difference (the difference between the true monitor and baseline velocities) shown in Figure 6.2 (**c**) is produced taking into account the result of the fluid substitution. The velocity difference in this image ranges from 0.0 m/s to approximately -500 m/s; the different level of velocity change indicates the different level of CO_2 saturation. Each of the models has 2.8 km in the lateral direction and 1.2 km deep with a square grid size of 8 m by 8 m.

The synthetic inversion examples include four scenarios: repeatable, non-repeatable, 10% data decimation, and 25% data decimation. They are designed to mimic the non-repeatability challenges. A smooth starting velocity model is shown in Figure 6.3 is used for the inversions in those four scenarios. Each inversion was run for 14 frequencies. The entire frequency ranges from 2.5 Hz to 30.0 Hz. The internal algorithm runs a maximum iteration of 81 for each frequency.

6.4.1 Scenario One: Repeatable

In this synthetic experiment example, the acquisition geometries for both baseline and monitor cases were the same, i.e., perfectly repeatable. The true velocity models in Figure 6.2 (**a**) and Figure 6.2 (**b**), and a Ricker wavelet with central frequency 10 Hz for source excitation were used to generate synthetic data using the constant density acoustic wave equation modelling presented in Equation 2.55. These data have a total of 70 shots and 350 receivers for each shot: 40 m shot separation and 8 m receiver separation.

Figures 6.4(**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by independent method, respectively. Figures 6.5 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and their difference by double difference method, respectively. Figures 6.6 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint method, respectively. Figures 6.7 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint reparametrized method, respectively.

Figures 6.8 (**a**), (**b**), (**c**), and (**d**) are vertical profiles of the monitor inversion by independent, double difference, joint and joint reparametrized methods, respectively. Each section of Figure 6.8 depicts a comparison of inverted monitor velocity model against the true and initial velocity models. Figure 6.9 shows a representative mean horizontal profile errors for each method. The mean absolute errors between the true and estimated velocity differences were calculated for all lateral positions by averaging in depth.

Figures 6.10 (**a**), (**b**) and (**c**) are comparison of the behavior of the cost function and computational time for the four time-lapse inversion methods in repeatable case. The

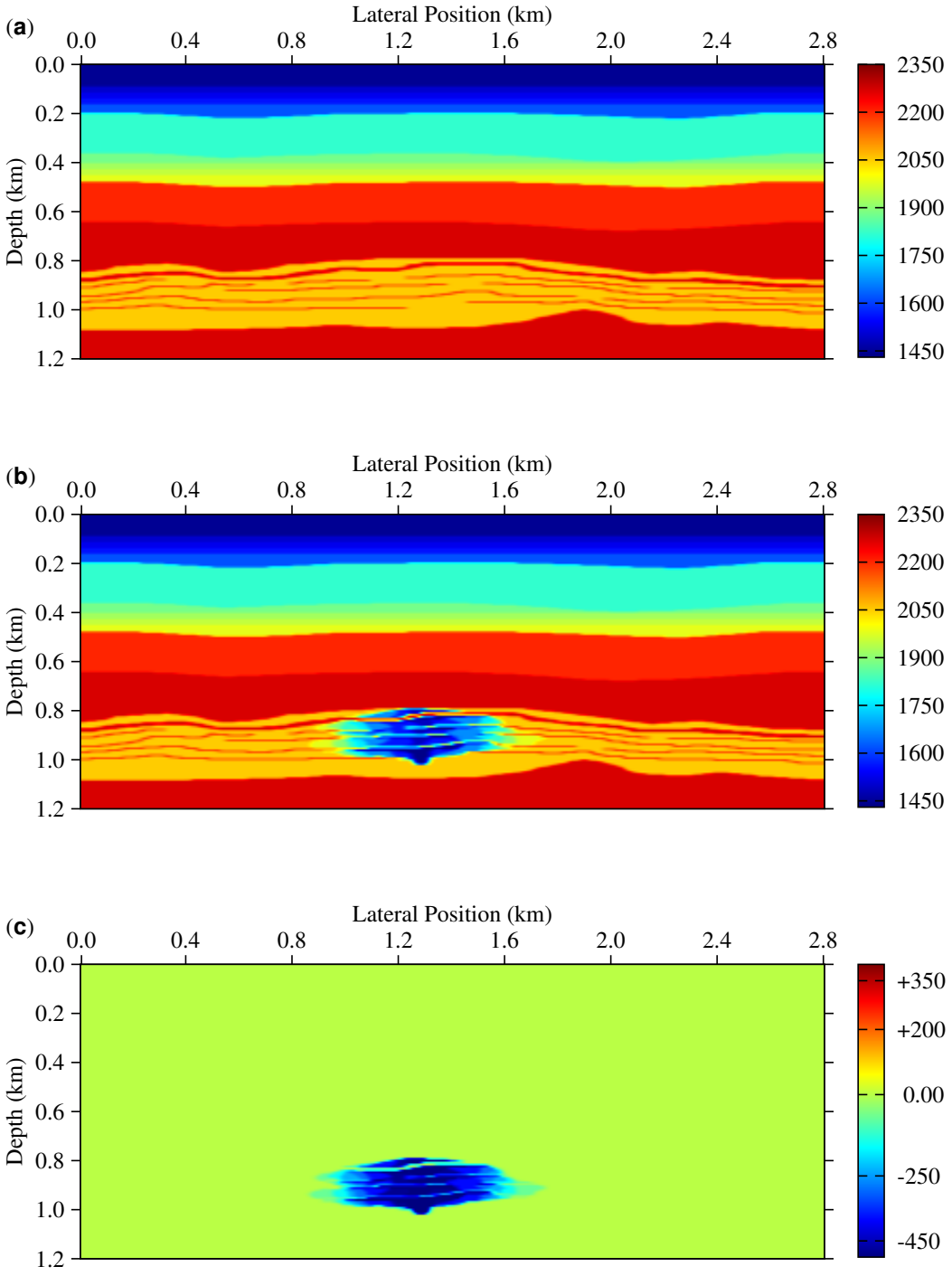


Figure 6.2: (**a**) True baseline velocity model (**b**) true monitor velocity model and (**c**) true time lapse velocity change.

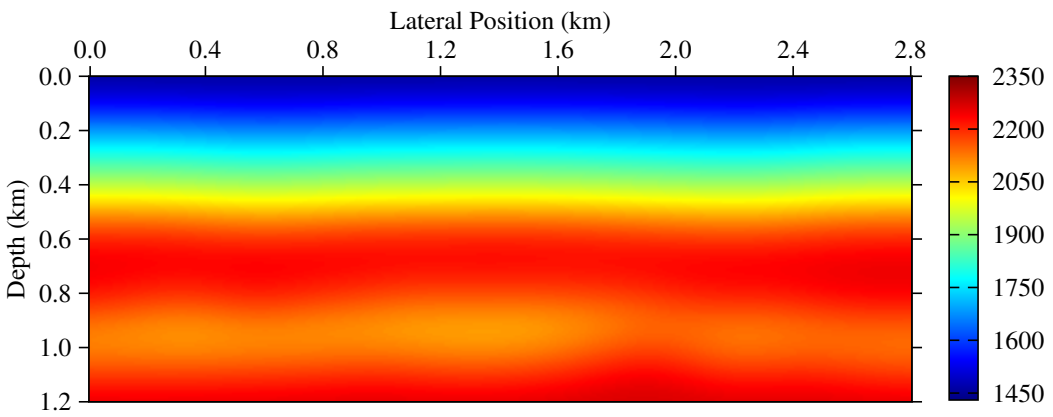


Figure 6.3: Initial velocity model used for time-lapse inversion.

normalized CPU time is plotted against the number of iteration as shown in Figure 6.10 (**a**). Figure 6.10 (**b**) depicts the normalized cost function as a function of iteration. Figure 6.10 (**c**) shows the cost function as a function of normalized CPU time. The cost functions and the CPU time at each iteration are aggregates over the 14 frequencies.

6.4.2 Scenario Two: Non-Repeatable

In this synthetic experiment example, the acquisition geometries for both baseline and monitor cases were different, i.e., a non-repeatable experiment. The true velocity models Figure 6.2 (**a**) and Figure 6.2 (**b**), and a Ricker wavelet with central frequency 10 Hz for source excitation were used to generate synthetic data. These data have a total of 70 shots and 350 receivers for each shot: 40 m shot separation and 8 m receiver separation. To mimic the non-repeatability, six shots of the monitor survey were shifted horizontally by 8 m such that every shifted source and corresponding receivers for each shot has acquisition difference of 8 m from normal baseline survey.

Figures 6.11(**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by independent method, respectively. Figures 6.12 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and their difference by double difference method, respectively. Figures 6.13 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint method, respectively. Figures 6.14 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint reparametrized method, respectively. Figures 6.15 (**a**), (**b**), (**c**), and, (**d**) are vertical profiles of the monitor inversion by independent, double

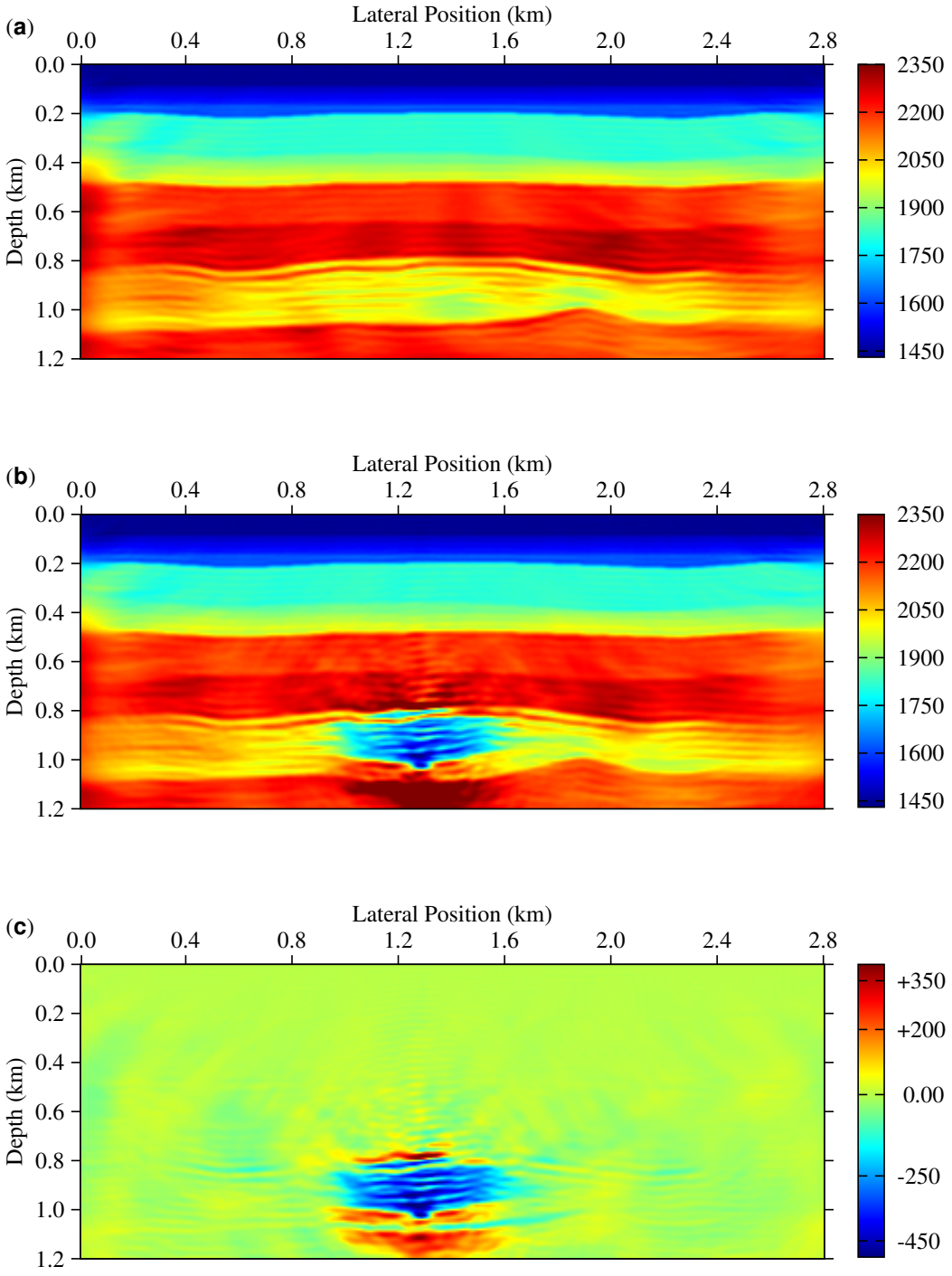


Figure 6.4: A repeatable example: inversion results by the independent method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

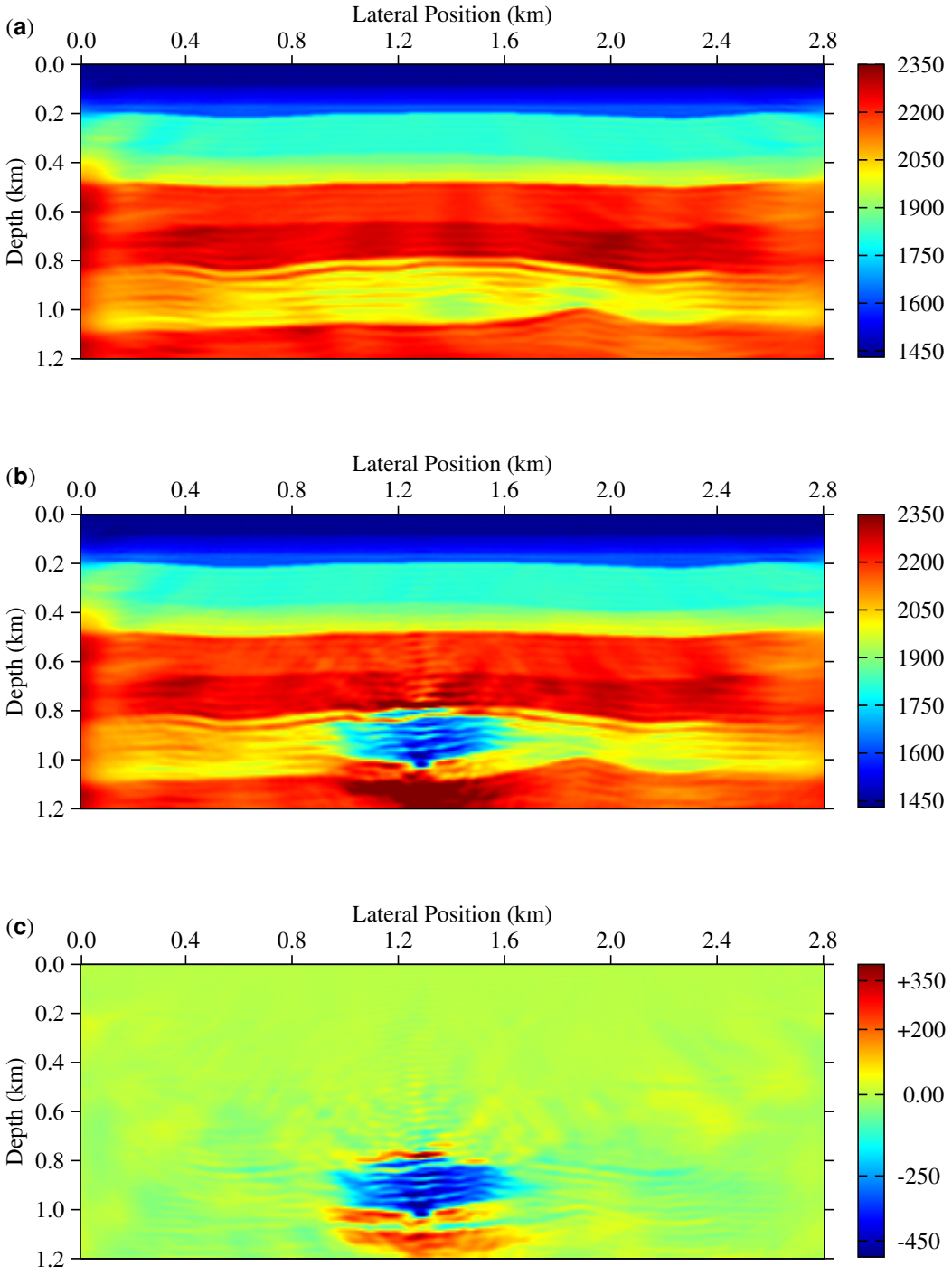


Figure 6.5: A repeatable example: inversion results by the double difference method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

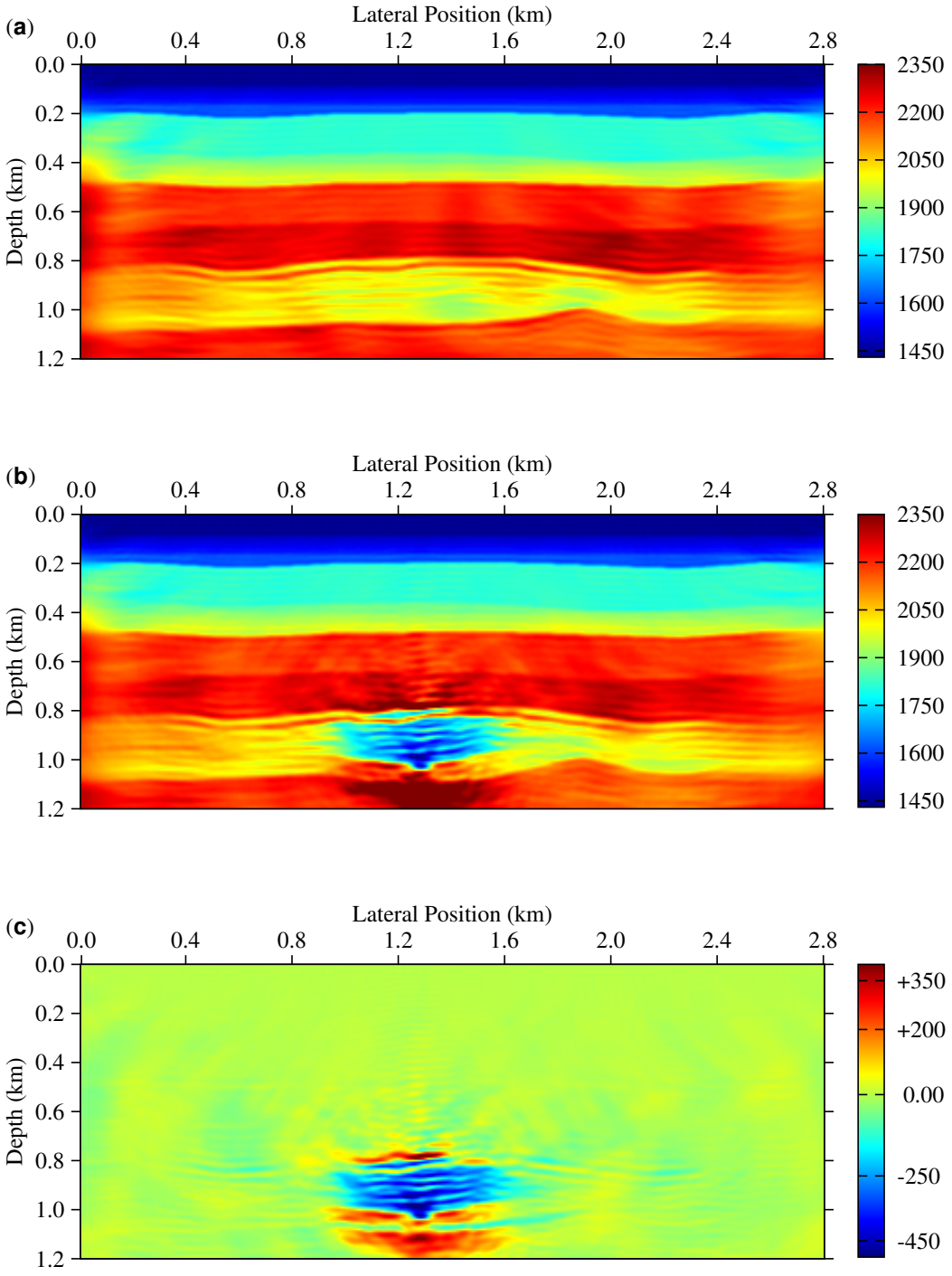


Figure 6.6: A repeatable example: inversion results by the joint method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

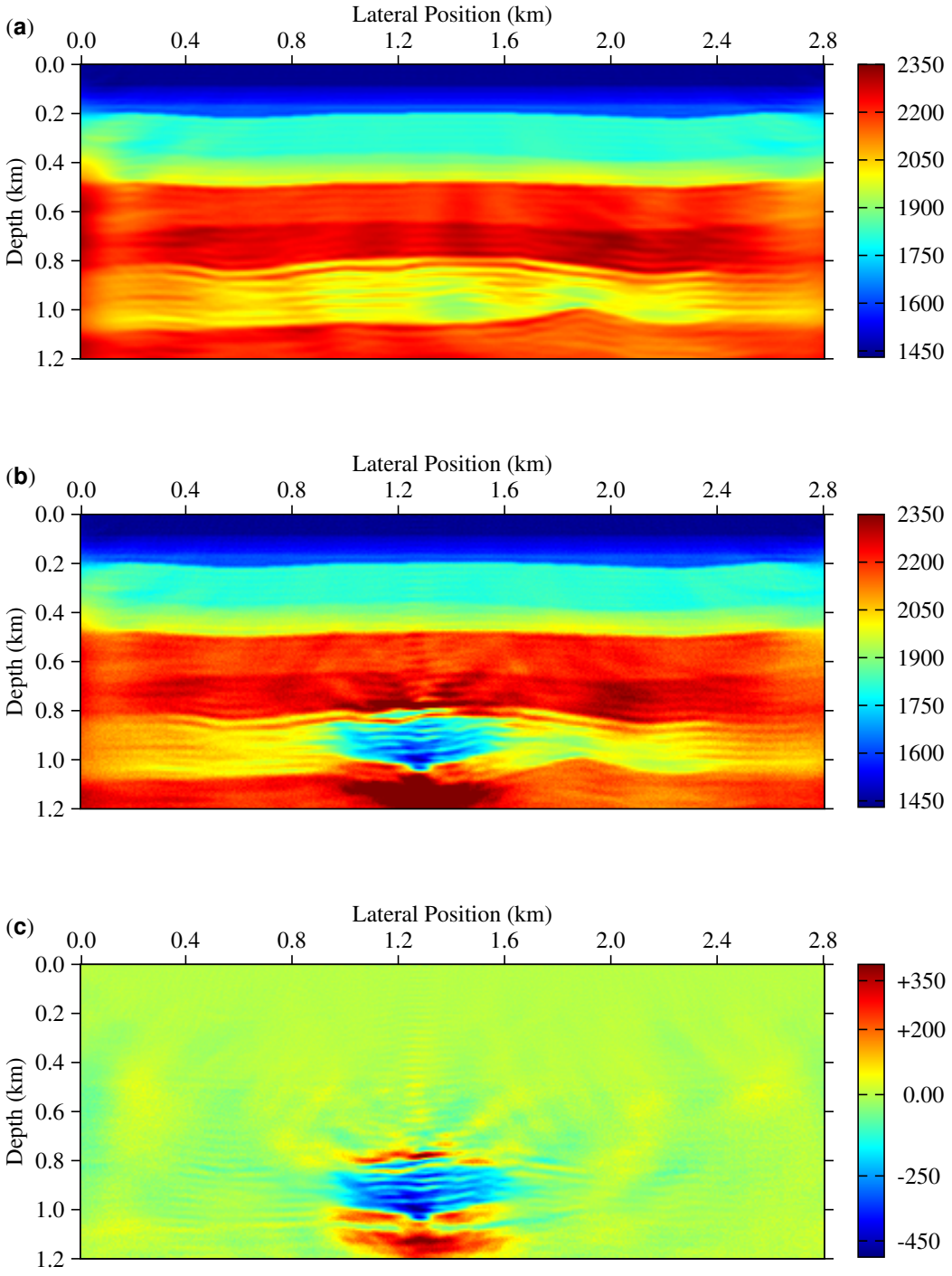


Figure 6.7: A repeatable example: inversion results by the joint reparameterized method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

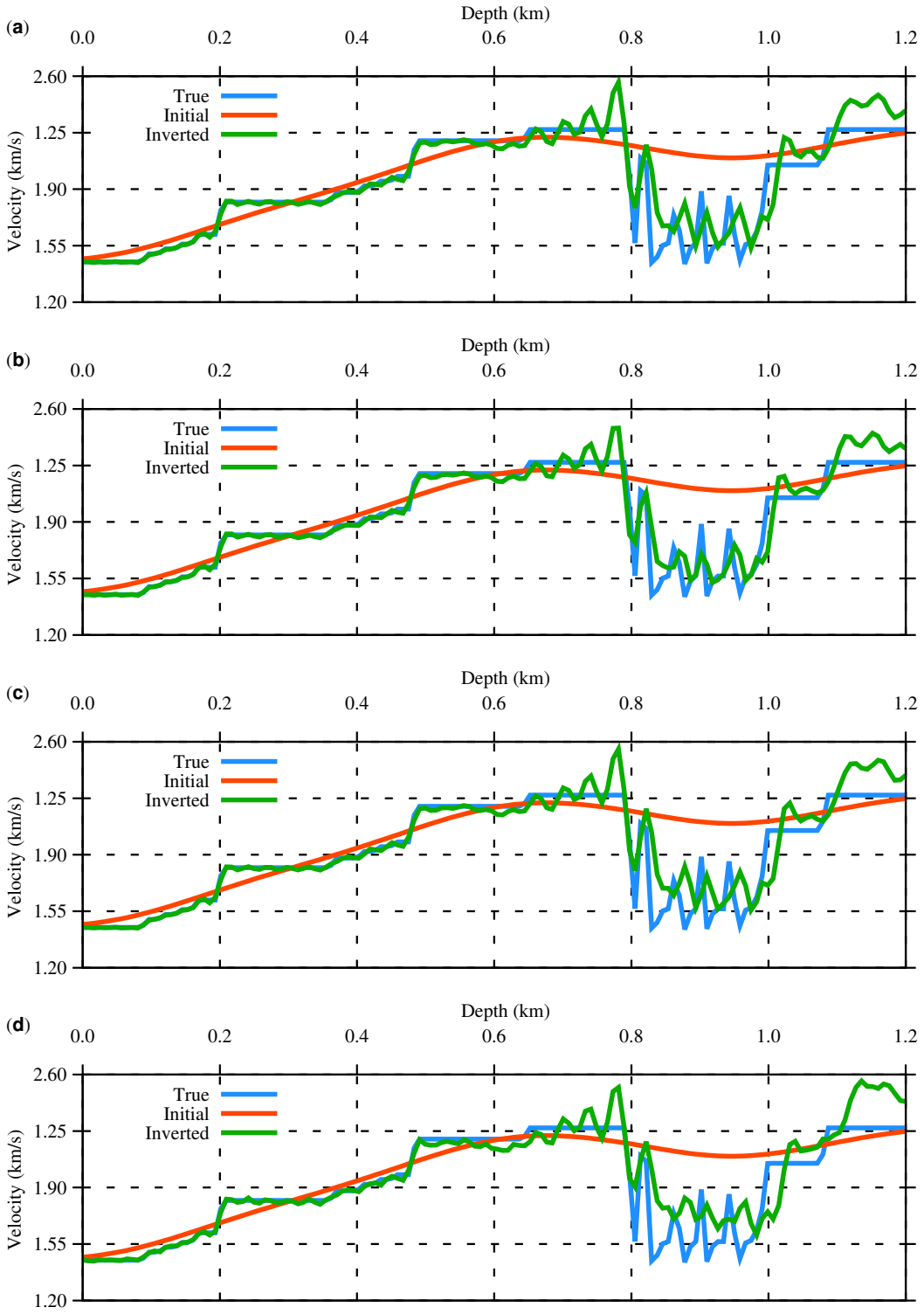


Figure 6.8: A repeatable example: comparison of vertical profiles of inverted monitor velocity with the true and initial velocity models (**a**) independent, (**b**) double difference, (**c**) joint, and (**d**) joint reparametrized.

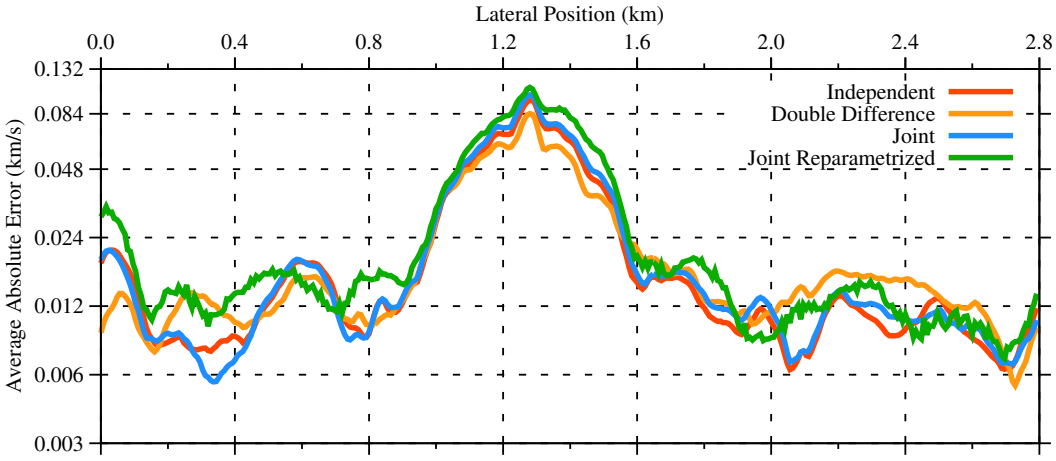


Figure 6.9: A repeatable example: error comparison of the four methods. The average absolute errors are representatives of horizontal profile errors that were averaged vertically. The vertical axes are plotted in log-scale.

difference, joint and joint reparametrized methods, respectively. Each section of Figure 6.15 depicts a comparison of inverted monitor velocity model against the true and initial velocity models. Figure 6.16 shows a representative mean horizontal profile errors for each method. The mean absolute errors between the true and estimated velocity differences were calculated for all lateral positions by averaging in depth.

Figures 6.17 (**a**), (**b**) and (**c**) are comparison of the behavior of the cost function and computational time for the four time-lapse inversion methods in non-repeatable case. The normalized CPU time is plotted against the number of iteration as shown in Figure 6.17 (**a**). Figure 6.17 (**b**) depicts the normalized cost function as a function of iteration. Figure 6.17 (**c**) shows the cost function as a function of normalized CPU time. The cost functions and the CPU time at each iteration are aggregates over the 14 frequencies.

6.4.3 Scenario Three: 10% of monitor survey data are decimated

In this particular synthetic example, the acquisition geometries for both baseline and monitor cases were the same. However, 10% of the monitor survey data were decimated to mimic data mismatch between baseline and monitor survey. Similar to scenario one and two, the true velocity models Figure 6.2 (**a**) and Figure 6.2 (**b**), and a Ricker wavelet with central frequency 10 Hz for source excitation were used to generate synthetic data. These data have a total of 70 shots and 350 receivers for each shot: 40 m shot separation and 8 m receiver separation.

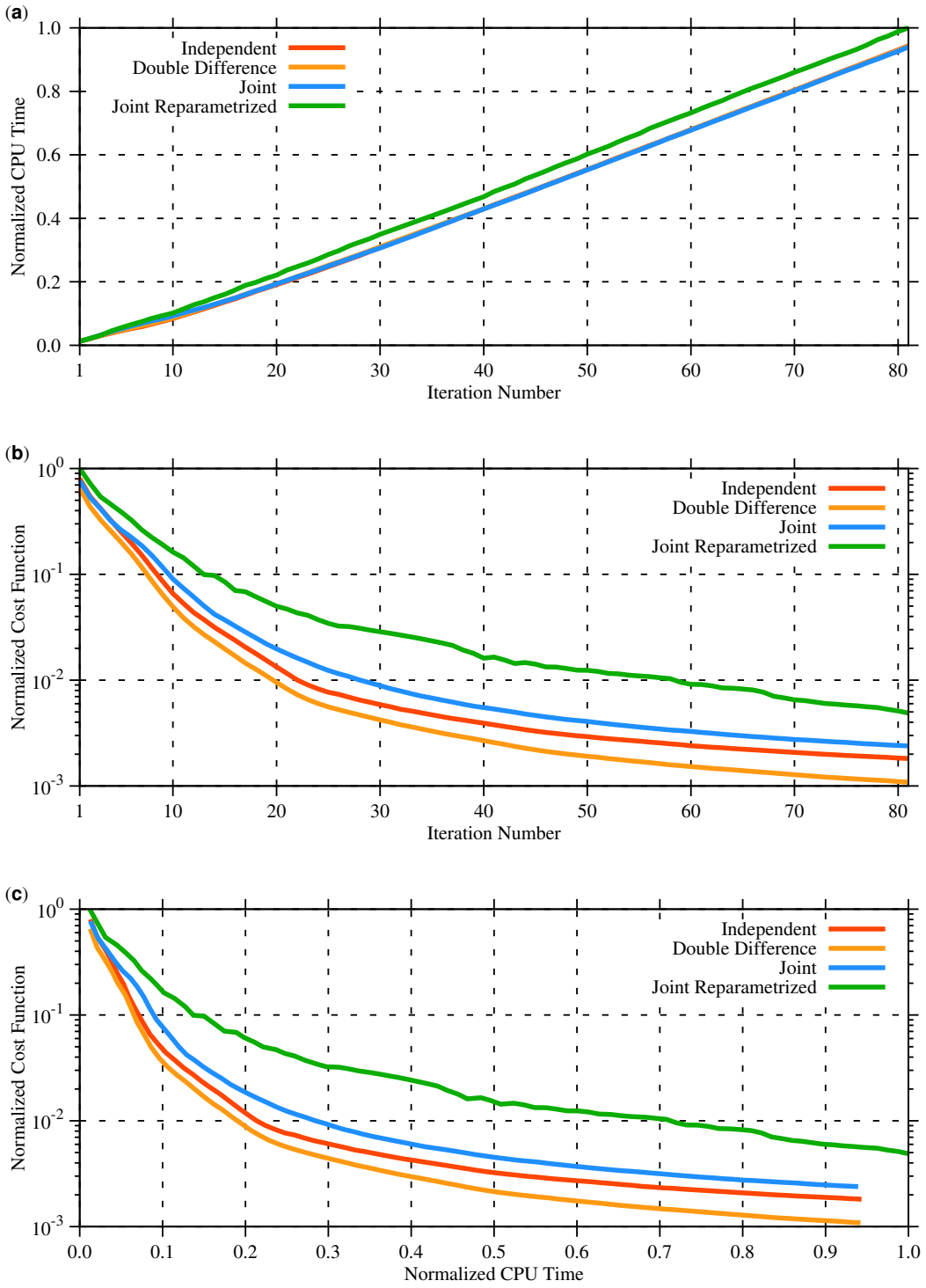


Figure 6.10: A repeatable example: (**a**) normalized computation time versus iteration number (**b**) normalized cost function versus iteration number, (**c**) normalized cost function versus normalized computation time. The vertical axes are plotted in log-scale.

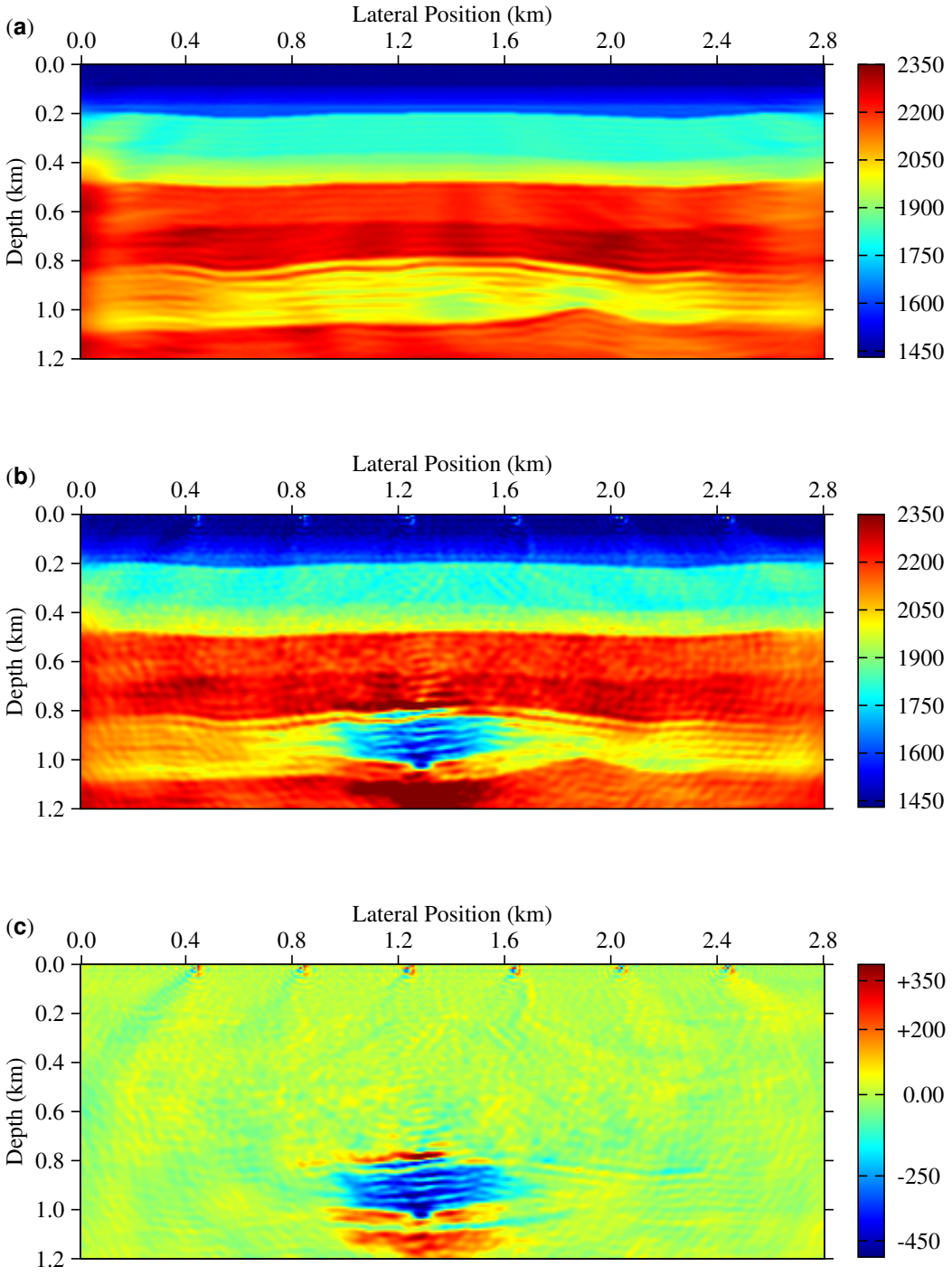


Figure 6.11: A non-repeatable example: inversion by the independent method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

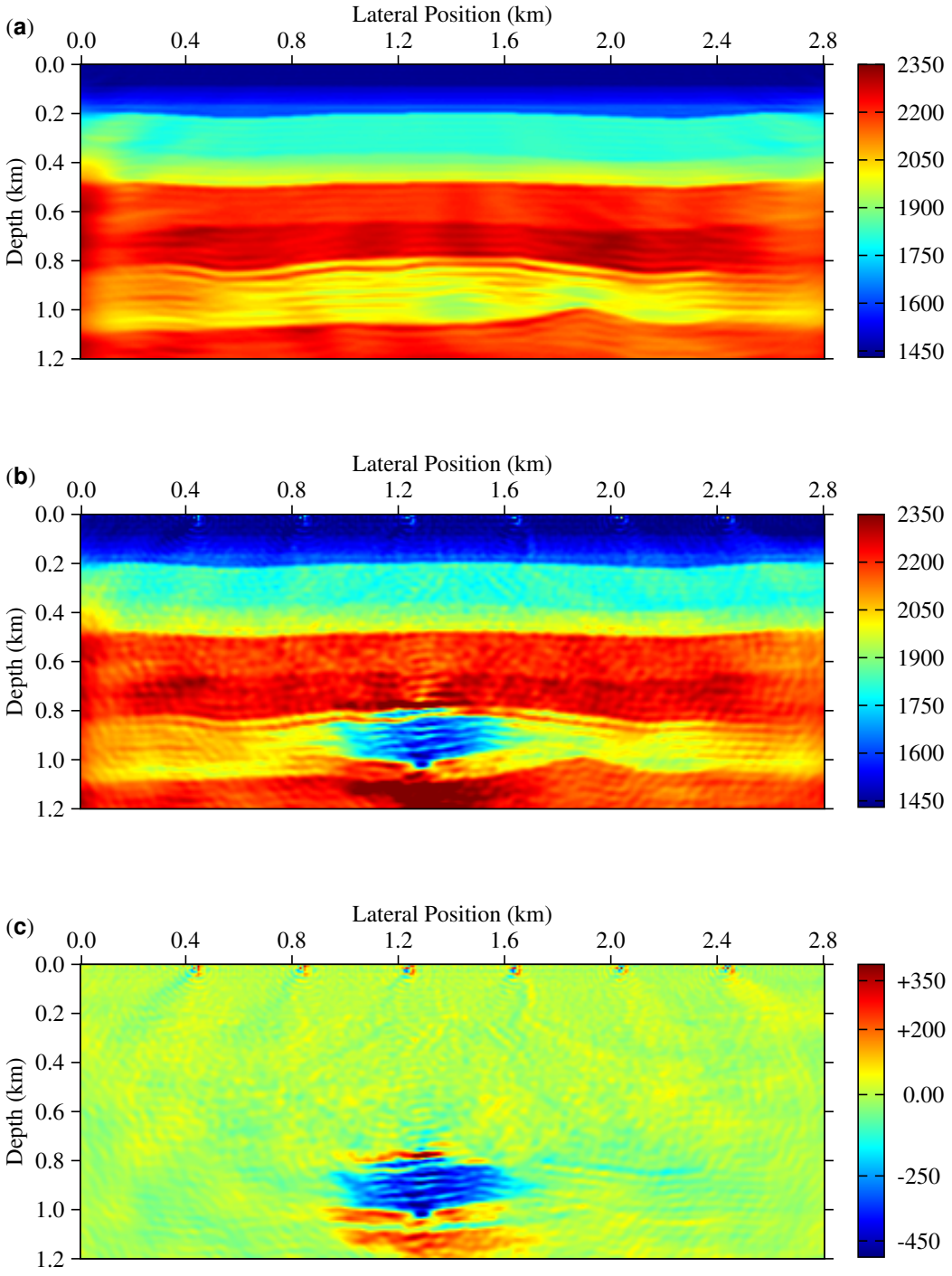


Figure 6.12: A non-repeatable example: inversion by the double difference method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

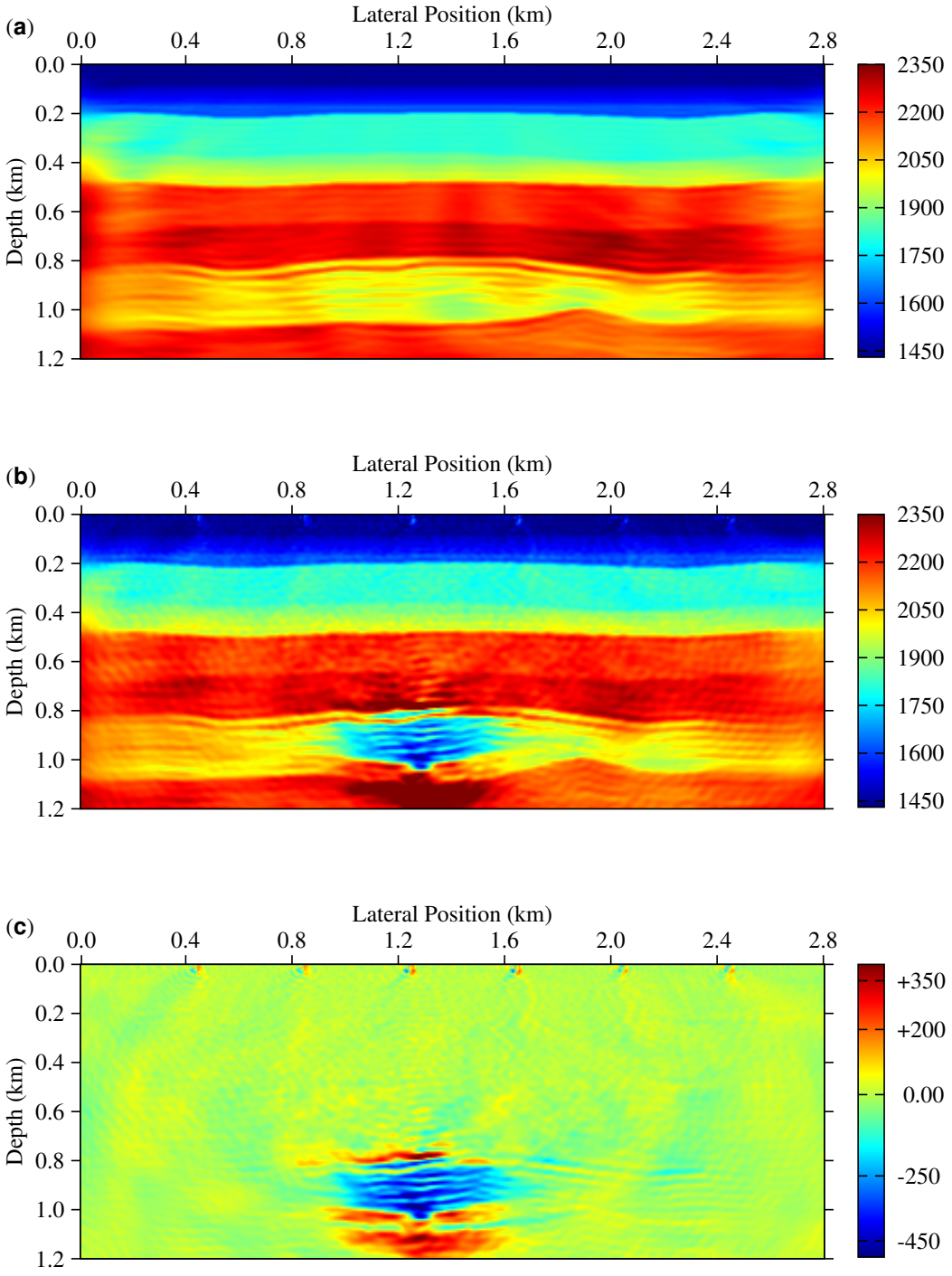


Figure 6.13: A non-repeatable example: inversion by the joint method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

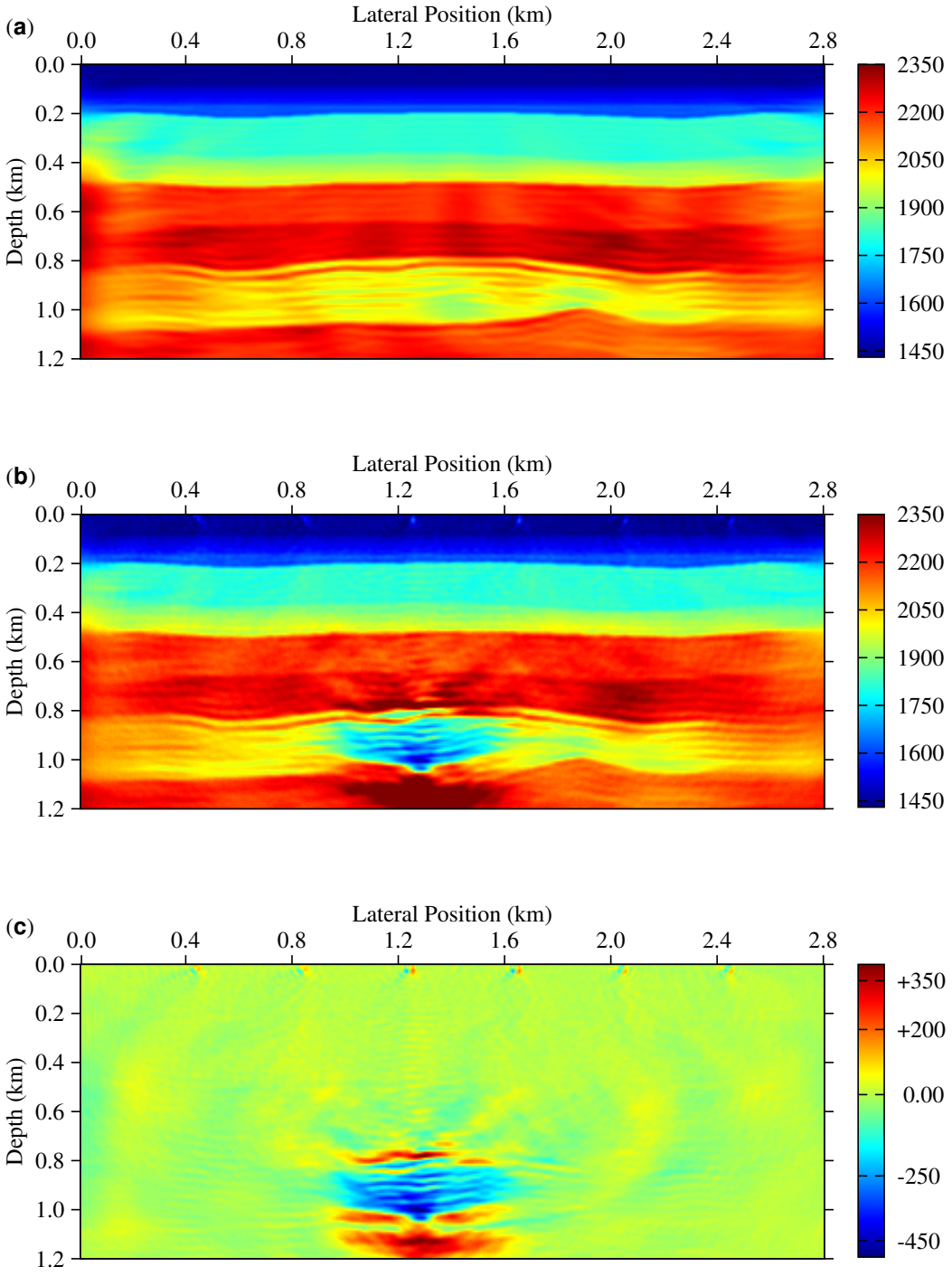


Figure 6.14: A non-repeatable example: inversion by the joint reparameterized method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

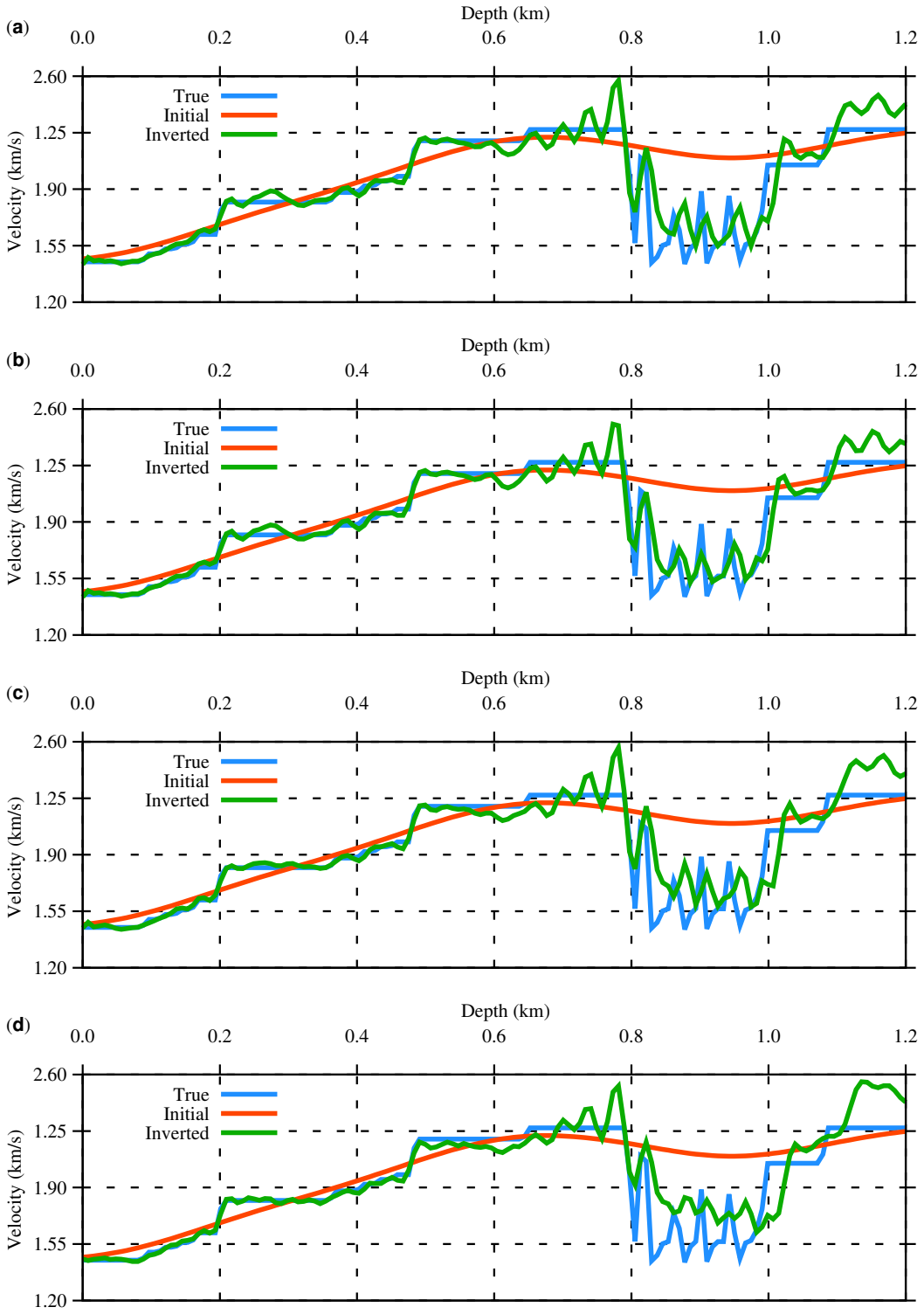


Figure 6.15: A non-repeatable example: comparison of vertical profiles of inverted monitor velocity with the true and initial velocity models (**a**) independent, (**b**) double difference, (**c**) joint, and (**d**) joint reparametrized.

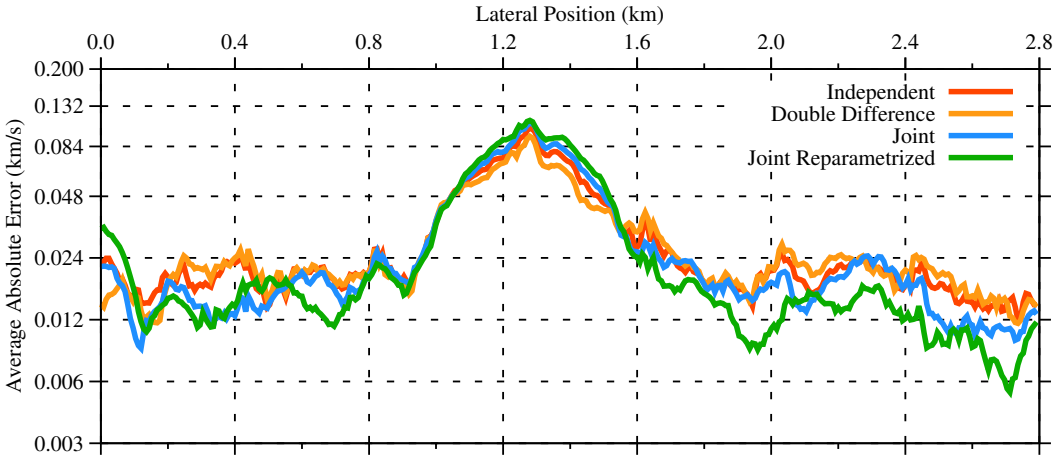


Figure 6.16: A non-repeatable example: error comparison of the four methods. The average absolute errors are representatives of horizontal profile errors that were averaged vertically. The vertical axes are plotted in log-scale.

Figures 6.18(**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by independent method, respectively. Figures 6.19 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and their difference by double difference method, respectively. Figures 6.20 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint method, respectively. Figures 6.21 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint reparametrized method, respectively.

Figures 6.22 (**a**), (**b**), (**c**), and (**d**) are vertical profiles of the monitor inversion by independent, double difference, joint and joint reparametrized methods, respectively. Each section of Figure 6.22 depicts a comparison of inverted monitor velocity model against the true and initial velocity models. Figure 6.23 shows a representative mean horizontal profile errors for each method. The mean absolute errors between the true and estimated velocity differences were calculated for all lateral positions by averaging in depth.

Figures 6.24 (**a**), (**b**) and (**c**) are comparison of the behavior of the cost function and computational time for the four time-lapse inversion methods in non-repeatable case. The normalized CPU time is plotted against the number of iteration as shown in Figure 6.24 (**a**). Figure 6.24 (**b**) depicts the normalized cost function as a function of iteration. Figure 6.24 (**c**) shows the cost function as a function of normalized CPU time. The cost functions and the CPU time at each iteration are aggregates over the 14 frequencies.

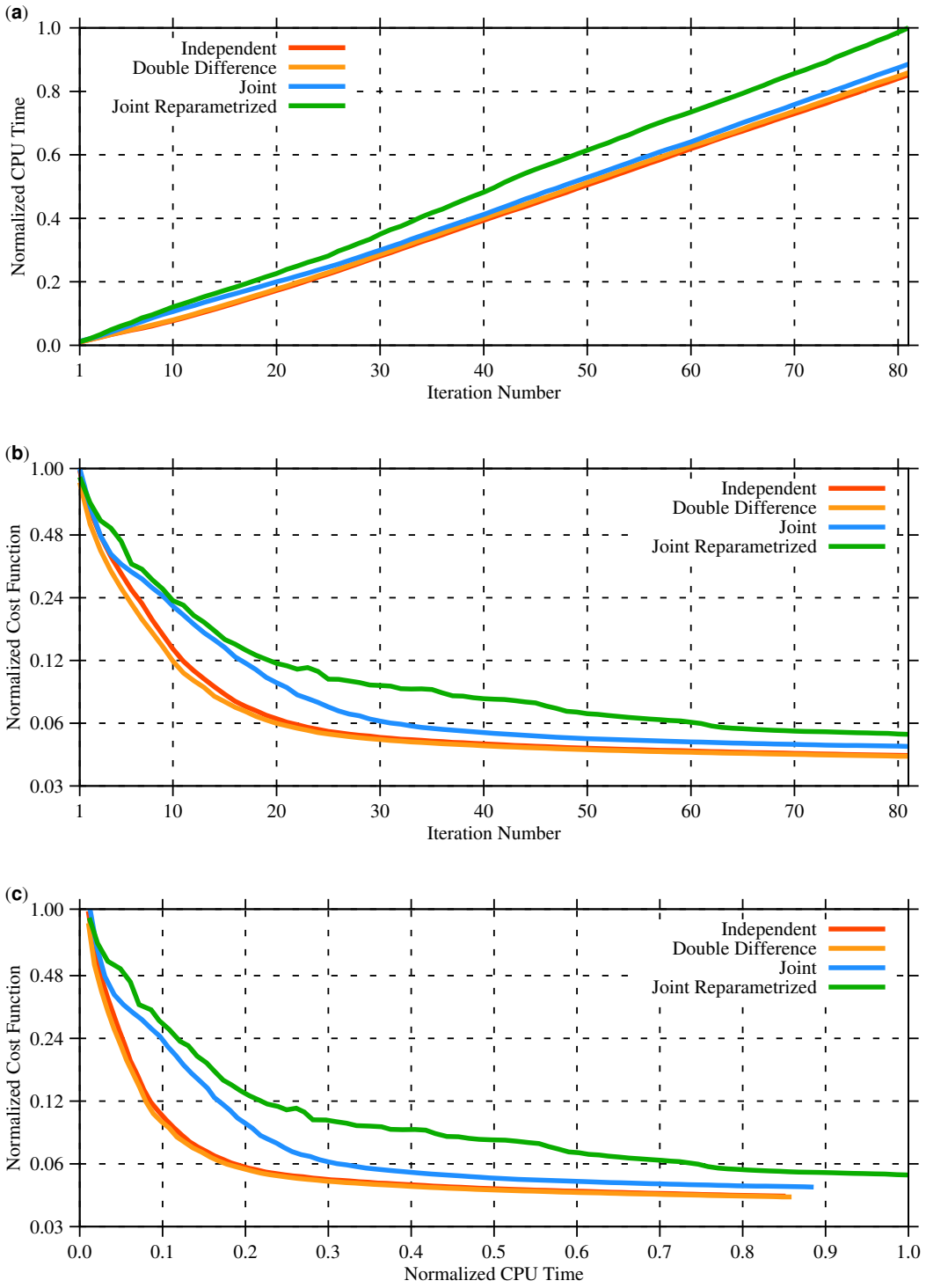


Figure 6.17: A non-repeatable example: (**a**) normalized computation time versus iteration number (**b**) normalized cost function versus iteration number, (**c**) normalized cost function versus normalized computation time. The vertical axes are plotted in log-scale.

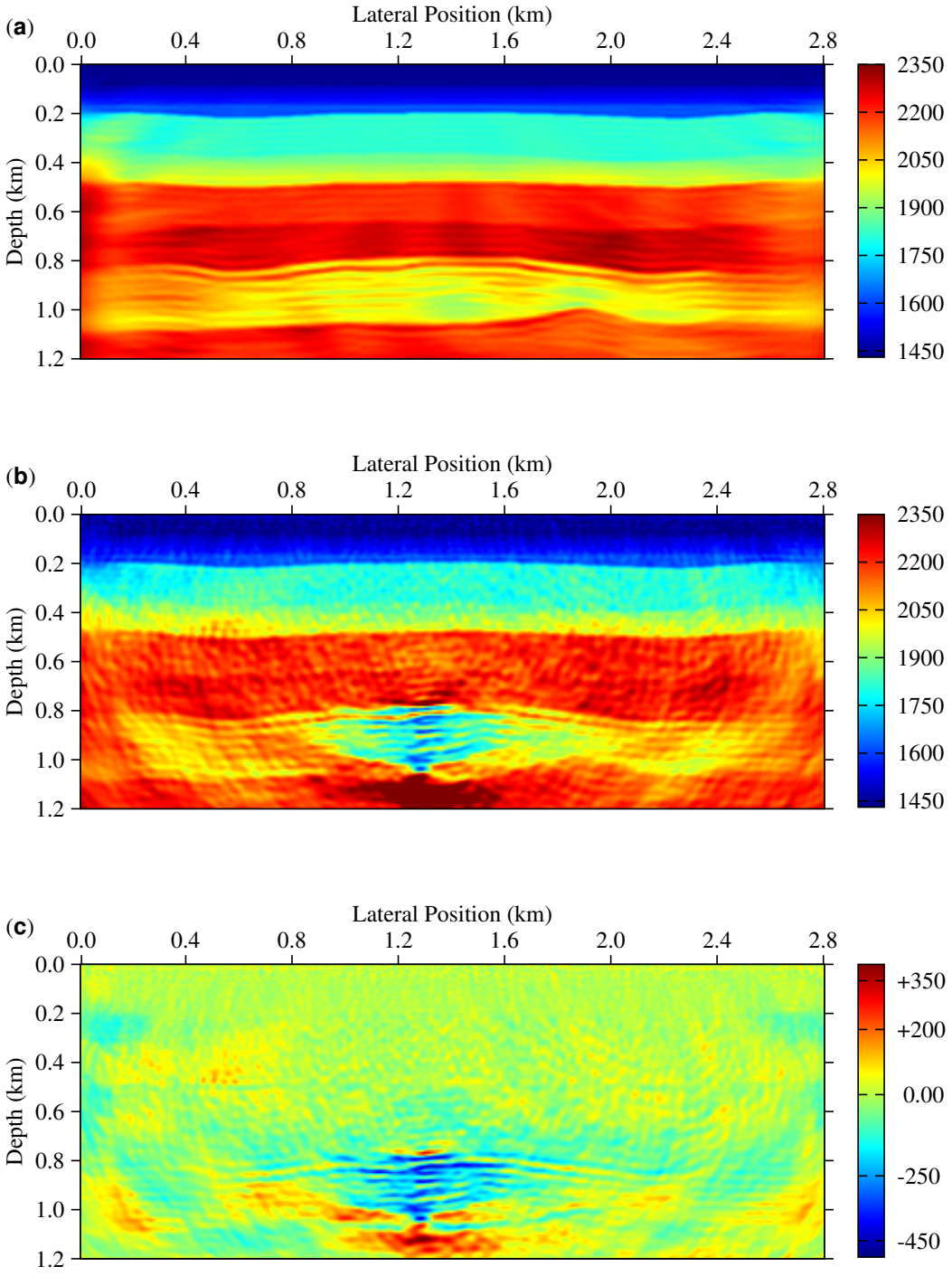


Figure 6.18: Data decimated (10%) example: inversion by the independent method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

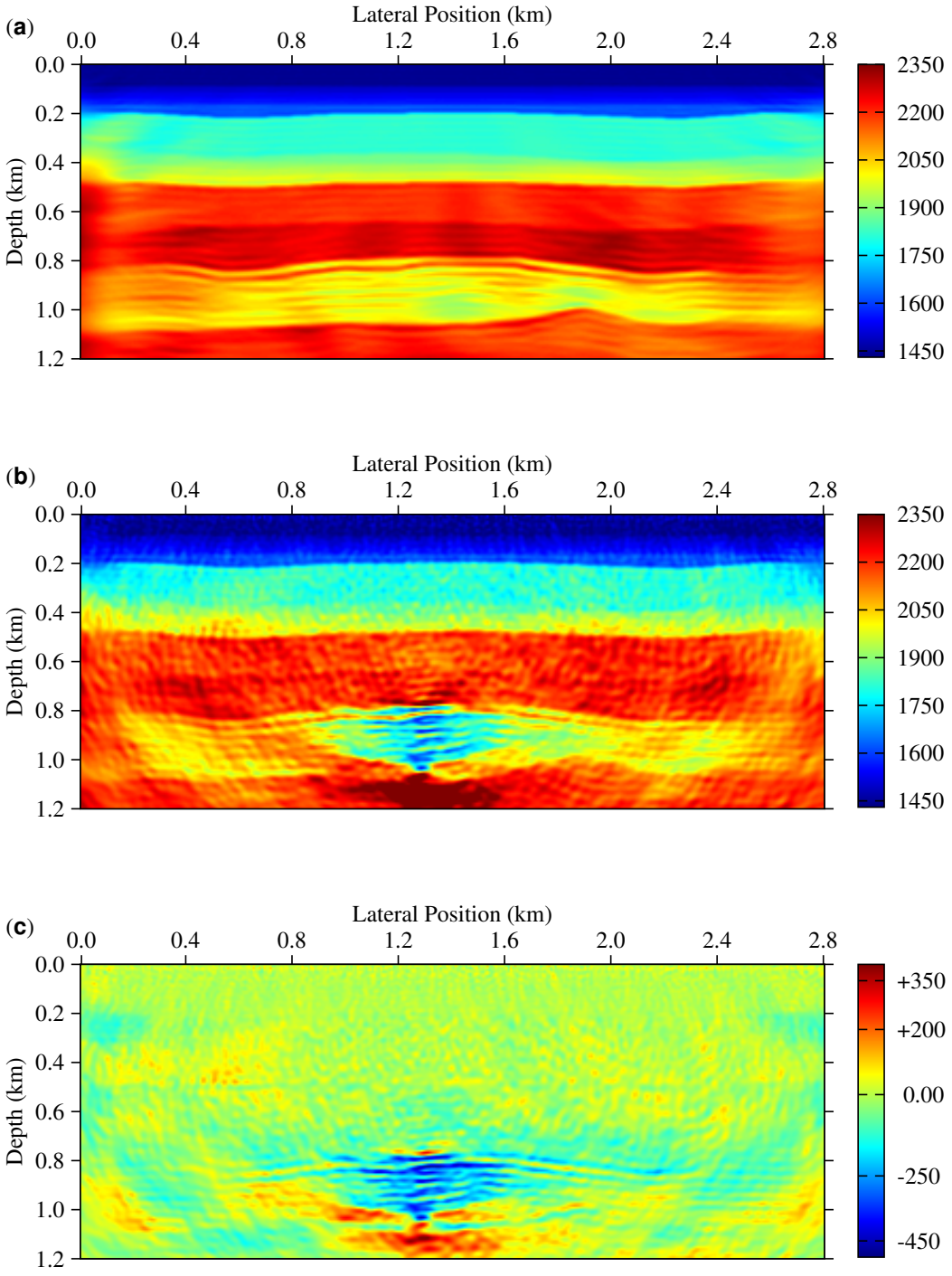


Figure 6.19: Data decimated (10%) example: inversion by the double difference method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

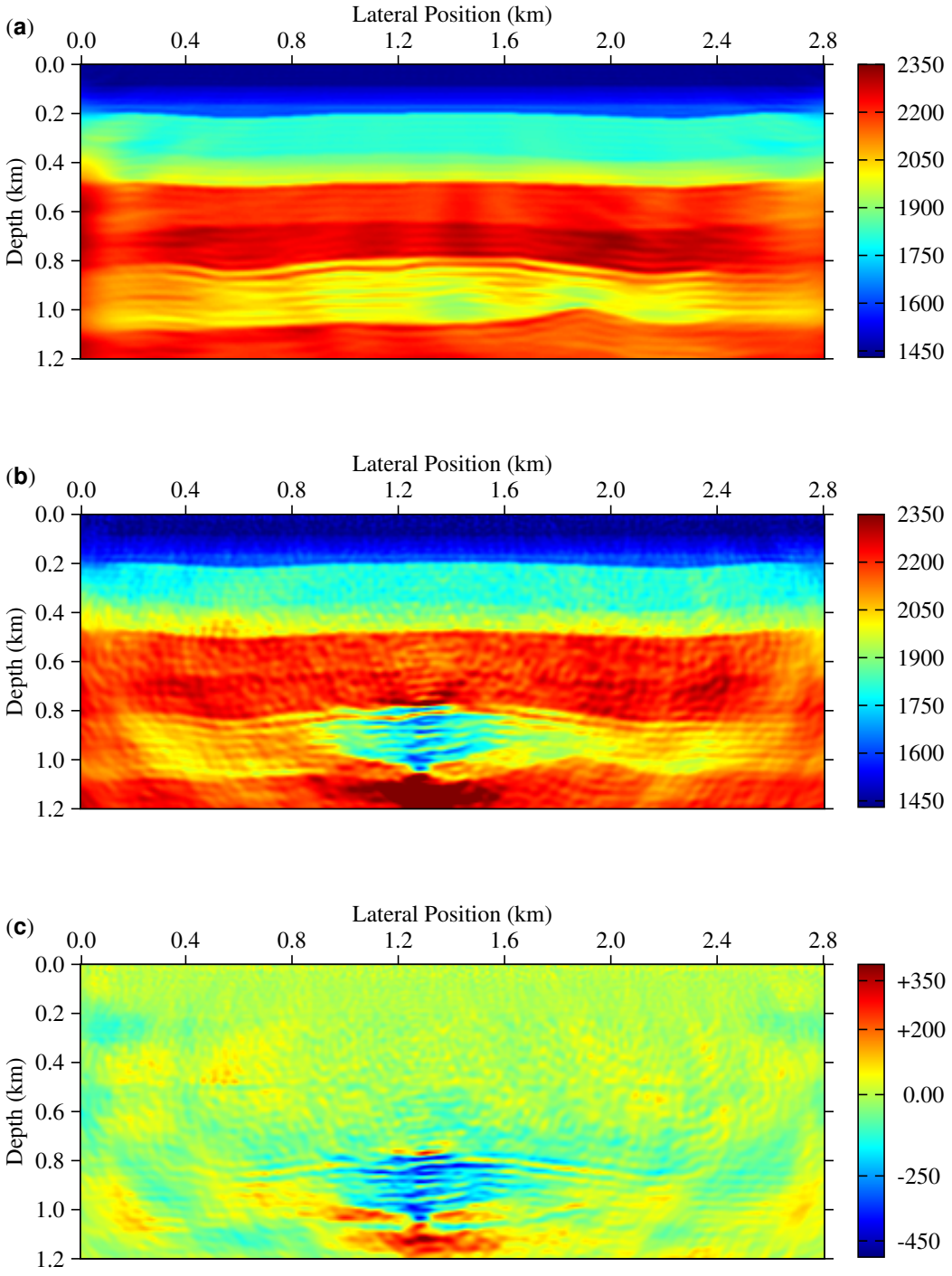


Figure 6.20: Data decimated (10%) example: inversion by the joint method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

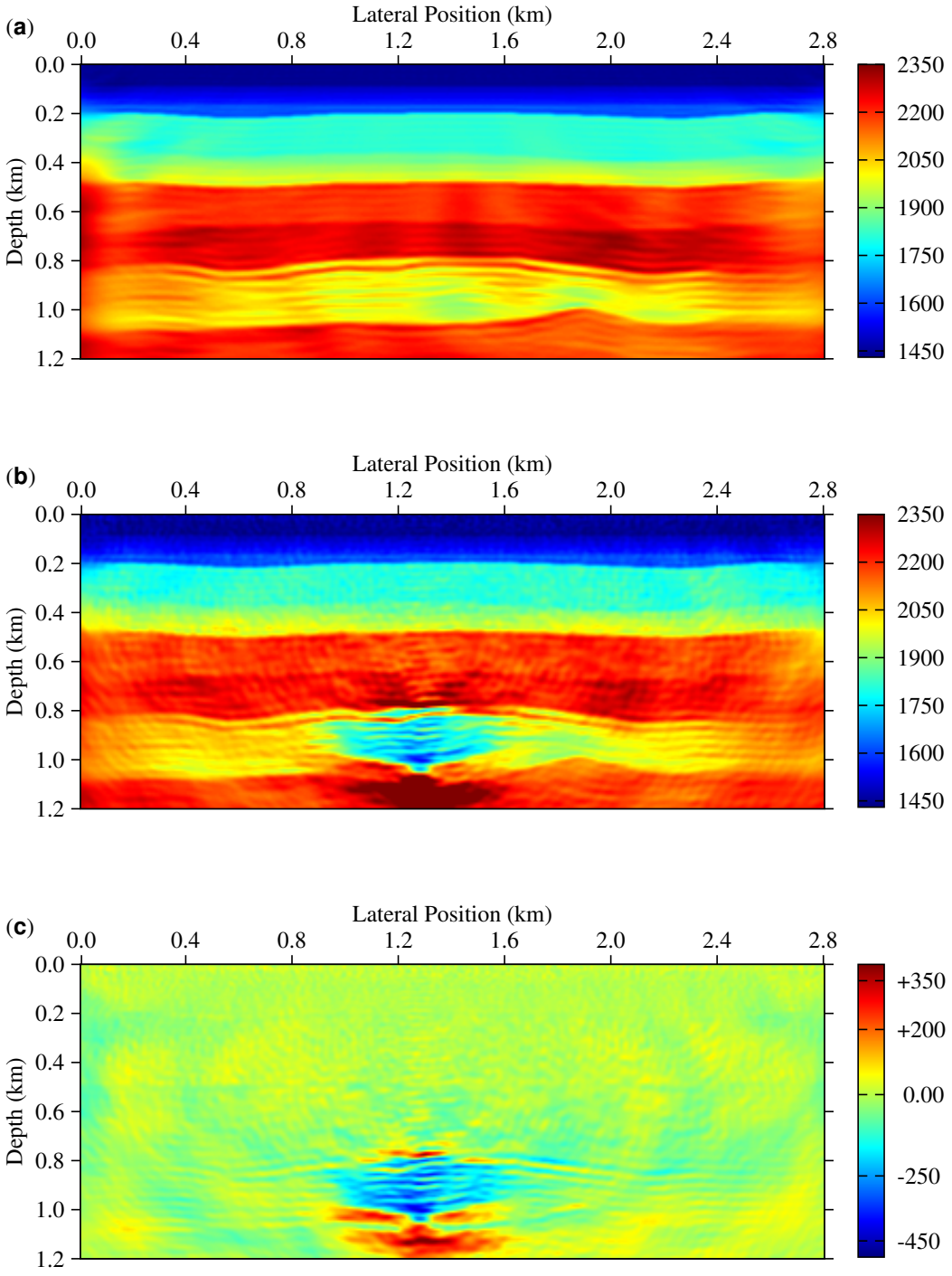


Figure 6.21: Data decimated (10%) example: inversion by the joint reparameterized method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

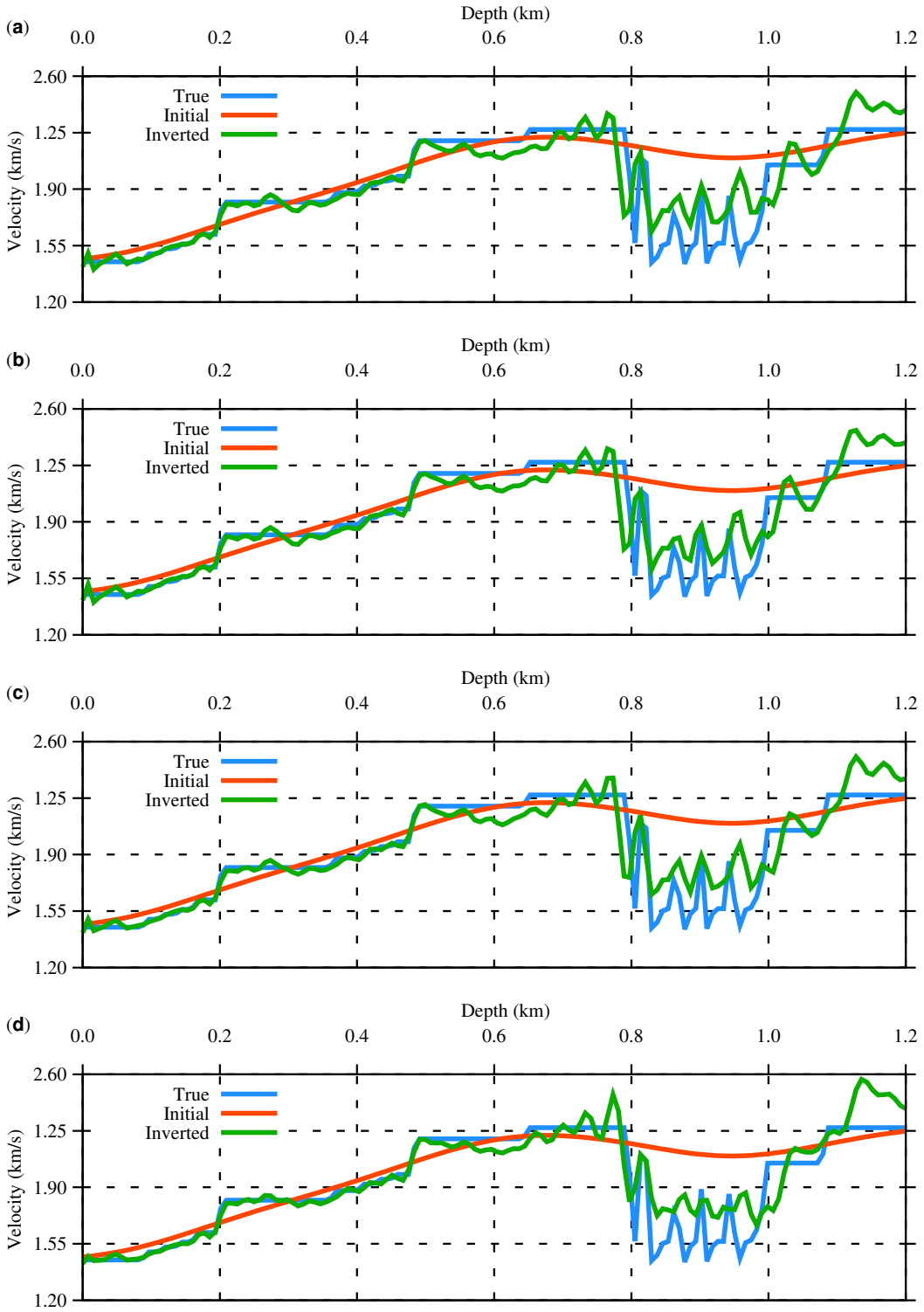


Figure 6.22: Data decimated (10%) example: comparison of vertical profiles of inverted monitor velocity with the true and initial velocity models (**a**) independent, (**b**) double difference, (**c**) joint, and (**d**) joint reparametrized.

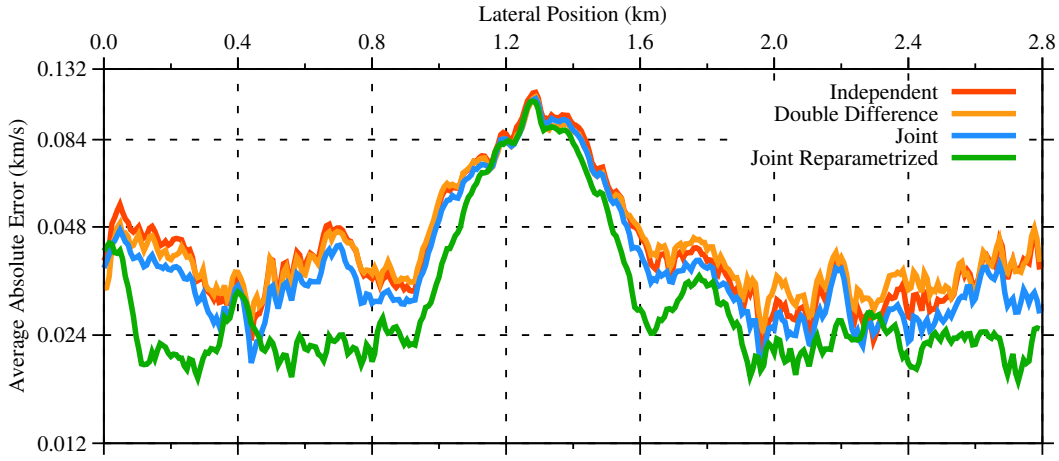


Figure 6.23: Data decimated (10%) example: error comparison of the four methods. The average absolute errors are representatives of horizontal profile errors that were averaged vertically. The vertical axes are plotted in log-scale.

6.4.4 Scenario Four: 25% of monitor survey data are decimated

In this particular synthetic example, the acquisition geometries for both baseline and monitor cases were the same. However, 25% of the monitor survey data were decimated to mimic data mismatch between baseline and monitor survey. Similar to the three previous scenarios, the true velocity models Figure 6.2 (**a**) and Figure 6.2 (**b**), and a Ricker wavelet with central frequency 10 Hz for source excitation were used to generate synthetic data. These data have a total of 70 shots and 350 receivers for each shot: 40 m shot separation and 8 m receiver separation.

Figures 6.25(**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by independent method, respectively. Figures 6.26 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and their difference by double difference method, respectively. Figures 6.27 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint method, respectively. Figures 6.28 (**a**), (**b**) and (**c**) are inversion results of the baseline velocity, the monitor velocity and the difference by joint reparametrized method, respectively. Figures 6.29 (**a**), (**b**), (**c**), and (**d**) are vertical profiles of the monitor inversion by independent, double difference, joint and joint reparametrized methods, respectively. Each section of Figure 6.29 depicts a comparison of inverted monitor velocity model against the true and initial velocity models. Figure 6.30 shows a representative mean horizontal profile errors for each method. The mean absolute errors between the true and estimated velocity differences were calculated for all lateral positions by averaging in depth.

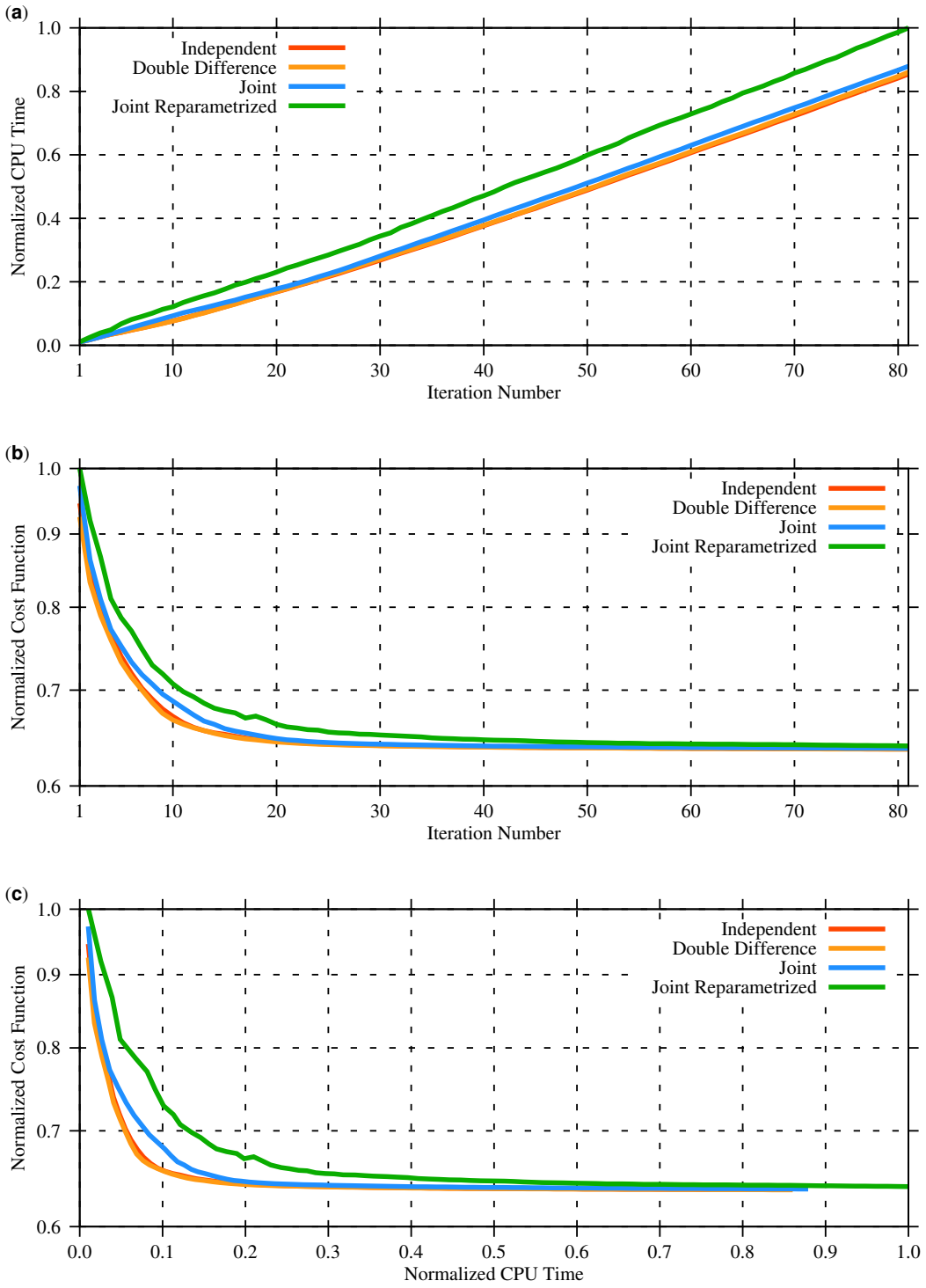


Figure 6.24: Data decimated (10%) example: (**a**) normalized computation time versus iteration number (**b**) normalized cost function versus iteration number, (**c**) normalized cost function versus normalized computation time. The vertical axes are plotted in log-scale.

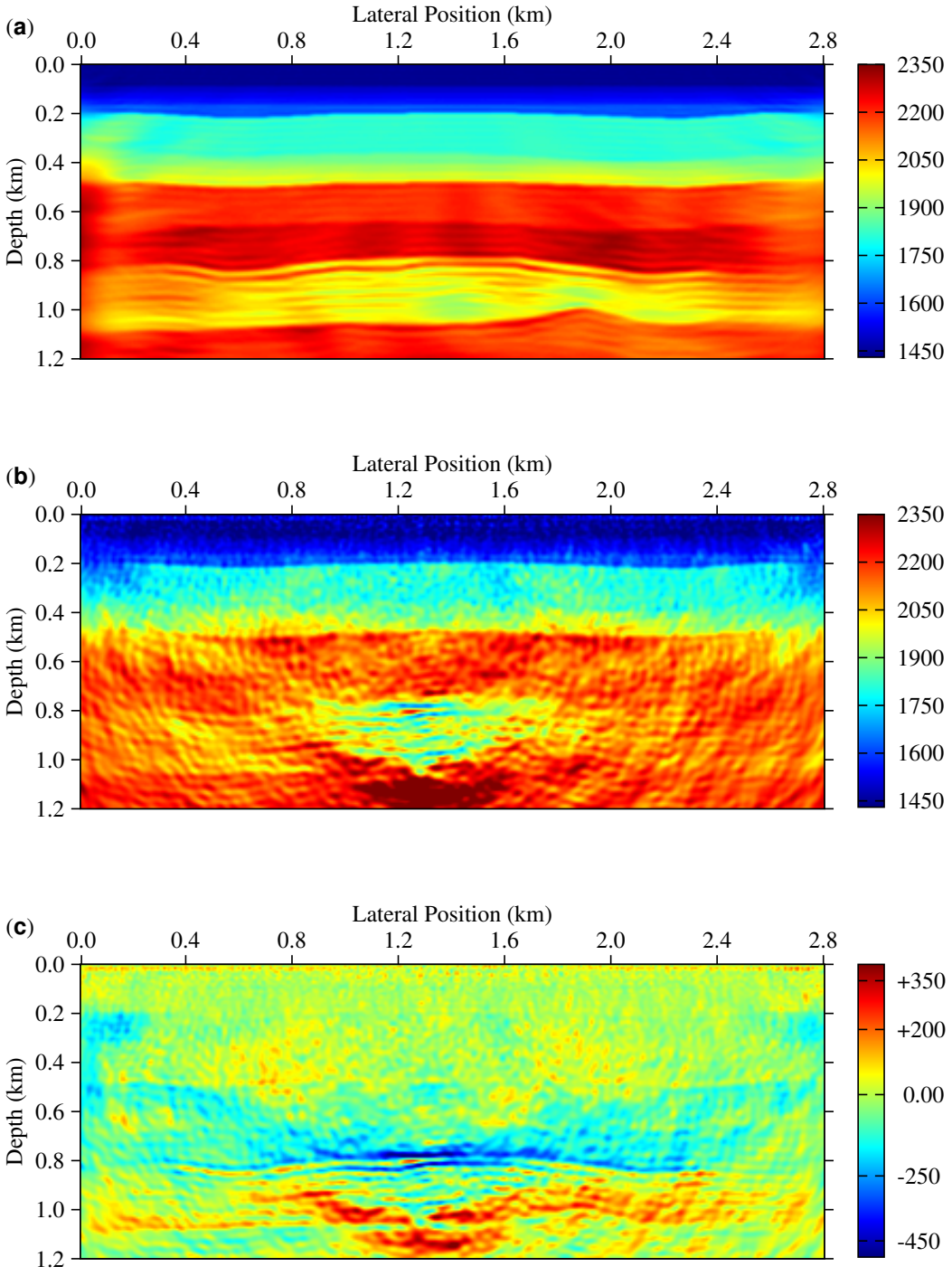


Figure 6.25: Data decimated (25%) example: inversion by the independent method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

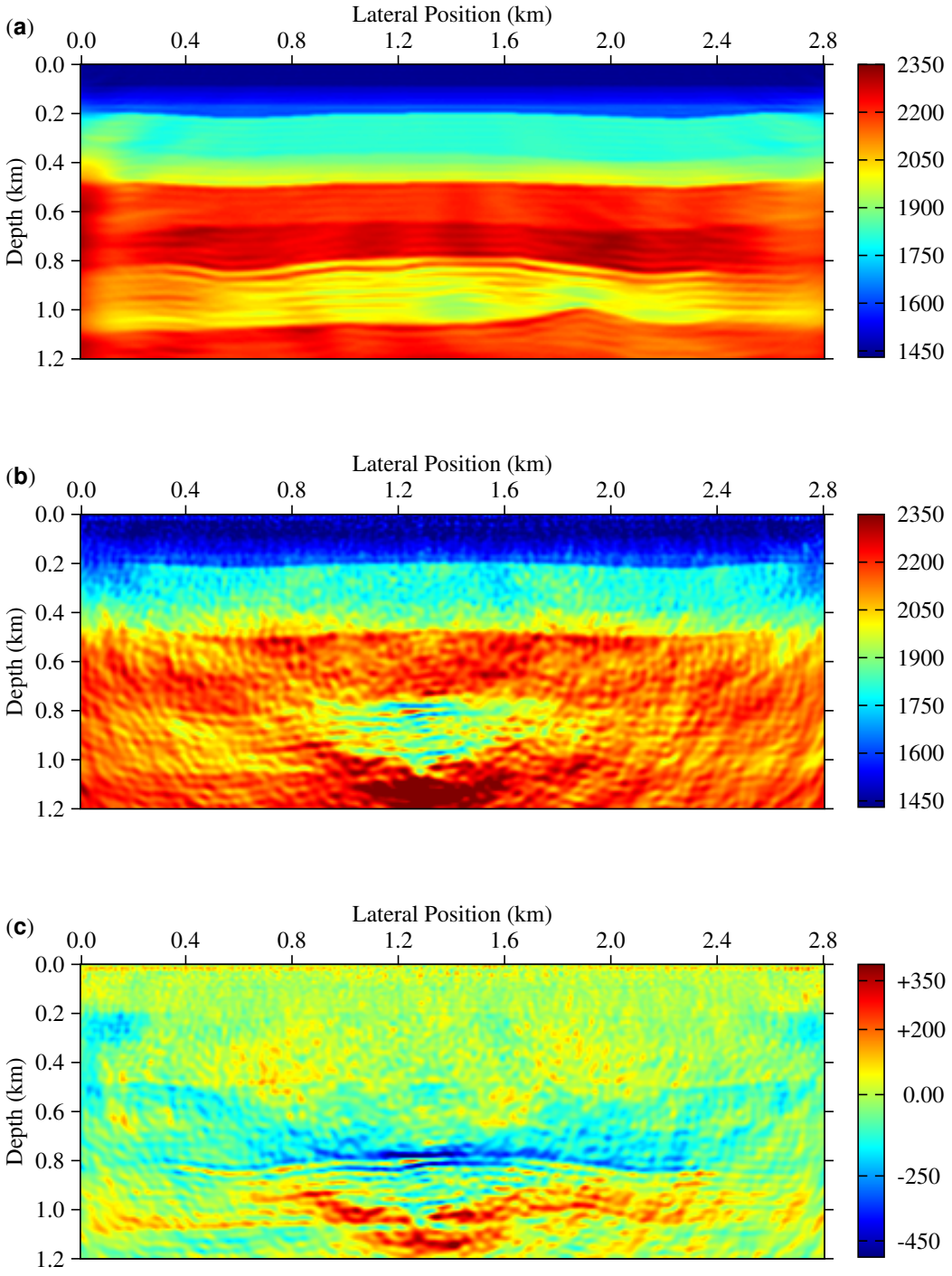


Figure 6.26: Data decimated (25%) example: inversion by the double difference method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

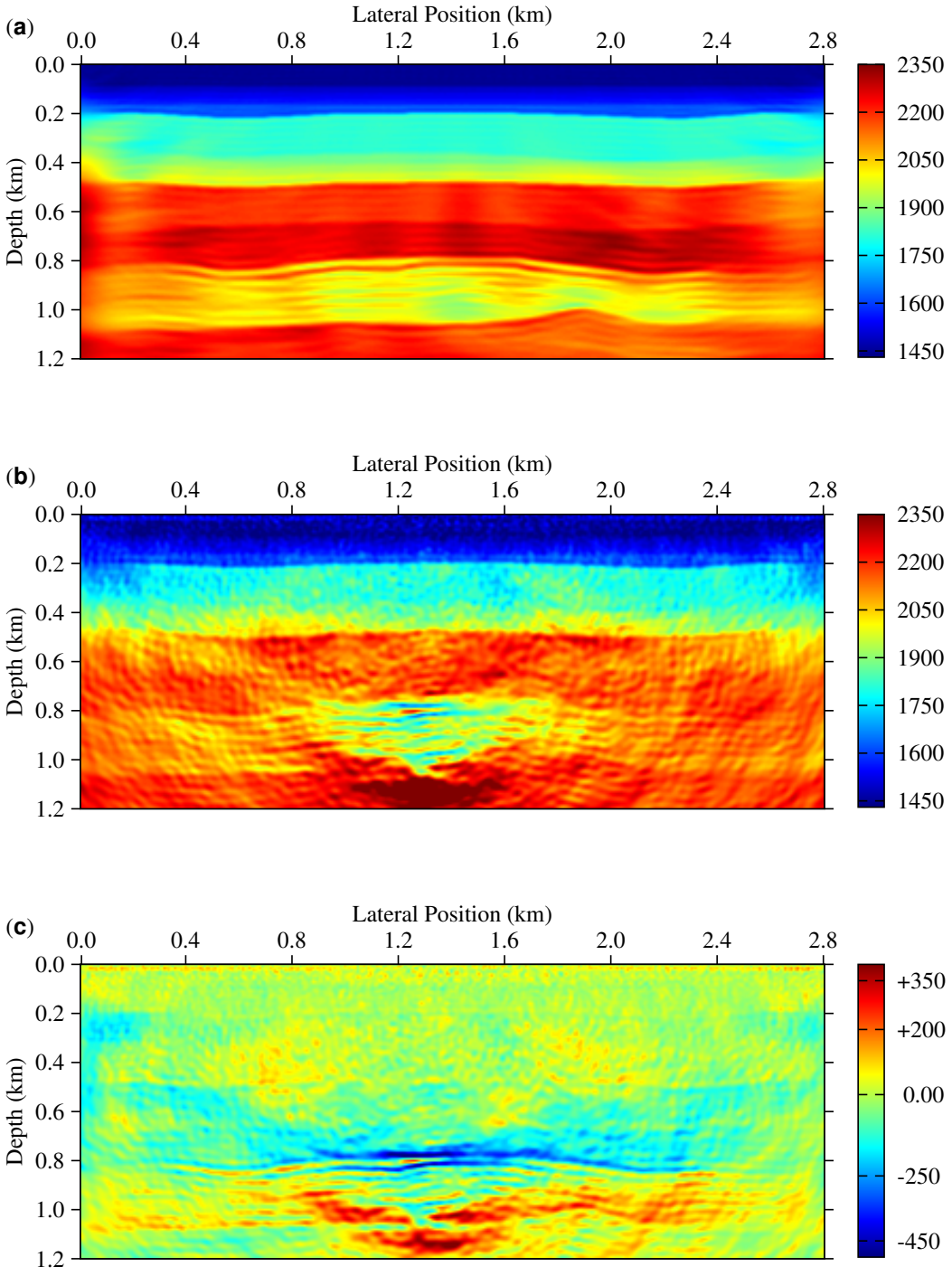


Figure 6.27: Data decimated (25%) example: inversion by the joint method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

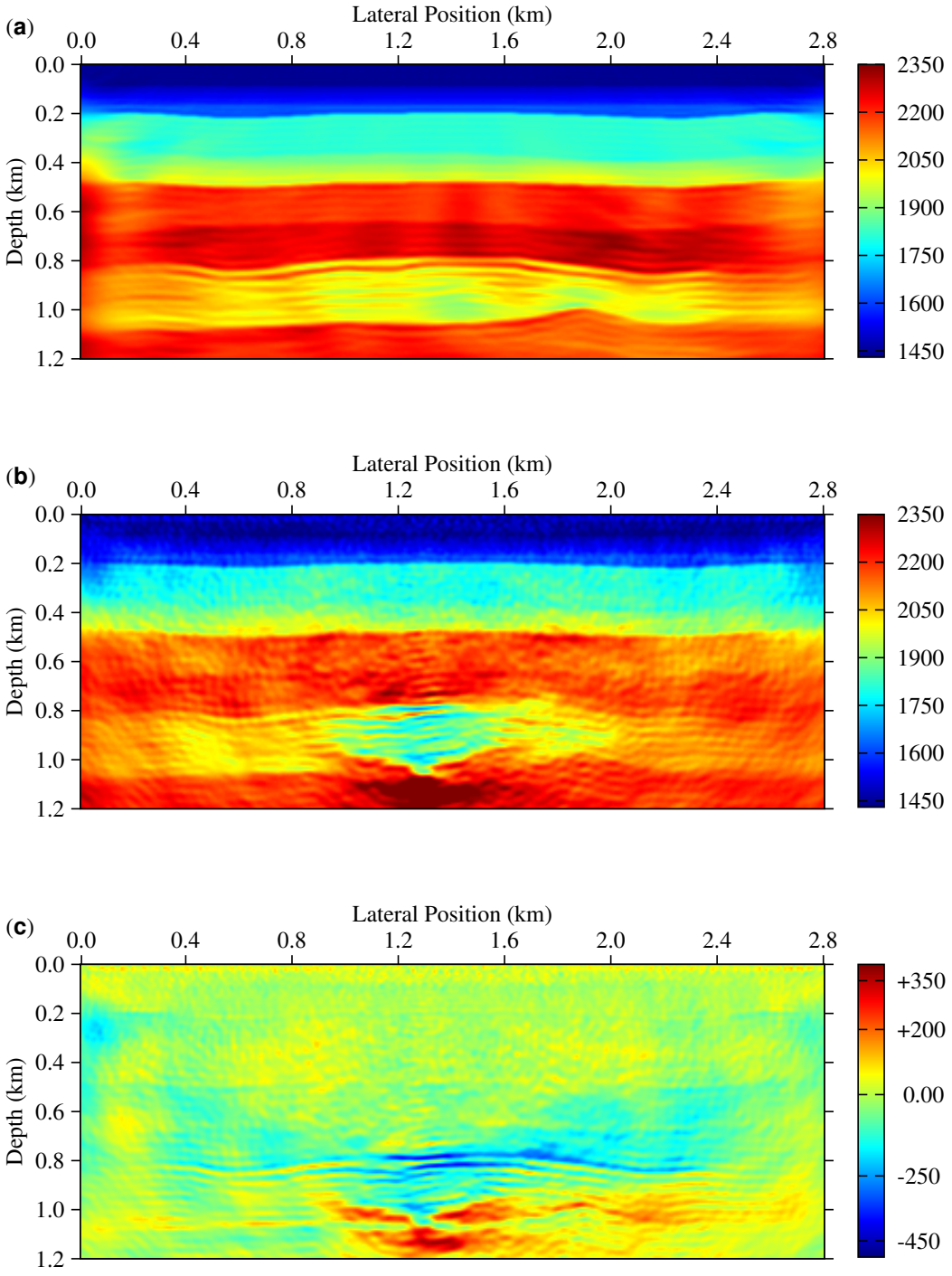


Figure 6.28: Data decimated (25%) example: inversion by the joint reparameterized method (**a**) baseline velocity, (**b**) monitor velocity and (**c**) the time-lapse difference.

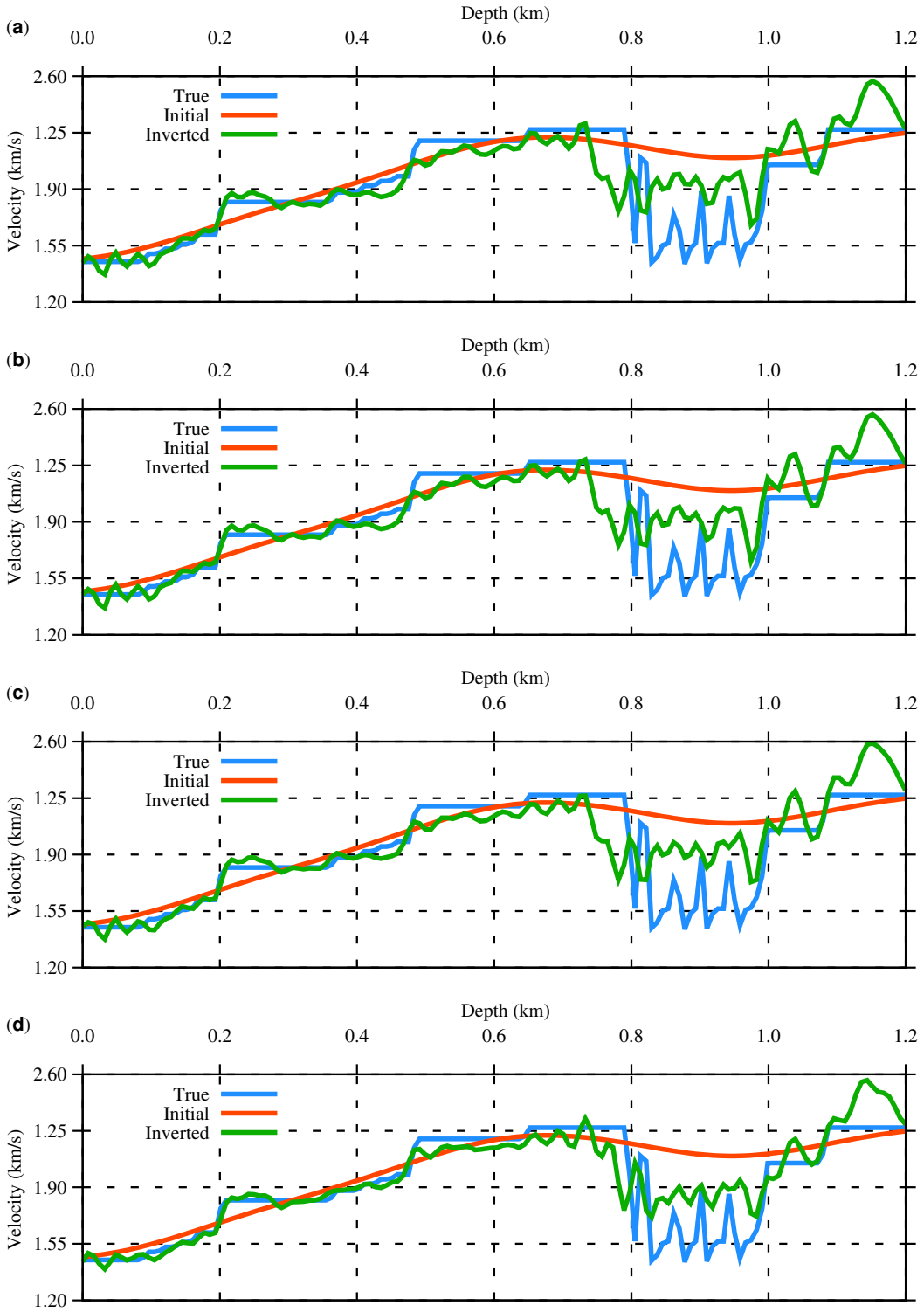


Figure 6.29: Data decimated (25%) example: comparison of vertical profiles of inverted monitor velocity with the true and initial velocity models (**a**) independent, (**b**) double difference, (**c**) joint, and (**d**) joint reparametrized.

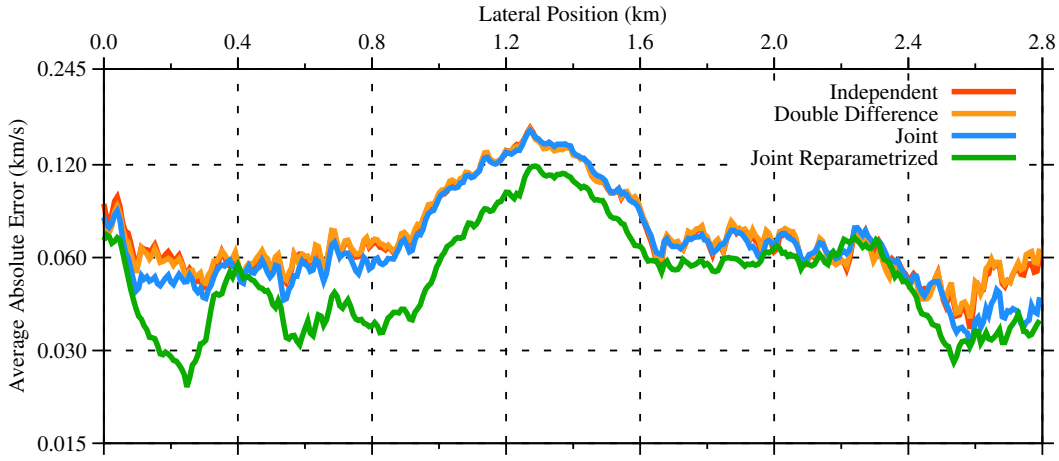


Figure 6.30: Data decimated (25%) example: comparison of vertical profiles of inverted monitor velocity with the true and initial velocity models (**a**) independent, (**b**) double difference, (**c**) joint, and (**d**) joint reparametrized.

Figures 6.31 (**a**), (**b**) and (**c**) are comparison of the behavior of the cost function and computational time for the four time-lapse inversion methods in non-repeatable case. The normalized CPU time is plotted against the number of iteration as shown in Figure 6.31 (**a**). Figure 6.31 (**b**) depicts the normalized cost function as a function of iteration. Figure 6.31 (**c**) shows the cost function as a function of normalized CPU time. The cost functions and the CPU time at each iteration are aggregates over the 14 frequencies.

6.5 Discussion and Summary

In this chapter, the synthetic CO_2 sequestration model for four various scenarios was considered. The velocity change in the monitor model follows the P-wave velocity change that occurs due to a realistic modification of the reservoir parameters using Gassmann's fluid substitution. It is important to mention that I have not taken into account detailed reservoir parameter changes such as temperature, pressure and rock frame. The substitution model was adopted to understand the percentage change in the P-wave velocity on account of the reservoir parameters given in Table 6.1 and to illustrate the time-lapse inversion methods. The velocity was modified at the target depth from approximately -5% to -23% as a result of increasing CO_2 saturation.

The first synthetic example is a hypothetical scenario where the experiments are repeatable i.e. same acquisition geometry for both the baseline and monitor cases. Artifacts due to

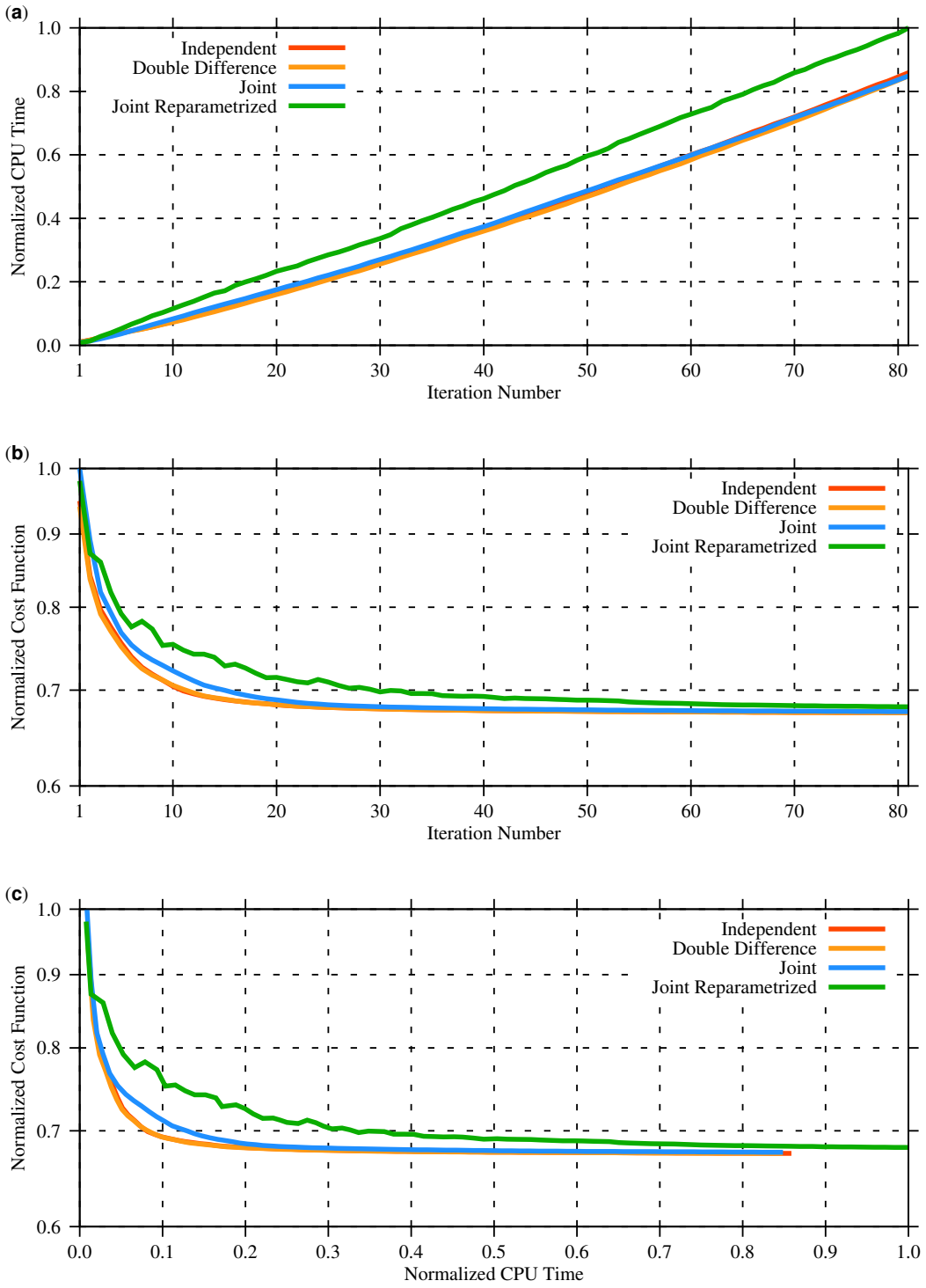


Figure 6.31: Data decimated (25%) example: (**a**) normalized computation time versus iteration number (**b**) normalized cost function versus iteration number, (**c**) normalized cost function versus normalized computation time. The vertical axes are plotted in log-scale.

non-repeatability are not expected; only minor computational artifacts are likely to show up. Thus, there is no much difference in performance among the time-lapse inversion methods as shown in the error analysis in Figure 6.9. In the second scenario, acquisition difference was incorporated by shifting ten shots each moved horizontally by the grid size, which results in acquisition discrepancy for source-receiver positions for the selected shots. In this examples shown in Figure 6.16, the performances of the time-lapse inversion methods are also relatively the same.

In the third scenario, there was no acquisition difference rather 10% of the monitor data were decimated. As depicted in the error analysis shown in Figure 6.23, the joint reparametrized inversion gives relatively improved result. In other words, this inversion technique tolerates the data decimation in comparison to the other three methods. Results in the fourth scenario, where 25% of the monitor data were decimated, also show similar performance. In the fourth scenario, the 25% decimation is too much; as a result, the overall time-lapse signature of the four inversion methods are not acceptable. But the joint reparametrized inversion has a better tolerance of the data discrepancy as compared to the independent, double difference and joint inversion methods.

The modified Marmousi model in chapter five has the higher complexity than the Sleipner CO_2 sequestration model covered in this chapter. Consequently, the performances of the time-lapse inversion methods are much clear in the Marmousi model than the Sleipner model. With simple acquisition difference, it was possible to show the quality that differentiates the four inversion methods. In the Sleipner model, the simple acquisition difference provides similar results from the four inversion methods. The data decimation examples are able to give a better comparison. The joint reparametrized method does provide improvement if the model has high complexity and high discrepancy in the monitor data due to acquisition difference or other sources of noise. If not the best, the proposed inversion method is an alternative to the existing traditional time-lapse inversion methods.

As pointed out in the theory, the joint reparametrized method acts as a regularization. It is a systematic way of putting the baseline model information to the monitor inversion. Hence, search direction towards the minimum is an average of the search direction of the baseline and monitor model; it takes relatively long time to reach a reasonable minimum. Furthermore, it prevents over-fitting to minimize time lapse velocity artifacts. Figure 6.10, Figure 6.17, Figure 6.24 and Figure 6.31 show this fact. In other words, the results obtained by joint reparametrized method honor both the prior information and the input data.

The objective of seismic monitoring is to detect and delineate the dynamics of fluid in a target reservoir. In particular, leak detection from subsurface CO_2 storage is the crucial part of CO_2 sequestration monitoring. An artifact-free inversion is necessary to quantify the fluid movement reliably. However, this can not be achieved with certainty due to limitations in

processing and inversion algorithms. In other words, the non-zero artifacts in the inversion results may jeopardize the correct interpretation of CO_2 contentment. Therefore, apart from developing efficient data seismic processing and inversion algorithms that can improve time-lapse signals, an interdisciplinary interpretation approach on the time-lapse results could help to minimize the risks.

CHAPTER 7

Conclusions

The goal of this thesis is to extend the traditional FWI algorithm to time-lapse inversion algorithm. In particular, the new time-lapse FWI algorithm namely JRTL FWI is proposed and compared to the existing time-lapse inversion algorithms such as independent inversion and double difference inversion with various numerical examples. To this end, acoustic wave equation modeling of seismic waves in the frequency domain with second order finite difference approximation was discussed in detail in chapter two. The effectiveness of PML boundary condition relative to ABC boundary condition was demonstrated with examples. I have shown that the PML boundary condition has superior performance over the ABC boundary condition. However, PML boundary condition is not perfect; although the amplitude of boundary reflection is negligible as compared to the main amplitude of interest. It implies that there is still room for improvement. The imperfection of the PML boundary condition could partly be a result of not assigning enough PML region and a proper choice of the decay function and parameters involved in its implementation.

Besides, the advantage of frequency domain modeling by the direct and efficient implementation of wave-field computation using the MUMPS solver was presented with examples. The computational benefit using the efficient method relative to the direct method is pronounced. It is a result of using a frequency domain modeling and LU decomposition. The frequency domain implementation helps to set up the wave-field computation as a system of linear equations. The LU decomposition allows factorizing the resulting forward operator matrix. Factorized matrix for same frequency and velocity can be used over and over again for a various number of sources. The given numerical examples show this advantage. The direct method for a different number of sources shows huge variation in computational time as the number of sources increase. It is due to the huge computational time in matrix factorization

for each source; the factorizing phase is computationally expressive than the solving phase. This computational advantage even becomes more important in the implementation of the forward modeling in FWI algorithm where a number of forward operations are needed to compute wave-field, gradient, and Hessian at each iteration.

Since the time-lapse FWI is based on the traditional FWI, the mathematical formulation in the least square sense was derived in chapter three. Migration via Born approximation relative to RTM was discussed. The relationship between RTM and FWI in frequency domain was established. The framework of FWI was used to test RTM with numerical examples. In particular, two approximate Hessian scaling namely diagonal Hessian and Gauss-Newton were tested. I have shown that the diagonal Hessian scaling is computationally inexpensive relative to the Gauss-Newton case because it does not need extra computational time as the diagonal Hessian can be computed along with the gradient. On the other hand, the Gauss-Newton approach uses the Conjugate Gradient algorithm to compute the search direction that needs two extra forward computation at each iteration. The solution using the conjugate gradient algorithm does not converge smoothly and hence takes a longer time. With the aid of the pre-conditioner, it is possible to speed up the convergence at the cost of its ability to sharpen the reflectors. It was shown that the smaller values of μ lead to nonsmooth curves that need a large number of iterations. As the values of μ increase, the square of the residual decreases linearly with fewer iterations. It was also noted in chapter three that the computation of the search direction at first iteration in the framework of full waveform inversion gives a migrated image. In RTM, the velocity is not updated instead the sum of search directions from various shots and frequencies provides the final migrated image.

In chapter four, four full waveform inversion optimization algorithms namely gradient descent, approximate diagonal Hessian, Gauss-Newton, and Quasi-Newton were reviewed. The performances of these algorithms were assessed with numerical examples by employing three velocity models with a different degree of complexity. Since the strategies to compute search direction using the diagonal Hessian and Gauss-Newton were discussed in Chapter 3, one more additional method namely the L-BFGS method was derived and reviewed in detail in chapter four. The inversion results with error analysis and computational time efficiencies of each optimization methods were analyzed with the behavior of cost function against iteration number and time. The first model is the 'Mac' model, which is a simple layered model with an anomaly. The second model is the Amoco model with trust structure that is more complex than the 'Mac' model. The third example is the Marmousi model, which has the higher degree of complexity as compared to the first two models.

From the three model examples, it is evident that the diagonal Hessian approach produces sharper velocity contrasts but substantial artifacts in high-velocity regions as compared

to the gradient descent. However, the vertical profile errors are significant because of the contribution of strong artifacts at higher velocity regions. On the other hand, the Gauss-Newton and the L-BFGS algorithms give better and competitive results. Furthermore, the L-BFGS algorithm for three different values of the number of maximum updates (Nm) was compared by calculating the average vertical profile error. The errors tend to decrease with increasing Nm. Given the large differences set in Nm, the improvements are marginal. The second class of examples is data with noise SNR = 30%. In these examples, only the L-BFGS method with smoothing and TV regularization was considered. For optimal values of regularization parameters (μ), the noise was able to be removed and no regularization artifacts. The average vertical profile errors show less error with the optimal parameter but significant errors for the smaller and larger values of μ. Using the smoothing regularization with a large value of μ, the velocity model becomes smoother, unlike the TV regularization that preserves the edges. From the comparisons of average vertical profiles for the optimal value of μ, it is apparent that both regularization methods reduce noise on the velocity image. However, TV regularization does perform better than the smoothing regularization in reducing error without compromising edges in the image.

The velocity images that can be obtained via the four optimizations are reliable under the condition that the velocity images do honor the data and the prior information which minimizes the defined objective function. Depending on the complexity of the model and data, the performance of each optimization methods differ. Given several variables problem, in some cases, the quality of the velocity image obtained by one method might be superior to the other. On the other hand, all the four optimization methods could also give similar velocity images as long as each minimization leads to a global minimum with their corresponding different search directions. One other important factor that hugely distinguishes one optimization method to the other is computational time.

From the computational time analysis for the three models, one can observe that the Gauss-Newton method takes an extremely long time to reach a minimum of the objective function as compared to gradient descent, diagonal Hessian, and L-BFGS. This observation is not surprising since the Gauss-Newton method uses the conjugate gradient method that needs at least two forward computations at each conjugate gradient iteration in addition to the forward computations required to compute source wavefield and gradient. Note that the conjugate gradient loop is within the main iteration at a given frequency. Besides, finding the optimal step length also puts more computational time burden. Unlike the Gauss-Newton method, the gradient descent, diagonal Hessian, and L-BFGS only need to use the efficient computation method to obtain source wave field and gradient. Thus, computational time burden in those three methods is the computation of source wavefield and gradient along with finding the optimal step length. The primary factor that separates the three methods is determining the optimal step length which needs one or more forward computation at

each iteration if the first step length fails to satisfy the Wolfe conditions. Depending on the accuracy of the Hessian scaling, the step length determines how far to go in that given direction to reach global minimum appropriately. It implies that finding the optimal step length for the gradient descent and diagonal Hessian which do not have a proper Hessian scaling like that of the Gauss-Newton and L-BFGS methods takes more time. As a result, they take small steps to wards the minimum. From the normalized cost function versus normalized computational time curves, one can observe that the L-BFGS algorithm reaches the best minimum with relatively shorter time than the three methods. It is surely at the cost of memory that depends on the size of the model and number of maximum updates. The effect of Nm on the behavior of the cost function against time was also considered for the three models. Increasing the number of maximum updates means incorporating more information to the Hessian from the gradients of the previous iterations. Thus, a larger value of maximum updates corresponds to relatively minimum cost function at a relatively small computational time as compared to the lower values. This fact can be observed from the examples although the differences are not significant.

The behavior of the cost function against computational time for the 'Mac', Amoco, and Marmousi models for noisy data using L-BFGS algorithm along with TV regularization were also presented. The velocity image obtained with a small value of μ results in over-fitting. In other words, the noise signature is visible in velocity image. In this case, the convergence is faster and gives relatively lower minimum cost function as compared to velocity images obtained by higher μ values. This finding is expected because the objective of adding regularization is to honor both the input data and the prior information (minimizing the total variation). Therefore, the optimal value of μ should give the desired result. But it takes a longer time to converge, and the evaluated cost function is a little higher than with smaller value of μ. All in all, the L-BFGS algorithm, if not best, is a very reasonable alternative optimization method for FWI in comparison to the Gauss-Newton method. Thus, L-BFGS algorithm is exploited in later chapters in the formulation of the time-lapse full waveform inversion algorithms.

In chapter five, four time-lapse inversion strategies namely independent inversion, double difference inversion, joint inversion, and joint reparametrized inversion were explored with three possible synthetic scenarios using the modified Marmousi velocity model. The strength of the artifacts on time-lapse signatures in the three synthetic examples is consistent with my expectations. In all cases, artifacts attributed to the nonlinear optimization procedure in time-lapse settings are expected. Different level of artifacts likely manifest on the velocity difference as the nature of the algorithms are not the same. In other words, a small velocity difference between the baseline and monitor results in differencing artifacts that have strength comparable to the signal of interest. It is on this basis that the idea of jointly inverting baseline and monitor data sets is important. In addition, non-repeatability brings more

artifacts that are responsible for intensifying differencing artifacts.

The four inversion strategies provided very close relative performance in both case 1 and case 2 repeatable experiment scenarios. The independent and joint inversions showed quite a similar degree of performance. The double difference inversion has the strongest artifacts, whereas the joint reparametrized inversion showed solution with the least energetic artifacts. In the synthetic case 3, the artifacts on the velocity differences are stronger as compared to those in case 1 and case 2. This outcome is expected for time-lapse data from non-repeatable surveys. Unlike case 1 and 2, the independent and joint inversion methods in case 3 showed results similar to those obtained via double difference inversion. In other words, both methods were unable to reduce non-repeatability artifacts. The joint reparameterized inversion outperformed all three methods as depicted in the provided error analysis and through visual inspection of velocity differences. The improvement using joint reparameterized inversion in case 1, case 2, and case 3 is due to the combined effect of the introduction of the same step length γ in the inversion process for the baseline and monitor parameters as well as the reparametrization technique. In other words, JRTL FWI assumes that time-lapse seismic data sets are strongly correlated.

It is also worth pointing out that the double difference method provided the best estimate of time-lapse signature on the target area. However, it yielded the strongest artifacts elsewhere. Strong artifacts were the consequence of time shifts in the data in conjunction with the use of different initial velocities for the baseline and monitor inversion which biases the monitor results. On the other hand, the joint reparameterized inversion provided a reasonable time-lapse signature at the target area and significantly reduced artifacts outside the target area.

Regarding the convergence and computational time, the cost function of the double difference inversion decreased faster than the cost function of other three methods. This observation is apparent as the data residual of the baseline inversion was subtracted in the definition of the cost function. In the joint inversion, the cost function decreased slower than in the other three methods. The joint reparametrized inversion shows faster convergence than the joint inversion and slower convergence than the independent and double difference inversion methods. This result can be explained by the existence of a trade-off between data fitting and the strong correlation assumption that was made to mitigate the artifacts on the velocity differences.

In chapter six, one typical synthetic model based on CO_2 sequestration was considered. The time-lapse inversion strategies that are covered in chapter five were implemented to test this model in detail with acquisition difference as well as data decimation that mimic non-repeatability. In the synthetic examples, the velocity change of the monitor model follows the P-wave velocity change that occurs due to a realistic modification of the reservoir

parameters using the Gassmann's fluid substitution model. The velocity was modified to decrease ranging from approximately 5% to 23% as a result of increasing CO_2 saturation.

The first synthetic scenario was a hypothetical scenario where the experiments are repeatable i.e. same acquisition geometry for both the baseline and monitor cases. In the second scenario, acquisition difference is incorporated by shifting ten shots each moved horizontally by the grid size, which results in acquisition discrepancy for source-receiver positions for the selected shots. In both the first and second scenarios, the performance of each time-lapse inversion methods is relatively the same.

In the third scenario, there is no acquisition difference rather 10% of the monitor data were decimated. In this case, the joint reparametrized inversion produced relatively improved results. In other words, this inversion technique tolerates the data decimation in comparison to the other three methods. Results in the fourth scenario, where 25% of the monitor data are decimated also showed similar performance. However, the 25% decimation is too much data decimation as a result, the overall results of the four inversion methods are not acceptable. But the joint reparametrized inversion has a better tolerance of the data discrepancy as compared to the independent, double difference and joint inversion methods.

As pointed out in the theory, JRTL FWI algorithm acts as a regularization technique. It is a systematic way of putting the baseline model information to the monitor inversion. Hence, search direction towards the minimum is an average of the search direction of the baseline and monitor model. As a result, it takes relatively long time to reach a reasonable minimum. In addition, it prevents over-fitting to minimize time lapse velocity artifacts. In other words, the results obtained by joint reparametrized method honor both the prior information and the input data.

The fundamental difference between the modified Marmousi model in chapter five and the Sleipner CO_2 sequestration model in chapter six was that the former has higher complexity than the later. Consequently, the performance of the time-lapse methods is much evident in the Marmousi model than the Sleipner model. With simple acquisition difference, it was possible to show the quality that differentiates the four inversion methods. In the Sleipner model, the simple acquisition difference results in similar results from the four inversion methods. The data decimation examples were able to give a better comparison. In general, the joint reparametrized method does provide improvement if the model has high complexity and a significant discrepancy in the monitor data due to acquisition difference or other noise sources. If not the best, the proposed inversion method is an alternative to the existing traditional time-lapse inversion methods.

The application of JRTL FWI is not limited to 2D time-lapse inversion; it is extendable to 4D inversion with more than two surveys, and the technique can also be adopted to elastic time-lapse inversion. It is worth pointing out that the proposed inversion method

takes advantage of the modified L-BFGS algorithm that uses the history of the difference of gradients. Although the idea of joint inversion approach can work in any optimization methods, joint reparametrized inversion may be challenging to implement via methods other than L-BFGS.

Future Directions

The main contribution of the thesis is the development of the Joint Reparameterized Time-Lapse Full Waveform Inversion algorithm. The algorithm was demonstrated with two synthetic models with a variety of scenarios that mimic the non-repeatability of seismic experiments. The examples indicate that the JRTL FWI improves the quality of the time-lapse velocity signatures as compared to the traditional time-lapse inversion algorithms. Although the synthetic examples have allowed testing the principle of time-lapse inversion, more realistic scenarios with exhaustive conditions should be tested. These include:

- coupling the synthetic modeling with rock physics and geo-mechanical simulation to mimic a more realistic time-lapse velocity change,
- increasing the number of time-lapse frames from two to three and more,
- testing the algorithm with a variety of data acquisition settings, and
- testing the algorithm with real time-lapse data sets in which seismic monitoring is reported successful.

In addition, if the acoustic version of the algorithm is proven successful with real data sets,

- extending the algorithm to joint reparametrized elastic time-lapse full waveform inversion and
- incorporating regularization techniques to the elastic time-lapse full waveform inversion that takes into account a prior information and correlation relation between the elastic parameters through a covariance matrix

are logical follow-ups towards a more practical application.

Bibliography

Adam, L., M. Batzle, and I. Brevik, 2006, Gassmann's fluid substitution and shear modulus variability in carbonates at laboratory seismic and ultrasonic frequencies: Geophysics, **71**, F173–F183.

Alemie, W., 2014, Time-lapse full waveform inversion application to seismic monitoring of CO_2 sequestration: 2014 CSEG CSPG CWLS Annual Convention. CSEG.

Alemie, W. and M. Sacchi, 2016, Joint reparametrized time-lapse full-waveform inversion, SEG Technical Program Expanded Abstracts 2016: Society of Exploration Geophysicists, 1309–1314.

Alford, R. M., K. R. Kelly, and D. M. Boore, 1974, Accuracy of finite-difference modeling of the acoustic wave equation: Geophysics, **39**, 834–842.

Amestoy, P. R., I. S. Duff, J. Koster, and J.-Y. L'Excellent, 2001, A fully asynchronous multifrontal solver using distributed dynamic scheduling: SIAM Journal on Matrix Analysis and Applications, **23**, 15–41.

Andrei, M., M. DeSimoni, A. Delbianco, P. Cazzani, and L. Zanibelli, 2010, Enhanced oil recovery with CO_2 capture and sequestration: Small Oilfield In Italy (Armatella) Report.

Armijo, L., 1966, Minimization of functions having lipschitz continuous first partial derivatives: Pacific J. Math., **16**, 1–3.

Arts, R., A. Chadwick, and O. Eiken, 2005, Recent time-lapse seismic data show no indication of leakage at the Sleipner CO_2 injection site: Proceedings of the 7th International Conference on Greenhouse Gas Technologies (GHGT-7).

Arts, R., O. Eiken, A. Chadwick, P. Zweigel, B. Van der Meer, and G. Kirby, 2004a, Seismic monitoring at the Sleipner underground CO_2 storage site (North Sea): Special Publication - Geological Society of London, **233**, 181 – 192.

Arts, R., O. Eiken, A. Chadwick, P. Zweigel, L. Vander Meer, and B. Zinszner, 2004b, Monitoring of CO_2 injected at Sleipner using time-lapse seismic data: Energy, **29**, 1383 – 1392. 6th International Conference on Greenhouse Gas Control Technologies.

Asnaashari, A., R. Brossier, S. Garambois, F. Audebert, and J. Virieux, 2013, Target-oriented time-lapse imaging using FWI with prior model information: 75th EAGE Conference & Exhibition incorporating SPE EUROPEC 2013. EAGE.

Ayeni, G. and B. Biondi, 2011, Wave-equation inversion of time-lapse seismic data sets: SEG Technical Program Expanded Abstracts, **30**, 4149–4154.

Bachu, S., D. Bonijoly, J. Bradshaw, R. Burruss, S. Holloway, N. P. Christensen, and O. M. Mathiassen, 2007, CO_2 storage capacity estimation: Methodology and gaps: International Journal of Greenhouse Gas Control, **1**, 430–443.

Bachu, S., K. Michael, and J. Adams, 2003, Effects of in situ conditions on aquifer capacity for CO_2 sequestration in solution: Proceedings of Second National Conference on Carbon Sequestration, 5–8.

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

Batzle, M., R. Hofmann, D.-H. Han, and J. Castagna, 2001, Fluids and frequency dependent seismic velocity of rocks: The Leading Edge, **20**, 168–171.

Batzle, M. and Z. Wang, 1992, Seismic properties of pore fluids: Geophysics, **57**, 1396–1408.

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

Benson, S. and C. Doughty, 2006, Estimation of field-scale relative permeability from pressure transient tests: Proceedings, CO_2 Site Characterization Conference, Lawrence Berkeley National Laboratory, Berkeley, California.

Berenger, J. P., 1994, A perfectly matched layer for the absorption of electromagnetic waves: J. Computational Physics, **114**, 185–200.

Berryman, J. G., 1999, Origin of Gassmann's equations: Geophysics, **64**, 1627–1629.

Bertsekas, D. P., 1982, Enlarging the region of convergence of newton's method for constrained optimization: Journal of Optimization Theory and Applications, **36**, 221–252.

——— 1999, Nonlinear Programming: Athena Scientific, 2nd edition.

Biot, M., 1956a, Theory of propagation of elastic waves in fluid-saturated porous solid Part I: low frequency range: Journal of the Acoustical Society of America, **28**, 168–178.

Biot, M. A., 1956b, Theory of propagation of elastic waves in a fluid-saturated porous solid: Higher frequency range.: Journal of the Acoustical Society of America, **28**, 179–191.

——— 1956c, Theory of propagation of elastic waves in a fluid-saturated porous solid: Low-frequency range.: Journal of the Acoustical Society of America, **28**, 168–178.

Born, M., 1923, Quantum mechanics of impact processes: Phys, 803–827.

Brain, J., M. Coogan, J. Floricich, H. Goudswaard, H. Hake, K. Hunt, and P. McDonnell, 2009, 4D imaging beneath a gas chimney for quick identification of development opportunities: 71st Conference and Exhibition, EAGE, Expanded Abstracts, **Y001**.

Brossier, R., 2011, Two-dimensional frequency-domain visco-elastic full waveform inversion: Parallel algorithms, optimization and performance: Computers and Geosciences, **37**, 444 – 455.

Brown, A. R., 2004, Interpretation of three-dimensional seismic data. sixth edition, AAPG

memoir 42, SEG investigations in geophysics: American Association of Petroleum Geologists, Annual Meeting, Denver, **Six Edition**.

Brown, R. J. S. and J. Korringa, 1975, On the dependence of the elastic properties of a porous rock on the compressibility of the pore fluid: Geophysics, **40**, 608–616.

Broyden, C. G., 1970, The convergence of a class of double-rank minimization algorithms 1. general considerations: IMA Journal of Applied Mathematics, **6**, 76–90.

Buland, A. and Y. El Ouair, 2006, Bayesian time-lapse inversion: Geophysics, **71**, R43–R48.

Burkhart, T., A. R. Hoover, and P. B. Flemings, 2000, Time-lapse (4D) seismic monitoring of primary production of turbidite reservoirs at south timbalier block 295, offshore louisiana, gulf of mexico: Geophysics, **65**, 351–367.

Calvetti, D., S. Morigi, L. Reichel, and F. Sgallari, 2000, Tikhonov regularization and the l-curve for large discrete ill-posed problems: Journal of computational and applied mathematics, **123**, 423–446.

Chadwick, A., V. Clochard, N. Delepine, K. Labat, S. Sturton, M. L. Buddensiek, M. Dillen, M. Nickel, A. L. Lima, G. Williams, F. Neele, R. Arts, and G. Rossi, 2010, 4quantitative analysis of time-lapse seismic monitoring at the Sleipner CO_2 storage operation: The Leading Edge, **29**, 170–177.

Chadwick, R., R. Arts, C. Bernstone, F. May, S. Thibeau, and P. Zweigel, 2008, Best Practice For The Storage of CO_2 In Saline Aquifers: British Geological Survey Occasional Publication.

Chadwick, R., R. Arts, O. Eiken, G. Kirby, E. Lindeberg, and P. Zweigel, 2004, 4D seismic imaging of CO_2 bubble at Sleipner Field, Central North Sea: Geological society, London, 311–320.

Chew, W. and Q. Liu, 1996, Perfectly matched layers for elastodynamics: A new absorbing boundary condition: Journal of Computational Acoustics, **04**, 341–359.

Chin, L., L. Thomas, J. Sylte, and R. Pierson, 2002, Iterative coupled analysis of geomechanics and fluid flow for rock compaction in reservoir simulation: Oil & Gas Science and Technology, **57**, 485–497.

Chu, D. and G. A. Gist, 2010, Inversion of 4D seismic data: Google Patents.

Claerbout, J. F., 1971, Toward a unified theory of reflector mapping: Geophysics, **36**, 467–481.

——— 1976, Fundamentals of geophysical data processing: McGraw-Hill.

Clayton, R. and B. Engquist, 1977, Absorbing boundary conditions for acoustic and elastic wave equations: Bulletin of the Seismological Society of America, **67**, 1529–1540.

Clochard, V., N. Delépine, K. Labat, and P. Ricarte, 2009, Post-stack versus pre-stack stratigraphic inversion for CO_2 monitoring purposes: a case study for the saline aquifer of the Sleipner Field: Society of Exploration Geophysicists.

Cole, S., D. Lumley, M. Meadows, and A. Tura, 2002, Pressure and saturation inversion of

4D seismic data by rock physics forward modeling: Society of Exploration Geophysicists.

Daley, T. M., L. R. Myer, J. E. Peterson, E. L. Majer, and G. M. Hoversten, 2008, Time-lapse crosswell seismic and VSP monitoring of injected CO_2 in a brine aquifer: Environmental Geology, **54**, 1657–1665.

Davidon, W. C., 1991, Variable metric method for minimization: SIAM Journal on Optimization, **1**, 1–17.

Davis, T. L., M. J. Terrell, R. D. Benson, R. Cardona, R. R. Kendall, and R. Winarsky, 2003, Multicomponent seismic characterization and monitoring of the CO_2 flood at weyburn field, saskatchewan: The Leading Edge, **22**, 696–697.

Denli, H. and L. Huang, 2009, Double-difference elastic waveform tomography in the time domain: SEG Technical Program Expanded Abstracts, 2302–2306.

Drossaert, F. H. and A. Giannopoulos, 2007, A nonsplit complex frequency-shifted pml based on recursive integration for fdtd modeling of elastic waves: Geophysics, **72**, T9–T17.

EIA, 2004, System for the analysis of global energy markets: International Energy Agency.

Eikeberg, O. and R. Elde, 2002, Time lapse & reservoir monitoring: The Leading Edge.

Engl, H. W., M. Hanke, and A. Neubauer, 1996, Regularization of inverse problems, volume **375**: Springer Science & Business Media.

Engquist, B. and A. Majda, 1977, Absorbing boundary conditions for numerical simulation of waves: Proceedings of the National Academy of Sciences, **74**, 1765–1766.

Fabriol, H., A. Bitri, B. Bourgeois, M. Delatre, J. Girard, G. Pajot, and J. Rohmer, 2011, Geophysical methods for CO_2 plume imaging: Comparison of performances: Energy Procedia, **4**, 3604 – 3611.

Fletcher, R., 1970, A new approach to variable metric algorithms: The computer journal, **13**, 317–322.

Gale, J. J. and P. Freund, 2001, Coal bed methane enhancement with CO_2 sequestration-worldwide potential: Environmental Geosciences, **8**, 71–72.

Gassmann, F., 1951, Elastic Waves Through a Packing of Spheres: Geophysics, **16**, 673.

Goldfarb, D., 1970, A family of variable-metric methods derived by variational means: Mathematics of computation, **24**, 23–26.

Guitton, A., B. Kaelin, and B. Biondi, 2006, Least-squares attenuation of reverse-time-migration artifacts: Geophysics, **72**, S19–S23.

Han, D.-h. and M. L. Batzle, 2004, Gassmann's equation and fluid-saturation effects on seismic velocities: Geophysics, **69**, 398–405.

Hansen, P. C., 1992, Analysis of discrete ill-posed problems by means of the l-curve: SIAM Review, **34**, 561–580.

Hashin, Z. and S. Shtrikman, 1963, A variational approach to the elastic behaviour of multiphase materials: Journal of the Mechanics and Physics of Solids, **11**, 127–140.

Helgerud, M. B., U. Tiwari, S. Woods, P. Homonko, A. Bucki, B. Laugier, E. Hicks, H.

Hoeber, and J. Khan, 2011, Optimizing seismic repeatability at ringhorne, ringhorne east, balder and forseti with qc driven time-lapse processing: Society of Exploration Geophysicists.

Hestenes, M. R. and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems, volume **49**: NBS.

Hill, R., 1952, The elastic behaviour of a crystalline aggregate: Proc. Phys. Soc. London Ser. A, **65**, 349–354.

Hitchom, B., 1984, Geothermal gradients, hydrodynamics, and hydrocarbon occurrences, Alberta, Canada: AAPG Bulletin, **68**, 713–743.

Holberg, O., 1997, Measurement strategies and techniques for seismic reservoir monitoring: Offshore Technology Conference.

Holloway, S., 1997, An overview of the underground disposal of carbon dioxide: Energy Conversion Management, **38**, S193–S198.

——— 2001, Storage of fossil fuel-derived carbon dioxide beneath the surface of the earth: Annual Review of Energy and the Environment, **26**, 145–166.

Holt, T., J.-I. Jensen, and E. Lindeberg, 1995, Underground storage of CO_2 in aquifers and oil reservoirs: Energy Conversion and Management, **36**, 535 – 538. Proceedings of the Second International Conference on Carbon Dioxide Removal.

Hovorka, S. D., S. M. Benson, C. Doughty, B. M. Freifeld, S. Sakurai, T. M. Daley, Y. K. Kharaka, M. H. Holtz, R. C. Trautz, H. S. Nance, L. R. Myer, and K. G. Knauss, 2006, Measuring permanence of CO_2 storage in saline formations: the frio experiment: Environmental Geosciences, **13**, 105–121.

Hu, W., A. Abubakar, and T. M. Habashy, 2009, Simultaneous multifrequency inversion of full-waveform seismic data: Geophysics, **74**, R1–14.

Hudson, J. A. and J. R. Heritage, 1981, The use of the born approximation in seismic scattering problems: Geophysical Journal International, **66**, 221–240.

Hustedt, B., S. Operto, and J. Virieux, 2004, Mixed-grid and staggered-grid finite-difference methods for frequency-domain acoustic wave modelling: Geophysical Journal International, **157**, 1269–1296.

Innanen, K. A., M. Naghizadeh, and S. T. Kaplan, 2014, Perturbation methods for two special cases of the time-lapse seismic inverse problem: Geophysical Prospecting, **62**, 453–474.

IPPC, 2001, Climate change 2001: Working group i: The scientific basis: Intergovernmental Panel on Climate Change.

——— 2005, Climate change 2005: Working group i: The scientific basis: Intergovernmental Panel on Climate Change.

Isaac, J. H. and D. C. Lawton, 2006, A case history of time-lapse 3D seismic surveys at cold lake, alberta, canada: Geophysics, **71**, B93–B99.

Jabbari, S., J. Wong, and K. A. Innanen, 2015, A theoretical and physical modeling analysis of the coupling between baseline elastic properties and time-lapse changes in determining difference amplitude variation with offset: Geophysics, **80**, N37–N48.

Jack, G. B. and R. O. Andrew, 2000, Exposing the 4d seismic time-lapse signal imbedded in the foinaven active reservoir management project: Offshore Technology Conference.

Jiang, H., 1999, Global Convergence Analysis of the Generalized Newton and Gauss-Newton Methods of the Fischer-Burmeister Equation for the Complementarity Problem: Mathematics of Operations Research, **24**, 529–543.

Johnson, J. W., J. J. Nitao, and K. G. Knauss, 2004, Reactive transport modelling of CO_2 storage in saline aquifers to elucidate fundamental processes, trapping mechanisms and sequestration partitioning: Geological Society, London, Special Publications, **233**, 107–128.

Johnston, D. H., 2013, Practical applications of time-lapse seismic data: Society of Exploration Geophysicists.

Kaelin, B. and A. Guitton, 2006, Imaging condition for reverse time migration: Society of Exploration Geophysicists, 2594–2598.

Kato, A. and R. R. Stewart, 2010, Joint AVO inversion for time-lapse elastic reservoir properties: Hangingstone heavy oilfield, alberta: SEG Technical Program Expanded Abstracts, 4432–4437.

Kloosterman, H. J., R. S. Kelly, J. Stammeijer, M. Hartung, J. van Waarde, and C. Chajecki, 2003, Successful application of time-lapse seismic data in shell expro's gannet fields, central north sea, UKCS: Petroleum Geoscience, **9**, 25–34.

Komatitsch, D. and R. Martin, 2007, An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation: Geophysics, **72**, SM155–SM167.

Koster, K., P. Gabriels, M. Hartung, J. Verbeek, G. Deinum, and R. Staples, 2000, Time-lapse seismic surveys in the north sea and their business impact: The Leading Edge, **19**, 286–293.

Kragh, E. and P. Christie, 2002, Seismic repeatability, normalized rms, and predictability: The Leading Edge, **21**, 640–647.

Kvalheim, A., I. Sandø, S. Skogland, V. Vinje, and M. Carpenter, 2007, Impact of time and depth imaging methods on quantitative 4D reservoir management: Presented at the 69th EAGE Conference & Exhibition.

Lafet, Y., B. Roure, P. Doyen, and H. Buran, 2009, 4D global seismic inversion and time-lapse fluid classification: Expanded Abstracts, 3830–3834.

Lailly, P., 1983, The seismic inverse problems as a sequence of before stack migration: Conference on inverse scattering, theory and application: Society of Industrial and Applied Mathematics, Expanded Abstracts, 206–220.

Landrø, M., 2006, Future challenges and unexplored methods for 4D seismic analysis: CSEG Recorder, **31**.

Landrø, M., H. Veire, K. Duffaut, and N. Najjar, 2003, Discrimination between pressure and fluid saturation changes from marine multicomponent time-lapse seismic data: Geophysics, **68**, 1592–1599.

Langston, M. V., S. F. Hoadley, and D. N. Young, 1988, Definitive CO_2 flooding response in the sacroc unit: Society of Petroleum Engineers SPE.

Lazaratos, S. K. and B. P. Marion, 1997, Crosswell seismic imaging of reservoir changes caused by CO_2 injection: The Leading Edge, **16**, 1300–1308.

Li, Y. E. and L. Demanet, 2016, Full-waveform inversion with extrapolated low-frequency data: Geophysics, **81**, R339–R348.

Lines, L. R., R. Slawinski, and R. P. Bording, 1999a, A recipe for stability of finite-difference wave-equation computations: Geophysics, **64**, 967–969.

——— 1999b, A recipe for stability of finite-difference wave-equation computations: Geophysics, **64**, 967–969.

Lumley, D., 1995, Seismic time-lapse monitoring of subsurface fluid flow:: Ph.D. Thesis, Standford University, Stanford, CA.

——— 2001, Time-lapse seismic reservoir monitoring: Geophysics, **66(1)**, 50–53.

——— 2010, 4D seismic monitoring of CO_2 sequestration: The Leading Edge, **29**, 150–155.

Ma, Y. and D. Hale, 2012, Quasi-newton full-waveform inversion with a projected hessian matrix: Geophysics, **77**, R207–R216.

Maharramov, M., B. Biondi, and M. Meadows, 2015, Simultaneous TV-regularized time-lapse FWI with application to field data: SEG Technical Program Expanded Abstracts, 1236–1241.

Marfurt, K. J., 1984, Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations: Geophysics, **49**, 533–549.

Marquardt, D. W., 1963, An algorithm for least-squares estimation of nonlinear parameters: Journal of the society for Industrial and Applied Mathematics, **11**, 431–441.

Martínez, J. M. and D. R. Schmitt, 2013, Anisotropic elastic moduli of carbonates and evaporites from the Weyburn-Midale reservoir and seal rocks: Geophysical Prospecting, **61**, 363–379.

Matter, J. M., T. Takahashi, and D. Goldberg, 2007, Experimental evaluation of in situ CO_2 -water-rock reactions during CO_2 injection in basaltic rocks: Implications for geological CO_2 sequestration: Geochemistry, Geophysics, Geosystems, **8**.

Mavko, G. and D. Jizba, 1991, Estimating grain scale fluid effects on velocity dispersion in rocks: Geophysics, **56**, 1940–1949.

Mavko, G. and T. Mukerji, 1995, Seismic pore space compressibility and Gassmann's relation: Geophysics, **60**, 1743–1749.

Mavko, G., T. Mukerji, and J. Dvorkin, 1998, The rock physics handbook: Cambridge.

McGillivray, P. and D. Oldenburg, 1990, Methods for calculating fréchet derivatives and sensitivities for the non-linear inverse problem: A comparative study: Geophysical Prospecting, **38**, 499–524.

McGrail, B. P., H. T. Schaef, Y. J. Chien, J. Dooley, and K. K. Humphreys, 2006, Potential carbon dioxide sequestration in flood basalts: Journal of Geophysical Research. Solid Earth, **B12**, 111–201.

Métivier, L., R. Brossier, Q. Merigot, E. Oudet, and J. Virieux, 2016, An optimal transport approach for seismic tomography: Application to 3D full waveform inversion: Inverse Problems, **32**, 115008.

Minkoff, S. E., C. M. Stone, S. Bryant, and M. Peszynska, 2004, Coupled geomechanics and flow simulation for time-lapse seismic modeling: Geophysics, **69(1)**, 200–211.

Nash, S. G. and J. Nocedal, 1991, A numerical study of the limited memory BFGS method and the truncated-newton method for large scale optimization: SIAM Journal on Optimization, **1**, 358–372.

Njiekak, G., D. R. Schmitt, H. Yam, and R. S. Kofman, 2013, CO_2 rock physics as part of the Weyburn-Midale geological storage project: International Journal of Greenhouse Gas Control, **16**, S118–S133.

Nocedal, J., 1980, Updating quasi-newton matrices with limited storage: Mathematics of computation, **35**, 773–782.

Nooner, S. L., O. Eiken, C. Hermanrud, G. S. Sasagawa, T. Stenvold, and M. A. Zumberge, 2007, Constraints on the in-situ density of CO_2 within the utsira formation from time-lapse seafloor gravity measurements: International Journal of Greenhouse Gas Control, **1**, 198–214.

O'Brien, J., F. Kilbride, and F. Lim, 2004, Time-lapse VSP reservoir monitoring: The Leading Edge, **23**, 1178–1184.

Oelkers, E. H., S. R. Gislason, and J. Matter, 2008, Mineral carbonation of CO_2: Elements, **4**, 333–337.

Pan, W., K. Innanen, G. Margrave, and S. Keating, 2016, Mitigate cycle-skipping for full-waveform inversion by band-limited impedance inversion and POCS: 2016 GeoConvention. CSEG.

Perkins, E., I. Czernichowski-Lauriol, M. Azaroual, and P. Durst, 2005, Long term predictions of CO_2 storage by mineral and solubility trapping in the weyburn midale reservoir: and solubility trapping in the Weyburn Midale Reservoir. Proceedings of the 7th International Conference on Greenhouse Gas Control Technologies (GHGT-7), **v.II**, 2093–2096.

Plessix, R. E. and W. A. Mulder, 2004, Frequency-domain finite-difference amplitude-preserving migration: Geophys J. Int, **157**, 975–987.

Pratt, R. G., 1999, Seismic waveform inversion in the frequency domain. part i: Theory and

verification in a physical scale model: Geophysics, **64**, 888–901.

Pratt, R. G., C. Shin, and G. J. Hick, 1998, Gauss-Newton and full newton methods in frequency-space seismic waveform inversion: Geophys. J Int., **133**, 341–362.

Pujol, J., B. Fuller, and S. Smithson, 1989, Interpretation of a vertical seismic profile conducted in the columbia plateau basalts: Geophysics, **54**, 1258–1266.

Queißer, M. and S. Singh, 2010, Time lapse seismic monitoring of CO_2 sequestration at Sleipner using time domain 2D full waveform inversion: SEG Technical Program Expanded Abstracts, **29**, 2875 – 2879.

Queißer, M. and S. C. Singh, 2013, Full waveform inversion in the time lapse mode applied to CO_2 storage at Sleipner: Geophysical Prospecting, **61**, 537–555.

Raef, A. E., R. D. Miller, E. K. Franseen, A. P. Byrnes, W. L. Watney, and W. E. Harrison, 2005, 4D seismic to image a thin carbonate reservoir during a miscible CO_2 flood: Hall-Gurney Field, Kansas, USA: The Leading Edge, **24**, 521–526.

Reuss, A., 1929, Berechnung der flieSSgrenze von mischkristallen auf grund der plastizitätsbedingung für einkristalle: ZAMM - Journal of Applied Mathematics and Mechanics - Zeitschrift für Angewandte Mathematik und Mechanik, **9**, 49–58.

Rhett, D. W., 1998, Ekofisk revisited: A new model of Ekofisk reservoir geomechanical behavior: Society of Petroleum Engineers.

Rickett, J. and D. Lumley, 2001, Cross-equalization data processing for time-lapse seismic reservoir monitoring: A case study from the gulf of mexico: Geophysics, **66**, 1015–1025.

Rigg, A. and J. Bradshaw, 2001, The geodisc program: Research into geological sequestration of CO_2 in australia: Environmental Geosciences, **8**, 73–74.

Rochelle, C. A., I. Czernichowski-Lauriol, and A. E. Milodowski, 2004, The impact of chemical reactions on CO_2 storage in geological formations: a brief review: Geological Society, London, Special Publications, **233**, 87–106.

Roden, J. A. and S. D. Gedney, 2000, Convolutional pml (CPML): An efficient FDTD implementation of the CFS-PML for arbitrary media: Microwave and optical technology letters, **27**, 334–338.

Rogers, K. L., P. S. Neuhoff, A. K. Pedersen, and D. K. Bird, 2006, CO_2 metasomatism in a basalt-hosted petroleum reservoir, nuussuaq, west greenland: Lithos, **92**, 55 – 82.

Ross, C., G. Cunningham, and D. Weber, 1996, Inside the crossequalization black box: The Leading Edge, **15**, 1233–1240.

Routh, P., G. Palacharla, I. Chikichev, and S. Lazaratos, 2012, Full wavefield inversion of time-lapse data for improved imaging and reservoir characterization: Society of Exploration Geophysicists.

Rubino, J. G., D. R. Velis, and M. D. Sacchi, 2011, Numerical analysis of wave-induced fluid flow effects on seismic data: Application to monitoring of CO_2 storage at the Sleipner field: Journal of Geophysical Research: Solid Earth, **116**, 2156–2202.

Rudin, L., S. Osher, and E. Fatemi, 1992, Nonlinear total variation based noise removal algorithms: Physica, **D**, 259–268.

Saeed, A.-N., 2014, Time−lapse AVO inversion: MSc thesis. University of Calgary.

Sakai, A., 2006, 4D seismic monitoring of the carbon dioxide geologic sequestration onshore in japan: Proceedings of the 8th International Workshop on Greenhouse Gas Technology,Trondheim, Norway.

Sarkar, D., A. Bakulin, and R. L. Kranz, 2003, Anisotropic inversion of seismic data for stressed media: Theory and a physical modeling study on berea sandstone: Geophysics, **68**, 690–704.

Sava, P. and S. Fomel, 2006, Time-shift imaging condition in seismic migration: Geophysics, **71**, S209–S217.

Sava, P. and I. Vasconcelos, 2011, Extended imaging conditions for wave-equation migration: Geophysical Prospecting, **59**, 35–55.

Schaback, R., 1985, Convergence analysis of the general Gauss-Newton algorithm: Numerische Mathematik, **46**, 281–309.

Schaef, H., B. McGrail, and A. Owen, 2010, Carbonate mineralization of volcanic province basalts: International Journal of Greenhouse Gas Control, **4**, 249 – 261.

Schmitt, D. R., 2015, Geophysical Properties of the Near Surface Earth: Seismic Properties, *in* Schubert, G., ed., Treatise on Geophysics, 43 – 87. Elsevier, Second edition.

Sei, A. and W. Symes, 1995, Dispersion analysis of numerical wave propagation and its computational consequences: Journal of Scientific Computing, **10**, 1–27.

Sen, V. and A. Settari, 2005, Coupled geomechanical and flow modeling of compacting reservoirs: The Leading Edge, **24**, 1284–1286.

Shah, N., M. Warner, T. Nangoo, A. Umpleby, I. Stekl, J. Morgan, and L. Guasch, 2012, Quality assured full-waveform inversion: Ensuring starting model adequacy, *in* SEG Technical Program Expanded Abstracts 2012, 1–5. Society of Exploration Geophysicists.

Shanno, D. F., 1970, Conditioning of quasi-newton methods for function minimization: Mathematics of computation, **24**, 647–656.

Sherman, J. and W. J. Morrison, 1950, Adjustment of an inverse matrix corresponding to a change in one element of a given matrix: The Annals of Mathematical Statistics, **21**, 124–127.

Shewchuk, J. R., 1994, An introduction to the conjugate gradient method without the agonizing pain: School of Computer Science, Carnegie Mellon University, Pittsburgh.

Shin, C., K. Yoon, K. J. Marfurt, K. Park, D. Yang, H. Y. Lim, S. Chung, and S. Shin, 2001, Efficient calculation of a partial derivative wavefield using reciprocity for seismic imaging and inversion: Geophysics, **66**, 1856–1863.

Sigit, R., P. Morse, and K. Kimber, 1999, 4d seismic that works: A successful large scale application, duri steamflood, sumatra, indonesia: SEG Technical Program Expanded

Abstracts 1999, 2055–2058.

Simbeck, D., 2004, CO_2 capture economics: Technologies and Policies for a low-Carbon Future, Pew Center on Global Climate Change and the national Commission.

Simon, C. P. and L. Blume, 1994, Mathematics for economists, volume **7**: Norton New York.

Smith, T. M., C. H. Sondergeld, and C. S. Rai, 2003, Gassmann fluid substitutions: A tutorial: Geophysics, **68**, 430–440.

Spetzler, J., Z. Xue, H. Saito, and O. Nishizawa, 2008, Case story: time-lapse seismic crosswell monitoring of CO_2 injected in an onshore sandstone aquifer: Geophysical Journal International, **172**, 214–225.

Stockwell, J., 2009, Geophysical image processing with seismic unix: Center for Wave Phenome.

Stoll, R. D., 1998, Sediment acoustics: Technical report, DTIC Document.

Stopin, A., P. J. Hatchell, C. Corcoran, E. Beal, C. Gutierrez, and G. Soto, 2011, First obs to obs time lapse results in the mars basin: Society of Exploration Geophysicists.

Strong, D. and T. Chan, 2003, Edge-preserving and scale-dependent properties of total variation regularization: Inverse Problems, **19**, S165.

Sun, Y. and G. Shuster, 1992, Hierarchic optimizations for smoothing and cooperative inversion, 745–748. SEG Technical Program Expanded Abstracts.

Tao, Y., M. K. Sen, R. Zhang, and K. T. Spikes, 2013, A new stochastic inversion workflow for time-lapse data: hybrid starting model and double-difference inversion: Journal of Geophysics and Engineering, **10**, 035011.

Tarantola, A., 1984a, Inversion of seismic reflection data in the acoustic approximation: Geophysics, 1259–1266.

——— 1984b, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266.

Tarantola, A. and B. Valette, 1982, Generalized nonlinear inverse problems solved using the least squares criterion: Rev. Geophys. Space Phys, **20**, 219–232.

Tikhonov, A., 1963, Solution of incorrectly formulated problems and the regularization method: Soviet Math. Dokl., **5**, 1035–1038.

Torp, T. A. and J. Gale, 2004, Demonstrating storage of CO_2 in geological reservoirs: The Sleipner and SACS projects: Energy, **29**, 1361 – 1369.

Van der Meer, J. W., 1998, Wave run-up and overtopping: Dikes and Revetments: Design, Maintenance and Safety Assessment, ed. KW Pilarczyk (AA Balkema, Rotterdam, The Netherlands), 145–159.

Vanorio, T., G. Mavko, S. Vialle, and K. Spratt, 2010, The rock physics basis for 4D seismic monitoring of CO_2 fate : Are we there yet?: The Leading Edge, **29**, 156–162.

Vedanti, N. and M. K. Sen, 2009, Seismic inversion tracks in situ combustion: A case study from balol oil field, india: Geophysics, **74**, B103–B112.

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

Vidal, S., A. Jardin, and F. Huguet, 2001, Feasibility study of time-lapse parameters estimate for mean effective stress and saturation changes in gas storage: SEG Technical Program Expanded Abstracts 2001.

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

Vogel, C. R. and M. E. Oman, 1996, Iterative methods for total variation denoising: SIAM Journal on Scientific Computing, **17**, 227–238.

Voigt, W., 1889, Ueber die Beziehung zwischen den beiden Elasticitàtsconstanten isotroper Kórper: Annalen der Physik, **274**, 573–587.

Watanabe, T., S. Shimizu, E. Asakawa, and T. Matsuoka, 2004, Differential waveform tomography for time-lapse crosswell seismic data with application to gas hydrate production monitoring: SEG Technical Program Expanded Abstracts, 2323–2326.

Watson, M., C. J. Boreham, and P. Tingate, 2004, Carbon dioxide and carbonate elements in the otway basin: implications for geological storage of carbon dioxide: The APPEA Journal, **44**, 703–720.

Watts, G., D. Jizba, D. Gawith, and P. Gutteridge, 1996, Reservoir monitoring of the magnus field through 4D time-lapse seismic analysis: Petroleum Geoscience, **2**, 361–372.

Wilson, M. and M. Monea, 2005, Iea ghg weyburn monitoring and storage project, summary report, 2000-2004. petroleum technology research center, regina sk, canada: Proceedings of the 7th International Conference on Greenhouse Gas Control Technologies (GHGT-7), Vancouver, Canada, **III**.

Wolfe, P., 1969, Convergence conditions for ascent methods: SIAM Review, **11**, 226–235.

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

Zhang, H. and C. H. Thurber, 2003, Double-difference tomography: The method and its application to the hayward fault, california: Bulletin of the Seismological Society of America, **93**, 1875–1889.

Zhang, Z. and L. Huang, 2013, Double-difference elastic-waveform inversion with prior information for time-lapse monitoring: Geophysics, **78**, R259–R273.

Zhdanov, M. S., 1993, Tutorial: Regularization in inversion theory: Colorado School of Mines.

Zheng, Y., P. Barton, and S. Singh, 2011, Strategies for elastic full waveform inversion of time-lapse ocean bottom cable (OBC) seismic data: SEG Technical Program Expanded Abstracts, 4195–4200.

Zhou, R., L. Huang, J. T. Rutledge, and Michael, 2010, Coda-wave interferometry analysis of

time-lapse VSP data for monitoring geological carbon sequestration: International Journal of Greenhouse Gas Control, **4**, 679 – 686.

Zweigel, P., R. Arts, T. Bidstrup, R. Chadwick, O. Eiken, U. Gregersen, M. Hamborg, P. Johansessen, G. Kirby, L. Kristensen, and E. Lindeberg, 2001, Results and experiences from the first industrial-scale underground CO_2 sequestration case, at Sleipner Field, central north sea: American Association of Petroleum Geologists, Annual Meeting, Denver, **6p**.

APPENDIX A

Adjoint State Method

A.1 Gradient via Adjoint State Method

The second approach for efficient gradient computation is using first order Adjoint state method via Lagrangian formulation. In Lagrangian formulation, the cost function defined by Equation 3.8 restated in a constrained optimization form,

$$\min_{m} \frac{1}{2} \sum_{j_{\omega_g}}^{n_{\omega_g}} \sum_{j_s}^{n_s} \Delta\mathbf{d}(\omega_{j_{\omega_g}}, X_{j_s})^{\dagger} \Delta\mathbf{d}(\omega_{j_{\omega_g}}, X_{j_s}), \ \textit{subject to } \mathbf{A}(\omega)\mathbf{u}_{j_s}(\omega) = \mathbf{s}_{j_s}(\omega), \quad \text{(A.1)}$$

The constraint in this case is the forward equation. The corresponding Lagrangian equation can be written as

$$L(\mathbf{m}, \mathbf{u}, \boldsymbol{\lambda}) = \frac{1}{2} \sum_{j_{\omega_g}}^{n_{\omega_g}} \sum_{j_s}^{n_s} \Delta\mathbf{d}(\omega_{j_{\omega_g}}, X_{j_s})^{\dagger} \Delta\mathbf{d}(\omega_{j_{\omega_g}}, X_{j_s}) + \Re(\mathbf{A}(\omega)\mathbf{u}_{j_s}(\omega) - \mathbf{s}_{j_s}(\omega), \boldsymbol{\lambda}), \quad \text{(A.2)}$$

where $\boldsymbol{\lambda}$ is the adjoint variable. If the solution to the forward equation is $\overline{\mathbf{u}}_m$, the Lagrangian L for this condition becomes equal to the $J(m)$ (cost function),

$$L(\mathbf{m}, \overline{\mathbf{u}}_m, \boldsymbol{\lambda}) = J(\mathbf{m}), \quad \text{(A.3)}$$

Taking the partial derivative of Equation A.3 with respect to $\mathbf{m}$, we have

$$\frac{\partial L(\mathbf{m}, \overline{\mathbf{u}}_m, \boldsymbol{\lambda})}{\partial \mathbf{m}} = \nabla_m J(\mathbf{m}), \quad \text{(A.4)}$$

Since $\mathbf{A}$ is explicitly depend on $\mathbf{m}$ and $\overline{\mathbf{u}}_m$ is an implicitly function of $\mathbf{m}$, Equation A.4, can be rewritten as

$$\frac{\partial L(\mathbf{m}, \overline{\mathbf{u}}_m, \boldsymbol{\lambda})}{\partial \mathbf{u}} \frac{\overline{\mathbf{u}}_m}{\partial \mathbf{m}} + \Re\left\{ \left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} \overline{\mathbf{u}}_m \right) \circ \left(\boldsymbol{\lambda} \right) \right\} = \nabla_m J(\mathbf{m}). \tag{A.5}$$

The symbol $\circ$ refers to element by element scalar product. The By defining adjoint state $\overline{\boldsymbol{\lambda}}$ such that the partial derivative of the Lagrangian function L with respect to $\mathbf{u}$ to be zero,

$$\frac{\partial L(\mathbf{m}, \overline{\mathbf{u}}_m, \overline{\boldsymbol{\lambda}})}{\partial \mathbf{u}} = \mathbf{0}. \tag{A.6}$$

Equation A.6 leads to the following result

$$\mathbf{A}^{\dagger} \overline{\boldsymbol{\lambda}} = R^T (\mathbf{d} - R^T \overline{\mathbf{u}}). \tag{A.7}$$

On account of Equations A.6 and A.7, the gradient can be put in a form

$$\nabla_m J(\mathbf{m}) = \Re\left\{ \left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}} \overline{\mathbf{u}}_m \right) \circ \left(\overline{\boldsymbol{\lambda}} \right) \right\}. \tag{A.8}$$

Equation A.8 implies the cross-correlation of the source wave-field and receiver wave-field, which is the same expression with the one obtained in Chapter 3 and shown in Equation 3.18.

A.2 Hessian via Adjoint State Method

Recall that we used the first order adjoint state method to derive the gradient for efficient computation. For Gauss Newton, both the gradient and the approximate Hessian use the first order adjoint approach. The first order adjoint approach, similar to the previous approach, helps us to efficiently compute the product of the approximate Hessian and a vector ($\mathbf{H}_a \mathbf{v}$). First we define the following

$$G_v(\mathbf{m}) = \mathbf{v}^T \nabla J \tag{A.9}$$

such that

$$\nabla G_v(\mathbf{m}) = \mathbf{H}_a \mathbf{v}. \tag{A.10}$$

Then, for an arbitrary $\mathbf{w}$ the problem is defined as a constrained optimization, that is

$$\min_m G_w(\mathbf{m}) \ = \mathbf{w}^T \mathbf{u}(\mathbf{m}) \ \textit{subject to} \ \mathbf{A}(\omega)\mathbf{u}_{j_s}(\omega) = \mathbf{s}_{j_s}(\omega), \tag{A.11}$$

such that the Gradient of $G_w(\mathbf{m})$ is

$$\nabla G_w(\mathbf{m}) = \Re\{\mathbf{J}_{jac}^{\dagger}\mathbf{w}\}. \tag{A.12}$$

The corresponding Lagrangian of the constrained optimization of Equation A.11 is given by

$$L(\mathbf{m}, \mathbf{u}, \boldsymbol{\eta}) = \mathbf{w}^T\mathbf{u} + \Re\{\boldsymbol{\eta}^T\big(\mathbf{A}(\omega)\mathbf{u}_{j_s}(\omega) - \mathbf{s}_{j_s}(\omega)\big)\}, \tag{A.13}$$

where $\boldsymbol{\eta}$ is an adjoint state variable. Employing the first order adjoint state approach, we have

$$\nabla G_w(\mathbf{m}) = \Re\{\boldsymbol{\eta}^{\dagger}\frac{\partial \mathbf{A}}{\partial \mathbf{m}}\mathbf{u}^*\} \tag{A.14}$$

with

$$\frac{\partial L_w(\mathbf{m}, \mathbf{u}^*, \boldsymbol{\eta}^*)}{\partial \mathbf{u}} = \mathbf{0}, \tag{A.15}$$

that result in

$$\mathbf{A}^T\boldsymbol{\eta}^* = -\mathbf{w}. \tag{A.16}$$

For $\mathbf{w}$,

$$\mathbf{w} = \mathbf{R}^T\mathbf{R}\mathbf{J}_{jac}\mathbf{v}, \tag{A.17}$$

where $\overline{\boldsymbol{\alpha}} = \mathbf{J}_{jac}\mathbf{v}$ is the solution of a forward problem with source represented by,

$$\boldsymbol{\Phi}_v(\mathbf{u}) = -\sum_{j=1}^{m}\left(\frac{\partial \mathbf{A}}{\partial m_j}v_j\right)\mathbf{u}. \tag{A.18}$$

On account of Equation A.17 and A.18, we have

$$\mathbf{A}^T\overline{\boldsymbol{\xi}} = -\mathbf{R}^T\mathbf{R}\overline{\boldsymbol{\alpha}}. \tag{A.19}$$

Then the action of the Hessian in Gauss Newton formulation onto an arbitrary vector $\mathbf{v}$ becomes

$$\mathbf{H}_a\mathbf{v} = \Re\left\{\left(\frac{\partial \mathbf{A}}{\partial \mathbf{m}_j}\overline{\mathbf{u}}(\mathbf{m})\right) \circ \left(\overline{\boldsymbol{\xi}}(\mathbf{m})\right)\right\}. \tag{A.20}$$

Equation A.20 is similar to Equation 4.6 derived in Chapter 3 except that A.20 represents product of the Hessian with a vector whereas Equation 4.6 is simply the Hessian.

APPENDIX B

Total Variation Regularization

B.1 Total Variation Regularization: Gradient

Let the penalty function for a total variation regularization be $J^{TV}(\mathbf{m})$ where $\mathbf{m}$ is a two dimensional model parameter $\mathbf{M}$ with size $(N_x+1)\times(N_z+1)$ rearranged in a vector form of size $(N_x+1)\times(N_z+1)\times 1$. The discrete form of total variation penalty function can be written as (Vogel and Oman, 1996),

$$J^{TV}(\mathbf{M}) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_z}\sqrt{(\mathbf{D1}^x_{i,j}\mathbf{M})^2 + (\mathbf{D1}^z_{i,j}\mathbf{M})^2 + \beta}, \tag{B.1}$$

where $\mathbf{D1}^x$ and $\mathbf{D1}^z$ are differential operator matrices such that

$$\mathbf{D1}^x_{i,j}\mathbf{M} = M_{i,j} - M_{i-1,j}, \qquad \mathbf{D1}^z_{i,j}\mathbf{M} = M_{i,j} - M_{i,j-1}. \tag{B.2}$$

Note that $\Delta x = \Delta z = 1$ is assumed for simplicity. The indices are $i = 0, 1, 2, ..., N_x$ and $j = 0, 1, 2, ..., N_z$. For fixed value of spacial steps Δx and Δz can be absorbed into the regularization parameter so that the gradient derivation can still be valid by letting the spacial steps to 1. Let $J^{TV}(\mathbf{M}+\tau\mathbf{V})$ be the penalty function for τ and arbitrary matrix $\mathbf{V}$. To find the gradient of the penalty function with respect to the model parameter $\mathbf{m}$, first the function $J^{TV}(\mathbf{M}+\tau\mathbf{V})$ is differentiated with respect to τ,

$$\frac{\partial J^{TV}(\mathbf{M}+\tau\mathbf{V})}{\partial\tau}|_{\tau=0} = 2\sum_{i=1}^{N_x}\sum_{j=1}^{N_z} q'_{i,j}(t_v)[(\mathbf{D1}^x_{i,j}\mathbf{M})((\mathbf{D1}^x_{i,j}\mathbf{V})) + (\mathbf{D1}^z_{i,j}\mathbf{M})((\mathbf{D1}^z_{i,j}\mathbf{V}))], \tag{B.3}$$

where $q'(t_v)$ is the derivative of a function $q(t_v)$,

$$q(t_v) = \sqrt{t_v^2 + \beta} \tag{B.4}$$

for $t_v^2 = (\mathbf{D1}_{i,j}^x \mathbf{M})^2 + (\mathbf{D1}_{i,j}^z \mathbf{M})^2 = (\mathbf{D}_{k,l}^x \mathbf{m})^2 + (\mathbf{D}_{k,l}^z \mathbf{m})^2$ is given by,

$$q'(t_v) = \frac{1}{\sqrt{t_v^2 + \beta}}. \tag{B.5}$$

Using the vector notations $\mathbf{m}$ and $\mathbf{v}$ as well as $\mathbf{D}^x$ and $\mathbf{D}^z$ which are large matrices with block matrix elements $\mathbf{D1}^x$ and $\mathbf{D1}^z$ respectively, Equation B.3 can be rewritten as

$$\frac{\partial J^{TV}(\mathbf{m} + \tau\mathbf{v})}{\partial \tau}|_{\tau=0} = 2(\langle \mathbf{QD}^x\mathbf{m}, \mathbf{D}^x\mathbf{v}\rangle + \langle \mathbf{QD}^z\mathbf{m}, \mathbf{D}^z\mathbf{v}\rangle), \tag{B.6}$$

where $\mathbf{Q}$ is a diagonal matrix with elements given by $q't_v$. The derivative of the penalty function$J^{TV}(\mathbf{m} + \tau\mathbf{v})$ with respect to τ and the resulting function evaluated $\tau = 0$ gives the following relationship between the gradient of interest $\mathbf{g}^{TV}$ with that of the arbitrary vector $\mathbf{v}$,

$$\frac{\partial J^{TV}(\mathbf{m} + \tau\mathbf{v})}{\partial \tau}|_{\tau=0} = \mathbf{v}^T\mathbf{g}^{tv}(\mathbf{m}). \tag{B.7}$$

On account of Equation B.6 and Equation B.7, one can show that

$$\mathbf{v}^T\mathbf{g}^{TV}(\mathbf{m}) = 2\mathbf{v}^T((\mathbf{D}^x)^T\mathbf{QD}^x\mathbf{m} + (\mathbf{D}^z)^T\mathbf{QD}^z\mathbf{m}). \tag{B.8}$$

Equation B.8 implies that

$$\begin{aligned} \mathbf{g}^{TV}(\mathbf{m}) &= 2[(\mathbf{D}^x)^T\mathbf{QD}^x\mathbf{m} + (\mathbf{D}^z)^T\mathbf{QD}^z\mathbf{m}] \\ &= 2[(\mathbf{D}^x)^T\mathbf{QD}^x + (\mathbf{D}^z)^T\mathbf{QD}^z]\mathbf{m} \\ &= 2L(\mathbf{m})\mathbf{m}. \end{aligned} \tag{B.9}$$

where

$$L(\mathbf{m}) = 2[(\mathbf{D}^x)^T\mathbf{QD}^x\mathbf{m} + (\mathbf{D}^z)^T\mathbf{QD}^z]. \tag{B.10}$$

B.2 Total Variation Regularization: Hessian

The Hessian of the penalty function J^{TV} is the first order derivative of the gradient $\mathbf{g}^{TV}$ given in Equation B.9 or Equation B.10. That is,

$$\frac{\partial \mathbf{g}^{TV}(\mathbf{m})}{\partial \mathbf{m}} = \mathbf{H}^{TV}(\mathbf{m}) = 2\frac{\partial}{\partial \mathbf{m}}\Big(L(\mathbf{m})\mathbf{m}\Big), \tag{B.11}$$

which leads to

$$\mathbf{H}^{TV}(\mathbf{m}) = 2\Big[L(\mathbf{m}) + \Big(\frac{\partial L(\mathbf{m})}{\partial \mathbf{m}}\Big)\mathbf{m}\Big]. \tag{B.12}$$

The first term in Equation B.12 is straight forward. Only the second term is to be determined. The second term has the following form

$$\begin{aligned}\Big(\frac{\partial L(\mathbf{m})}{\partial \mathbf{m}}\Big)\mathbf{m} &= (\mathbf{D}^x)^T\mathbf{Q}_1\mathbf{D}^x + (\mathbf{D}^x)^T\mathbf{Q}_2\mathbf{D}^z + (\mathbf{D}^z)^T\mathbf{Q}_3\mathbf{D}^x + (\mathbf{D}^z)^T\mathbf{Q}_4\mathbf{D}^z \\ &= (\mathbf{D}^x)^T\mathbf{Q}_1\mathbf{D}^x + (\mathbf{D}^z)^T\mathbf{Q}_4\mathbf{D}^z + (\mathbf{D}^x)^T\mathbf{Q}_2\mathbf{D}^z + (\mathbf{D}^z)^T\mathbf{Q}_3\mathbf{D}^x \end{aligned} \tag{B.13}$$

where

$$\begin{aligned}\mathbf{Q}_1 &= diag[2(\mathbf{D}^x\mathbf{m})^2 q''(t_v)], \\ \mathbf{Q}_2 &= diag[2(\mathbf{D}^x\mathbf{m})(\mathbf{D}^z\mathbf{m})q''(t_v)], \\ \mathbf{Q}_3 &= diag[2(\mathbf{D}^z\mathbf{m})(\mathbf{D}^x\mathbf{m})q''(t_v)], \\ \mathbf{Q}_4 &= diag[2(\mathbf{D}^z\mathbf{m})^2 q''(t_v)]. \end{aligned} \tag{B.14}$$

The last two terms in Equation B.13 introduces negative definiteness to the Hessian. Thus, ignoring these two terms, Equation B.13 becomes

$$\Big(\frac{\partial L(\mathbf{m})}{\partial \mathbf{m}}\Big)\mathbf{m} = (\mathbf{D}^x)^T\mathbf{Q}_1\mathbf{D}^x + (\mathbf{D}^z)^T\mathbf{Q}_4\mathbf{D}^z. \tag{B.15}$$

Plugging Equation B.10 and Equation B.15 into Equation B.12, the approximate Hessian of the total variation penalty function becomes,

$$\mathbf{H}^{TV}(\mathbf{m}) = 4[(\mathbf{D}^x)^T\mathbf{Q}\mathbf{D}^x + (\mathbf{D}^z)^T\mathbf{Q}\mathbf{D}^z] + 2[(\mathbf{D}^x)^T\mathbf{Q}_1\mathbf{D}^x + (\mathbf{D}^z)^T\mathbf{Q}_4\mathbf{D}^z]. \tag{B.16}$$

APPENDIX C

The Interpretation of the Joint Reparametrization

C.1 Interpretation

The proposed reparametrization technique for time-lapse full waveform inversion assumes that baseline and monitor velocity are similar. This assumption becomes important when the monitor data is contaminated with non-repeatability and other forms of noise. Let g_0 and g_1 be gradients with respect to m_0 and m_1, respectively. The gradient g_0 is computed via expression 3.18. Then we introduce a new parameter $\Theta = m_1 - m_0$, and we call its corresponding gradient g_Θ. Following the reparametrization technique discussed in the main body of the paper, the solution update at $k+1^{th}$ iteration is given by,

$$\begin{bmatrix} \mathbf{m}_{t_0} \\ \mathbf{\Theta}_{t_1} \end{bmatrix}^{k+1} = \begin{bmatrix} \mathbf{m}_{t_0} \\ \mathbf{\Theta}_{t_1} \end{bmatrix}^{k} + \gamma^k \begin{bmatrix} \mathbf{p}_{t_0} \\ \mathbf{p}_{\Theta_{t_1}}, \end{bmatrix}^{k} \tag{C.1}$$

where

$$\begin{bmatrix} \mathbf{p}_{t_0} \\ \mathbf{p}_{\Theta} \end{bmatrix}^{k} = - \begin{bmatrix} \mathbf{H}_{t_0}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\Theta}{}^{-1} \end{bmatrix}^{k} \begin{bmatrix} \mathbf{g}_{t_0} \\ \mathbf{g}_{\Theta} \end{bmatrix}^{k}. \tag{C.2}$$

Equation C.2 can be rewritten in terms of the original gradients, $\mathbf{g}_{t_0}$ and $\mathbf{g}_{t_1}$ as,

$$\begin{bmatrix} \mathbf{p}_{t_0} \\ \mathbf{p}_{\Theta} \end{bmatrix}^{k} = - \begin{bmatrix} \mathbf{H}_{t_0}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\Theta}{}^{-1} \end{bmatrix}^{k} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{g}_{t_0} \\ \mathbf{g}_{t_1} \end{bmatrix}^{k}, \tag{C.3}$$

where $\mathbf{I}$ is a block diagonal identity matrix. Let,

$$\mathbf{B} = \begin{bmatrix} \mathbf{H}_{t_0}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{\Theta}{}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix}. \tag{C.4}$$

Using Equation C.4 in Equation C.3, we obtain,

$$\begin{bmatrix} \mathbf{p}_{t_0} \\ \mathbf{p}_{\Theta} \end{bmatrix}^k = -\mathbf{B}^k \begin{bmatrix} \mathbf{g}_{t_0} \\ \mathbf{g}_{t_1} \end{bmatrix}^k. \tag{C.5}$$

Substituting Equation C.5 into Equation C.1 leads to the following expression,

$$\begin{bmatrix} \mathbf{m}_{t_0} \\ \mathbf{\Theta}_{t_1} \end{bmatrix}^{k+1} = \begin{bmatrix} \mathbf{m}_{t_0} \\ \mathbf{\Theta}_{t_1} \end{bmatrix}^k + \gamma^k \mathbf{B}^k \begin{bmatrix} \mathbf{g}_{t_0} \\ \mathbf{g}_{t_1}, \end{bmatrix}^k. \tag{C.6}$$

Equation C.6 can be interpreted as the update of the solution using the gradient descent method with a pre-conditioner matrix $\mathbf{B}$.

APPENDIX D

Error Calculations

D.1 Velocity Error

In every synthetic examples, the errors are calculated to compare various inversion strategies. The calculated vertically averaged absolute value represent horizontal profile error between a true velocity and inverted velocity models. Let a two dimensional true velocity model be V^t and a corresponding inverted velocity model be V^r. Let the size of the models be M by N. A representative horizontal profile error averaged vertically, EV^{avg}, can be calculated as,

$$EV_j^{avg} = \frac{1}{M}\sum_{i}^{M} |V_{i,j}^t - V_{i,j}^r|, \tag{D.1}$$

where i represents the rows (depth), j represents the columns (lateral), and E_j^{avg} is average errors for each column. This gives a one dimensional average horizontal profile error. In ideal case, where the true velocity and inverted velocity are exactly same, the average horizontal profile error is zero. Therefore, the degree of success for an inversion method is measured how the error values are close to zero.

D.2 Time-Lapse Velocity Error

In time-lapse inversion case, the horizontal profile error comparison between the true time-lapse velocity difference and the inverted time-lapse velocity difference are calculated similar

to Equation D.1. Let the true baseline velocity model be V^{t0}, the true monitor velocity model be V^{t1}, the inverted baseline velocity model be V^{r0} and the inverted monitor velocity model be V^{r1}. First the time-lapse velocity difference for each time-lapse frame are calculated.

$$DV^t_{i,j} = V^{t1}_{i,j} - V^{t0}_{i,j} \tag{D.2}$$

and

$$DV^r_{i,j} = V^{r1}_{i,j} - V^{r0}_{i,j}, \tag{D.3}$$

where DV^0 is the difference between the true and inverted baseline velocity models and DV^1 is the difference between the true and inverted monitor velocity models. Then, the horizontal profile error for the time lapse signature, EDV^{avg}, can be calculated as

$$EDV^{avg}_j = \frac{1}{M}\sum_i^M |DV^t_{i,j} - DV^r_{i,j}|. \tag{D.4}$$