SHAPE BASED METHODS FOR SEISMIC FULL-WAVEFORM INVERSION

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2019

Yifan Wu

School of Mathematics

Contents

Word count: 43250

List of Figures

Abstract

This thesis concentrates on a shape based approach for seismic full waveform inversion, especially a novel level set based shape estimation method for elastic waveform inversion. Full waveform inversion is a numerical data processing technique aiming at the reconstruction of subsurface structure of the earth using collected seismic reflective data. However, traditional techniques using an acoustic or Helmholtz wave equation as forward model are faced with the limitation of using simulated non-elastic wave data to numerically fit elastic waveform data, which is physically incorrect and practically prone to obtaining wrong estimates; the correct scheme for modelling seismic waves is using an elastic wave equation. We construct an elastic waveform inversion algorithm using a symmetric-hyperbolic scheme, and a time-reversal adjoint-state method; in addition, we introduce a Sobolev gradient method as a regularization method, with the goal to smooth the gradient function and thereby obtain more regular boundaries of the reconstructed shapes. However, the procedure of elastic waveform inversion is a multi-parameter estimation, which will lead to the numerical error of 'cross-talk'; this phenomenon is particularly severe in high-contrast situations and irregular shape boundary reconstruction problems, as expected to face in salt dome estimation problems. Therefore we introduce a shape based method using a level set technique to tackle specific seismic reconstruction problems, instead of more traditional pixel-based schemes. We also introduce a stochastic gradient descent method as an alternative to traditional gradient line search techniques for level sets, in order to increase efficiency and to avoid certain local minima in large-scale inversion problems. In addition, we introduce an additional integrated internal-value reconstruction scheme; this will prove to be an interesting and possibly necessary expansion of the shape based approach in order to deal with more realistic 3D seismic reconstruction problems.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including or administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420`), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see `http://www.library.manchester.ac.uk/about/regulations`) and in The University's policy on Presentation of Theses.

Acknowledgements

- I would like to thank my Supervisor, Oliver Dorn for guiding me in research, presenting in conferences and publishing during the four year session; he does not only help me with the project research as my supervisor, but also teaches me how to be an honest, decent and conscientious researcher in the future; this is even a more vital lesson than the work itself.

- I would also like to thank the University of Manchester, for providing me the school scholarship to support me with my postgraduate session, right after my master degree is finalized.

- I would like to thank my friended colleagues as well, who gave me important support during my PhD session: Aaron, Jake, Jonathan, Matthew, Puneet, Chico; thank you for allowing me to share an office and get on with our study together; even that we are not working on the same project, not even similar.

- I would also like to thank my distant colleagues in academia, whom I referred to in my project for help; from the authors in the reference, I would specially like to thank the following: Daniel Köhn, Brian Cox, Bradley Treeby, Jiri Jaros, Dominique Lesselier, Tristan Van Leeuwen, Ajinkya kadu, Rossmary Villegas, Alex Hiles, who offers either direct help for my research, or propose important ideas or inspiration of my work.

- Finally, I would like to thank to my family, including my father Wu Fang, my mother Li Kaili; my brothers and sisters, Li Ying, Jie Xiao, Wei Li, Li Ming, Wang Jing, Wu Tianyi, Zhang Ziyang, and all other family members. Being alone in UK, away from family is never easy, especially that the study of PhD is adds up the challenge. You not only support me financially with the cost of living, but also encourages me, and be there for me when I am down and fragile, which make me strong, and brave enough to overcome all the difficulties.

1 Introduction

Full-waveform inversion (FWI) is a data processing technique to reconstruct the quantitative characteristics of the sub-surface from seismic wave propagation data. The numerical estimation procedure often relies on minimising the misfit between the model data and real data. The procedure is to start with an initial guess on a parameterized model, recording the wave propagation data and calculating the misfit to the real data; then an update to the model is calculated with a line search technique, so that the misfit is reduced. By repeating the same procedure we obtain the final approximation of the true model.

The idea of full-waveform inversion has originally been developed by A. Tarantola in the 1980s; the inverse problem theory for acoustic and elastic waves were separately developed by a number of researchers, among them A. Tarantola [70] and J. Virieux [75]. They reported the various challenges of FWI; due to the restriction of seismic reflection data, the optimization does not always reconstruct the true sub-earth model adequately; the non-linearity of the inverse wave equation problem often gives rise to local minima in the process of model updating. Only with a sufficiently accurate initial model does the inversion provide a satisfactory approximation to the real sub-earth model. Therefore, different techniques of regularization have been developed to refine the model result [4], such as a penalty method [41], TV regularization [23], and others.

Historically, the numerical implementation of FWI first concentrated on acoustic and Helmholtz equations, for simplicity in modelling and numerical programming. Compared to that, the mathematical formulation of elastic wave equation contains more variables and parameters, therefore the numerical implementation requires more time and computing resources. In FWI which requires large scale, and hundred or thousands times of wave propagation, the application of elastic wave equations would increase the total computational time significantly. Therefore, much of the recent technical research on FWI still uses either acoustic or Helmholtz equation as the forward model instead of full elastics. However, in geophysical applications of those simplified models, the elasticity of the true earth material brings an inevitable problem namely that the modelled and true data are constructed by different wave propagation models, therefore the correct inversion becomes theoretically and practically impossible. Since the sub-surface real data is assumed to be obtained from elastic media, the elasticity model of

elastic waveform inversion becomes the only physically correct model.

The elastic wave equation could be constructed in time-domain or frequency-domain, the same way as in acoustics. The time-domain and frequency-domain model both have their advantages and disadvantages: in a frequency-domain model, the acoustic wave model could be formulated by a Helmholtz equation; since the model is time-independent, the time-step iteration scheme is avoided, thus reducing the computational costs significantly if only few frequencies are used.

Usually the seismic data are recorded as time series, the transformation of data in time-domain to frequency-domain needs to be done by computers which might introduce computational inaccuracies which might affect the accuracy of the reconstruction; on the other hand, in a time-domain inversion method the recorded data can be applied directly in the inversion process; this is the reason why we opted for using a time-domain based inversion method in our approach. It comes with the additional benefit that back-propagation (a standard concept in waveform inversion) is directly linked to physical time-reversal, as we will point out later in this thesis.

In order to put an emphasis on this time-reversal aspect, we will apply a symmetric hyperbolic first order system formulation of waveform inversion as proposed in [57] as an alternative model to traditional formulations, which often are based on second order models [26]. This also has some links to first order numerical implementations for elastic systems [11]; an additional advantage of applying this particular formulation from [57] is that it contains a pressure term in the model, which directly agrees with measured data by direct physical inspection. No data conversion is needed, which should be a clear advantage for its link to practical application in seismic imaging.

One of the principal complexities in elastic FWI is that it is a multi-parameter estimation procedure; therefore updates of different parameters might compensate their effects on the data for each other, and finally lead to cross-talk and artificial local minimum [50]. In both acoustic and elastic waveform inversion, some of the recent research tries to avoid this difficulty; some authors choose to only recover the velocity, and set density to be constant. In our approach, we choose a different simplifying approach which is based on geological structure assumptions. In many physical applications, the topological (or geological) struc-

ture of three parameters can be assumed to be closely correlated in the sense that geological shapes of different parameters share the same boundaries. Therefore, our idea is to propose a shape based method. As a starting point, we we assume that the background values of parameters in different layers are approximately known, and our objective is reduced to identify and characterize the shapes and boundaries of embedded objects or bodies of interest inside each layer; in this way, instead of recovering the entire pixel-based multi-parameter profile, we only need to estimate the shape and boundary of embedded objects; moreover, the internal parameters inside those bodies could be estimated accordingly, reducing the difficulty in multi-parameter estimation.

A very flexible numerical simulation techniques for shape evolution is the level set method. This technique has first been developed by S. Osher and J. Sethian for the modeling in computational physics and image processing [52], [51], [61]; its objective is to automatically incorporate topology changes during computational shape evolution; It has been applied in various mathematical problems, such as image processing, computational geometry, and biophysics. In imaging problems, sometimes the model parameters have a large contrast between objects, and the boundary between objects and background represent an irregular shape. Traditional pixel based full waveform inversion methods are not well-suited to reconstruct such models, even with added standard regularization terms. The goal is to improve object reconstruction by applying a level set function formulation in the modelling. Previously, this technique has already been successfully applied in electromagnetic tomography [17], optical tomography [65], [53] or history matching [16], amongst many others. In seismic geophysical imaging, the main application for shape estimation is the reconstruction of salt bodies, starting by [31]; more research on different types of level set techniques has also have been developed in [36]. Their research shows a successful level-set based reconstruction which relies on the simplified model of a Helmholtz equation. So far the gap between the elastic data and non-elastic forward modelling therefore still remains.

Therefore, our main interest of study is to build up a level set based elastic wave inversion model based on a 2D time-domain grid, which could be applied in geophysical problems like salt reconstruction. This method has been applied for reconstruction based on Helmholtz equation, but not applied in elastic wave

equation as a forward model [30], [31], [36]. Certainly, some form of regularization might still be necessary even in this model based approach. For regularization, we apply a Sobolev gradient [49] based method as an alternative of traditional L^2 gradient, used as a smoothing process of gradient calculation and numerical implementation. This will have the effect of producing shapes with more regular boundaries.

We will employ a Kaczmarz type reconstruction technique in our approach in order to deal with the usually vast amount of data available in seismic inversion. However, traditional line search techniques for optimization methods such as Armijo, Wolfe or strong Wolfe conditions applied for elastic waveform inversion would not be well-suited since they rely on knowledge to the full gradient, whose calculation is highly time-consuming; this limits its application for future large scale problems, especially in 3D. Therefore, we will develop a novel line search method more suited to single step (Kaczmarz) FWI and in addition consider a stochastic gradient descent method (SGD) [13] in order to solve this large scale FWI problem. The resulting method shows high similarity to Landweber-Kaczmarz type methods [29] applied in non-linear inverse problems [38], [12]. The SGD method not only proves to be less time consuming, but also avoids certain local minima which might occur when using a cyclic choice of partial data sets for gradient estimation. We expect that its main power will become invisible when moving to realistic 3D applications where many more data are usually obtained which cannot be included in a small number of data packages. SGD makes sure that the information content of the available data are explored in an optimal way, without unnecessary repetition of the same data subsets multiple times which might cause local minima. It also guarantees some form of unbiased data selection during the inversion process. 3D applications, however, need to be left to future research due to their highly increased complexity and the need for significantly more expensive computing resources, going beyond a PhD project.

Concluding, we apply a SGD method in our model for level set full wave form inversion, and apply a cross-validation misfit estimation technique that should be able to detect if there occurs overfitting for our model. We also design an additional approach for internal parameter value reconstruction, which is sometimes required to estimate the internal object value as well as the shape of salt domes. This will help to analyse its properties more precisely.

The structure of the remaining parts of our thesis is as follows:

Section 2 describes the theoretical formulation of 2D elastic waveform inversion, including the elasticity theory, the symmetric hyperbolic scheme, the adjoint-state method for gradient computation, the different line search techniques, and their corresponding inverse problem algorithms. We also introduce a regularization method that is based on Sobolev norm and Sobolev gradient, that is applied as a smoothing term for level set gradient, where we will apply this this smoothing technique in the following model tests in Section 4, 5 and 6, as a method of regularization; We finally introduce different methods of line search, and their pros and cons in full waveform inversion.

Section 3 describes a pseudo-spectral scheme that has been applied in the k-wave package [71], [11]; we actually will modify this numerical scheme for our forward problem for symmetric hyperbolic elastic wave propagation; we also present a numerical technique called PML so as to simulate the 'free-surface' condition. We need to consider the stability condition of elastic wave simulation such as CFL and numerical dispersion; Finally, we implement our numerical model following a numerical example that has been applied in SOFI2D [8] as to verify our wave simulation.

Section 4 introduces the numerical test for our example used in a pixel based scheme; we apply a high-contrast model that simulates a salt-dome condition. After the numerical test, we analyse the advantages and limitations of this schemes based on the obtained results; we also analyse the specific phenomenon called 'cross-talk' that occurs in multi-parameter estimation such as elastic wave inversion.

Section 5 introduces the idea of a level set representation used in elastic wave inversion. Firstly, we introduce the concept of a level set function, and then develop a level set based steepest descent method for inverse problems, and a narrow-band technique for extended velocity fields for the evolution [19]; we then apply this technique in our inverse model, and design the corresponding numerical tests for our case. Finally we compare the advantage of level set representation to pixel based model, and analyse its limitations at this stage.

Section 6 introduces a stochastic gradient descent (SGD) method that is applied in our level set based inversion; we apply a pixel-count control as novel line search strategy in a level set formulation to monitor the gradient descent, and

then we apply a large scale model test case for the level set evolution in Chapter 5, in order to analyse the advantage of SGD method in level set based inversion. We apply a hold-out validation (cross-validation) method for monitoring the data misfit for each sweep, which helps to detect whether our reconstruction scheme avoids over-fitting.

Section 7 considers a reconstruction of the actual contrast of the objects to the background, in particular for salt domes; considering a multi-parameter scheme, we make use of the expected correlation of interfaces of the related shapes and apply an independent update technique for all the model parameters, in order to simplify the parameter evolution, and use the same SGD method for the internal parameter estimation.

Section 8 summarizes the results of our research described in this thesis, and outlines some possible future research directions.

2 The 2D time-domain elastic waveform inversion

2.1 Introduction and methodology

As outlined above, FWI is a model reconstruction technique that uses the recorded seismic data to recover the sub-surface material structure. In geophysical application, researchers use air-guns to produce certain types of waves that propagate inside the earth; while these waves propagate, receiving devices like geophones are used to record the seismic data at discrete positions (usually located at the surface or inside observation wells) as required. Full waveform inversion has long been a key interest for seismic imaging; large oil companies like BP, Shell, ExxonMobil or others are all building up their own research teams for high-dimension, large-data seismic imaging research, that is applied for oil and gas exploration.

We need to build up a model to facilitate the parameter estimation of the earth, given the information of source and receivers, and the recorded data. The FWI methodology proceeds as follows: first, we set up the forward model that simulates the wave propagation, where the wave propagation parameters and variables should provide quantitative characteristics of the experiment, like wave velocity, density, pressure; we assume a starting model parameter distribution, and implement the wave propagation on the starting model to obtain the model data. Then we calculate the residual of the data, where we are able to obtain the adjoint variable from adjoint-state equation using the Lagrange multiplier from constrained optimization; in full waveform inversion, this procedure is also known as backward propagation, since the adjoint state equation can be considered as a computer realization of a time-reversal wave propagation [74]. We then compute the gradient based on the adjoint variable with the method called adjoint-state method [55]; we apply a Sobolev regularization technique for smoothing the gradient, as a refinement technique for numerical implementation. Finally we propose a line search on the estimated parameter update, which results in an updated new guess for the unknown parameters; we repeat the above procedure till we finally obtain a sufficiently accurate data fit.

The structure of Chapter 2 is designed as follows: We start introducing the

elasticity theory before we formally construct our model, and then demonstrate the difference between acoustic and elastics, so to explain the importance of applying an elastic wave equation for seismic imaging. Then we introduce the basic model formulation for the forward and inverse problem. We then introduce the elastic wave equation in 2D, and derive its formulation in symmetric hyperbolic form, which will be our forward model; we will also explain the reason for choosing a symmetric hyperbolic form instead of traditional settings [26]. We apply this in the governing form of a least-squares optimization scheme, resulting in a time-reversal form of adjoint-state equation in a 'free-surface' condition; then we calculate the gradient using the adjoint variable. We later introduce the idea of Sobolev norm and Sobolev gradient, considered as an alternative gradient form for smoothing the data. Finally, we introduce different methods for the required line search, compare their advantages and disadvantages, and then show how we will be applying these methods into our model.

2.2 Elasticity theory

The forward problem in full waveform inversion is generally formulated as a wave equation problem. The most widely used wave equations are acoustic, Helmholtz and elastic wave equations. Many of the previous researchers concentrate on acoustic and Helmholtz equations due to their simplicity in numerical implementations; however, since the earth has elastic properties, it is generally accepted that the application of a elastic wave equation for forward modelling is vital when dealing with real data. Therefore we apply an elastic wave equation as our forward model. We will do this in 2D due to computational limitations as part of such a PhD project, mentioning however that general results are expected to carry over with only minor modifications to more realistic 3D situations (requiring more expensive computational environments for practical calculations). Regardless whether 2D or 3D is considered, we need to first understand the physics of elasticity for our model construction, which briefly will be outlined in the following for the convenience of the reader. More details can be found in the standard literature on elastic wave modelling.

In FWI, wave energy travels through the earth subsurface as seismic waves; seismic waves can be a result of artificial or natural phenomena like earthquakes, volcanic eruptions, landslides, or any sort of man-made releases of energy. Elastic

wave propagation is the type of energy propagation that travels through an elastic material or fluid, or its surfaces, without causing permanent structural or physical changes. Examples of elastic waves are: waves travelling through water, sound travelling through air, or elastic energy moving through solid materials, such as artificial materials or the earth subsurface. When seismic wave travels through the earth subsurface, it carries with it the energy to deform the medium it passes through; the medium then produces the force to resist the change, so that the medium will recover to its original shape; this property of the medium is called elasticity.

The formulation of seismic wave propagation is complicated. Normally, we divide seismic waves into body waves and surface waves. A surface wave is a type of wave that travels along the surfaces of the earth, and it diminishes quickly along the surface. Therefore there remains little energy when those waves reach the distant receivers; our main focus is therefore on the body waves, which travel through the interior of the earth. A body wave is a combination of many types of waves; and the two main parts of a body wave is the pressure wave, also known as primary wave, or P-wave; and the shear wave, also known as secondary wave, or S-wave.

Therefore, a P-wave is a compressional wave that travels longitudinal in nature, indicating that the displacement of the wave is in the same direction as, or opposite direction to, the wave travel direction. Compressional waves travel through any material made up of gas, liquids, or solids. Examples are sound waves or acoustic wave. P-waves travel fastest in a seismic environment (which is the main reason to often call it primary waves, since they usually arrive first at the receivers).

The S-wave is a type of transverse waves, indicating that the displacement is perpendicular to the wave travel direction. Unlike P-waves, S-waves do not travel in gases or fluids, but only in truly elastic media. In most elastic media, the speed of S-waves is approximately 60% of that of the P-waves; That is the reason why they are also often called secondary waves, since they usually arrive after the P-waves at the seismic receivers.

2.3 Model set

In this section we will provide some notation and prerequisites used in our model of FWI.

2.3.1 Time and space

We define Ω and $\mathcal{T}$ to be the model space and model time range; in physical applications, Ω can be defined in two or three dimensions, depending on the computational setup used; and $\mathcal{T}$ is normally defined as $\mathcal{T} = [0, T]$, whereas T represents the maximum model time.

2.3.2 Source and receiver notation

We define $\mathcal{S}$ to be set of wave source locations, usually produced by air-guns; and $s \in \mathcal{S}$ represents one single wave source with starting point at one single air-gun's location. Comparatively, we denote the set of all receivers as $\mathcal{R}$, and $r \in \mathcal{R}$ defines an individual receiver.

2.3.3 Variables

We define the dynamic wave-field model variables as $\boldsymbol{w} = (w_1, w_2, ..., w_n) \in U$, considering that each component $w_i = w_i(\boldsymbol{x}, t)$ with $i = 1, 2, ..., n$ satisfies that $w_i \in \Omega \times \mathcal{T}$. Then we have the variable set $\boldsymbol{U} = (\Omega \times \mathcal{T})^n$.

2.3.4 Parameter

We define $\mathcal{M}$ as the set of FWI static medium parameters m, such that $m \in \mathcal{M}$. Each element m should be considered as a space dependent function defined at each point of Ω. In the level set formulation, we will modify this assumption and instead use shapes and contrast values as model parameters.

2.3.5 Data space

The seismic data is usually recorded at different receiver locations throughout physical time. We define the data space $\mathcal{D}$, satisfying that $d = (d_1, d_2, ..., d_n)$ and typically recording all or part of the dynamic model components, or simple functions of those. Each component satisfies that $d_i = d_i(\boldsymbol{x}_r, t)$, where we apply

$\boldsymbol{x}_r$ to define the location of the receiver r, $\boldsymbol{x}_r \in \mathcal{X}_r$ is the set of data locations satisfying that $\mathcal{R} \subseteq \Omega$. This gives us the data space $\mathcal{D} = (\mathcal{X}_r \times \mathcal{T})^n$.

We also define the generalised variable-to-data mapping $\boldsymbol{P} : U \to \mathcal{D}$. The choice of mapping will be introduced in a later chapter.

2.3.6 Abstract wave equation formulation

Based on the previous setting, We define the general wave equation formulation as

$$\mathcal{L}(m)\boldsymbol{w} = \boldsymbol{q} \tag{1}$$

where $w = (w_i) \in \boldsymbol{U}$ is the partial differential variable, where $w_i \in \Omega \times \mathcal{T}$ for $i = 1, 2, ..., n$, and that $\boldsymbol{U} = (\Omega \times \mathcal{T})^n$ is defined as the variable space. $\mathcal{L}$ defines the partial differential operator, and $\boldsymbol{q}$ is defined as the vectorized source term satisfying that $\boldsymbol{q} = (q_1, q_2, ..., q_n)$, whereas for all i, $q_i \in \Omega \times \mathcal{T}$.

2.4 Elastic wave equation formulation

2.4.1 Generalized elastic wave equation

In this chapter we formulate the general equations for elastic wave propagation. We consider the situation when a force is applied to a continuum, where every point in such a continuum is influenced by this force. The force can be divided into internal and external forces. The external force, also known as body force, will lead to a deformation of the medium, which will result in the shape deformation. The internal force, also known as the surface force, will resist the deformation, and try to recover the shape towards the initial condition. In an elastic medium, this recovery of the original form is considered to be perfect.

We consider the above three basic physic laws: equation of motion, relation between strain and displacement, and Hooke's law, demonstrated as follows [5]:

$$\rho\ddot{\boldsymbol{u}} = \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{F} \quad \textit{(Equation of motion)} \tag{2}$$

$$\boldsymbol{\epsilon} = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T) \quad \textit{(Strain-displacement relation)} \tag{3}$$

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\epsilon} \quad \textit{(Hooke's law)} \tag{4}$$

where $\boldsymbol{F}$ is the body force per unit volume, ρ is the mass density, $\boldsymbol{\sigma}$ is the stress

tensor, $\boldsymbol{\epsilon}$ is the infinitesimal strain tensor, $\boldsymbol{u}$ is the wave displacement, and $\boldsymbol{C}$ is the fourth order stiffness tensor specifically defined for elastic waves.

In the following we will apply the Einstein summation convention for tensors. Consider (4) in the tensor form $\sigma_{ij} = C_{ijlm}\epsilon_{lm}$; in isotropic media, the stiffness tensor is formulated in tensor form as

$$C_{ijlm} = \lambda\delta_{ij}\delta_{lm} + \mu(\delta_{il}\delta_{jm} + \delta_{im}\delta_{jl}) \tag{5}$$

where λ and μ are defined to be known as Lamé's first and second parameters [64]; those two parameters have been described by the French mathematician G. Lamé (1795-1870), used to analyse the rock physics property; μ is also known as shear modulus. It is generally considered that first and second Lamé parameters have no direct physical interpretation. In this thesis, we restrict ourselves to modelling elastic wave propagation in isotropic media.

We rewrite (2), (3) and (4) in Einstein summation convention to obtain the general form of an elastic wave equation as

$$\begin{aligned} \rho\partial_{tt}u_i &= \sigma_{ji,j} + F_i \quad \textit{(Equation of motion)} \\ \epsilon_{ij} &= \frac{1}{2}(u_{i,j} + u_{j,i}) \quad \textit{(Strain-displacement relation)} \\ \sigma_{ij} &= \lambda\delta_{ij}\epsilon_{kk} + 2\mu\epsilon_{ij} \quad \textit{(Hooke's law)} \end{aligned} \tag{6}$$

In the two dimensional case, we replace the general axes i, j, k by a more intuitive

notation as x and y; then (6) gives us the two dimensional elastic wave formulation

$$\begin{aligned}
\rho\frac{\partial^2 u_x}{\partial t^2} &= \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + F_x \\
\rho\frac{\partial^2 u_y}{\partial t^2} &= \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + F_y \\
\epsilon_{xx} &= \frac{\partial u_x}{\partial x} \\
\epsilon_{yy} &= \frac{\partial u_y}{\partial y} \\
\epsilon_{xy} &= \frac{1}{2}(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}) \\
\sigma_{xx} &= (\lambda + 2\mu)\epsilon_{xx} \\
\sigma_{yy} &= (\lambda + 2\mu)\epsilon_{yy} \\
\sigma_{xy} &= 2\mu\epsilon_{xy}
\end{aligned} \tag{7}$$

Now we can derive the following form of a second-order wave equation which has been used frequently in the literature, for example in [37]

$$\begin{aligned}
\rho\frac{\partial^2 u_x}{\partial t^2} &= (\lambda + 2\mu)\frac{\partial^2 u_x}{\partial x^2} + \mu(\frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_y}{\partial x \partial y}) + F_x \\
\rho\frac{\partial^2 u_y}{\partial t^2} &= (\lambda + 2\mu)\frac{\partial^2 u_y}{\partial y^2} + \mu(\frac{\partial^2 u_y}{\partial^2 x^2} + \frac{\partial^2 u_x}{\partial x \partial y}) + F_y
\end{aligned} \tag{8}$$

2.5 A symmetric hyperbolic scheme

Much of the previous research in two dimensional elastic wave equation inversion uses the above formulation of (7), see for example [26], [37]; In particular, in [37] the forward model is defined as (8); but the disadvantage of this formulation in our setup is that its numerical implementation does not directly use wave pressure as dynamic variable, which which is one of the main observables in practical applications. Therefore, we prefer to introduce an alternative form of elastic wave equation, where we will directly take wave pressure into consideration. This way it is more convenient to extract pressure data as needed in practice.

Traditionally in wave propagation, the definition of wave pressure p is often defined as $p = \lambda \mathrm{div}(\boldsymbol{u})$ or similar [57]; such derivation is not directly achieved in a second-order formulation with wave displacement (8). Alternative schemes in first-order elastics are applied in [25], [11]; such a form has been applied for

forward propagation, and has the advantage of simplicity and being numerically less time-consuming; but its computation of adjoint-state operator for back propagation is not identical to forward problem, making the application for inverse problems difficult.

On the other hand, G. Papanicolaou *et al.* [57] have introduced a first-order symmetric hyperbolic scheme for elastic wave propagation, which directly incorporaates the wave pressure as a dynamic variable; in [18] this scheme has been linked to time-reversal where the adjoint state equation of this formulation is directly linked to physical time-reversal propagation, which adds to its convenience for applying adjoint-state schemes for later gradient computation. In the following chapter, we will use this formulation for deriving an inverse problem for the symmetric hyperbolic scheme as applied to seismic FWI.

2.5.1 First-order elastic wave equation

To start with we derive the first-order elastic wave equation as an intermediate form from (7). We introduce the velocity parameter $v = \partial_t u$ for axis x and y; therefore we obtain the first-order elastic wave equation as

$$\begin{aligned}
\frac{\partial v_x}{\partial t} &= \frac{1}{\rho}(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + F_x) \\
\frac{\partial v_y}{\partial t} &= \frac{1}{\rho}(\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + F_y) \\
\frac{\partial \sigma_{xx}}{\partial t} &= (\lambda + 2\mu)\frac{\partial v_x}{\partial x} + \lambda \frac{\partial v_y}{\partial y} \\
\frac{\partial \sigma_{yy}}{\partial t} &= (\lambda + 2\mu)\frac{\partial v_y}{\partial y} + \lambda \frac{\partial v_x}{\partial x} \\
\frac{\partial \sigma_{xy}}{\partial t} &= \mu(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x})
\end{aligned} \tag{9}$$

This is the scheme applied by B. Cox and B. Treeby in a k-wave toolbox [25], [11]. We will derive the symmetric hyperbolic system from this first-order elastic scheme.

2.5.2 Symmetric hyperbolic elastic equation

A key technique in the formulation of elastic waves in form of a symmetric hyperbolic system has been applied by G. Papanicolaou [57] in 1996; the idea is to separate the stress tensor into two components, where one component is linear to the term of pressure, and the other part is considered λ-independent. To achieve this, new variables are defined as (η_{ij}, p) which replace (σ_{ij}) in (9).

We will concentrate here on a 2D setup, where this new set of variables is related to wave displacement u_x, u_y by

$$\begin{aligned} v_x &= \partial_t u_x \\ v_y &= \partial_t u_y \\ p &= \lambda(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y}) \\ \eta_{xx} &= 2\mu\frac{\partial u_x}{\partial x} \\ \eta_{yy} &= 2\mu\frac{\partial u_y}{\partial y} \\ \eta_{xy} &= \mu(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}) \end{aligned} \tag{10}$$

By applying this new set of variables to (9) we obtain the new formulation

$$\begin{aligned} \frac{\partial v_x}{\partial t} &= \frac{1}{\rho}(\frac{\partial \eta_{xx}}{\partial x} + \frac{\partial p}{\partial x} + \frac{\partial \eta_{xy}}{\partial y} + F_x) \\ \frac{\partial v_y}{\partial t} &= \frac{1}{\rho}(\frac{\partial \eta_{xy}}{\partial x} + \frac{\partial p}{\partial y} + \frac{\partial \eta_{yy}}{\partial y} + F_y) \\ \frac{\partial \eta_{xx}}{\partial t} &= 2\mu\frac{\partial v_x}{\partial x} \\ \frac{\partial \eta_{yy}}{\partial t} &= 2\mu\frac{\partial v_y}{\partial y} \\ \frac{\partial \eta_{xy}}{\partial t} &= \mu(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y}) \\ \frac{\partial p}{\partial t} &= \lambda(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}) \end{aligned} \tag{11}$$

As already mentioned before, the advantage of this new formulation is not just being a symmetric hyperbolic system. In practical application, the seismic data are mostly recorded as wave pressure. However, in standard elastic wave equa-

tions, the pressure is not directly included in the set of dynamic variables; the most common way representing pressure is to use the average of diagonal terms of stress tensor. In our new form, on the other hand, the included wave parameter p is directly related to the physically measured wave pressure [57].

Let us consider now in general a multi-variable linear system with $U : \mathcal{R}^n \times (0, \infty) \rightarrow \mathcal{R}^m$, for $i = 1, 2, ..., n$; furthermore, Γ, Φ, A_i are $m \times m$ matrices, and $F : \mathcal{R}^n \times (0, \infty) \rightarrow \mathcal{R}^m$.

$$\Gamma U_t + \sum_{i=1}^{n} A_i U_{x_i} + \Phi U = F(\boldsymbol{x}, t) \tag{12}$$

The system is defined to be a symmetric hyperbolic if Γ is symmetric, positive definite; A_i are constant, symmetric matrices; and Φ is a semi-definite matrix. (11) can be transformed into a symmetric hyperbolic general form as

$$\Gamma(m)\boldsymbol{w_t} + D_x \boldsymbol{w_x} + D_y \boldsymbol{w_y} = \boldsymbol{q} \tag{13}$$

where $m = (\lambda, \mu, \rho)$ is the static parameter set for elastic wave propagation; $\boldsymbol{q}$ is the source term $\boldsymbol{q} = (F_x/\rho, F_y/\rho, 0, 0, 0, 0)^T$; $\boldsymbol{w}$ is the dynamic variable set where $\boldsymbol{w} = (v_x, v_y, \eta_{xx}, \eta_{yy}, \eta_{xy}, p)^T \in \boldsymbol{U}$ as $\boldsymbol{U} = (\Omega \times \mathcal{T})^6$. The precise formulations of Γ, D_x, D_y are

$$\Gamma(m) = \begin{bmatrix} \rho & & & & & \\ & \rho & & & & \\ & & 1/2\mu & & & \\ & & & 1/2\mu & & \\ & & & & 1/\mu & \\ & & & & & 1/\lambda \end{bmatrix} \tag{14}$$

$$D_x = - \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$D_y = -\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

This constitutes our principal forward model for the remainder of the thesis.

2.6 From governing form to gradient derivation

2.6.1 Governing form

Now we will consider the inverse problem with the given forward model. Usually, FWI is mathematically modelled as the minimization of a L^2 norm of misfit for seismic data obtained from each source wave, recorded at all receivers. By definition, the data misfit calculates the difference between observed data and calculated data, which is based on the calculated dynamic wave variables. The standard form is provided as follows:

$$min_m E(m) = \frac{1}{2} \sum_s \sum_r \| d^s_{cal}(\boldsymbol{x}_r, t) - d^s_{obs}(\boldsymbol{x}_r, t) \|^2_{\mathcal{D}} \tag{15}$$

for all sources $s \in \mathcal{S}$, where we have the variable-to-data mapping

$$\boldsymbol{P}_{\mathcal{R}} \boldsymbol{w}^s = d^s_{cal} \tag{16}$$

and constraints

$$\mathcal{L}(m) \boldsymbol{w}^s = \boldsymbol{q}^s$$

where d^s_{cal}, d^s_{obs} separately represents the calculated, and observed measurement at receivers $r \in \mathcal{X}_r$ according to source s. $\mathcal{L}$ is the partial differential operator for our symmetric hyperbolic system (11), (13), and $E(m) : \Omega \to \mathrm{R}$ is the energy functional defined to compute the 2-norm of data misfit.

We briefly consider the initial and boundary conditions for the forward model. In seismic imaging, we can assume that the considered domain is sufficiently large, such that the seismic wave never reaches its boundary; that is, from time zero to

the maximum time T, the boundary values for the dynamic wave variables are assumed to be zero; [18] defines such survey condition as 'free-surface'. Moreover, in practical application, we define that the initial time for the dynamic wave variables is zero as well. Therefore, we add appropriate initial and boundary conditions:

$$\begin{aligned} \mathcal{L}(m)\boldsymbol{w}^s &= \boldsymbol{q} \\ \boldsymbol{w}^s(\boldsymbol{x}, 0) &= 0 \\ \boldsymbol{w}^s(\partial\Omega, t) &= 0 \end{aligned} \tag{17}$$

Combined with (15), this results in our final form of least square optimization:

$$min_m E(m) = \frac{1}{2} \sum_s \sum_r \|\boldsymbol{P}_{\mathcal{R}} \boldsymbol{w}^s(\boldsymbol{x}, t) - d^s_{obs}(\boldsymbol{x}_r, t)\|^2_{\mathcal{D}} \tag{18}$$

with constraints

$$\mathcal{L}(m)\boldsymbol{w}^s = \boldsymbol{q}^s$$

and the initial and boundary conditions

$$\begin{aligned} \boldsymbol{w}^s(\boldsymbol{x}, 0) &= 0 \\ \boldsymbol{w}^s(\partial\Omega, t) &= 0 \end{aligned} \tag{19}$$

The above is our final governing form for the inverse problem considered in the following chapters.

2.6.2 An adjoint-state method for gradient computation

We apply an adjoint-state method to calculate the gradient of our least square optimization problem (18), see for example [55] for a recent general overview; in that paper, the Lagrangian technique is applied to obtain the state and adjoint variables and to obtain the gradient from them; we apply the same technique here to obtain the gradient for our (different) model.

Consider the governing form (18). We rewrite $E(m) = E(\boldsymbol{w}^s)$. Then we are able to obtain the Lagrangian as $L(\boldsymbol{w}^s, \boldsymbol{v}^s) = E(\boldsymbol{w}^s) - \sum_{s\in\mathcal{S}} \langle \boldsymbol{v}^s, \mathcal{L}\boldsymbol{w}^s - \boldsymbol{q}^s \rangle_U$; here $\boldsymbol{w}^s$ is considered as the variable, and $\boldsymbol{v}^s$ is said to be the Lagrange multiplier.

We need to calculate the minimum of the above functional; such a minimum

should satisfy the following conditions for all $s \in S$:

$$\begin{aligned} \frac{\delta L}{\delta \boldsymbol{v}^s} = 0 \rightarrow \boldsymbol{\mathcal{L}}\boldsymbol{w}^s = \boldsymbol{q}^s \quad \textit{(state equation)} \\ \frac{\delta L}{\delta \boldsymbol{w}^s} = 0 \rightarrow \boldsymbol{\mathcal{L}}^*\boldsymbol{v}^s = \boldsymbol{P}_{\mathcal{R}}^*(\boldsymbol{P}_{\mathcal{R}}^*\boldsymbol{w}^s - d_{obs}^s) \quad \textit{(adjoint-state equation)} \end{aligned} \tag{20}$$

We call the second of these equations as the adjoint-state equation, and $\boldsymbol{v}^s$ the adjoint variable satisfying $\boldsymbol{v}^s \in \boldsymbol{U}$ for all s. Neglecting the source notation s, we have the following

Theorem Given that the forward problem (13) satisfies the 'free-surface' condition (19). Then the adjoint-state equation (20) is provided by [18]:

$$\begin{aligned} \Gamma(m)\boldsymbol{v}_t + D_x\boldsymbol{v}_x + D_y\boldsymbol{v}_y = -\boldsymbol{P}_{\mathcal{R}}^*(\boldsymbol{P}_{\mathcal{R}}\boldsymbol{w} - d_{obs}) \\ \boldsymbol{v}(\boldsymbol{x}, T) = 0 \\ \boldsymbol{v}(\partial\Omega, t) = 0 \end{aligned} \tag{21}$$

Proof The definition for $\boldsymbol{\mathcal{L}}^*$ is provided by $\langle \boldsymbol{\mathcal{L}}^*\boldsymbol{v}, \boldsymbol{w}\rangle_{\boldsymbol{U}} = \langle \boldsymbol{v}, \boldsymbol{\mathcal{L}}\boldsymbol{w}\rangle_{\boldsymbol{U}}$, as $\boldsymbol{\mathcal{L}}$ is considered to be the symmetric hyperbolic operator. This provides the following

$$\begin{aligned} \langle \boldsymbol{v}, \boldsymbol{\mathcal{L}}\boldsymbol{w}\rangle_{\boldsymbol{U}} &= \iint_\Omega \int_0^T \boldsymbol{v}^T[(\Gamma\partial_t + D_x\partial_x + D_y\partial_y)\boldsymbol{w}] \\ &= \iint_\Omega \int_0^T \boldsymbol{v}^T\Gamma\partial_t\boldsymbol{w} + \iint_\Omega \int_0^T \boldsymbol{v}^T(D_x\partial_x + D_y\partial_y)\boldsymbol{w} \end{aligned} \tag{22}$$

The first term of (22) yields

$$\begin{aligned} &\iint_\Omega \int_0^T \boldsymbol{v}^T\Gamma\partial_t\boldsymbol{w} \\ &= \iint_\Omega [\boldsymbol{v}^T\Gamma\boldsymbol{w}]\Big|_0^T - \iint_\Omega \int_0^T \boldsymbol{w}^T\Gamma\partial_t\boldsymbol{v} \end{aligned} \tag{23}$$

The second term gives rise to

$$\begin{aligned} &\iint_\Omega \int_0^T \boldsymbol{v}^T(D_x\partial_x + D_y\partial_y)\boldsymbol{w} \\ &= \iint_\Omega \int_0^T (\boldsymbol{v}^T D_x, \boldsymbol{v}^T D_y)\cdot\nabla\boldsymbol{w} \end{aligned} \tag{24}$$

Denote that $V = (\boldsymbol{v}^T D_x, \boldsymbol{v}^T D_y)$. Consider the boundary condition $\boldsymbol{w}(\partial\Omega, t) = 0$. Then from the divergence theorem, we obtain that

$$\begin{aligned}
&\iint_\Omega \int_0^T (\boldsymbol{v}^T D_x, \boldsymbol{v}^T D_y) \cdot \nabla \boldsymbol{w} \\
&= \iint_\Omega \int_0^T (V \cdot \nabla \boldsymbol{w}) \\
&= \int_{\partial\Omega} \int_0^T \boldsymbol{w}^T [V \cdot \boldsymbol{n}] \\
&- \iint_\Omega \int_0^T \boldsymbol{w}^T \nabla \cdot V \\
&= - \iint_\Omega \int_0^T \boldsymbol{w}^T (D_x \boldsymbol{v}_x + D_x \boldsymbol{v}_y)
\end{aligned} \tag{25}$$

Consider the stated boundary condition. Notice that the condition ensures that at the initial time $\boldsymbol{w}(x, y, 0) = 0$. Now if we provide a time-reversal initial condition for the adjoint-state such that $\boldsymbol{v}(x, y, T) = 0$, then the first term of (23) will also be zero. Combine (22), (23), (24), (25); we obtain the following form

$$\begin{aligned}
\langle \boldsymbol{v}, \mathcal{L}\boldsymbol{w} \rangle_{\boldsymbol{U}} &= - \iint_\Omega \int_0^T \boldsymbol{w}^T \Gamma \partial_t \boldsymbol{v} - \iint_\Omega \int_0^T \boldsymbol{w}^T (D_x^T \partial_x + D_y^T \partial_y) \boldsymbol{v} \\
&= \iint_\Omega \int_0^T \boldsymbol{w}^T (-\Gamma \partial_t - D_x \partial_x - D_y \partial_y) \boldsymbol{v} = \langle -\mathcal{L}\boldsymbol{w}, \boldsymbol{v} \rangle_{\boldsymbol{U}}
\end{aligned} \tag{26}$$

Given that $\langle \boldsymbol{v}, \mathcal{L}\boldsymbol{w} \rangle_{\boldsymbol{U}} = \langle \mathcal{L}^* \boldsymbol{v}, \boldsymbol{w} \rangle_{\boldsymbol{U}}$, the adjoint operator is derived such that $\mathcal{L}^* = -\mathcal{L}$. Combining this with the adjoint-state equation

$$\mathcal{L}^* \boldsymbol{v} = \boldsymbol{P}_{\mathcal{R}}^* (\boldsymbol{P}_{\mathcal{R}} \boldsymbol{w} - d_{obs})$$

we obtain the adjoint-state equation to be

$$- (\Gamma(m) \partial_t + D_x \partial_x + D_y \partial y) \boldsymbol{v} = \boldsymbol{q}_v \tag{27}$$

subject to

$$\begin{aligned}
&\boldsymbol{q}_v = -\boldsymbol{P}_{\mathcal{R}}^* (\boldsymbol{P}_{\mathcal{R}} \boldsymbol{w} - d_{obs}) \\
&\boldsymbol{v}(x, y, T) = 0
\end{aligned} \tag{28}$$

Therefore the adjoint operator satisfies that $\mathcal{L}^* = -\mathcal{L}$, indicating a skew-symmetric relation. This shows that the implementations of the forward and adjoint operator are basically identical, and the adjoint state equation can be considered as the time-reversal problem with its source term being given by the data residuals recorded at all receivers; this procedure is commonly also known as back propagation [74].

Next we will dervive an expression for the gradient from the state and adjoint state equations above. Related work in the literature includes Fitchner's derivation for gradient descent which satisfies a formation of symmetry [26]. Moreover, [37] applies the same technique in his model for 2D elastic wave inversion. However, we emphasize that Fitchner's derivation is based on a second order elastic wave equation which is fundamentally different from the formulation of our first-order scheme. Therefore we need to recalculate the gradient that is suited for our model.

The gradient related to our minimization problem (18) is defined by the functional derivative $\frac{\delta E}{\delta m}$, given as [22]:

$$\lim_{\epsilon \to 0} \frac{E(m + \epsilon \delta m) - E(m)}{\epsilon} = \iint_{\Omega} \frac{\delta E(m)}{\delta m(\boldsymbol{x})} \epsilon \delta(m(\boldsymbol{x})) \mathrm{d}\boldsymbol{x} \tag{29}$$

for any δm.

Notice that our previous definition is a general form valid for all possibilities of data measurements. For our more detailed model, we choose our observed data d_{obs} as the values of wave variables at receiver points, as in (11) provided from an initial model. This indicates that $\boldsymbol{P}_{\mathcal{R}}$ in (18) is a projection map from wave variables to their values at the receiver set $\mathcal{R}$. Also, we rewrite the dynamic variable $\boldsymbol{w} = (v_x, v_y, \eta_{xx}, \eta_{yy}, \eta_{xy}, p)$ as $\boldsymbol{w} = (w_1, ..., w_6)$, and we define that $d_{obs} = (d^1_{obs}, ..., d^6_{obs})$ accordingly. We therefore obtain the following derivation

$$E(m) = \frac{1}{2} \sum_{s \in S} \sum_{r \in \mathcal{R}} \sum_{i=1}^{6} \iint_{\Omega} \int_0^T (w_i^s(\boldsymbol{x}_r, t) - d_{obs}^{i,s}(\boldsymbol{x}_r, t))^2 \mathrm{d}\boldsymbol{x} \mathrm{d}t$$

Define the function $\mathcal{E}(m)$, such that $E(m) = \iint_{\Omega} \mathcal{E}(m) \mathrm{d}\boldsymbol{x}$. This gives rise to

$$\mathcal{E}(m) = \frac{1}{2} \sum_{s \in S} \sum_{r \in \mathcal{R}} \sum_{i=1}^{6} \int_0^T (w_i^s(\boldsymbol{x}_r, t) - d_{obs}^{i,s}(\boldsymbol{x}_r, t))^2 \mathrm{d}t$$

From previous definitions it follows that $\frac{\delta E}{\delta m} = \frac{\partial \mathcal{E}}{\partial m}$. This provides that

$$\frac{\partial \mathcal{E}}{\partial m} = \sum_{s \in S} \sum_{r \in \mathcal{R}} \sum_{i=1}^{6} \int_0^T (w_i^s - d_{obs}^{i,s})(\boldsymbol{x}_r, t) \frac{\partial w_i^s}{\partial m}(\boldsymbol{x}_r, t) \mathrm{d}t$$

Rewrite this in a form with $\boldsymbol{P}_{\mathcal{R}}$ in (16). We therefore obtain

$$\begin{aligned} \frac{\delta E}{\delta m} = \frac{\partial \mathcal{E}}{\partial m} &= \sum_{s \in \mathcal{S}} \langle \boldsymbol{P}_{\mathcal{R}}^* (\boldsymbol{P}_{\mathcal{R}} \boldsymbol{w}^s - d_{obs}^s), \frac{\partial \boldsymbol{w}^s}{\partial m} \rangle_{\boldsymbol{\tau}} \\ &= \sum_{s \in \mathcal{S}} \langle \mathcal{L}^* \boldsymbol{v}^s, \frac{\partial \mathcal{L}^{-1} \boldsymbol{q}_s}{\partial m} \rangle_{\boldsymbol{\tau}} \end{aligned}$$

given the inner product

$$\langle \boldsymbol{r_1}, \boldsymbol{r_2} \rangle_\tau = \sum_i \int_0^T \boldsymbol{r_1}^i \boldsymbol{r_2}^i \mathrm{d}t \tag{30}$$

for $\boldsymbol{r_1}, \boldsymbol{r_2} \in \boldsymbol{U}$. Consider the operator property

$$\begin{aligned} \mathcal{L}^{-1}\mathcal{L} &= \mathcal{I} \\ \frac{\partial}{\partial m}\mathcal{L}^{-1}\mathcal{L} &= 0 \\ \mathcal{L}^{-1}\frac{\partial \mathcal{L}}{\partial m} + \mathcal{L}\frac{\partial \mathcal{L}^{-1}}{\partial m} &= 0 \\ \frac{\partial \mathcal{L}^{-1}}{\partial m} &= -\mathcal{L}^{-1}\frac{\partial \mathcal{L}}{\partial m}\mathcal{L}^{-1} \end{aligned}$$

For every s, which yields

$$\begin{aligned} \langle \mathcal{L}^* \boldsymbol{v}^s, \frac{\partial \mathcal{L}^{-1} \boldsymbol{q}^s}{\partial m} \rangle_{\boldsymbol{\tau}} &= \langle \mathcal{L}^* \boldsymbol{v}^s, -\mathcal{L}^{-1} \frac{\partial \mathcal{L}}{\partial m} \mathcal{L}^{-1} \boldsymbol{q}^s \rangle_{\boldsymbol{\tau}} \\ &= -\langle \boldsymbol{v}^s, \frac{\partial \mathcal{L}}{\partial m} \boldsymbol{w}^s \rangle_{\boldsymbol{\tau}} \end{aligned}$$

Sum up the above for all source terms to obtain the gradient formulation

$$\frac{\delta E}{\delta m} = -\sum_{s \in \mathcal{S}} \langle \boldsymbol{v}^s, \frac{\partial \mathcal{L}}{\partial m} \boldsymbol{w}^s \rangle_{\boldsymbol{\tau}} \tag{31}$$

This derivation broadly follows the outline provided in R. Plessix [55]; However,

the difference is that in that paper, Plessix uses a second-order time-domain acoustic equation as example; In our case, we apply the general technique to our tailor made model which uses a first-order symmetric hyperbolic system form.

In particular, for the elastic wave equation, we need to calculate the partial derivative with respect to model parameters $m = (\lambda, \mu, \rho)$; the gradient $\frac{\delta E}{\delta m}$ can then be written as $\frac{\delta E}{\delta m} = (\frac{\delta E}{\delta \lambda}, \frac{\delta E}{\delta \mu}, \frac{\delta E}{\delta \rho})$. Therefore we have the following derivation from (31) for the chosen model parameters:

$$\begin{aligned}
\frac{\delta E}{\delta \lambda} &= -\sum_{s \in \mathcal{S}} \langle \boldsymbol{v}^s, \frac{\partial \mathcal{L}}{\partial \lambda} \boldsymbol{w}^s \rangle_{\boldsymbol{\tau}} \\
\frac{\delta E}{\delta \mu} &= -\sum_{s \in \mathcal{S}} \langle \boldsymbol{v}^s, \frac{\partial \mathcal{L}}{\partial \mu} \boldsymbol{w}^s \rangle_{\boldsymbol{\tau}} \\
\frac{\delta E}{\delta \rho} &= -\sum_{s \in \mathcal{S}} \langle \boldsymbol{v}^s, \frac{\partial \mathcal{L}}{\partial \rho} \boldsymbol{w}^s \rangle_{\boldsymbol{\tau}}
\end{aligned} \tag{32}$$

Recall now the symmetric hyperbolic elastic scheme (11). We deduce that $\mathcal{L}(m) = \Gamma(m)\partial_t + D_x\partial_x + D_y\partial_y$; the above yields

$$\frac{\partial \mathcal{L}}{\partial m} = \frac{\partial \Gamma}{\partial m} \partial_t$$

We obtain the partial derivative for each parameter of m where $m = (\lambda, \mu, \rho)$:

$$\frac{\partial \Gamma}{\partial \lambda} = \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & 0 & & & \\ & & & 0 & & \\ & & & & 0 & \\ & & & & & -1/\lambda^2 \end{bmatrix}$$

$$\frac{\partial \Gamma}{\partial \mu} = \begin{bmatrix} 0 & & & & & & \\ & 0 & & & & & \\ & & -1/2\mu^2 & & & & \\ & & & -1/2\mu^2 & & & \\ & & & & -1/\mu^2 & & \\ & & & & & 0 \end{bmatrix}$$

$$\frac{\partial \Gamma}{\partial \rho} = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 0 & & & \\ & & & 0 & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix}$$

Recall the definition of the state variable $\boldsymbol{w} = (v_x, v_y, \eta_{xx}, \eta_{yy}, \eta_{xy}, p)$; now define the adjoint-variable $\boldsymbol{v} = (v_x^*, v_y^*, \eta_{xx}^*, \eta_{yy}^*, \eta_{xy}^*, p^*)$. With this notation we obtain the gradient from (32) in the form

$$\begin{aligned}
\frac{\delta E}{\delta \lambda} &= \sum_{s \in \boldsymbol{S}} \langle \boldsymbol{v}^s, \frac{\partial \mathcal{L}}{\partial \lambda} \boldsymbol{w}^s \rangle_\tau = \sum_{s \in \boldsymbol{S}} \int_0^T (\boldsymbol{v}^s)^T \frac{\partial \mathcal{L}}{\partial \lambda} \boldsymbol{w}^s \mathrm{d}\boldsymbol{x}\mathrm{d}t \\
&= -\frac{1}{\lambda^2} \sum_{s \in \mathcal{S}} \int_0^T (p^s)^* \frac{\partial p^s}{\partial t} \mathrm{d}\boldsymbol{x}\mathrm{d}t \\
\frac{\delta E}{\delta \mu} &= \sum_{s \in \boldsymbol{S}} \langle \boldsymbol{v}^s, \frac{\partial \mathcal{L}}{\partial \mu} \boldsymbol{w}^s \rangle_\tau = \sum_{s \in \boldsymbol{S}} \int_0^T (\boldsymbol{v}^s)^T \frac{\partial \mathcal{L}}{\partial \mu} \boldsymbol{w}^s \mathrm{d}\boldsymbol{x}\mathrm{d}t \\
&= -\frac{1}{2\mu^2} \sum_{s \in \mathcal{S}} \int_0^T ((\eta_{xx}^s)^* \frac{\partial \eta_{xx}^s}{\partial t} + (\eta_{yy}^s)^* \frac{\partial \eta_{xx}^s}{\partial t} + 2(\eta_{xy}^s)^* \frac{\partial \eta_{xy}^s}{\partial t}) \mathrm{d}\boldsymbol{x}\mathrm{d}t \\
\frac{\delta E}{\delta \rho} &= \sum_{s \in \boldsymbol{S}} \langle \boldsymbol{v}^s, \frac{\partial \mathcal{L}}{\partial \rho} \boldsymbol{w}^s \rangle_\tau = \sum_{s \in \boldsymbol{S}} \int_0^T (\boldsymbol{v}^s)^T \frac{\partial \mathcal{L}}{\partial \rho} \boldsymbol{w}^s \mathrm{d}\boldsymbol{x}\mathrm{d}t \\
&= \sum_{s \in \mathcal{S}} \int_0^T ((v_x^s)^* \frac{\partial v_x^s}{\partial t} + (v_y^s)^* \frac{\partial v_y^s}{\partial t}) \mathrm{d}\boldsymbol{x}\mathrm{d}t
\end{aligned} \tag{33}$$

We define now $\boldsymbol{g}_m = (\frac{\delta E}{\delta \lambda}, \frac{\delta E}{\delta \mu}, \frac{\delta E}{\delta \rho})$ to represent the gradient, which will be applied in the following chapters.

2.6.3 A wave velocity gradient parameterization

The elastic wave has two main parts: the compressional wave , or P-wave, with wave displacements in the same direction as the wave propagates. This part is broadly related to pressure. The P-wave velocity is denoted as V_p. The other type of wave at the shear waves, or S-waves, with displacements perpendicular to the wave direction; this wave force comes from the shear strength of elastic medium. The S-wave velocity is denoted as V_s.

In Appendix 1 (9.1) we show that V_p and V_s can be derived from lamé pa-

rameters:

$$
\begin{aligned}
V_p &= \sqrt{\frac{\lambda + 2\mu}{\rho}} \\
V_s &= \sqrt{\frac{\mu}{\rho}}
\end{aligned}
\tag{34}
$$

This derivation is crucial in numerical modelling. In numerical simulation, the Lamé parameters λ and μ do not have a direct physical interpretation. Therefore, to monitor the medium properties most of tests choose to use wave velocity V_p, V_s instead of the Lamé parameters as measurements; therefore, a relationship between those quantities would be very convenient for geophysicists to estimate the composition from approximated seismic images.

We consider therefore now an alternative parameter set (V_p, V_s, ρ'). The parameter transformation (34) provides

$$
\begin{aligned}
\lambda &= \rho(V_p^2 - 2V_s^2) \\
\mu &= \rho V_s^2 \\
\rho &= \rho'
\end{aligned}
$$

We apply a chain-rule to obtain the corresponding gradient for (V_p, V_s, ρ') as

$$
\begin{aligned}
\frac{\delta E}{\delta V_p} &= 2V_p\rho\frac{\delta E}{\delta \lambda} \\
\frac{\delta E}{\delta V_s} &= -4V_s\rho\frac{\delta E}{\delta \lambda} + 2V_s\rho\frac{\delta E}{\delta \mu} \\
\frac{\delta E}{\delta \rho'} &= (V_p^2 - 2V_s^2)\frac{\delta E}{\delta \lambda} + V_s^2\frac{\delta E}{\delta \mu} + \frac{\delta E}{\delta \rho}
\end{aligned}
\tag{35}
$$

In our model update, we apply (33) as the gradient to compute λ, μ, ρ, then compute and demonstrate V_p, V_s, ρ' correspondingly. Numerical comparison of such different model parametrizations are for example provided in [37]; although D. Köhn's numerical example claims that the model set with wave velocity looks better than with Lamé parameters, the approximation is still far away from the true model; moreover, the data misfit does not show a very satisfactory convergence. Due to time limitations, in our model we only consider the model set with Lamé parameters, and apply V_p and V_s as parameter measurements, as to better

simulate practical seismic surveys. We will leave the application with (35) as a possibility for future work.

2.7 Regularization with Sobolev gradient

In full waveform inversion, the use of standard gradients for the calculation of descent directions (neglecting any artificial regularization terms) often gives rise to very rough updates. In shape based models, this corresponds to rough boundaries, especially in high-contrast models; Therefore, we should consider applying regularization methods to refine our numerical computation. Different regularization methods have been applied in several other areas of research. In our thesis, we consider the idea of Sobolev gradient [49] that is used for calculating smoothed versions of gradients based on function space projections. Those gradients are also often called Sobolev gradients. Such techniques have been applied elsewhere in the literature with good success, see for example a level set approach to electromagnetic tomography in [20].

2.7.1 Sobolev gradient

The definition of Sobolev gradient uses the concepts of a Sobolev space containing functions satisfying a Sobolev norm condition [2]; in our application, its primary use is to apply a least-squares functional as a solution to a nonlinear differential equation, but define the norm to be a Sobolev norm instead of an L^2 norm (18). The corresponding gradient based on such a variational form is called Sobolev gradient [49]. The Sobolev gradient has been used historically a lot in the numerical solution for partial differential equations. However, it has been widely applied also for the solution of inverse problems in various fields, like optical tomography [53], history matching [58], magnetic induction tomography [20], image processing [60], e.t.c.

In the following we briefly outline the use of the Sobolev gradient in our application.

We recall the Sobolev space $H_1(\Omega) := \{m \in \Omega, \partial m \in \Omega\}$ (notice that here, m is defined differently from previous sections without risk of confusion); then the Sobolev norm is provided that

$$\|m\|_{H_1} = (\|m\|_\Omega^2 + \|\nabla m\|_\Omega^2)^{1/2} \tag{36}$$

and the inner product

$$\langle m_1, m_2\rangle_{H_1} = \langle m_1, m_2\rangle_\Omega + \langle \nabla m_1, \nabla m_2\rangle_\Omega$$

In particular, we can introduce a weight parameter $\gamma > 0$ for obtaining a moderated Sobolev-norm

$$\|m\|_{H_1,\gamma} = (\|m\|_\Omega^2 + \gamma\|\nabla m\|_\Omega^2)^{1/2} \tag{37}$$

and the associated inner product

$$\langle m_1, m_2\rangle_{H_1,\gamma} = \langle m_1, m_2\rangle_\Omega + \gamma\langle \nabla m_1, \nabla m_2\rangle_\Omega$$

We apply the above variational form in (18), (64) and compute the Sobolev gradient from the well-understood L^2 gradient. We apply the Sobolev norm condition for the state variable $\boldsymbol{w}$ and adjoint-state variable $\boldsymbol{v}$, such that

$$\|\boldsymbol{w}\|_{H_1(\boldsymbol{U}),\gamma} = (\|\boldsymbol{w}\|_U^2 + \gamma\|\nabla \boldsymbol{w}\|_{\boldsymbol{U}}^2)^{1/2} \tag{38}$$

and the corresponding inner product

$$\langle \boldsymbol{w}_1, \boldsymbol{w}_2\rangle_{H_1(\boldsymbol{U}),\gamma} = \langle \boldsymbol{w}_1, \boldsymbol{w}_2\rangle_{\boldsymbol{U}} + \gamma\langle \nabla \boldsymbol{w}_1, \nabla \boldsymbol{w}_2\rangle_{\boldsymbol{U}}$$

Firstly, we compute the corresponding adjoint operator $L^*_{H_1,\gamma}$. Recall (18); this time, we define the Lagrangian form corresponding to Sobolev norm as

$$L(\boldsymbol{w}, \boldsymbol{v})_{H_1(\boldsymbol{U}),\gamma} = E(\boldsymbol{w}) - \sum_{s\in\mathcal{S}} \langle \boldsymbol{v}^s, \mathcal{L}(m)\boldsymbol{w}^s - \boldsymbol{q}^s\rangle_{H_1(\boldsymbol{U}),\gamma} \tag{39}$$

Define $\boldsymbol{r}^s = \mathcal{L}(m)\boldsymbol{w}^s - \boldsymbol{q}^s$, with the boundary condition $\boldsymbol{r}^s(\partial\boldsymbol{U}) = 0$; then the Sobolev norm satisfies

$$\begin{aligned}
\langle \boldsymbol{v}^s, \boldsymbol{r}^s\rangle_{H_1(\boldsymbol{U}),\gamma} &= \langle \boldsymbol{v}^s, \boldsymbol{r}^s\rangle_{\boldsymbol{U}} + \gamma\langle \nabla \boldsymbol{v}^s, \nabla \boldsymbol{r}^s\rangle_{\boldsymbol{U}} \\
&= \langle \boldsymbol{v}^s, \boldsymbol{r}^s\rangle_{\boldsymbol{U}} + \gamma(\int_{\partial\boldsymbol{U}} \nabla \boldsymbol{v}^s \cdot \boldsymbol{r}^s - \int_{\boldsymbol{U}} \Delta \boldsymbol{v}^s \cdot \boldsymbol{r}^s) \\
&= \langle (I - \gamma\Delta)\boldsymbol{v}^s, \boldsymbol{r}^s\rangle_{\boldsymbol{U}}
\end{aligned} \tag{40}$$

Substitute this into (39); we obtain the Sobolev-based adjoint-state equation that

$$\mathcal{L}^*_{H_1,\gamma}\boldsymbol{v} = -(I-\gamma\Delta)^{-1}\boldsymbol{P}^*_{\mathcal{R}}(\boldsymbol{P}^*_{\mathcal{R}}\boldsymbol{w}^s - d^s_{obs}) \tag{41}$$

It could also be deduced that $\mathcal{L}^*_{H_1,\gamma} = (I-\gamma\Delta)^{-1}\mathcal{L}^* = -(I-\gamma\Delta)^{-1}\mathcal{L}$, which provides a link to the L^2 adjoint operator. We apply the same gradient derivation (31) to the Sobolev gradient, defined as $\boldsymbol{g}^{H_1,\gamma}$; therefore the Sobolev gradient can be obtained from the standard gradient by a simple post-processing step [19]:

$$\boldsymbol{g}^{H_1,\gamma} = (I-\gamma\Delta)^{-1}\boldsymbol{g}_m \tag{42}$$

as $\boldsymbol{g}_m$ is the calculated gradient provided by (32). Practically, it can be shown that this postprocessing step can be implemented in form of a modified heat-kernel solution, which will be outlined in Appendix 2 (9.2). In the following chapter we will apply both, the Sobolev gradient and the L^2 gradient, for a numerical model update, and compare their numerical results in order to demonstrate the smoothing affect of the Sobolev gradient.

2.8 Techniques on line search

In optimization approaches for the solution of nonlinear inverse problems, the next step after gradient computation usually is the design of a line search for model update, to obtain a good descent direction and step size. For parameter updates, a line search method can be written in the form [47]:

$$m_{n+1} = m_n + \alpha_n \boldsymbol{p}_n$$

where m is the parameter, n is the iteration step, α is the step length or step size, and $\boldsymbol{p}$ is the descent direction. The line search method aims to compute the optimum of α_n and $\boldsymbol{p}_n$ as to obtain the best reduction in cost for a given search direction; This procedure is started with an initial guess of parameter m_0.

The most widely applied line search methods are based on first-order gradient schemes such as the steepest descent method, the conjugate gradient method, and high-order methods such as Newton method and Quasi-Newton method. Next we will introduce all these methods and analyse their pros and cons as to our model of elastic wave inversion.

Even though those techniques are classical and well understood, we will implement some of them here for reference and for verifying whether the above derived gradient directions actually do perform well in such a classical setup. This will ensure that they also will work when integrated in our novel level set based schemes.

2.8.1 Steepest descent methods

The steepest descent method can be derived by considering a perturbation on the energy functional $E(m_0 + \Delta m)$. The first order Taylor expansion provides that

$$E(m_0 + \Delta m) = E(m_0) + \frac{\delta E}{\delta m} \cdot \Delta m + o(\Delta m)^2 \tag{43}$$

In FWI, for any iteration n, we consider a line search that $m_{n+1} = m_n + \alpha_n \boldsymbol{p}_n$, and that $\boldsymbol{g}_n = \frac{\delta E}{\delta m}(m_n)$. Then the model update is given by

$$E(m_{n+1}) \approx E(m_n) + \alpha_n \boldsymbol{p}_n^T \boldsymbol{g}_n$$

where $\boldsymbol{p}_n$ is considered to be the normalised descent direction, and α is considered to be the step length; the descent direction must satisfy the descent property $\boldsymbol{p}_n^T \boldsymbol{g}_n < 0$.

The most straightforward technique is the steepest descent method with the descent direction $\boldsymbol{p}_n = -\boldsymbol{g}_n$. Here $\boldsymbol{p}_n^T \boldsymbol{g}_n = -\boldsymbol{g}_n^T \boldsymbol{g}_n$ which guarantees a descent direction. Combining this with a suitable choice of step sizes provides us with the steepest descent algorithm. Recall that all the following algorithms for different line search techniques are applied with the choice of a Sobolev gradient instead

of the standard gradient.

Algorithm 1: Steepest descent algorithm

Initial m_0, $\boldsymbol{g}_0$, α, N;
for $n = 0, 1, 2, ..., N$ **do**
  Forward propagation;
  Adjoint-state backward propagation;
  Compute gradient $\boldsymbol{g}_n$;
  if *Apply Sobolev gradient* **then**
    Obtain Sobolev gradient $\boldsymbol{g}_n^{H_1,\gamma}$;
    Apply that $\boldsymbol{g}_n = \boldsymbol{g}_n^{H_1,\gamma}$;
  end
  Define the descent direction $\boldsymbol{p}_n = -\boldsymbol{g}_n$;
  Compute step length α_n;
  Update $m_n = m_{n-1} + \alpha_n \boldsymbol{p}_n$;
  Check the stopping criterion;
end

In this algorithm, we consider the Sobolev gradient as the regularization method; the same procedure also applies for the following line search techniques.

The line step calculation could be derived from back-tracking scheme with an Armijo, Wolfe or stong Wolfe condition. For shape based methods, a derivation for steepest descent method in level set update could be referred to in [19]; we will apply this derivation in the level set iteration in Chapter 5, but for the moment concentrate on pixel based schemes.

We mention that single-step Kaczmarz type variants of the gradient method have been developed as well in the literature using steepest descent directions for subsets of the full data set as descent directions. This is also closely related to the co-called Stochastic Gradient Method. Those techniques will be considered as well in this thesis and will be discussed in more details in later chapters for shape based inversion problems.

2.8.2 Conjugate gradient methods

The steepest descent method is widely considered the most straightforward method in line search techniques. However, it also has a major disadvantage:

the convergence speed is usually relatively low. An alternative for that is the conjugate gradient method; a conjugate gradient method for linear optimization reaches exact minimum in a finite number of iterations (neglecting imperfect precision of operations). This is not the case for nonlinear schemes, and often a reinitialization is required. However, even in the non-linear case, conjugate gradient methods normally converge much faster than the steepest descent method (A good example is named by Rosenbrock's valley).

The conjugate gradient method can be implemented as follows:

Algorithm 2: Conjugate gradient algorithm

Initialise m_0, $\boldsymbol{p}_0 = -\boldsymbol{g}_0$, α, β, N;
for $n = 0, 1, 2, ..., N$ **do**
  Forward propagation;
  Adjoint-state backward propagation;
  Compute gradient $\boldsymbol{g}_n$;
  if *Apply Sobolev gradient* **then**
    Obtain Sobolev gradient $\boldsymbol{g}_n^{H_1,\gamma}$;
    Apply that $\boldsymbol{g}_n = \boldsymbol{g}_n^{H_1,\gamma}$;
  end
  Compute descent $\boldsymbol{p}_n = -\boldsymbol{g}_n + \beta_n \boldsymbol{p}_{n-1}$;
  Compute step length α_n;
  Update $m_n = m_{n-1} + \alpha_n \boldsymbol{p}_n$;
  Check the stopping criterion;
end

where β_n is a scalar guarantees that $\boldsymbol{p}_n$ are approximately conjugate to each other. In practical applications, different formulas for β can be applied:

$$\beta_n^{FR} = \frac{\Delta m_n^T \Delta m_n}{\Delta m_{n-1}^T \Delta m_{n-1}} \quad \text{(Fletcher-Reeves)};$$

$$\beta_n^{PL} = \frac{\Delta m_n^T (\Delta m_n - \Delta m_{n-1})}{\Delta m_{n-1}^T \Delta m_{n-1}} \quad \text{(Polak-Ribière)};$$

$$\beta_n^{HS} = -\frac{\Delta m_n^T (\Delta m_n - \Delta m_{n-1})}{p_{n-1}^T (\Delta m_n - \Delta m_{n-1})} \quad \text{(Hestenes-Stiefel)};$$

$$\beta_n^{DL} = -\frac{\Delta m_n^T \Delta m_n}{p_{n-1}^T (\Delta m_n - \Delta m_{n-1})} \quad \text{(Dai-Yuan)}$$

However, despite the increased efficiency, conjugate gradient methods also have some of the same weaknesses as other gradient based schemes; the descent directions show high sensitivity to the initial condition and the iterations can easily fall into a local minimum. In full waveform inversion, such result is not sufficient. Also, in non-linear inverse problem, when reaching a certain step of iteration, the conjugate gradient will reach a point where it loses conjugacy.

Therefore, recent research on the line search technique improvement has been divided into two research fields; one method is by setting up a preconditioning matrix to each descent direction to accelerate the convergence speed, known as preconditioned conjugate gradient method, or PCG; another research field concentrates on the second order convergence based Newton method.

We will apply conjugate a gradient method in Chapter 4, as a comparison to steepest descent method in pixel-based full waveform inversion; but in level set evolution, a conjugate gradient scheme would require more sophisticated considerations (with uncertain gains in efficiency) due to its shape-based nature such that we will not apply a conjugate gradient scheme there. More details are provided in Chapter 5.

2.8.3 Newton-type and Quasi-Newton methods

The first order line search technique is insufficient in many cases of numerical implementation; so some research considers the application of higher-order methods for line search techniques, such as the Newton method, which is briefly outlined here for completeness.

Consider the nonlinear optimization equation $E(m)$. A second order Taylor expansion is considered

$$E(m + \Delta m) \approx E(m) + \nabla E(m)\Delta m + \Delta m^T H \Delta m$$

where $\nabla E(m)$ is the gradient functional, and H is considered to be the Hessian matrix. Minimising $E(m + \Delta m) - E(m)$, we obtain the relation that

$$\Delta m \approx -H^{-1}\nabla E(m)$$

where H is a symmetric matrix. Specifically, the case where H equal to identity implies a steepest descent line search.

The direct calculation for Hessian matrix is extremely difficult, especially for nonlinear large scale optimization problems, where computation of second-order derivative is difficult and extremely time-consuming; Newton based algorithms therefore concentrate on the approximation of the Hessian matrix. We define that B is the approximation of the Hessian matrix, where it updates for every iteration. Therefore the algorithm is constructed as follows:

Algorithm 3: Newton method algorithm

Initialise m_0, $\boldsymbol{g}_0$, B_0, α, N;
for $n = 0, 1, ..., N$ **do**
  Forward propagation;
  Adjoint-state backward propagation;
  Compute gradient $\boldsymbol{g}_n$;
  if *Apply Sobolev gradient* **then**
    Obtain Sobolev gradient $\boldsymbol{g}_n^{H_1,\gamma}$;
    Apply that $\boldsymbol{g}_n = \boldsymbol{g}_n^{H_1,\gamma}$;
  end
  Update B_n (or B_n^{-1});
  Compute descent direction $\boldsymbol{p}_n = -B_n^{-1}\boldsymbol{g}_n$;
  Compute α_n;
  Update parameter $m_{n+1} = m_n + \alpha_n \boldsymbol{p}_n$;
  Check the stopping criterion;
end

The Newton method is mostly avoided in practical large-scale inverse problems, due to the cost of repeated calculation of the inverse of the Hessian matrix. An alternative method is known as Gauss-Newton algorithm; however, for nonlinear systems, the calculation of Gauss-Newton steps still involves the computation of large scale matrices, often involving adjoint techniques; for large scale full waveform inversion, the computational cost is still considerable but it has been applied in the literature for pixel or voxel based FWI.

One popular alternative method is known as a Quasi-Newton method, aiming to approximate the Hessian matrix and updating this approximation in each iteration based on first order derivatives only. For Quasi-Newton methods, we define a gradient difference $y_k = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$, and $\Delta m_k = m_{k+1} - m_k$. Also,

it requires an initial set up for B_0 symmetric, and positive definite; normally, choosing B_0 as identity matrix would be adequately sufficient. The most popular iteration formula for the inverse of B_k are as follows:

$$
\begin{aligned}
B_{k+1}^{-1} &= B_k^{-1} + \frac{\Delta m_k \Delta m_k^T}{\Delta m_k^T y_k} - \frac{B_k^{-1} y_k y_k^T B_k^{-1}}{y_k^T B_k^{-1} y_k} \\
&\quad \text{(Davidon - Fletcher - Powell)} \\
B_{k+1}^{-1} &= (I - \frac{\Delta m_k y_k^T}{y_k^T \Delta m_k} B_k^{-1} (I - \frac{y_k \Delta m_k^T}{y_k^T \Delta m_k}) + \frac{\Delta m_k \Delta m_k^T}{y_k^T \Delta m_k} \\
&\quad \text{(BFGS)} \\
B_{k+1}^{-1} &= B_k^{-1} + \frac{(\Delta m_k - B_k^{-1} y_k) \Delta m_k B_k^{-1}}{\Delta m_k B_k^{-1} y_k} \\
&\quad \text{(Broyden)} \\
B_{k+1}^{-1} &= B_k^{-1} + \frac{(\Delta m_k - B_k^{-1} y_k)(\Delta m_k - B_k^{-1} y_k)^T}{(\Delta x_k - B_k^{-1} y_k)^T B_k^{-1}} \\
&\quad \text{(SR1)}
\end{aligned}
\tag{44}
$$

Substituting this into Algorithm 3 we obtain the different forms of a Quasi-Newton algorithm. The advantage of the Quasi-Newton method is that the computation of B_k avoids extra computation of a second order Hessian matrix in wave propagation. It only requires gradient and function evaluations in successive steps. Therefore it could prove to be an efficient method for FWI. Recent research compares a limited memore form of the quasi Newton method, L-BFGS, to other methods [44] in FWI; [54] applies a Gauss-Newton type method into frequency based FWI, but only consider small-scale examples for the numerical analysis. Since our focus is on shape based methods, where such Quasi-Newton approaches are not yet well-understood, we will not consider the corresponding Quasi-Newton scheme in this thesis.

3 Numerical implementation on forward modelling

3.1 Brief introduction

In the previous chapter, we constructed a symmetric hyperbolic system based elastic waveform inversion model in a 2D setup. Iterative inversion methods require the repeated solution of forward problems which are numerical realizations of the underlying PDEs. The implementation for such a symmetric hyperbolic system on a computer is another challenge. Due to the complexity and computational cost of an elastic wave model, only few wave modelling open source codes could be found at the moment of writing this thesis. None of them exactly matched the setup which we planned to follow here. However, some were close and served as a basis for what we will explain in the following.

There exists a variety of traditional numerical schemes for wave equations, amongst them finite difference methods, finite element methods, or spectral methods. Of the above, finite difference methods are the most straightforward techniques; first order and higher order finite difference schemes have been widely applied to all types of PDE simulations. They result in fast codes in either time-domain or frequency-domain FWI. However, finite difference methods are not considered the most accurate numerical schemes. Some other research applies high-order finite difference methods instead [37]; this formulation increases the accuracy of computation, but might also cause significant error in discontinuous regions.

Finite element methods, on the other hand, are formulated for solving a weak solution for partial differential equations by constructing finite element functions and solving a linear system for computing all coefficients of those elements to representing the solution. This is a popular numerical technique able to adjust easily to complicated geometries, but its advantage is that for large scale problems the calculations are relatively complex and slow.

An alternative to the above two methods are pseudo-spectral methods [26]; A pseudo-spectral method is a Fourier based numerical technique; it shows higher efficiency and accuracy in wave simulation [40], [27], [25]. However, Its main disadvantage is that it doesn't do well with the boundary value problem in irregular

domains; in that case, the grid refinement seems to be difficult. However, in our model of geophysical free-surface domains, the problem of boundary conditions and irregularity of the domain can be considered of minor importance.

Therefore, our numerical implementation will be based on a forward elastic wave propagation module of the k-wave toolbox [71], [11], which is available as an open source code; as already menioned, we will not be able to directly use that code due to the differences of the underlying hyperbolic PDEs. Instead, we will modify this tool-box module so that it is suited for our symmetric hyperbolic scheme for both forward and backward propagation. We will test this MATLAB based program on some example problems to analyse the results, and then compare our results to other research [25] to see if the forward propagation should be satisfactory for modelling elastic wave simulation.

3.2 Discretization of the wave equation

3.2.1 Numerical scheme

We need to discretize our forward model (11) for the numerical implementation of our first-order symmetric hyperbolic scheme. We assume that the spatial grid for Ω has $N_x \times N_y$ grid points, and that each grid has a size of $\Delta x \times \Delta y$ along the x and y axes; for time domain $[0, T]$, we provide a time difference Δt and the corresponding time grid point N_t such that $T = \Delta t(N_t - 1)$.

3.2.2 Staggered grid

In elastic wave modeling, one of the biggest challenges is to accurately denote and keep track of all the variables and parameters of the accurate displacements in a grid setting. In order to improve the accuracy of the forward modelling, J. Virieux introduced a staggered grid approach [33] for the discretization, reminiscent to similar codes in electromagnetics, such that the implementation is point-wise accurate; B. Cox *et al* [11] modified that so that it satisfies a first-order elastic wave equation scheme. We apply the staggered grid formulation from [11], and modify it so that it is suitable for the discretization of our symmetric hyperbolic system (11).

Recall the formulation (11); firstly, we obtain the staggered grid for the time

derivative. It deduces the following numerical scheme:

$$\begin{aligned}
\frac{v_x^{t+\Delta t/2} - v_x^{t-\Delta t/2}}{\Delta t} &= \frac{1}{\rho^t}\left(\frac{\partial \eta_{xx}^t}{\partial x} + \frac{\partial p^t}{\partial x} + \frac{\partial \eta_{xy}^t}{\partial y} + F_x^t\right) \\
\frac{v_y^{t+\Delta t/2} - v_y^{t-\Delta t/2}}{\Delta t} &= \frac{1}{\rho^t}\left(\frac{\partial \eta_{yy}^t}{\partial y} + \frac{\partial p^t}{\partial y} + \frac{\partial \eta_{xy}^t}{\partial x} + F_y^t\right) \\
\frac{\eta_{xx}^{t+\Delta t} - \eta_{xx}^t}{\Delta t} &= 2\mu^{t+\Delta t/2}\frac{\partial v_x^{t+\Delta t/2}}{\partial x} \\
\frac{\eta_{yy}^{t+\Delta t} - \eta_{yy}^t}{\Delta t} &= 2\mu^{t+\Delta t/2}\frac{\partial v_y^{t+\Delta t/2}}{\partial y} \\
\frac{\eta_{xy}^{t+\Delta t} - \eta_{xy}^t}{\Delta t} &= \mu^{t+\Delta t/2}\left(\frac{\partial v_x^{t+\Delta t/2}}{\partial y} + \frac{\partial v_y^{t+\Delta t/2}}{\partial x}\right) \\
\frac{p^{t+\Delta t} - p^t}{\Delta t} &= \lambda^{t+\Delta t/2}\left(\frac{\partial v_x^{t+\Delta t/2}}{\partial x} + \frac{\partial v_y^{t+\Delta t/2}}{\partial y}\right)
\end{aligned} \tag{45}$$

such that for all parameters, the updated variables lie at the correct location. A clear view of the grid geometry for the corresponding parameters and variables on the spatial domain is displayed in the following graphic.

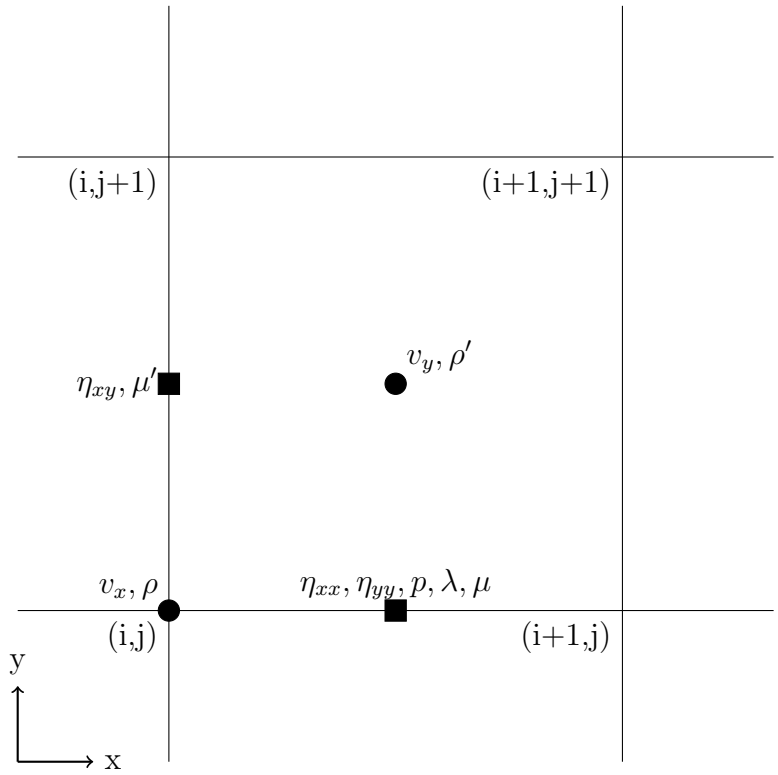


Figure 1: Grid geometry for a staggered grid in 2D Cartesian coordinates suited for implementing the discretized symmetric hyperbolic system

provided with the notation

$$
\begin{aligned}
v_x &= v_x^{i,j} \\
v_y &= v_y^{i+1/2,j+1/2} \\
\eta_{xx} &= \eta_{xx}^{i+1/2,j} \\
\eta_{yy} &= \eta_{yy}^{i+1/2,j} \\
\eta_{xy} &= \eta_{xy}^{i,j+1/2} \\
p &= p^{i+1/2,j} \\
\lambda &= \lambda^{i+1/2,j} \\
\mu &= \mu^{i+1/2,j} \\
\mu' &= \mu^{i,j+1/2} \\
\rho &= \rho^{i,j} \\
\rho' &= \rho^{i+1/2,j+1/2}
\end{aligned}
\tag{46}
$$

Define $i = 1, 2, ..., N_x$ on the x-axis, and $j = 1, 2, ..., N_y$ on the y-axis. We also denote $\xi_1^{t+\Delta t/2} = \xi^+, \xi_1^{t-\Delta t/2} = \xi^-$ for $\xi_1 \in (v_x, v_y)$, and $\xi_2^{t+\Delta t} = \xi_2^+, \xi_2^t = \xi_2^-$ for $\xi_2 \in (\eta_{xx}, \eta_{yy}, \eta_{xy}, p)$; combining the above, we obtain the final staggered grid formulation in the full domain:

$$
\begin{aligned}
\frac{v_x^{+,i,j} - v_x^{-,i,j}}{\Delta t} &= \frac{1}{\rho^{i,j}}(f_x^{-,i,j} + \frac{p^{-,i+1/2,j} - p^{-,i-1/2,j}}{\Delta x} \\
&+ \frac{\eta_{xx}^{-,i+1/2,j} - \eta_{xx}^{-,i-1/2,j}}{\Delta x} + \frac{\eta_{xy}^{-,i,j+1/2} - \eta_{xy}^{-,i,j-1/2}}{\Delta y}) \\
\frac{v_y^{+,i+1/2,j+1/2} - v_y^{-,i+1/2,j+1/2}}{\Delta t} &= \frac{1}{\rho^{i+1/2,j+1/2}}(f_y^{-,i+1/2,j+1/2} + \frac{p^{-,i+1/2,j+1} - p^{-,i+1/2,j}}{\Delta y} \\
&+ \frac{\eta_{xy}^{-,i+1,j+1/2} - \eta_{xy}^{-,i,j+1/2}}{\Delta x} + \frac{\eta_{yy}^{-,i+1/2,j+1} - \eta_{yy}^{-,i+1/2,j}}{\Delta y}) \\
\frac{\eta_{xx}^{+,i+1/2,j} - \eta_{xx}^{-,i+1/2,j}}{\Delta t} &= 2\mu^{i+1/2,j}\frac{v_x^{+,i+1,j} - v_x^{+,i,j}}{\Delta x} \\
\frac{\eta_{yy}^{+,i+1/2,j} - \eta_{yy}^{-,i+1/2,j}}{\Delta t} &= 2\mu^{i+1/2,j}\frac{v_y^{+,i+1/2,j+1/2} - v_y^{+,i+1/2,j-1/2}}{\Delta y} \\
\frac{\eta_{xy}^{+,i,j+1/2} - \eta_{xy}^{-,i,j+1/2}}{\Delta t} &= \mu^{i,j+1/2}(\frac{v_x^{+,i,j+1} - v_x^{+,i,j}}{\Delta y} + \frac{v_y^{+,i+1/2,j+1/2} - v_y^{+,i-1/2,j+1/2}}{\Delta x}) \\
\frac{p^{+,i+1/2,j} - p^{-,i+1/2,j}}{\Delta t} &= \lambda^{i+1/2,j}(\frac{v_x^{+,i+1,j} - v_x^{+,i,j}}{\Delta x} + \frac{v_y^{+,i+1/2,j+1/2} - v_y^{+,i+1/2,j-1/2}}{\Delta y})
\end{aligned}
\tag{47}
$$

We consider the forward and backward operator D^+ and D^-, respectively, on the spatial domain. For any variable u, these operators are applied to compute the spatial derivatives:

$$
\begin{aligned}
D_x^+ u_x &= \frac{u_{(x+\Delta x)} - u_x}{\Delta x} \\
D_x^- u_x &= \frac{u_x - u_{(x-\Delta x)}}{\Delta x} \\
D_y^+ u_y &= \frac{u_{(y+\Delta y)} - u_y}{\Delta y} \\
D_y^- u_y &= \frac{u_y - u_{(y-\Delta y)}}{\Delta y}
\end{aligned}
$$

We apply this formulation in (47) to obtain a simplified numerical scheme of the

forward model:

$$
\begin{aligned}
\frac{1}{\Delta t}(v_x^+ - v_x^-) &= \frac{1}{\rho}(f_x^- + D_x^- p^- + D_x^- \eta_{xx}^- + D_y^- \eta_{xy}^-) \\
\frac{1}{\Delta t}(v_y^+ - v_y^-) &= \frac{1}{\rho'}(f_x^- + D_y^+ p^- + D_x^+ \eta_{xy}^- + D_y^+ \eta_{yy}^-) \\
\frac{1}{\Delta t}(\eta_{xx}^+ - \eta_{xx}^-) &= 2\mu D_x^+ v_x^+ \\
\frac{1}{\Delta t}(\eta_{yy}^+ - \eta_{yy}^-) &= 2\mu D_y^- v_y^+ \\
\frac{1}{\Delta t}(\eta_{xy}^+ - \eta_{xy}^-) &= \mu'(D_y^+ v_x^+ + D_x^- v_y^+) \\
\frac{1}{\Delta t}(p^+ - p^-) &= \lambda(D_x^+ v_x^+ + D_y^- v_y^+)
\end{aligned}
\tag{48}
$$

This derivation gives us a more straight-forward implementation on a staggered grid based numerical scheme.

3.3 A k-wave modelling applying pseudo-spectral method

As mentioned, we will build our code on an open source forward modelling code known as k-wave designed by B. Cox, B. Treeby and J. Jaros [71]; This code has been initially constructed for acoustic modelling in ultrasound propagation in a biomedical application, aiming at simulating the wave propagation inside body vessels or tissue. An elastic forward propagation simulation has also been developed as part of that code in 2015, incorporating an elastic module into the k-wave toolbox [11].

The k-wave toolbox applies a pseudo-spectral method by using Fourier transforms to calculate the spatial derivatives for the wave variables, and the finite difference scheme for the time derivative; In this subsection, we will introduce the pseudo-spectral method for first-order spatial derivatives applied in the k-wave approach, and apply it in the numerical scheme in (48). We apply the same pseudo-spectral method algorithm to our own model implementation, and then alter the program code so that it is suitable for the symmetric hyperbolic form. We apply the modified code in our forward modelling, extend it to the adjoint or time-reversed setup, and then use both as part of our FWI gradient calculation.

3.3.1 A k-wave based spatial derivative

Before getting into technical details of the pseudo-spectral scheme, we must first understand the basic idea of pseudo-spectral methods. the k-wave toolbox [71] applies a pseudo-spectral method based on Fourier transformation in a non-periodic domain. Let us consider a 1D function $f(x)$, such that $f : [0, L] \rightarrow \mathcal{R}$, with an expansion

$$f(x) = \sum_{n=0}^{N-1} f_n \overline{\phi_n(x)}$$

with $f_n = \langle f, \phi_n \rangle$; here ϕ_n is an orthonormal basis. Consider the Fourier basis functions on $[0, L]$ such that

$$\phi_n(x) = \frac{1}{\sqrt{L}} e^{-ik_n x}$$

where k_n is considered to be the wave number with $k = [k_0, k_1, ..., k_{N-1}]$, such that

$$\begin{aligned} k &= [-\frac{N}{2}, -\frac{N}{2} + 1, ..., \frac{N}{2} - 1]\frac{2\pi}{L} \quad \textit{for } N \textit{ even} \\ k &= [-\frac{N-1}{2}, -\frac{N-1}{2} + 1, ..., \frac{N-1}{2}]\frac{2\pi}{L} \quad \textit{for } N \textit{ odd} \end{aligned} \tag{49}$$

Therefore we obtain the Fourier transform in the form

$$\mathcal{F}(k) = \frac{1}{\sqrt{L}} \int_0^L f(x) e^{-ikx} \mathrm{d}x$$

and

$$\mathcal{F}^{-1}(x) = \frac{1}{\sqrt{L}} \sum_{k=0}^{N-1} \mathcal{F}(k) e^{ikx}$$

The above formulation provides a Fourier based representation for first-order derivatives such that

$$\partial_x f = \mathcal{F}^{-1}(ik\mathcal{F}(k)) \tag{50}$$

We apply the pseudo-spectral method for first-order spatial differences, as an alternative to traditional finite difference methods. Notice that this is the same formulation as applied in k-wave toolbox [71].

3.3.2 Updated numerical scheme with pseudo-spectral method

Recall the staggered grid numerical scheme (48); consider the pseudo-spectral first order derivative (52); then we have that $\mathcal{F}(f(x+d)) = e^{-ikd}\mathcal{F}(f(x))$. From the above formulation, we introduce the pseudo-spectral based operators such that

$$\begin{aligned}
D_x^F v_x &= \mathcal{F}_x^{-1}\{ik_x \mathrm{e}^{ik_x\Delta x/2}\mathcal{F}_x\{v_x\}\} \\
D_x^F v_y &= \mathcal{F}_x^{-1}\{ik_x \mathrm{e}^{-ik_x\Delta x/2}\mathcal{F}_x\{v_y\}\} \\
D_y^F v_x &= \mathcal{F}_y^{-1}\{ik_y \mathrm{e}^{+ik_y\Delta y/2}\mathcal{F}_y\{v_x\}\} \\
D_y^F v_y &= \mathcal{F}_y^{-1}\{ik_y \mathrm{e}^{-ik_y\Delta y/2}\mathcal{F}_y\{v_y\}\} \\
D_x^F \eta_{xx} &= \mathcal{F}_x^{-1}\{ik_x \mathrm{e}^{+ik_x\Delta x/2}\mathcal{F}_x\{\eta_{xx}\}\} \\
D_y^F \eta_{yy} &= \mathcal{F}_y^{-1}\{ik_y \mathrm{e}^{+ik_y\Delta y/2}\mathcal{F}_y\{\eta_{yy}\}\} \\
D_y^F \eta_{xy} &= \mathcal{F}_y^{-1}\{ik_x \mathrm{e}^{-ik_y\Delta y/2}\mathcal{F}_y\{\eta_{xy}\}\} \\
D_y^F \eta_{xy} &= \mathcal{F}_y^{-1}\{ik_y \mathrm{e}^{-ik_y\Delta y/2}\mathcal{F}_y\{\eta_{xy}\}\} \\
D_x^F p_x &= \mathcal{F}_x^{-1}\{ik_x \mathrm{e}^{+ik_x\Delta x/2}\mathcal{F}_x\{p_x\}\} \\
D_y^F p_y &= \mathcal{F}_y^{-1}\{ik_y \mathrm{e}^{+ik_y\Delta y/2}\mathcal{F}_y\{p_y\}\}
\end{aligned} \tag{51}$$

where k_x, k_y are defined as the wave numbers computed for Fourier transforms along the x and y axes, with the corresponding grid numbers N_x, N_y from (49). The above provides the pseudo-spectral based elastic scheme

$$\begin{aligned}
v_x^+ &= v_x^- + \frac{\Delta t}{\rho}(f_x^- + D_x^F p^- + D_x^F \eta_{xx}^- + D_y^F \eta_{xy}^-) \\
v_y^+ &= v_y^- + \frac{\Delta t}{\rho'}(f_y^- + D_y^F p^- + D_x^F \eta_{xy}^- + D_y^F \eta_{yy}^-) \\
\eta_{xx}^+ &= \eta_{xx}^- + 2\Delta t\mu D_x^F v_x^+ \\
\eta_{yy}^+ &= \eta_{yy}^- + 2\Delta t\mu D_y^F v_y^+ \\
\eta_{xy}^+ &= \eta_{xy}^- + \Delta t\mu'(D_y^F v_x^+ + D_x^F v_y^+) \\
p^+ &= p^- + \Delta t\lambda(D_x^F v_x^+ + D_y^F v_y^+)
\end{aligned} \tag{52}$$

Therefore we obtain the discretization of the symmetric hyperbolic form using a pseudo-spectral method.

3.3.3 Perfectly matched layer

In our wave model, we assume that the elastic wave travels in domain with a free surface; this implies that there should be no reflection at the boundary of the domain. In order to simulate this, we will set up a layer such that the wave attenuates towards zero inside this layer, and the energy disappears at the boundary, so that there will be no reflected waves at the boundaries, and no wave energy can enter from the outside of the domain.

This method is known as the perfectly matched layer (PML) technique. Perfectly matched layers are defined as thin absorbing boundary layers surrounding the domain, that give rise to anisotropic absorption inside these layers. The construction of PML is designed to satisfy the following conditions

- The PML must provide sufficient absorption such that wave propagating toward its outer boundary contains almost no energy.
- The layer itself should not cause any wave reflection.

The setting of PML's requires two parameters: the layer width and the absorption coefficient. We define α to be the anisotropic absorption coefficient of the PML. In the k-wave toolbox [71], the setting of PML applies the split-field formulation by J. Berenger [6]; we apply the same approach in our model.

We assume that α is independent of the wave variable; the definition of the PML suggests that α is an attenuation coefficient inside PML, and zero elsewhere. Consider a generalised first-order partial differential equation with α as attenuation

$$V_t = M - \alpha V \tag{53}$$

where V, M are arbitrary real function. We recall the relation that

$$(\partial_t + \alpha)V = \mathrm{e}^{-\alpha t}\partial_t(\mathrm{e}^{\alpha t}V)$$

Consider the staggered grid finite difference scheme in time domain; we obtain that

$$\begin{aligned} \mathrm{e}^{\alpha(t+\Delta t/2)}V^{t+\Delta t/2} &= \mathrm{e}^{\alpha(t-\Delta t/2)}V^{t-\Delta t/2} + \mathrm{e}^{\alpha t}\Delta t M^t \\ V^{t+\Delta t/2} &= \mathrm{e}^{-\alpha\Delta t/2}(\mathrm{e}^{-\alpha\Delta t/2}V^{t-\Delta t/2} + \Delta t M^t) \end{aligned} \tag{54}$$

We should apply the same attenuation in the numerical scheme of (52). Substituting (54), we obtain the refined numerical scheme with PML setting as

$$\begin{aligned}
v_x^+ &= \mathrm{e}^{-\alpha\Delta t/2}(\mathrm{e}^{-\alpha\Delta t/2}v_x^- + \frac{\Delta t}{\rho}(f_x^- + D_x^F p^- + D_x^F \eta_{xx}^- + D_y^F \eta_{xy}^-)) \\
v_y^+ &= \mathrm{e}^{-\alpha\Delta t/2}(\mathrm{e}^{-\alpha\Delta t/2}v_y^- + \frac{\Delta t}{\rho'}(f_y^- + D_y^F p^- + D_x^F \eta_{xy}^- + D_y^F \eta_{yy}^-)) \\
\eta_{xx}^+ &= \mathrm{e}^{-\alpha\Delta t/2}(\mathrm{e}^{-\alpha\Delta t/2}\eta_{xx}^- + 2\Delta t\mu D_x^F v_x^+) \\
\eta_{yy}^+ &= \mathrm{e}^{-\alpha\Delta t/2}(\mathrm{e}^{-\alpha\Delta t/2}\eta_{yy}^- + 2\Delta t\mu D_y^F v_y^+) \\
\eta_{xy}^+ &= \mathrm{e}^{-\alpha\Delta t/2}(\mathrm{e}^{-\alpha\Delta t/2}\eta_{xy}^- + \Delta t\mu'(D_y^F v_x^+ + D_x^F v_y^+)) \\
p^+ &= \mathrm{e}^{-\alpha\Delta t/2}(\mathrm{e}^{-\alpha\Delta t/2}p^- + \Delta t\lambda(D_x^F v_x^+ + D_y^F v_y^+))
\end{aligned} \tag{55}$$

Notice that the parameter α should be chosen properly for the wave propagation. For very small α, the thickness of the PML layer might not be sufficient in order to attenuate the wave enough; on the other hand, a large value of α results in a large gradient between the PML boundary and the adjacent grid point inside the domain next to the PML boundary, which will cause a reflection of the incoming wave. One way to address this tradeoff is to set α as a function of position within the PML, where $\alpha_\xi = \alpha_\xi(\xi)$, where ξ is defined to be the coordinate inside the PML perpendicular to the PML-domain interface. The detailed choice of this functional relationship is a matter of ongoing research. One option of this PML definition by [72] has been applied in the k-wave toolbox [71]; we apply the same PML function in our model:

$$\alpha_\xi = \alpha_{max}(\frac{\xi - \xi_0}{\xi_{max} - \xi_0})^4 \tag{56}$$

where ξ_0 is the coordinate at the internal boundary of the PML, and ξ_{max} is the coordinate at the external boundary; α_{max} is the maximum of the wave absorption parameter. For convenience, we set $\alpha_{max} = 2$ in our simulations.

3.4 Stability analysis

In order to complete the wave propagation model, we need to consider also the stability of the numerical scheme. Two of the main conditions to be satisfied in FWI are the Courant instability and the numerical dispersion [26]. We will briefly discuss these two conditions in the following.

3.4.1 Courant instability

Our wave propagation should satisfy the condition of numerical convergence during the simulations; a necessary condition is called the Courant-Freidrichs-Lewy (CFL) condition. This condition states that for first-order schemes in two dimensions, the grid setting should satisfy

$$\sqrt{2}|V_{max}\frac{\Delta t}{\Delta x}| < 1$$

We define the Courant number

$$CFL = V_{max}\frac{\Delta t}{\Delta x}$$

where V_{max} is taken as the maximum value of compressional wave speed V_p. Theoretically, a good choice of Δx and Δt should satisfy $CFL \leq 1$, to guarantee the convergence of simulation; but practically, that is not sufficient. Numerical tests suggest that an upper bound for the CFL condition for convergence should be that $CFL \approx 0.45$.

3.4.2 Numerical dispersion

In addition to the Courant instability, numerical dispersion is also an important factor that may serious affect the accuracy of the elastic wave propagation [26]. In signal processing with discretized time steps and spatial grids, sometimes the signal may not be able to record all the information of the full waveform. The Nyquist-Shannon sampling theorem provides a criterion for a good choice for the grid length. It is $\Delta x = \lambda/2$, here λ is the wave length provided by $\lambda = V/f$. However, in most cases this condition is still insufficient for a good wave simulation. We choose a simple example to demonstrate the inaccuracy caused by numerical dispersion. Consider the wave displacement for the example $u_x = \sin(2\pi x/\lambda)$; At $x = 0$, the analytical calculation for $\partial u_x/\partial x$ is

$$\frac{\partial u_x}{\partial x}|_{x=0} = \frac{2\pi}{\lambda}\cos(2\pi x/\lambda)|_{x=0} = 2\pi/\lambda$$

The staggered grid numerical scheme requires

$$\frac{\partial u_x}{\partial x}|_{x=0} = \frac{\sin(2\pi(\Delta x/2)/\lambda) - \sin(2\pi(-\Delta x/2)/\lambda)}{\Delta x}$$

The grid setting with $\Delta x = \lambda/2$ acquires the result that $\frac{\partial u_x}{\partial x}|_{x=0} = \frac{4}{\lambda}$, which is incorrect compared to the analytical result of $\frac{2\pi}{\lambda}$. However, if we set that $n = \lambda\Delta x$ larger, the computational accuracy would be improved. For example, if we set $\Delta x = \lambda/16$, for $\lambda = 1$, the numerical result will be $\partial u_x/\partial x \approx 6.2429$, which amounts to a much higher accuracy.

In elastic wave simulation, We define the dispersion number $n = \frac{\lambda}{\Delta x} = \frac{V_{min}}{f\Delta x}$, as $n = 2$ satisfies the Nyquist criterion; in elastic wave propagation, we consider that V_{min} is the minimum of the shear wave velocity V_s.

3.5 Forward modelling test

3.5.1 A SOFI2D example

We test our model on a homogeneous medium example, borrowed from a test example in [8], where $V_p = 3500(m/s)$, $V_s = 2000(m/s)$, $\rho = 2000(kg/m^3)$; the grid size is set to be $54 \times 54(m^2)$, and the grid number is set to be 100×100. We consider as wave source a Ricker wavelet of the form

$$\begin{aligned} R(\tau) &= (1 - 2\tau^2)\exp(-\tau^2), \, as \\ \tau &= \pi f(t - t_d) - k\pi \end{aligned} \tag{57}$$

We select the fixed parameter $t_d = 0$, and the central frequency f alters with the grid dispersion number for cases that $n = 2$, $n = 4$, and $n = 8$; the corresponding central wave frequencies (Hz) are calculated as $f_2 = 18.5185$, $f_4 = 9.2593$, $f_8 = 4.6296$. k is defined according to f such that the starting time of the Ricker wave is approximately the same while f changes.

The time step is automatically set to fit the CFL number such that $CFL = 0.3$, below the numerical upper bound for numerical convergence that $CFL = 0.45$; the computation finally obtains that $\Delta t = 0.0046$. The time length is set to be 472, so that it guarantees the maximum time that elastic wave travels throughout the domain for any initial conditions. The parameter setting follows the formulation provided by (56).

In the following figures, we demonstrate the results of the simulations concentrating on wave pressure p as part of the numerical implementation of (11). In the following figures we display the wave-fronts at different time steps, with different wavelet set-ups corresponding to different grid dispersion numbers.

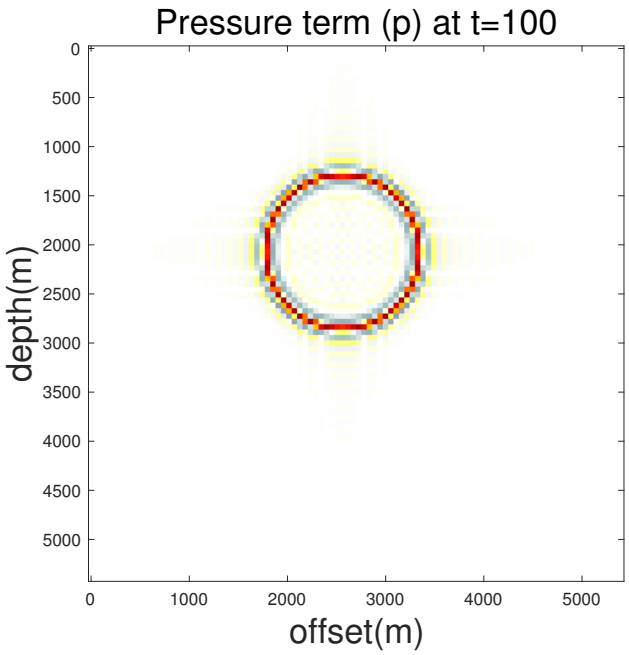


Figure 2: Wave propagation at t=100, n=2

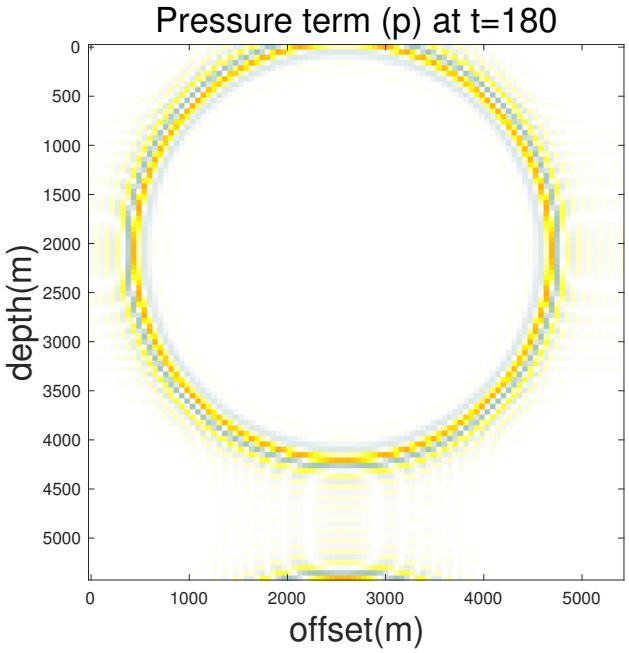


Figure 3: Wave propagation at t=180, n=2

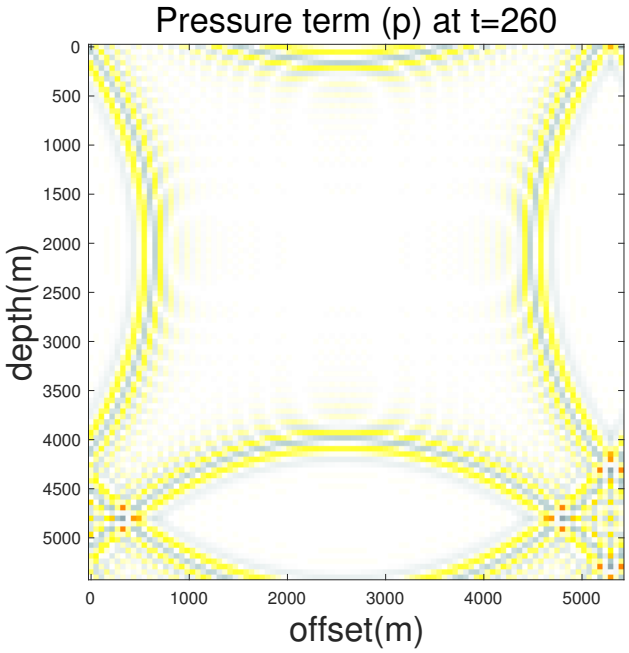


Figure 4: Wave propagation at t=260, n=2

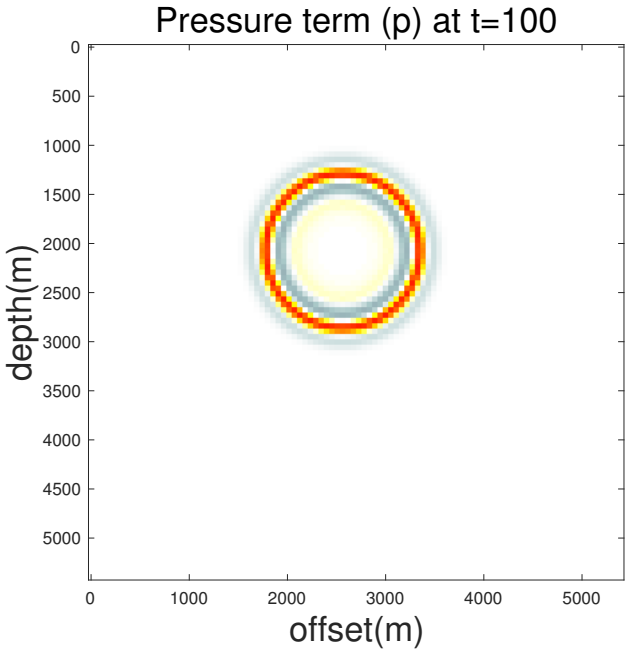


Figure 5: Wave propagation at t=100, n=4

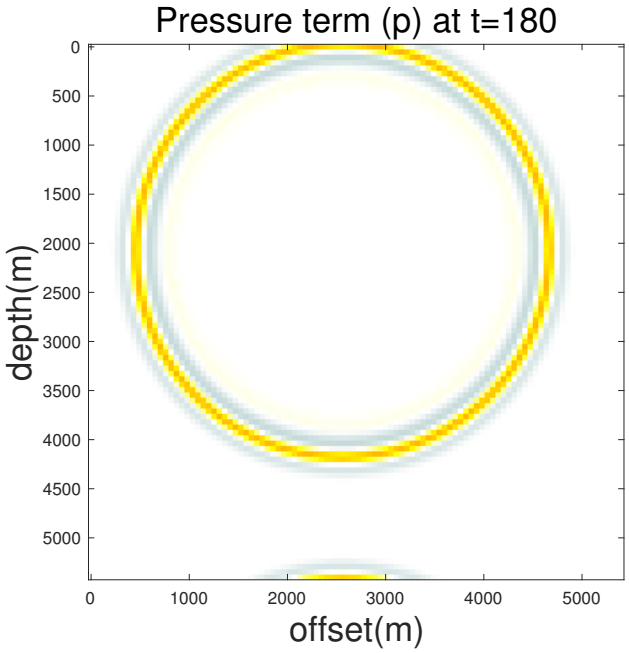


Figure 6: Wave propagation at t=180, n=4

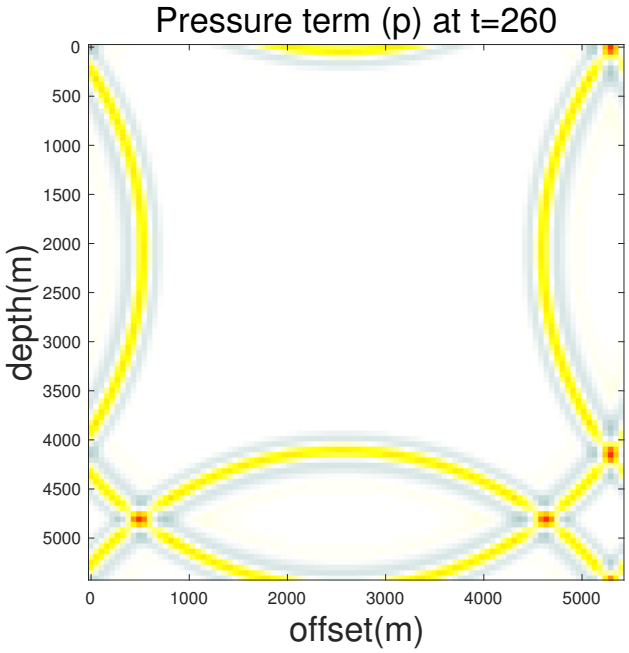


Figure 7: Wave propagation at t=260, n=4

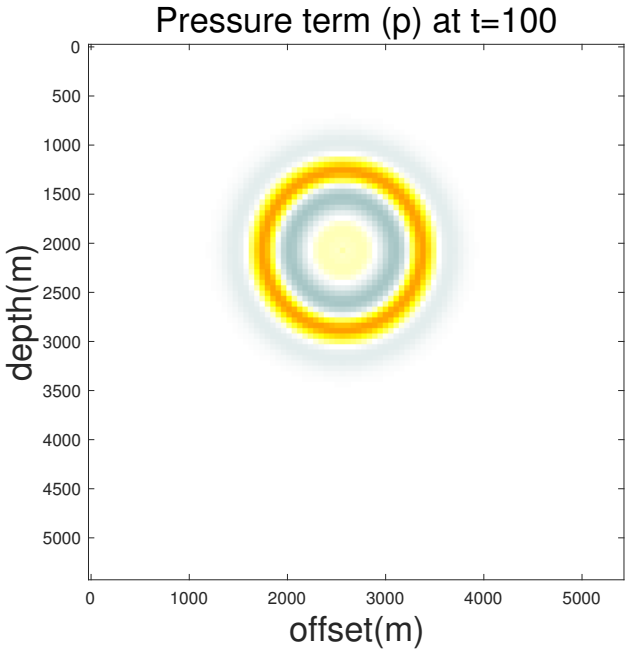


Figure 8: Wave propagation at t=100, n=8

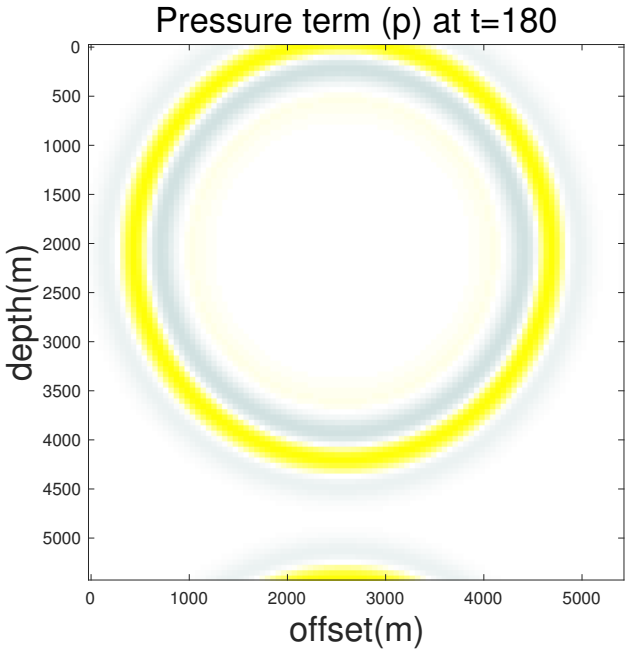


Figure 9: Wave propagation at t=180, n=8

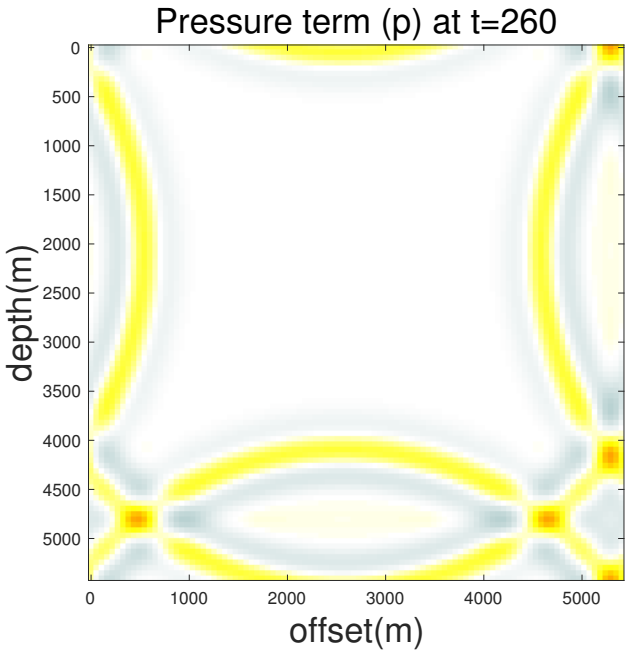


Figure 10: Wave propagation at t=260, n=8

We can see clearly that in our numerical experiments, when $n = 2$ the affect of numerical dispersion is relatively strong around the wave-front; while with

$n = 4$ and $n = 8$, the wavefront seems sufficiently sharp, and the affect of dispersion diminishes; this indicates that the wave propagation information has been relatively well recorded. This wave-front character shows good similarity to the wave-front simulation demonstrated in SOFI2D [8] with the same test example, demonstrating that the wave simulation accuracy increases when the dispersion number n increases. Even though this is not a strict proof of equivalence with the SOFI2D results, this test however tends to confirm the accuracy of our modified forward model.

3.5.2 PML test

In order to demonstrate the functioning of our PML approach, we will now compare the numerical results of the wave propagation with and without PML; our setting of PML is [10,10] in x and y axis.

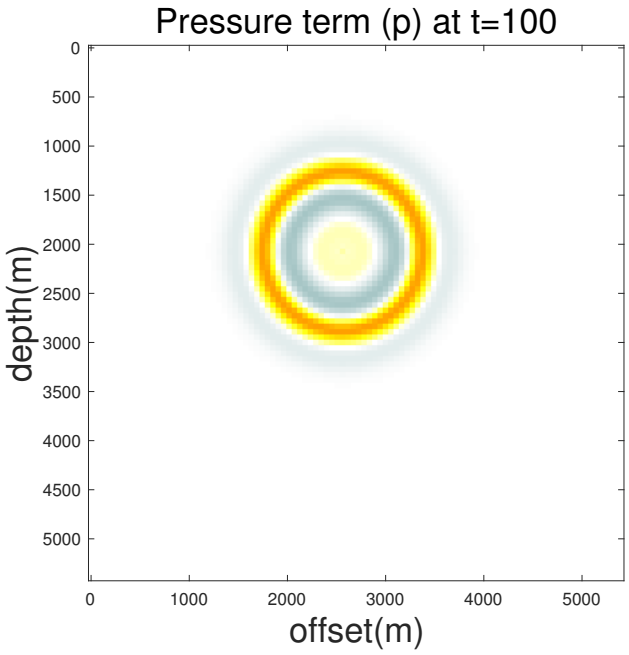


Figure 11: Wave propagation at t = 100, no PML

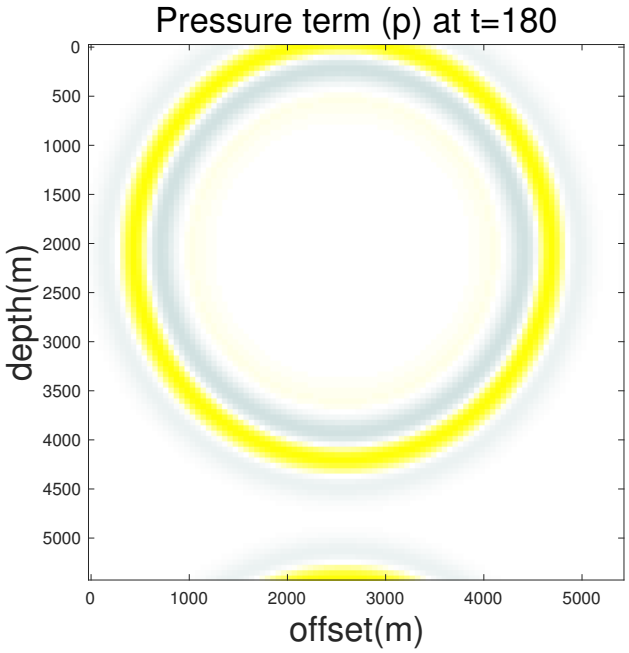


Figure 12: Wave propagation at t = 180, no PML

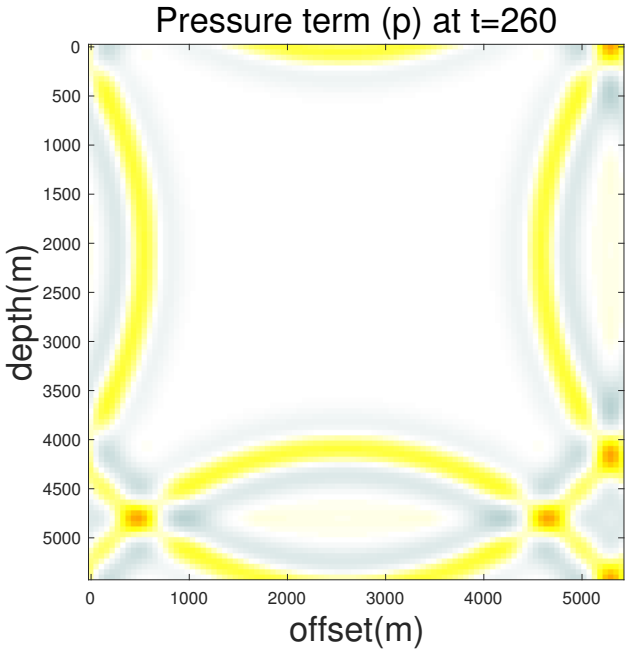


Figure 13: Wave propagation at t = 180, no PML

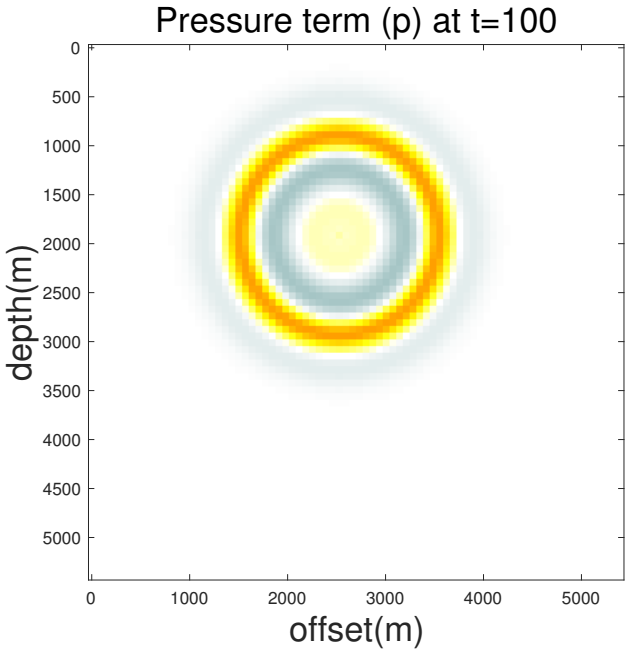


Figure 14: Wave propagation at t=100, PML = [10,10]

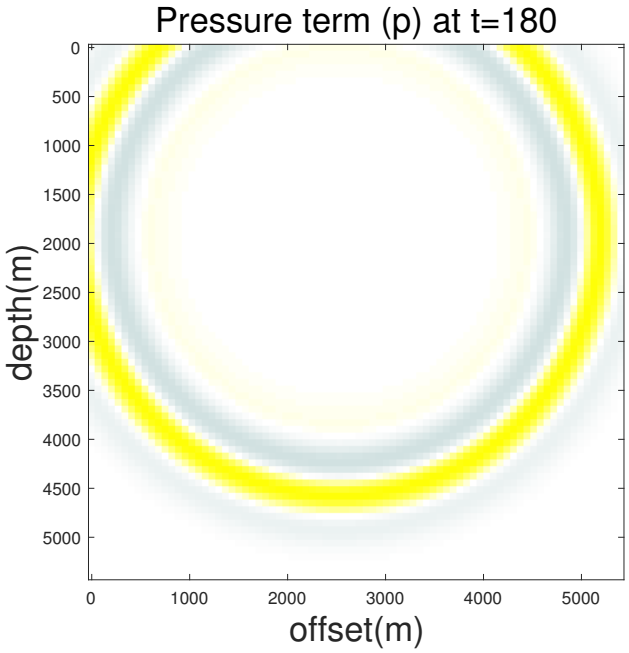


Figure 15: Wave propagation at t=180, PML = [10,10]

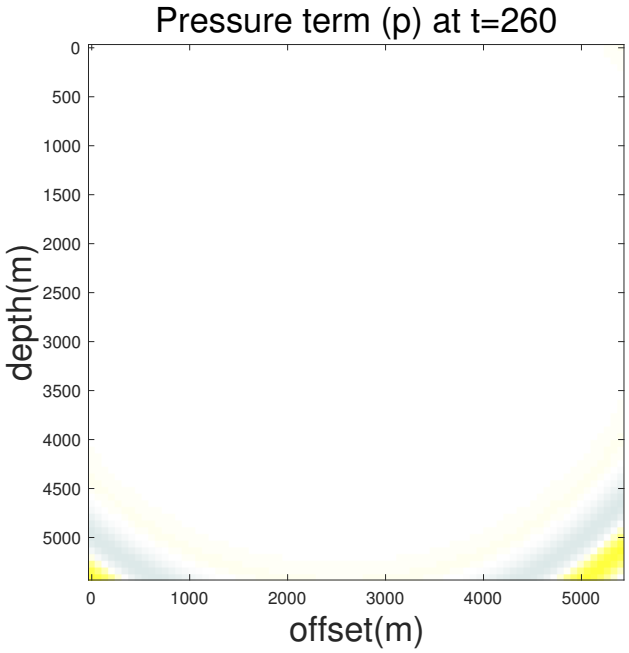


Figure 16: Wave propagation at t=260, PML = [10,10]

We compare the wave propagation with and without PML; in figures 11, 12 and 13, we present a simulation without a PML at the boundary of the domain; it can be observed that a reflective wave exists close to the boundary of the domain. In figures 14, 15 and 16, we introduce a PML with parameters $[10, 10]$ in the x-axis and y-axis. The results clearly demonstrate that reflections are suppressed at the boundary, which indicates the importance of using such a PML; the wave energy close to the boundary has been well absorbed, and therefore it produces no reflection wave back into the domain, so that it guarantees a good 'free-surface' simulation for the wave propagation.

3.5.3 Tests on numerical dispersion

Previous tests show that numerical dispersion can significantly affect the simulation for wave propagation in elastics wave modeling. We would like to be able to avoid any sort of artificial oscillations which would affect the accuracy of wavelet simulation; but the choice of the grid numbers should both consider computational cost and accuracy. Generally speaking, smaller dispersion numbers n indicate more grid and higher accuracy, but the closer n is to Nyquist number 2, the wavelet is more likely to be affected by dispersion. Faccioli [24] states that

a good choice of numerical dispersion should be that $n \geq 4$, so that less number is required to avoid grid dispersion. Therefore, our goal would be to determine a numerical condition as to avoid such grid dispersion.

E. Faccolini [24] presents a similar numerical test on elastic wave propagation using pseudo-spectral methods, also applying a Ricker wavelet as the source form; we should expect that our numerical test would obtain a similar result. Considering the survey being homogeneous, we should expect the wavelet form similar to that in [24].

Based on the previous tests, we alter the Ricker wavelet so that its central frequency satisfies the grid number $n = 2, 2.5, 3, 4, 8$. We then record the seismogram for all 6 wave variables from 11 at [60, 60]; with PML set up to be [10,10], this sensor is located close to the boundary region, and therefore the wave-front would not be affected. We demonstrate the following results:

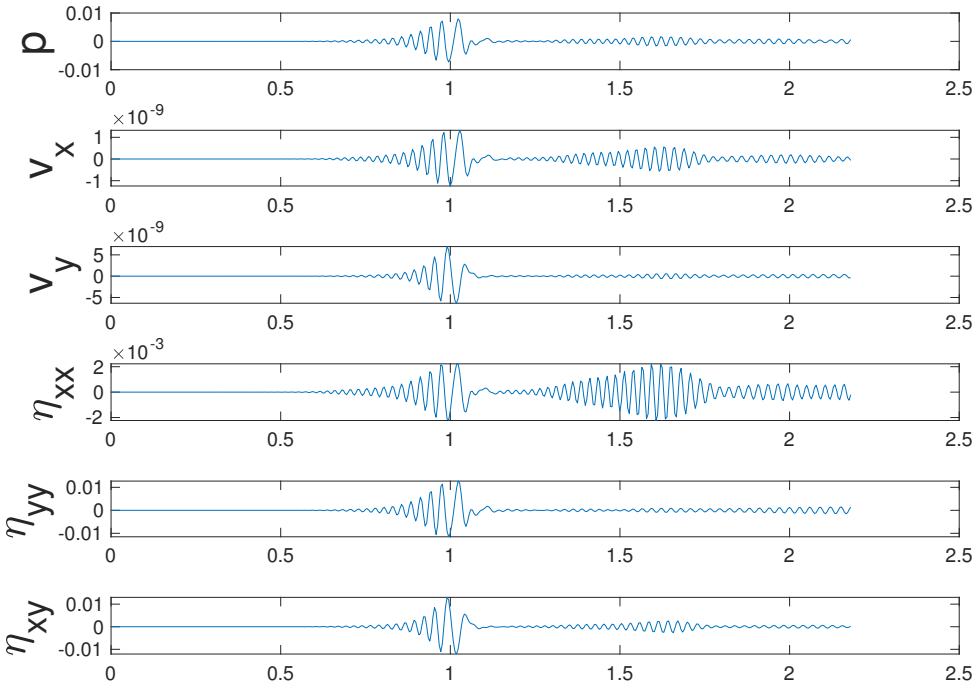


Figure 17: Seismogram of wave variables: from up to bottom are p, v_x, v_y, η_{xx}, η_{yy}, η_{xy}, with $n = 2$

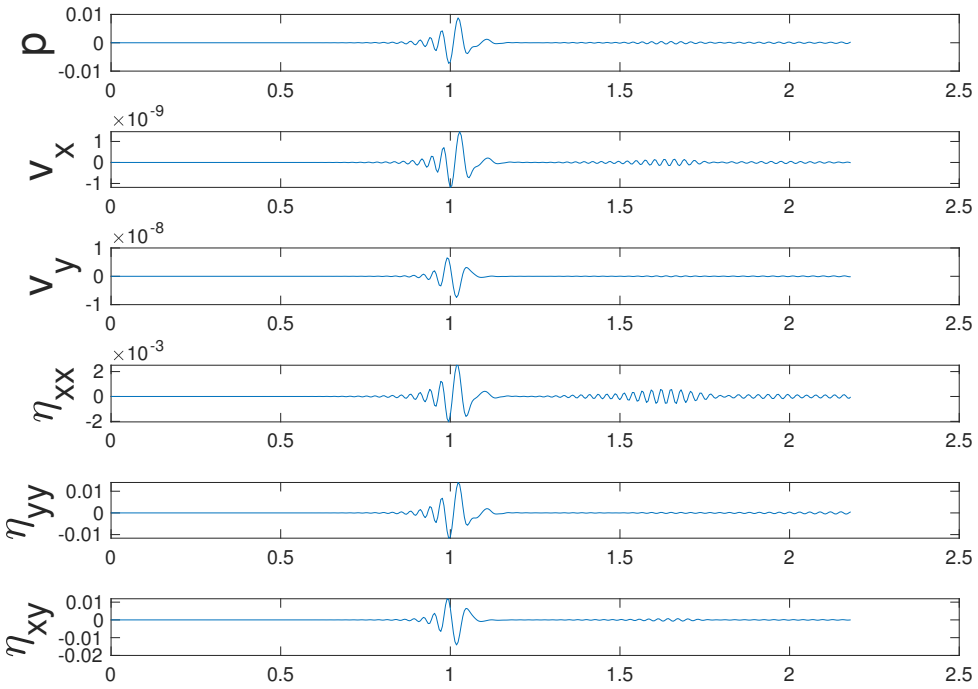


Figure 18: Seismogram of wave variables: from up to bottom are p, v_x, v_y, η_{xx}, η_{yy}, η_{xy}, with $n = 2.5$

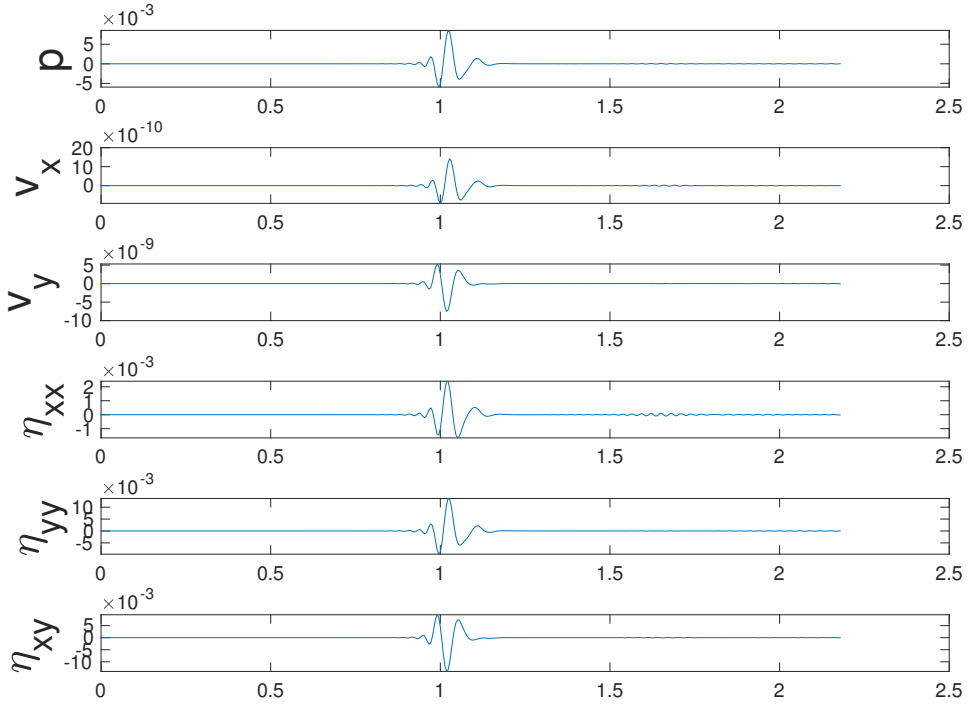


Figure 19: Seismogram of wave variables: from up to bottom are p, v_x, v_y, η_{xx}, η_{yy}, η_{xy}, with $n = 3$

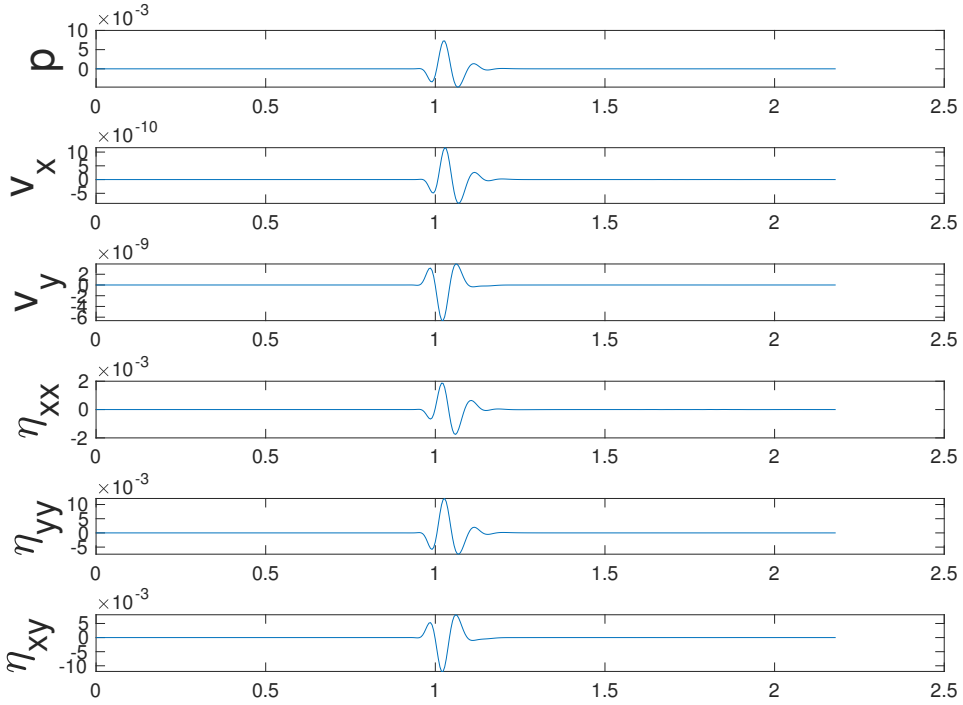


Figure 20: Seismogram of wave variables: from up to bottom are p, v_x, v_y, η_{xx}, η_{yy}, η_{xy}, with $n = 4$

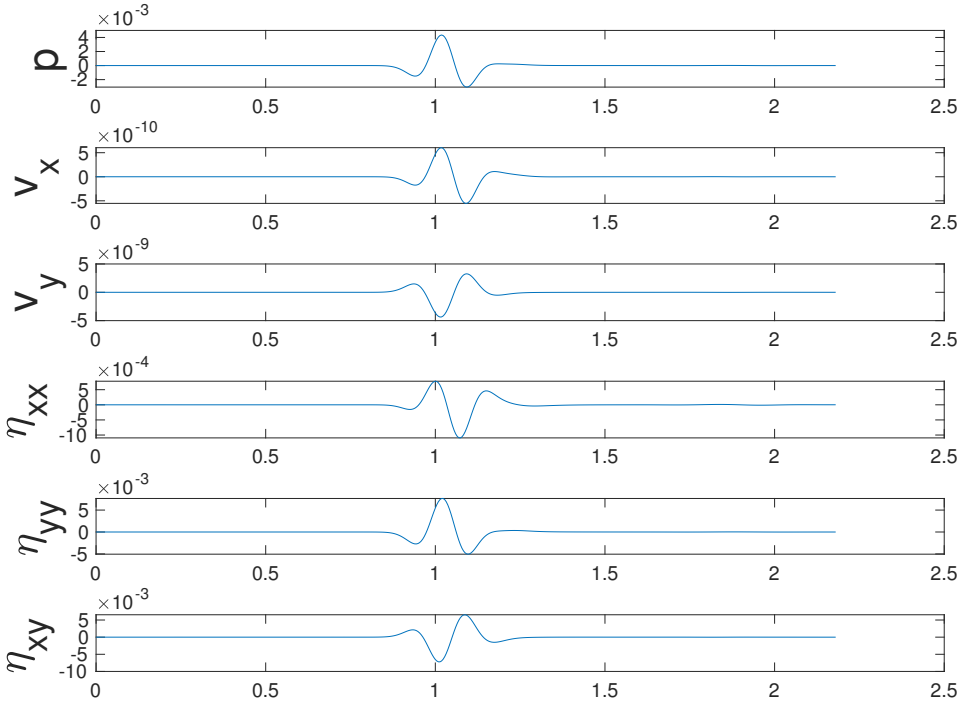


Figure 21: Seismogram of wave variables: from up to bottom are p, v_x, v_y, η_{xx}, η_{yy}, η_{xy}, with $n = 8$

For result of $n = 2, 2.5$, the seismogram shows clear wave oscillation, which would seriously affect the accuracy of our numerical result; with $n = 3$ the wavelet is relatively stable. As n increases, this phenomenon seems to be weaker. Therefore we could reach a conclusion that $n = 3$ is considered the lower boundary of good dispersion number. In FWI, however, we should consider the case where the good wave condition changes with the wave model updates; therefore to ensure avoiding wave dispersion, we would choose to set the central frequency so that $n \geq 4$, as to avoid the possible instability during the model update as numerical dispersion number n changes.

3.6 Conclusion

This section shows a numerical implementation for the elastic wave propagation using a modified version of a module available in the k-wave toolbox [71], [11]; we introduce the pseudo-spectral method in spatial derivatives, a staggered

grid, a perfectly matched layer (PML), the selection of the source wavelet, CFL and grid dispersion that might affect the accuracy of numerical implementation. Finally, we simulate some test cases using some examples from [8], and obtain similar results as presented there; we then compare our seismogram to other wave propagation tests [24], where we also obtain similar results. This indicates that our modification of the k-wave toolbox for our elastic wave forward model is successful.

In the next chapter, we will be concentrating on the numerical implementation of inverse problem based on the above forward model; we will introduce different groups of numerical tests on the simulation of seismic imaging that detect various objects inside the seismic background (salt domes), and compare different line search techniques and Sobolev gradients on a pixel based setup.

4 Inverse implementation for a pixel based case

4.1 Overview

In FWI, the inversion theory with respect to elastic wave equations has been developed in the 1980s [75]; however, due to the computational limitations and the complexity of elastic modelling, the development of elastic FWI codes has been slow. Even nowadays, most advanced research in inverse problems is still concentrating on simpler models such as the acoustic wave equation or the Helmholtz equation; in the previous chapters we have generated an elastic wave propagation model for the forward problem; now we will develop it further to be used as part of practical FWI codes.

The general mathematical model for solving the nonlinear FWI problem is displayed in the following chart:

Algorithm 4: Full waveform algorithm methodology

```
Model setting;
Initial condition;
for Iteration i do
    Forward propagation (Using k-wave);
    Adjoint-state Backward propagation (Using k-wave with time-reversal);
    Compute gradient;
    Apply line search technique;
    Update model parameter;
    Check stopping criterion;
end
Accept final approximation;
```

Based on the above workflow, we will implement groups of numerical tests for pixel based and level set based full waveform inversion for our model.

4.2 Model set

We discretize the 2D domain of interest into a grid of size 30×120 pixels each having dimension $\Delta x \times \Delta y = 100\text{m} \times 100\text{m}$. With this the modelled domain has a length of 12km and depth of 3km, which serves to simulate a simplified seismic setup.

In this thesis, we will focus on specific situations where our domain contains one or several objects with a high contrast of parameter values to the background; a practical example for such case can be found in salt dome reconstruction. To simulate a more practical seismic environment, we design the parameters of the background, V_p, V_s, and ρ, to be increasing linearly with the depth; and for the embedded objects, such as salt domes, we assume that the parameters are constant inside. In our test cases, the choice of parameters is defined as follows:

	Objective	Background at top	Bottom
$V_p(m/s)$	5000	3000	4000
$V_s(m/s)$	2700	1200	2000
$\rho(kg/m^3)$	3000	2100	2500

(58)

We design two separate test cases containing one or two objects within a given background; figure 22 shows the two test models.

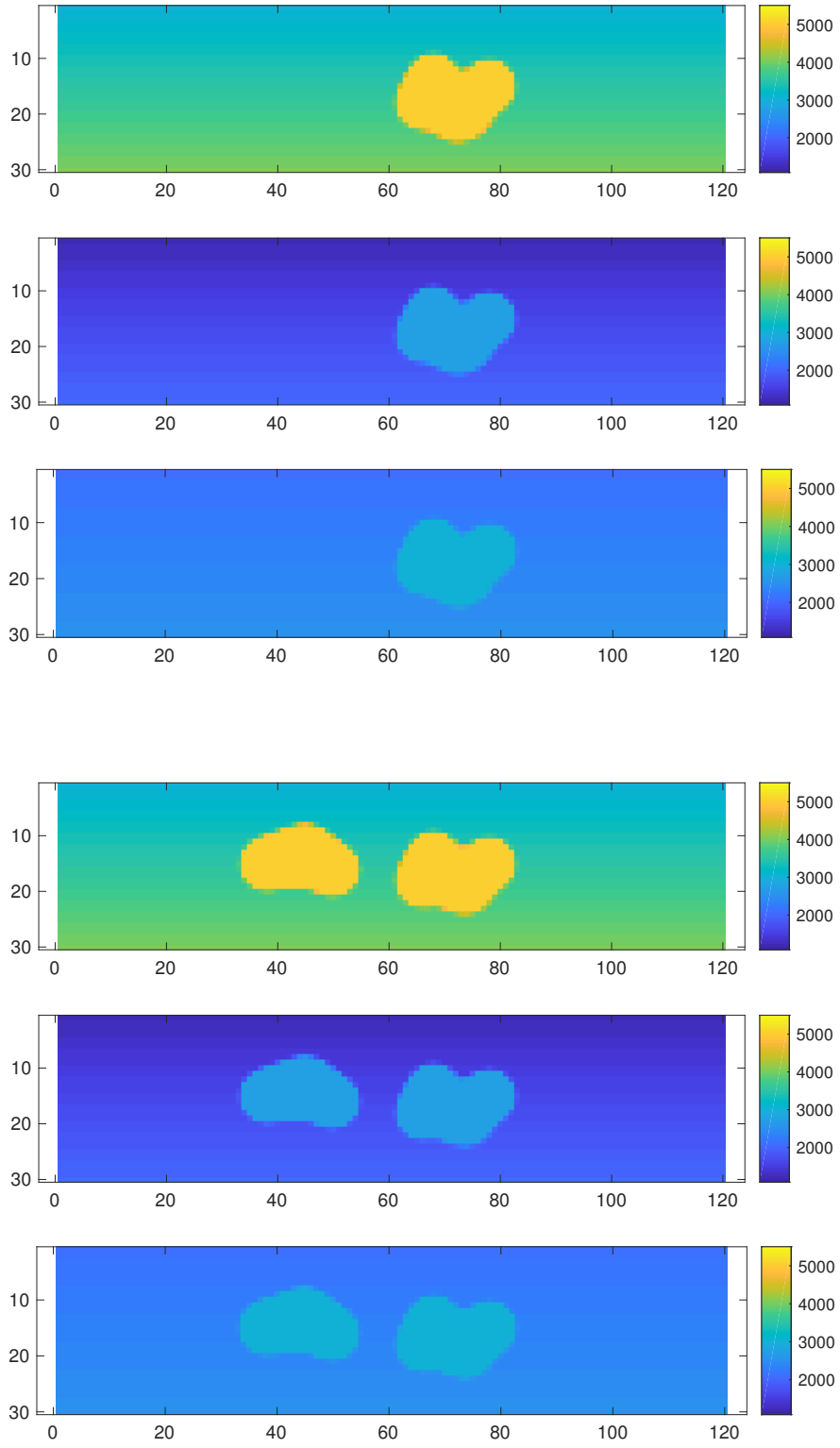


Figure 22: Up: Test model with one object. Down: Test model with two objects. From top to bottom subfigures denote the parameter V_p, V_s and ρ for both test models.

Given that Δx, Δy are defined as above, we define the time length Δt as to satisfy the CFL condition that $CFL = 0.3$ for the initial model; the time step is set automatically to satisfy that this condition is met. We set up this number in an appropriate way to make sure that even after several model updates the wave propagation still remains stable as the maximum of V_p updates.

4.3 Source and receiver

For testing our algorithms, we define two different sets of sources and receivers. In seismic imaging, practically source and receiver locations are limited to the surface positions and are in particular distributed on just one side of the domain. However, for verifying the general capabilities of a new algorithm, more general distributions of sources and receivers yield very useful insights as well. Therefore, we are actually considering two different setups of sources and receivers in this thesis.

In particular, we consider two groups of sources and receivers: Group 1 defines both sources and receivers are near the top of our survey (neglecting PML), on the same side of the object domain; in Group 2 we select the sources at the top, but the receivers are near the bottom, on the opposite side of the object. This case is only set to compare the general performance of our algorithms using only reflected or both, reflected and transmitted, waves.

To start with, we select 81 receivers, which continuously form a line in the survey near the surface, as to simulate a practical case in seismic imaging.

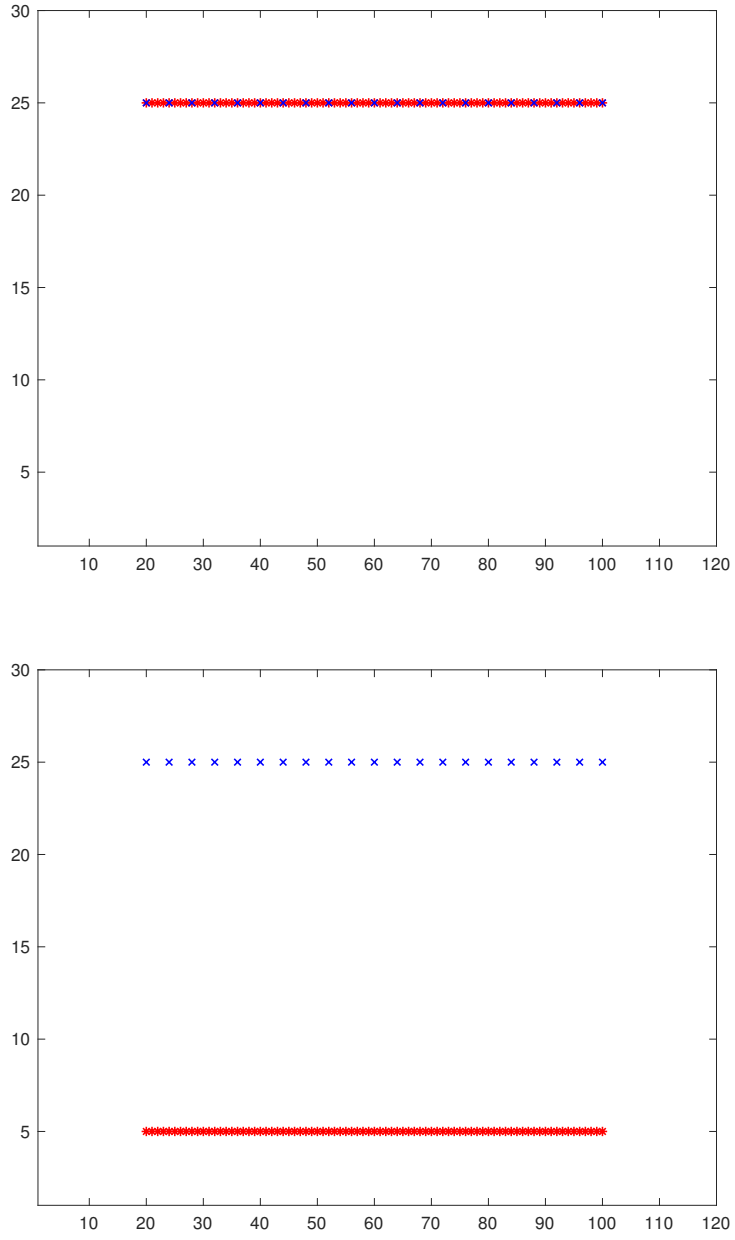


Figure 23: Up: Source and receivers both at the top. Down: Source and receivers separately located at top and bottom. The blue cross denotes the source location, and the red dot denotes the receiver acquisition.

We apply the Ricker wave formulation in equation (57), and define as central frequency $f = 1.5$, and $t_d = 0$, $k = 2$ as displayed in the following graph:

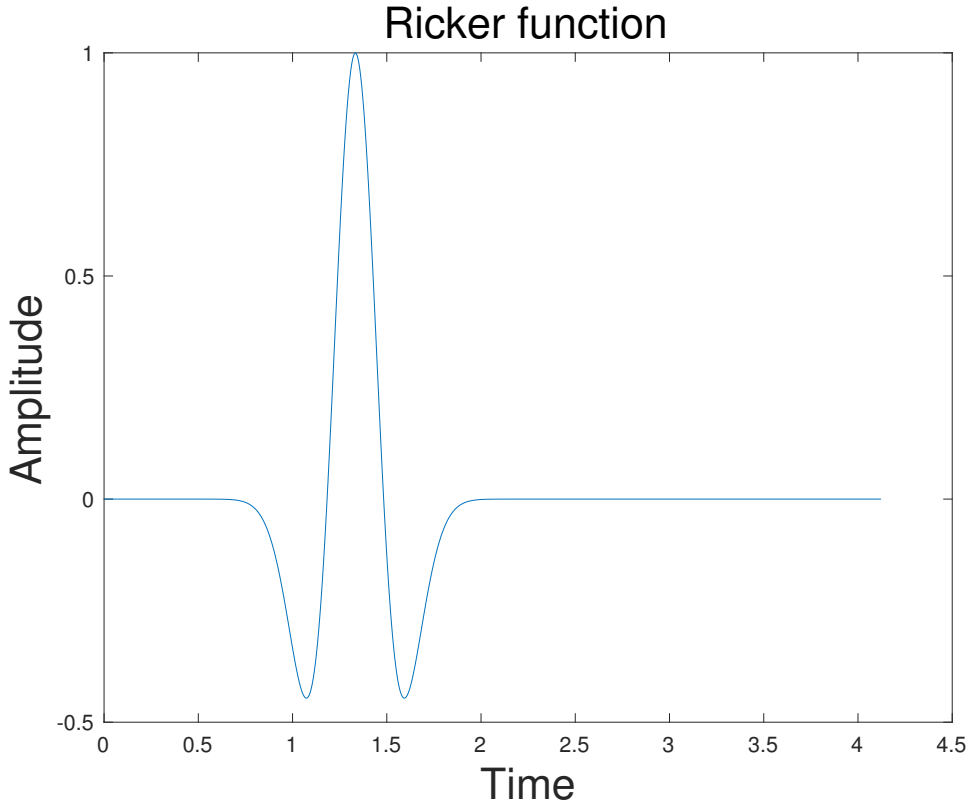


Figure 24: Corresponding Ricker wave

Notice that this Ricker wave central frequency also satisfies the numerical dispersion condition number with $n = 8$.

4.4 Numerical results

We implement the pixel based elastic FWI code following the algorithm in 4.1; for the steepest descent method, we construct the numerical model using Algorithm 1. We set up the line search step size to be $\alpha = 5 \times 10^{11}$, and fix the maximum iteration to be 100; the stopping criterion is that the relative error is smaller or equal to 5×10^{-4}. We also add in a 5% noise to the data to avoid the inverse crime. For both cases, we set up the initial model as the correct background without any object embedded in the domain.

4.4.1 Test results

We test the above 2 models by using 2 different source and sensor set-ups, and analyse the final results accordingly; the following experiments apply the pixel-based steepest descent method for all test cases.

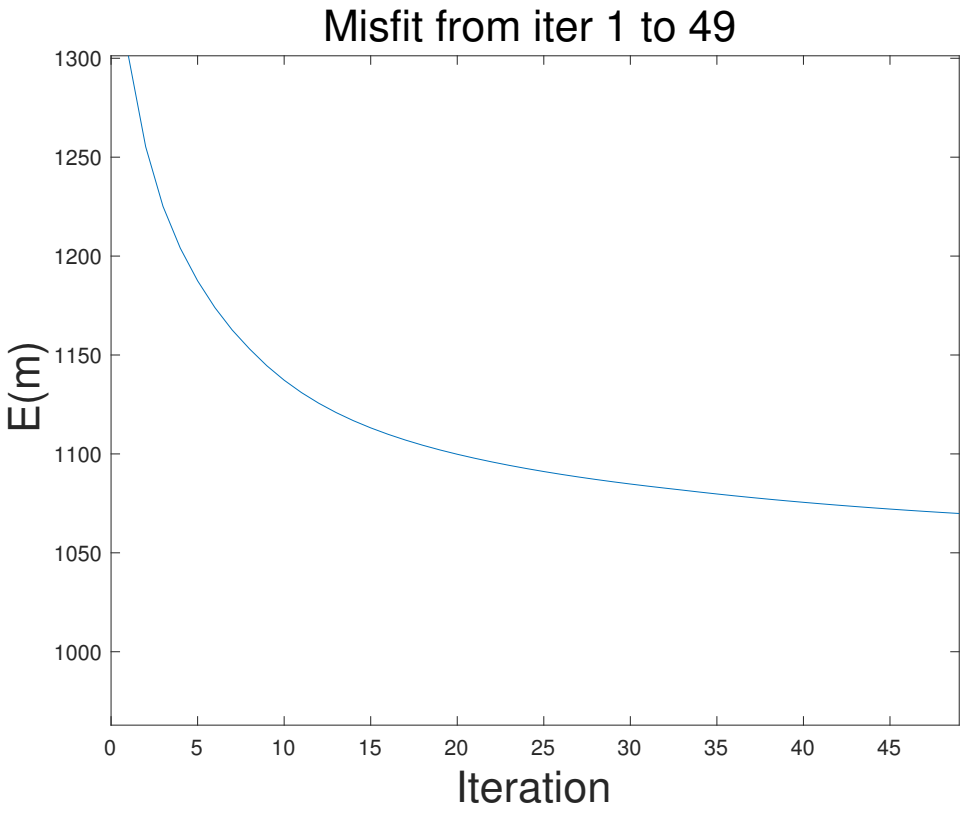


Figure 26: Data misfit; source and sensors located at the top; reached stopping criterion at iteration 49

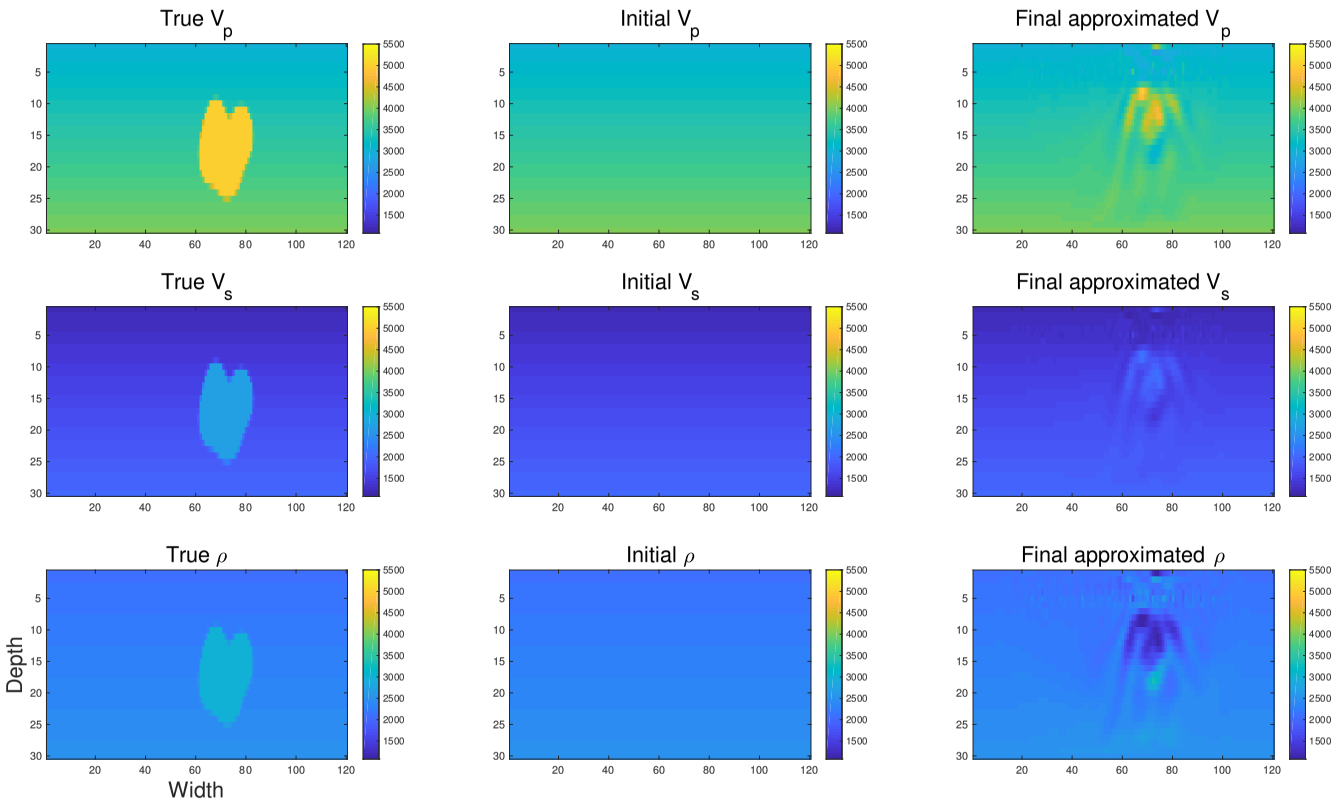


Figure 25: Final result for model 1: the subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

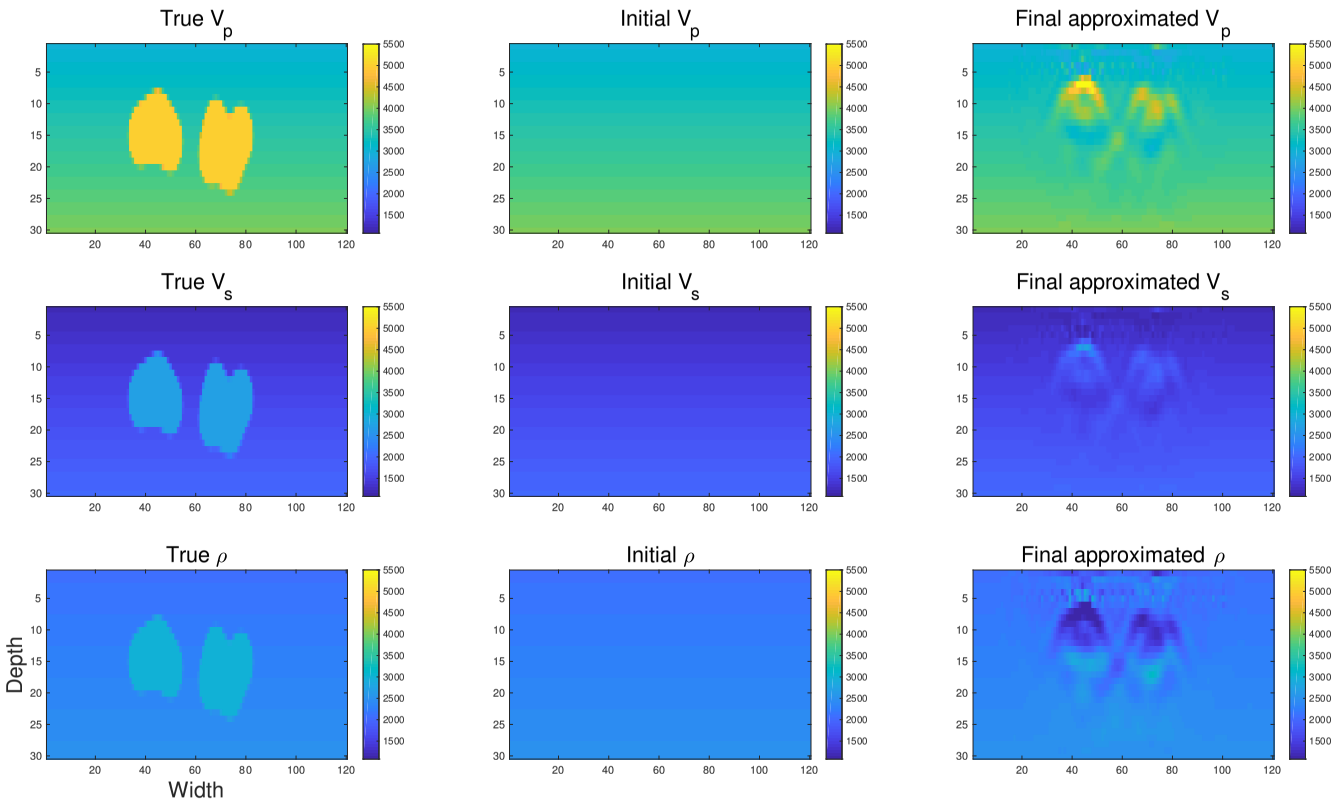


Figure 27: Final result for model 2: the subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

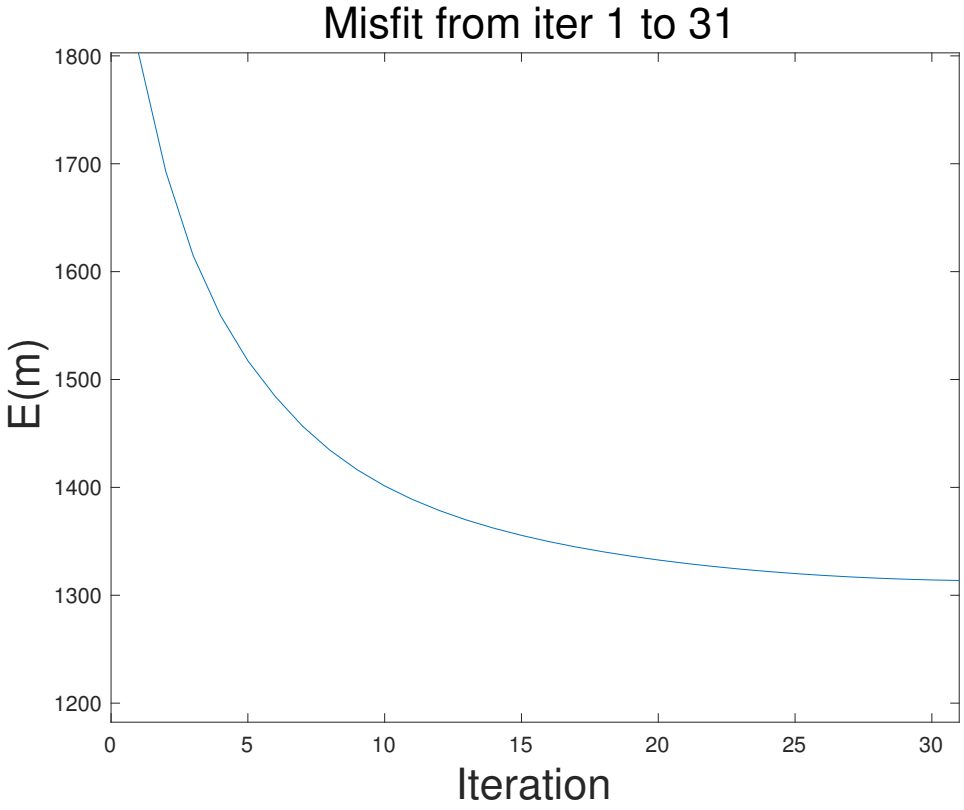


Figure 28: Data misfit; source located at the top, sensors at the top; reached stopping criterion at iteration 31

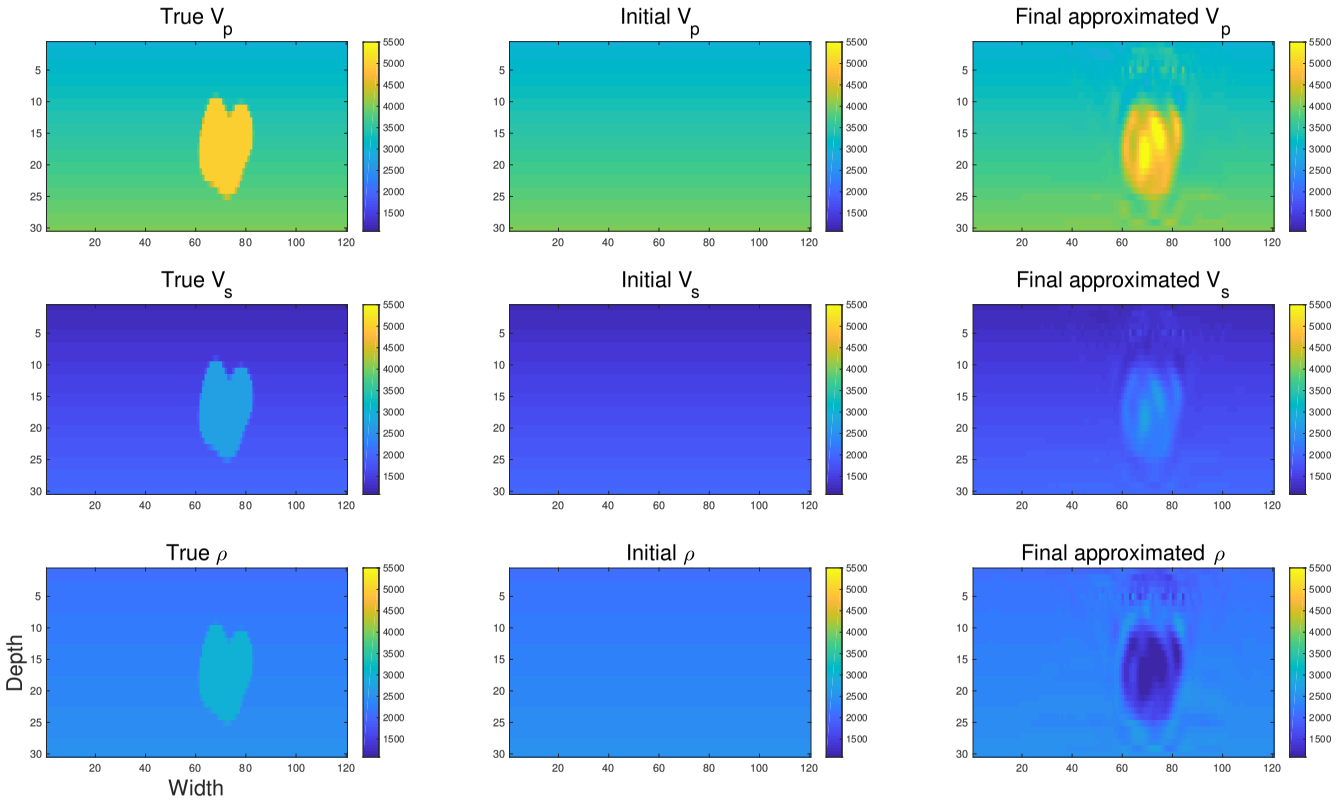


Figure 29: Final result for model 2: the subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

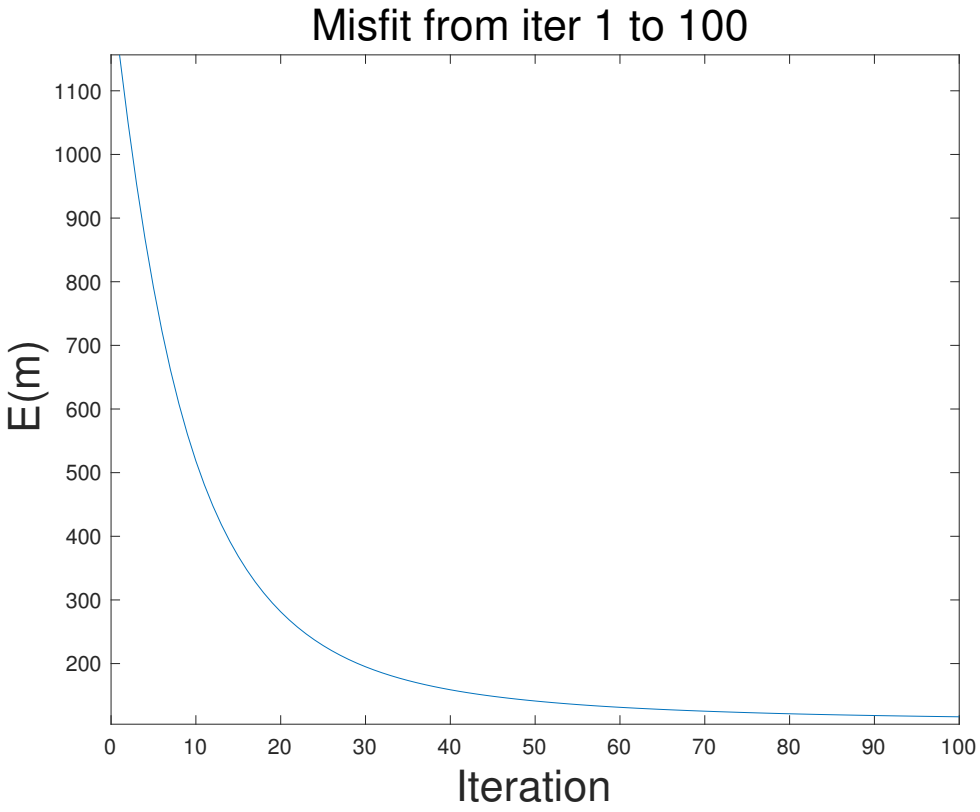


Figure 30: Data misfit; source and sensors located at the top

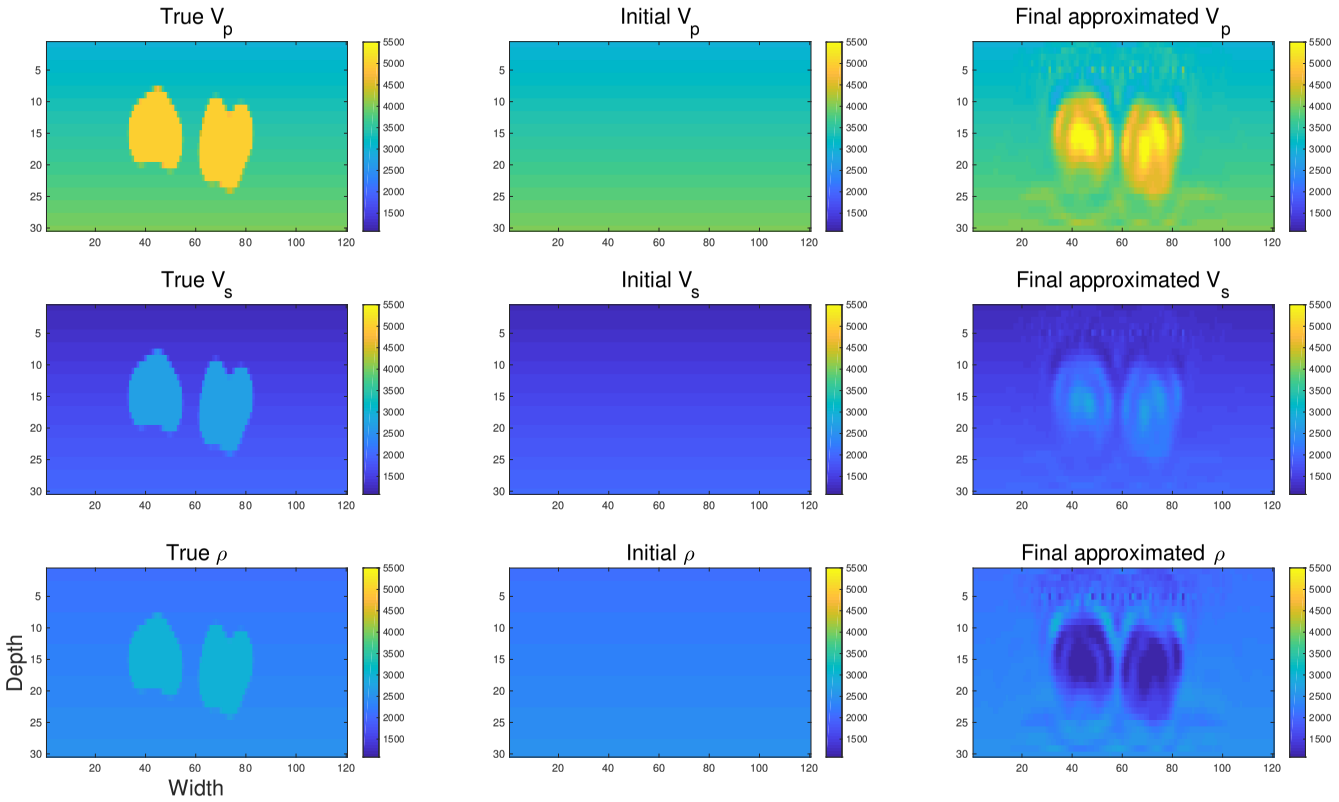


Figure 31: Final results for model 2: the subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

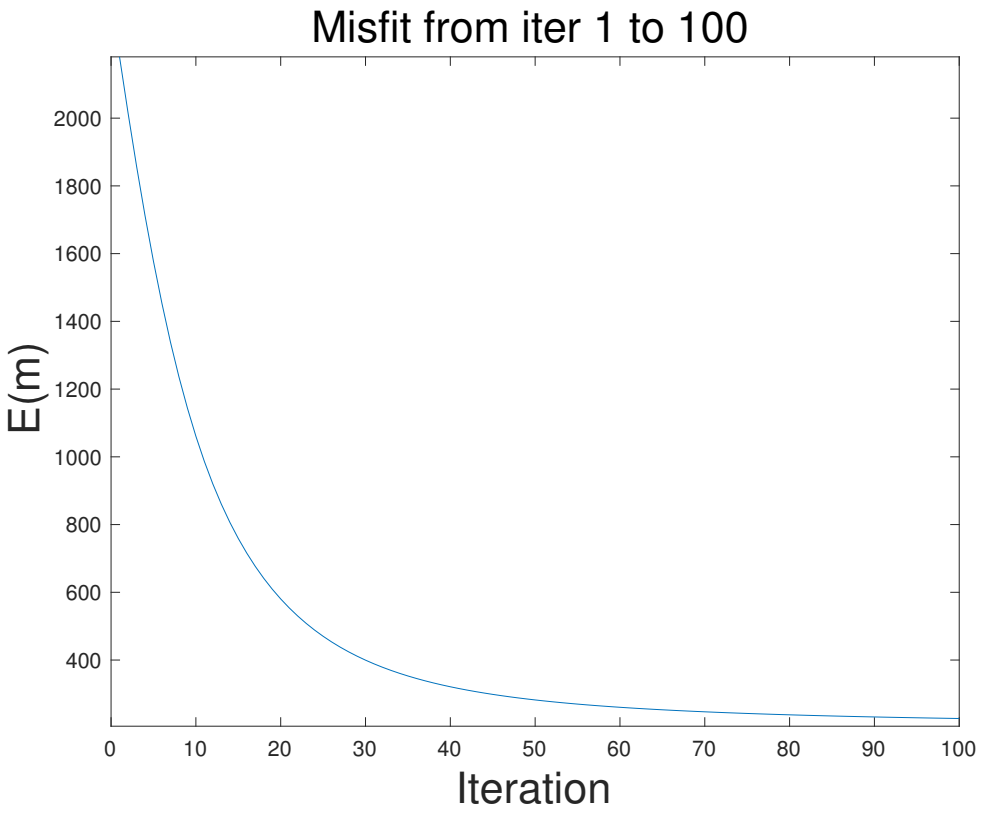


Figure 32: Data misfit; source located at the top, sensors at the bottom

Our numerical results show that the sensor locations do have a great impact on the numerical reconstructions; for tests in figures 25, 27, the sensor locations are at the top of the survey; the final results show a good estimation of the near-surface boundary of the object, but they fail to provide the correct information in deeper earth regions; it could be deduced that the model has reached a local minimum. For figures 29, 31, the shapes of the embedded objects seem generally close to the true model, although there still remains some blurs near the object boundaries. The final misfit seems to be sufficiently reduced.

However, in geophysical applications it is practically difficult to locate the sensors in the opposite direction of object domain (salt domes) (ignoring novel developments of horizontal wells); this means that we could only observe the seismic waves showing the reflected wave information close to the surface. As expected, in our test cases this proves to yield inferior reconstructions compared to those incorporating transmitted wave information. Also, there seems to be other problems with the final approximation related to the cross-talk of different

physical parameters; we will discuss that aspect in more details in Section 4.4.4.

4.4.2 Applying Sobolev gradient

Using the Sobolev gradient in the field of partial differential equations is considered to be a smoothing technique for numerical refinement; we also apply this technique in our inverse model to see how it affects the final solution, and compare the results with those obtained by using the normal L^2 gradient. For simplicity, we only compare the results with figures 29, 31.

A challenge when applying the Sobolev gradient is the choice of the weight parameter γ; a bad choice of this weight, either too small or too large, will either diminish the smoothing effect, or it overshadows the actual information of the data in the model updates. In our experiments, we choose two cases where $\gamma = 1$ and $\gamma = 10$;

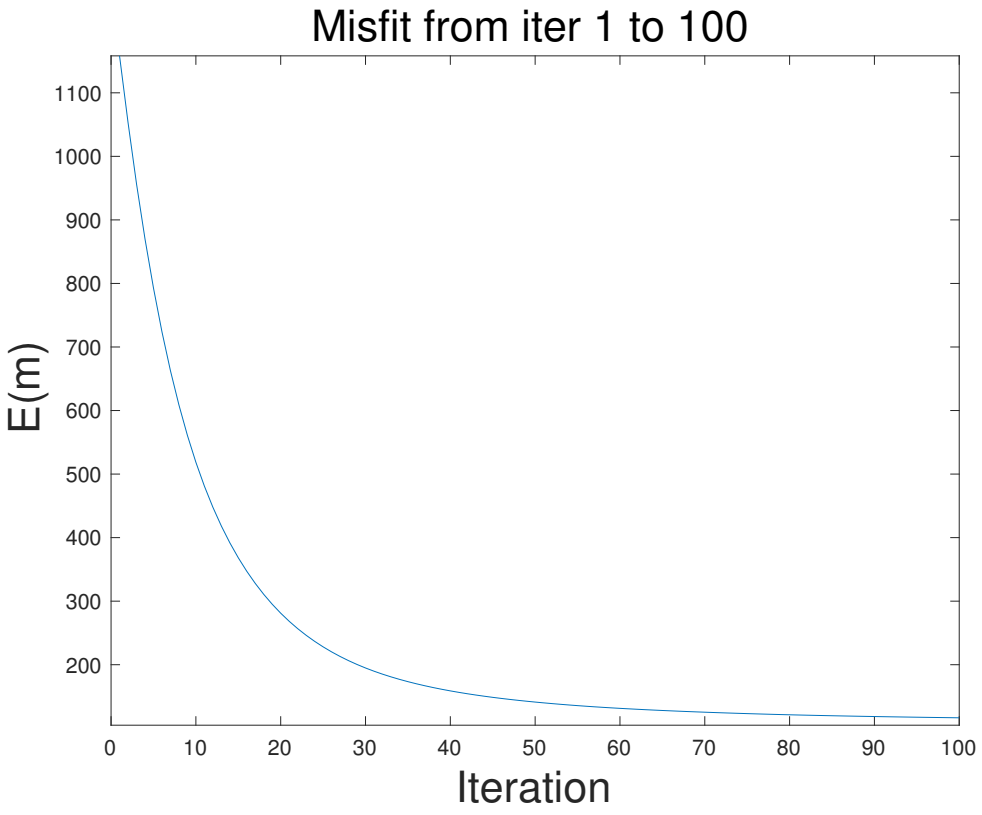


Figure 34: Misfit for model 1 test result using Sobolev gradient, with parameter $\gamma = 1$

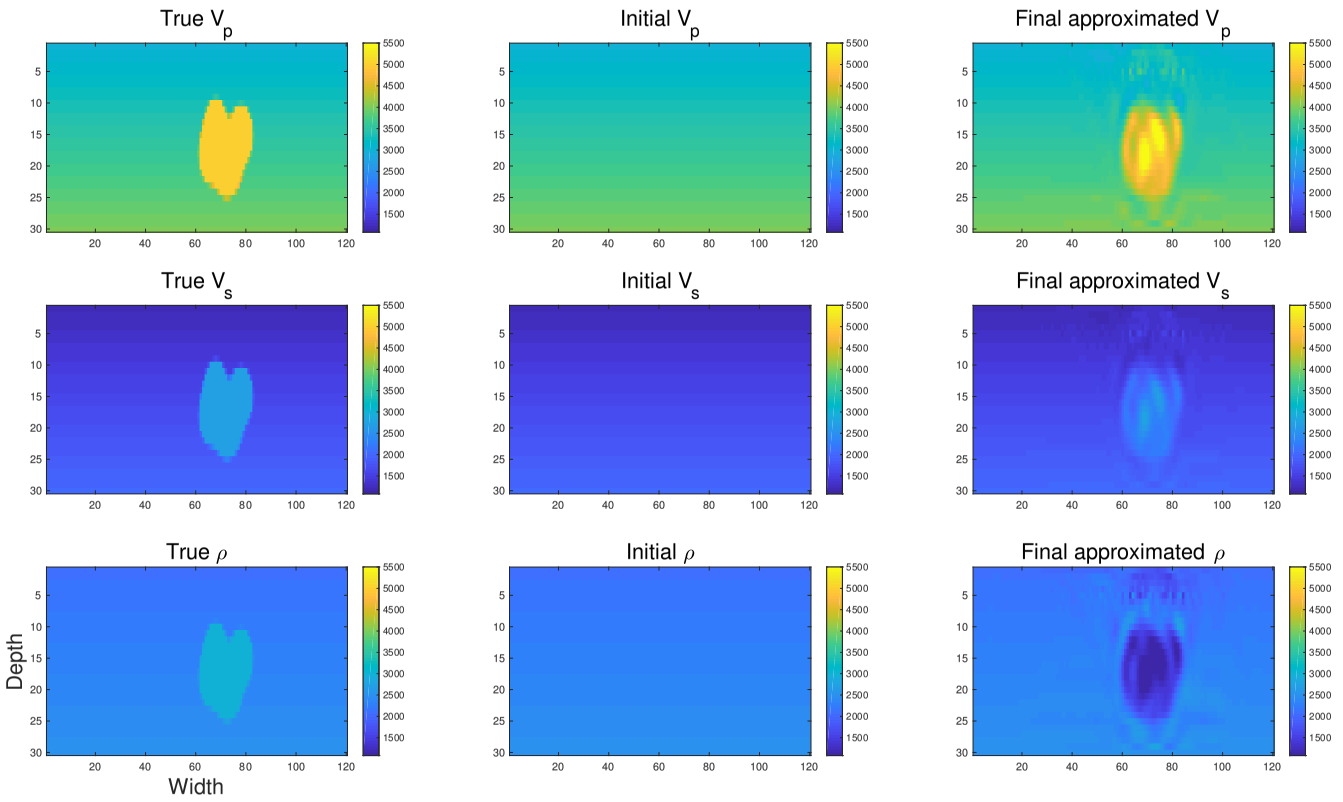


Figure 33: Test result on model 1: Using Sobolev gradient with a weight parameter $\gamma = 1$. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they refer to V_p, V_s and ρ

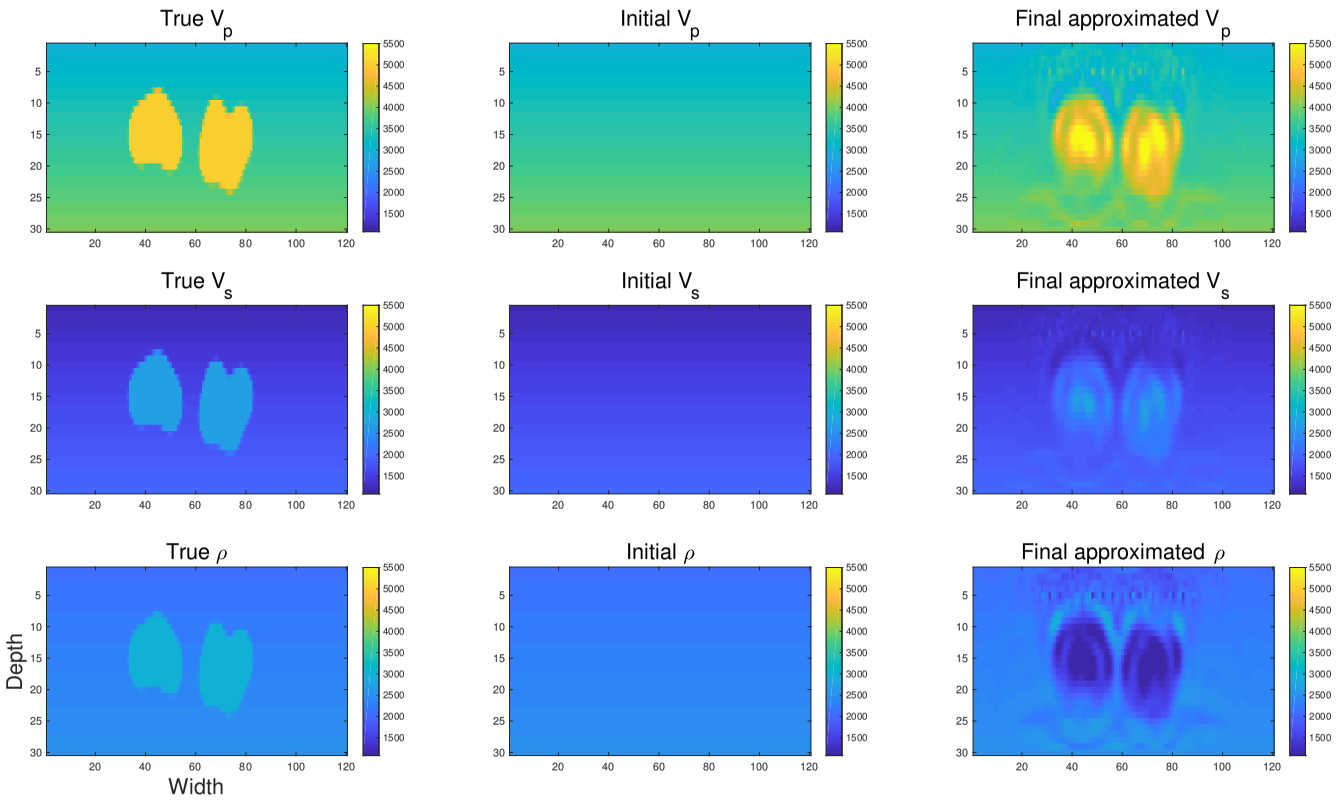


Figure 35: Test result on model 2: Using Sobolev gradient with weight parameter $\gamma = 1$. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they refer to V_p, V_s and ρ

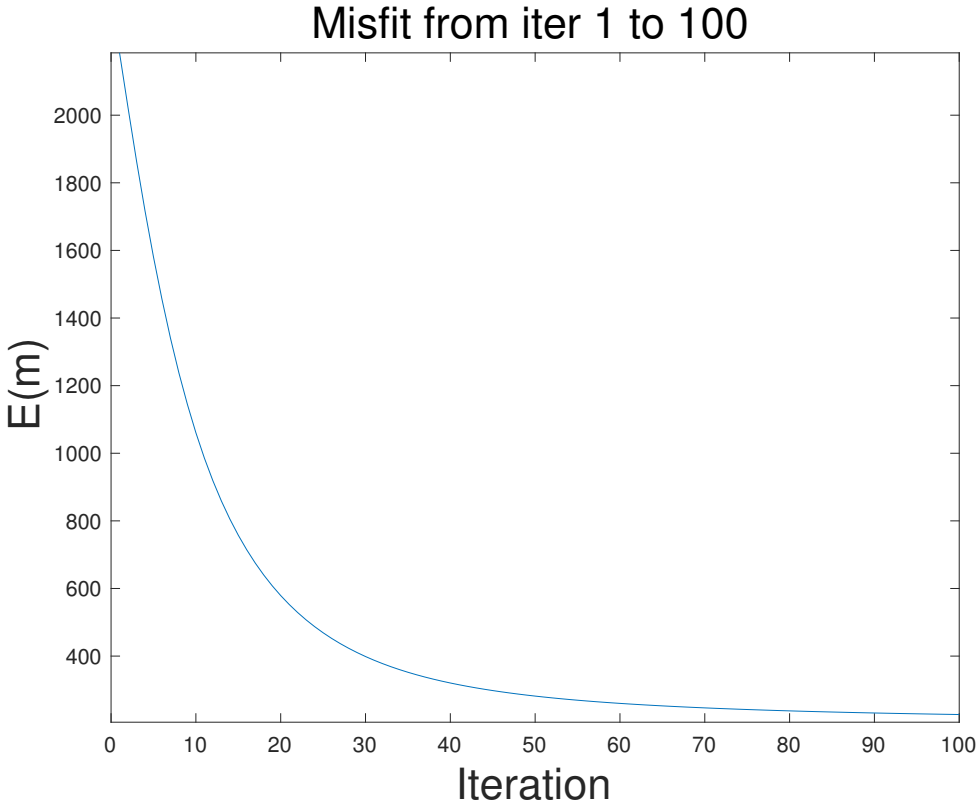


Figure 36: Misfit for model 2 test result, using Sobolev gradient, $\gamma = 1$

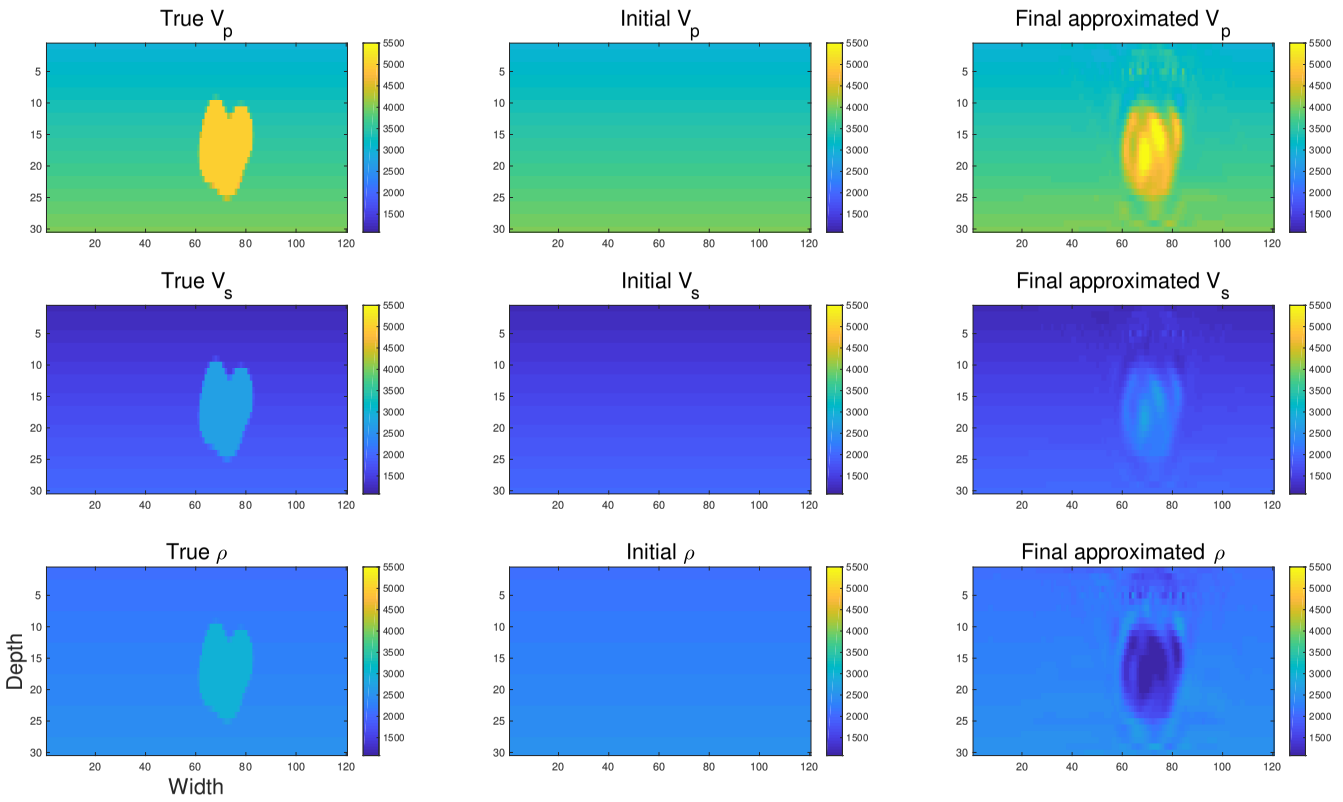


Figure 37: Test result on model 1: Using Sobolev gradient with weight parameter $\gamma = 10$. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they refer to V_p, V_s and ρ

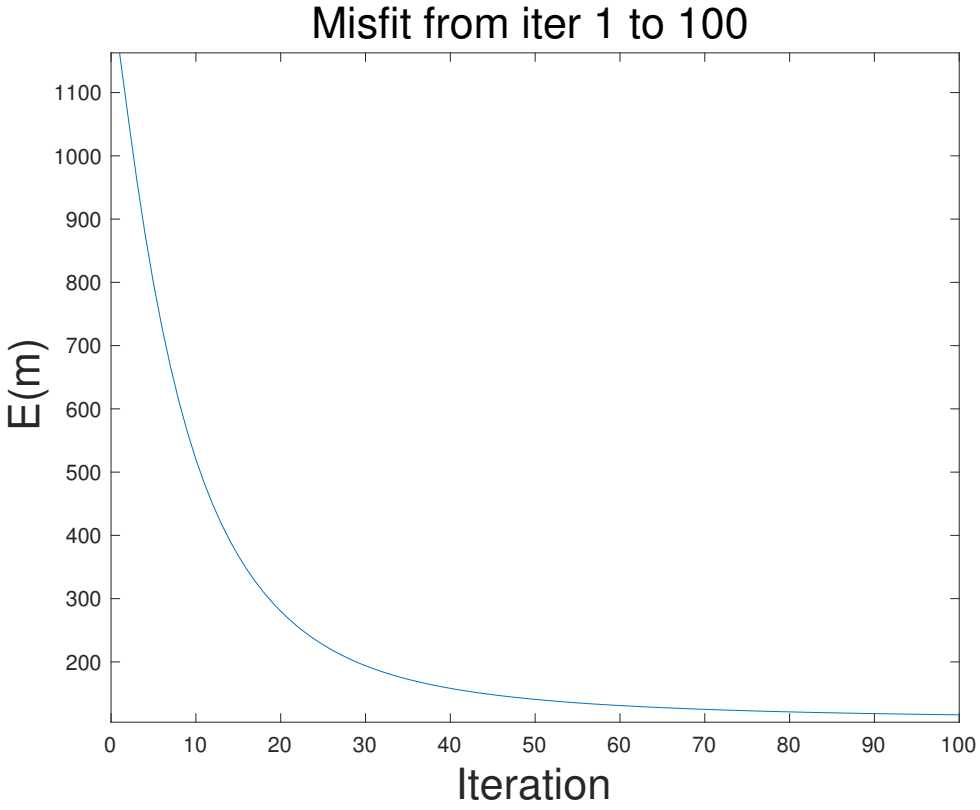


Figure 38: Misfit for model 1 test result, using Sobolev gradient, $\gamma = 10$

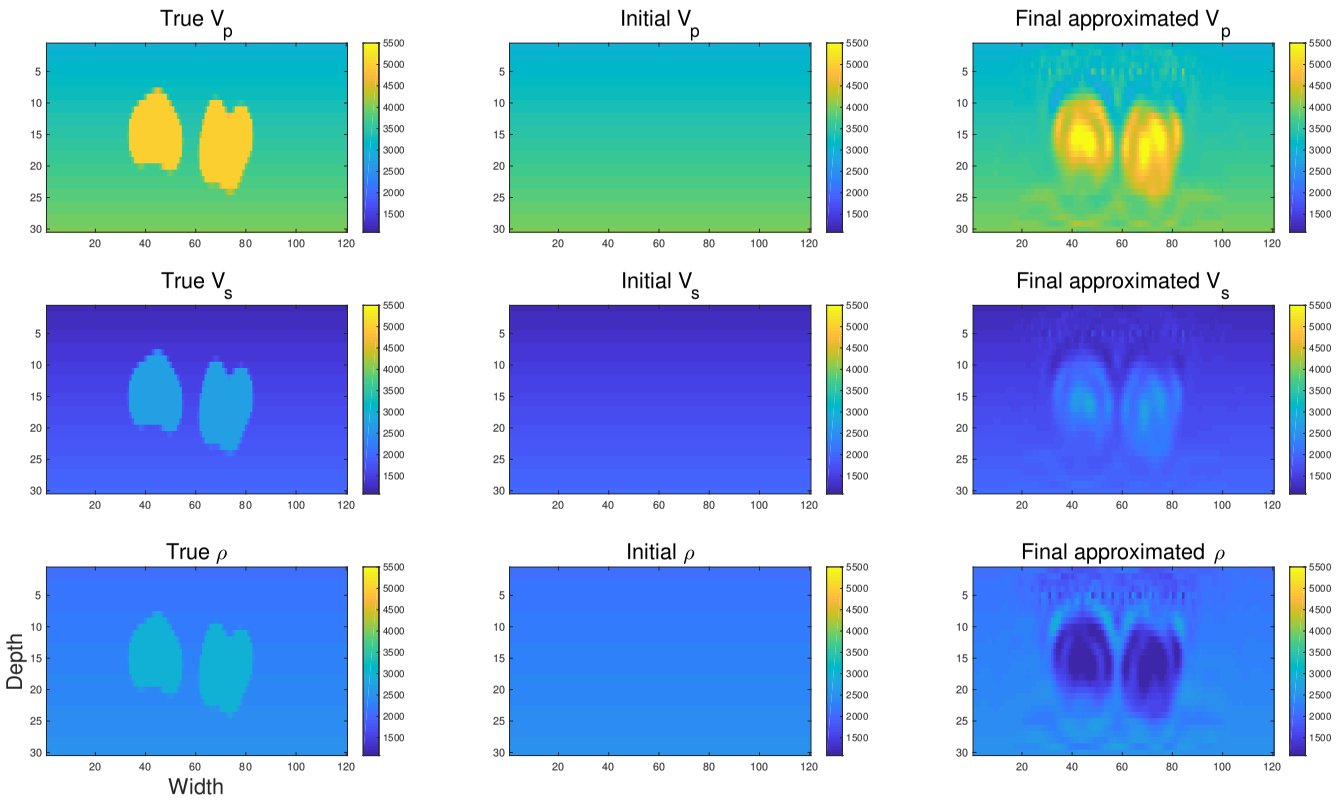


Figure 39: Test result on model 2: Using Sobolev gradient with weight parameter $\gamma = 10$. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they refer to V_p, V_s and ρ

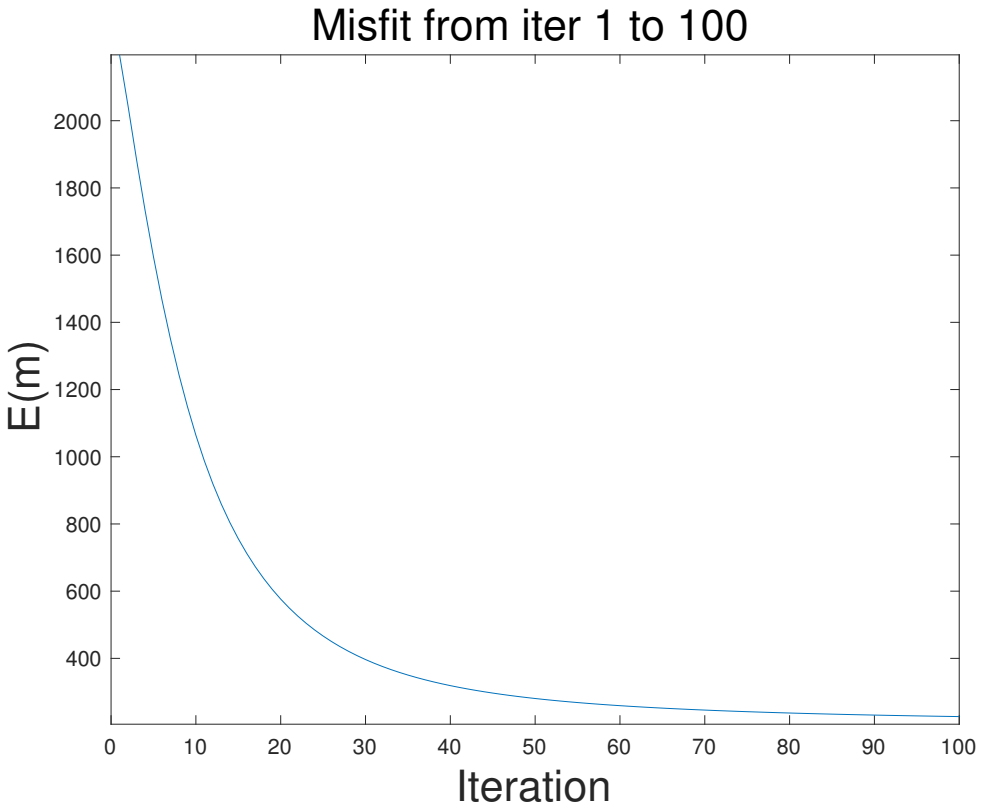


Figure 40: Misfit for model 2 test result, using Sobolev gradient, $\gamma = 10$

A deeper comparison of V_p and the misfit is demonstrated for two models, with Sobolev gradient and L^2 gradient:

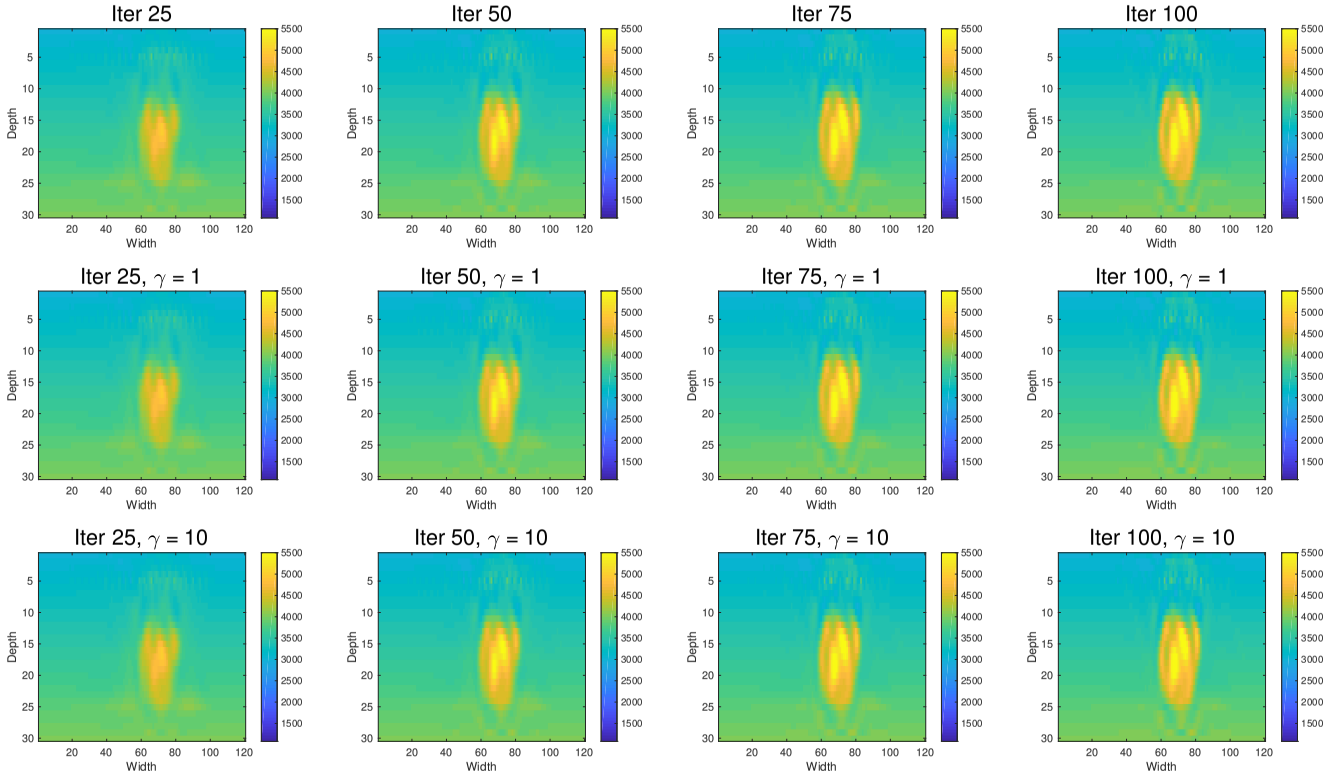


Figure 41: A comparison of V_p for model 1 tests with L^2 gradient, Sobolev gradient with $\gamma = 1$, Sobolev gradient with $\gamma = 10$. Top to bottom: results with L^2 gradient, $\gamma = 1$, $\gamma = 10$. Left to right: V_p at numerical iteration 25, 50, 75 and 100

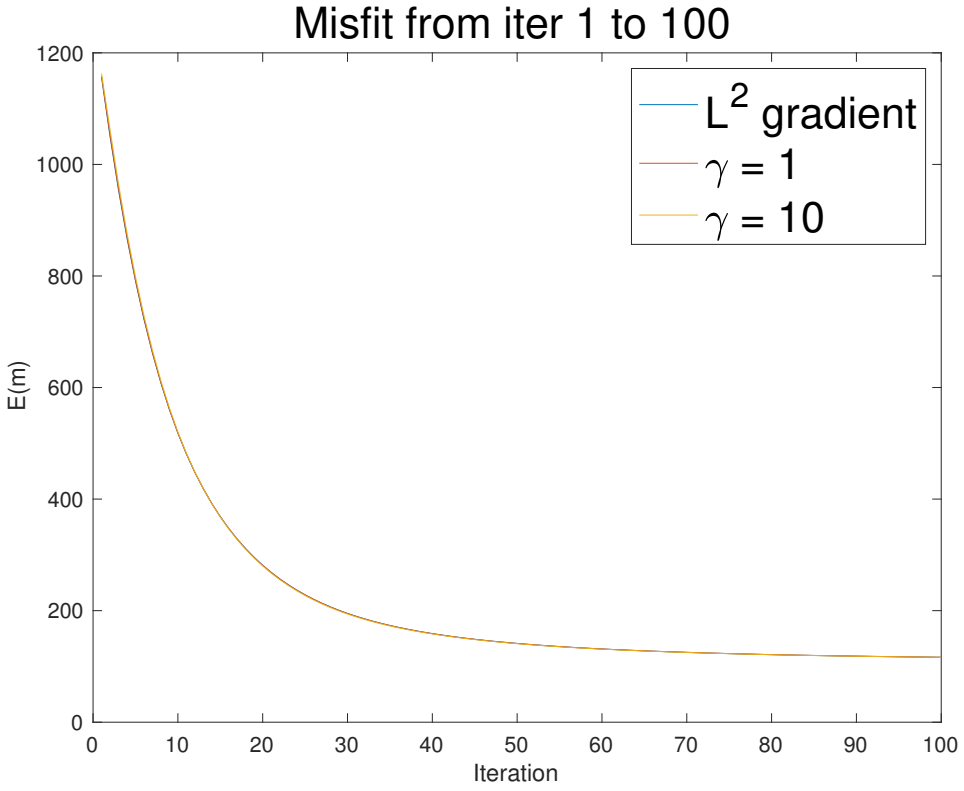


Figure 42: Data misfit for model 1 tests with L^2 gradient, Sobolev gradient with $\gamma = 1$, Sobolev gradient with $\gamma = 10$.

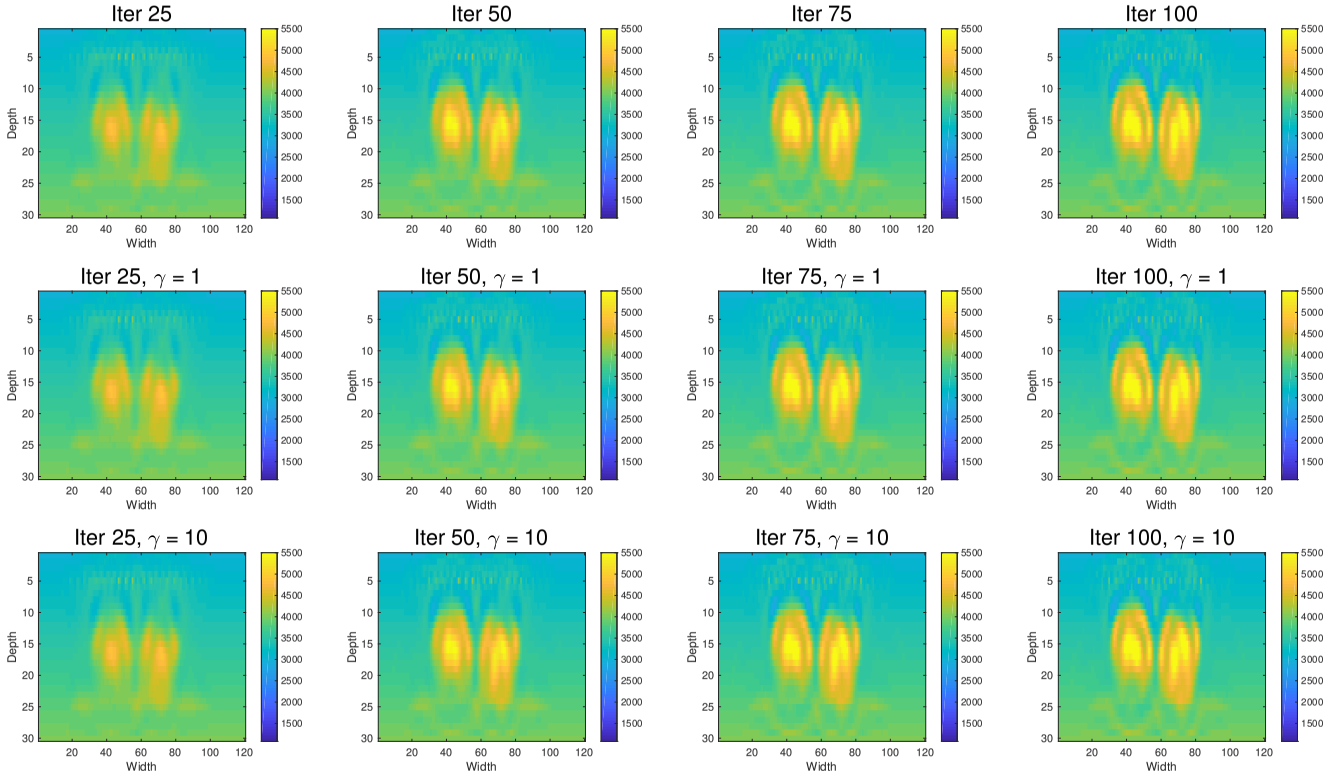


Figure 43: A comparison of V_p for model 2 tests with L^2 gradient, Sobolev gradient with $\gamma = 1$, Sobolev gradient with $\gamma = 10$. Top to bottom: results with L^2 gradient, $\gamma = 1$, $\gamma = 10$. Left to right: V_p at numerical iteration of 25, 50, 75 and 100

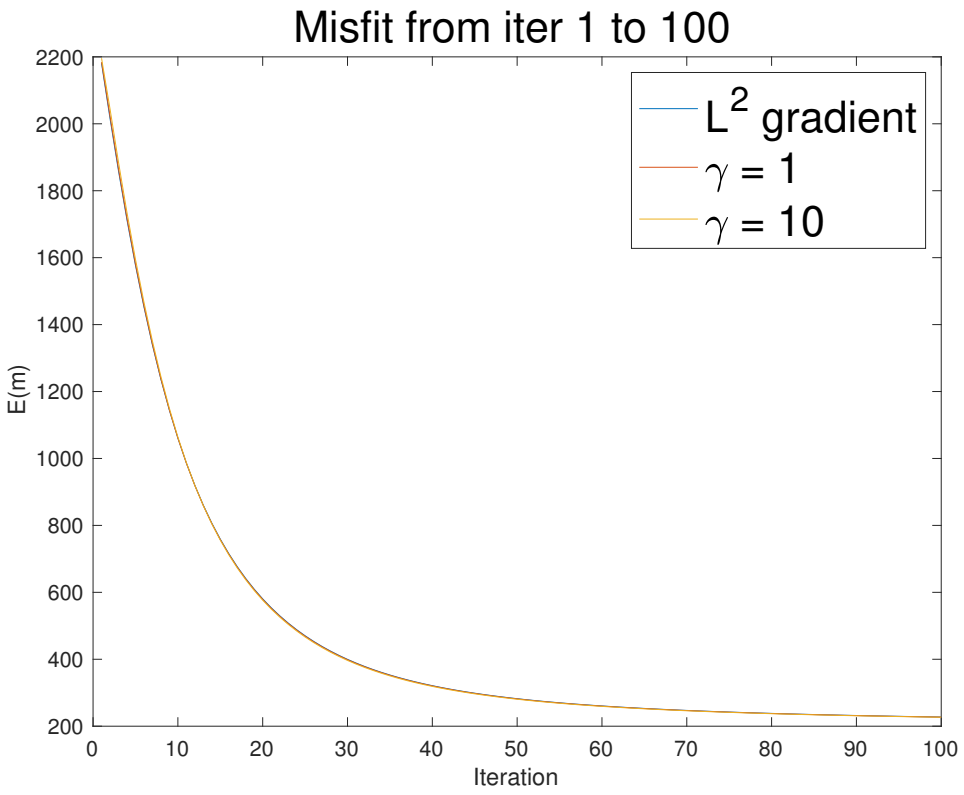


Figure 44: Data misfit for model 2 tests with L^2 gradient, Sobolev gradient with $\gamma = 1$, Sobolev gradient with $\gamma = 10$

The numerical results show that in pixel based FWI, the difference between applying Sobolev gradients and L^2 gradients is not very big. The key reason is that Sobolev gradients are constructed by using a form of Sobolev norm instead of L^2 norm. This tends to smoothen out the individual updates with increasing weight parameter. however in pixel based inversion, the model update is mostly sufficiently smooth already, such that the added smoothness of the gradients has little impact on the result.

4.4.3 A Conjugate gradient method comparison

In addition to the steepest descent method, we will also compare the numerical results on different line search approaches for other gradient based search directions. Theoretically, the conjugate gradient method is considered a much faster technique than steepest descent, especially in highly non-linear cases; but applying conjugate gradients carries also the risk of losing conjugacy at some

point. Our choice is to apply a reinitialization for conjugate gradients every 5 steps, and then re-initialize this process [47]; Our numerical implementation of the conjugate gradient line search scheme is based on Algorithm 2; we apply the Fletcher-Reeves form for scalar β.

In this example, we examine the difference of using the conjugate gradient and the steepest descent method. Both have been applied traditionally to FWI, but to our knowledge not in the form of the symmetric hyperbolic system as used here. We compare the results at different iterations for the same tests as figures 29, 31, and compare the compressional wave speeds obtained. We set the maximum iteration to be 100, and as line search step size we use $\alpha = 5 \times 10^{11}$ for both cases. In addition, the iteration stops at a relative error bound of 5×10^{-4}.

In the following test, we demonstrate the numerical results of a conjugate gradient method, compare V_p for in different line search techniques in different iteration steps, and compare the data misfit for both cases.

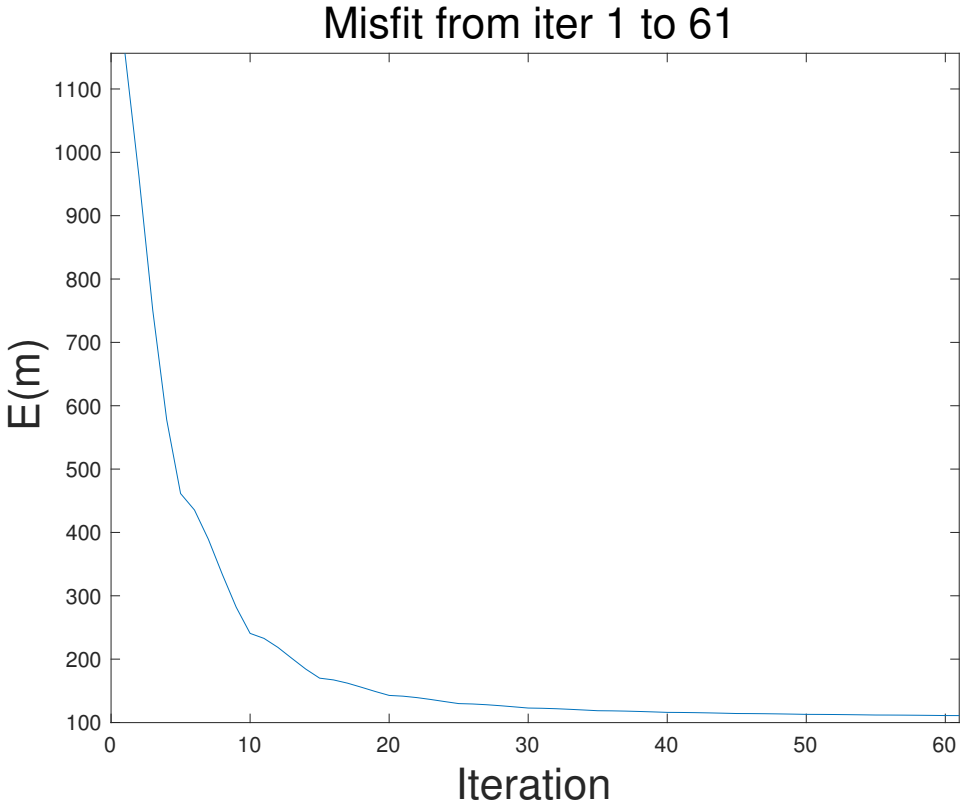


Figure 46: Numerical results for model 1 with conjugate gradient: Data misfit

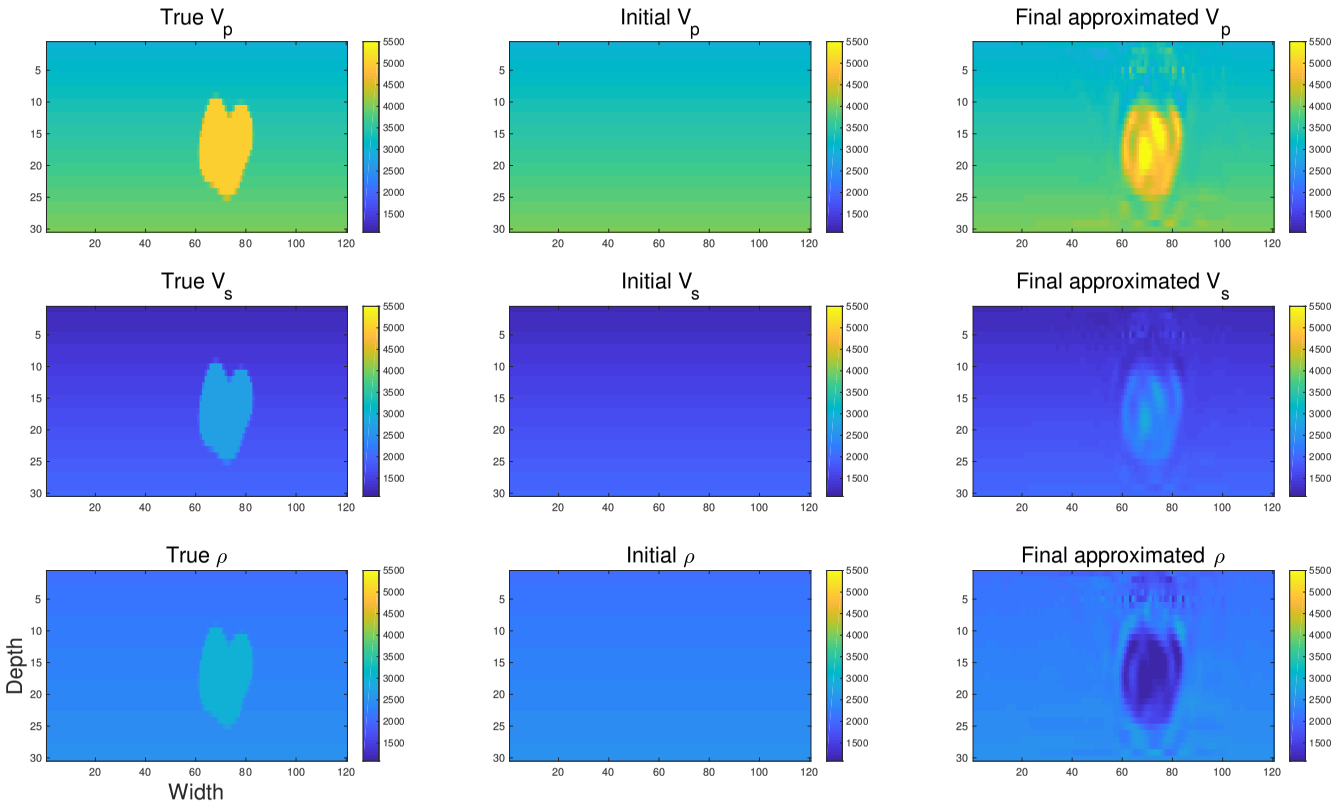


Figure 45: Numerical results for model 1 with conjugate gradients. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

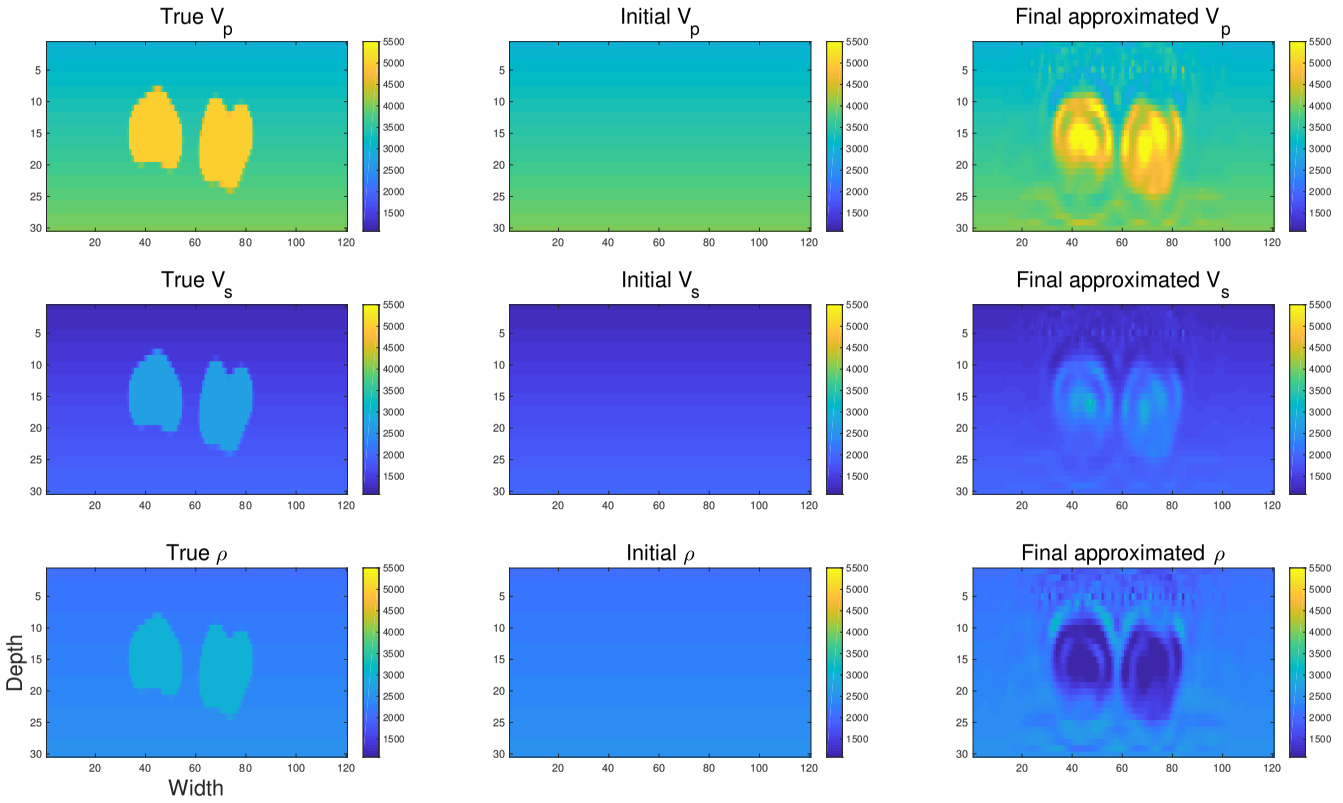


Figure 47: Numerical results for model 2 with conjugate gradients. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

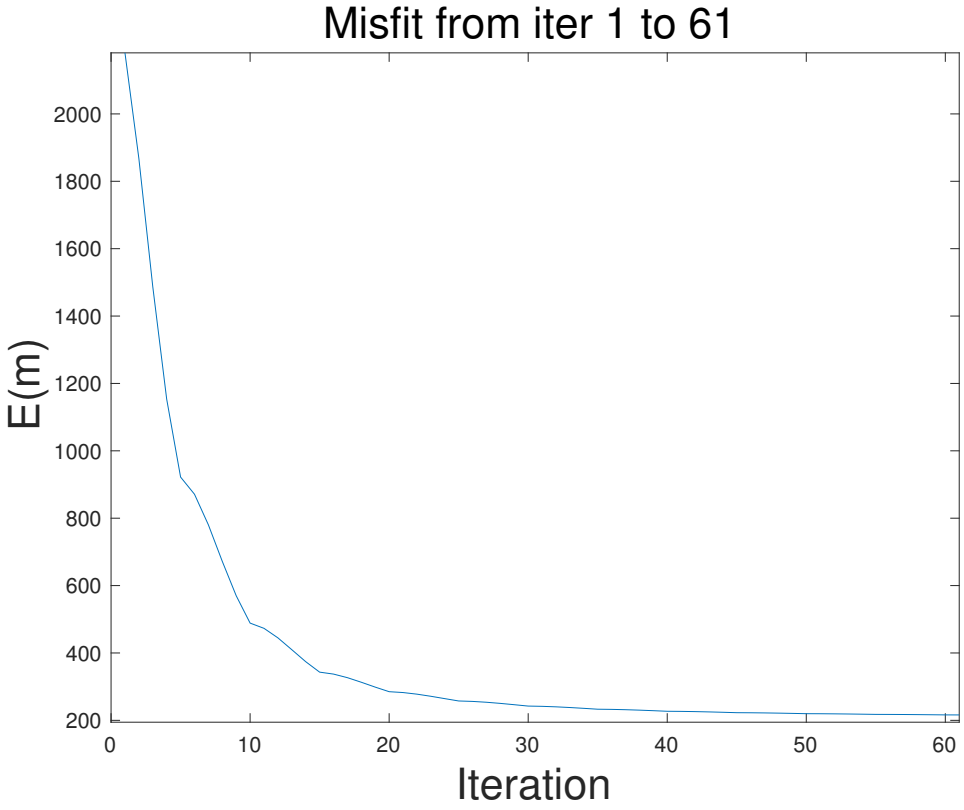


Figure 48: Numerical results for model 2 with conjugate gradient: Data misfit

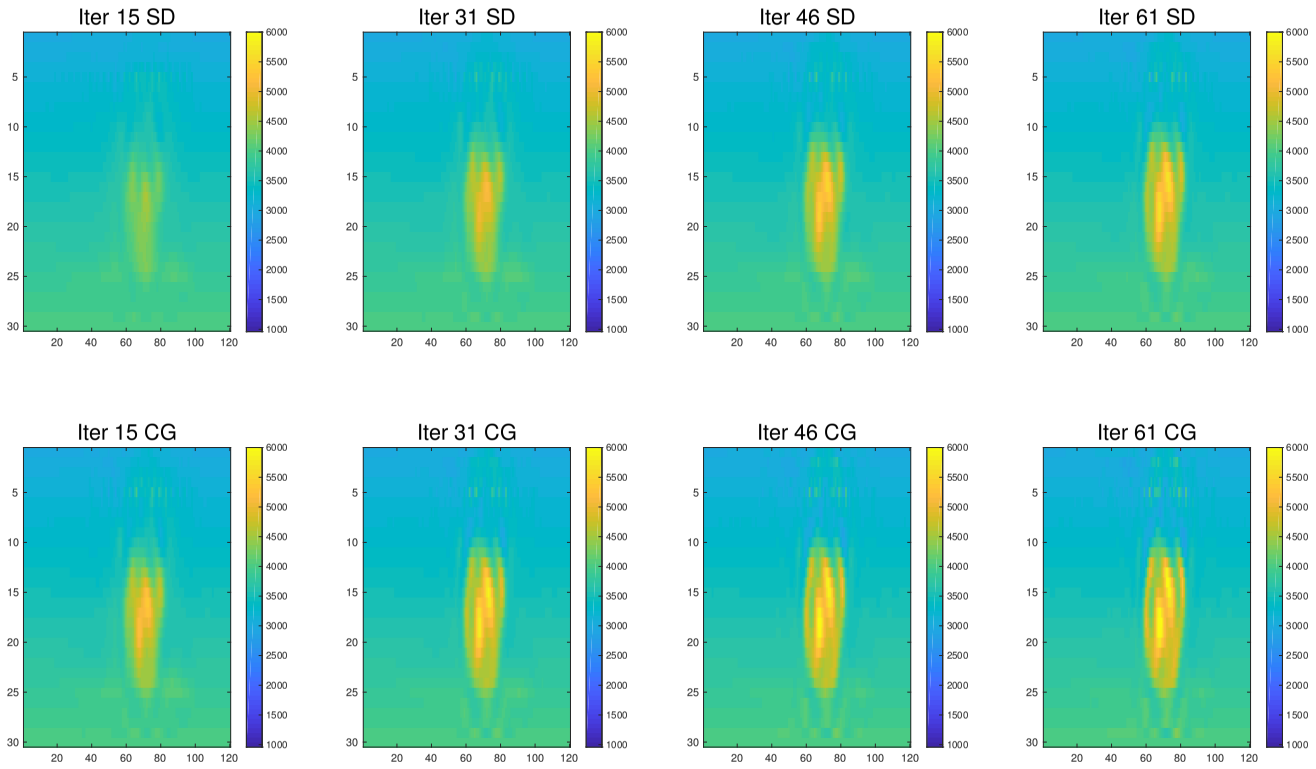


Figure 49: Line search comparison: Top figures show the value of V_p using steepest descent at iteration 13, 25, 38 and 50 from left to right; the bottom figures show the same result using conjugate gradients at the same steps.

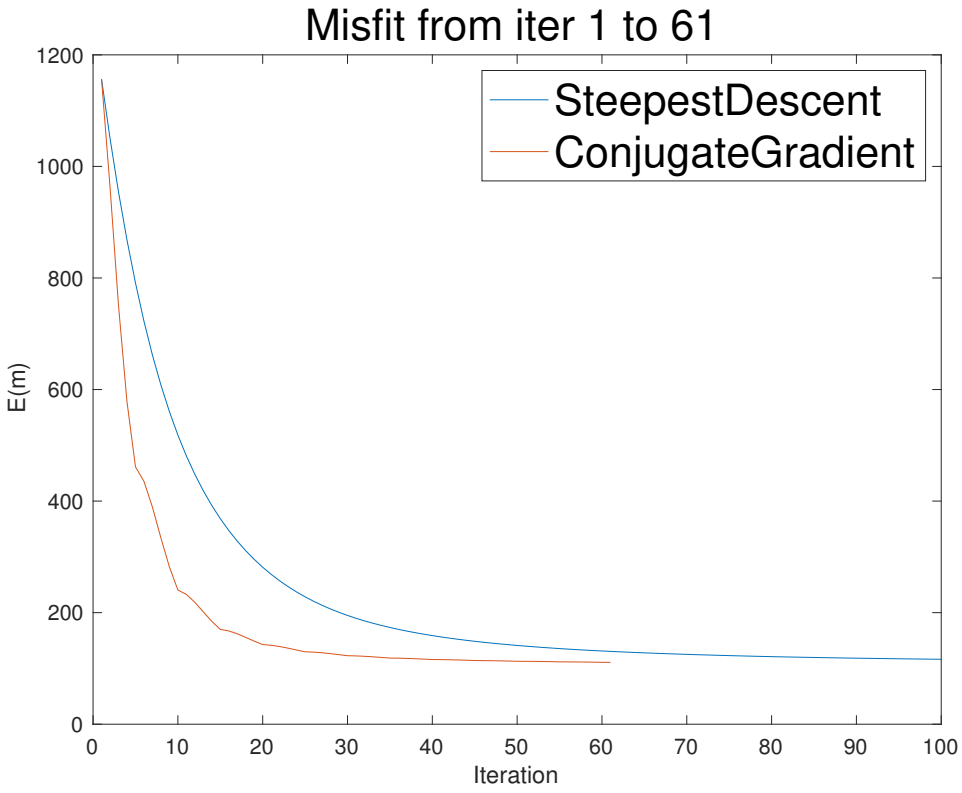


Figure 50: The data misfit using different line search techniques

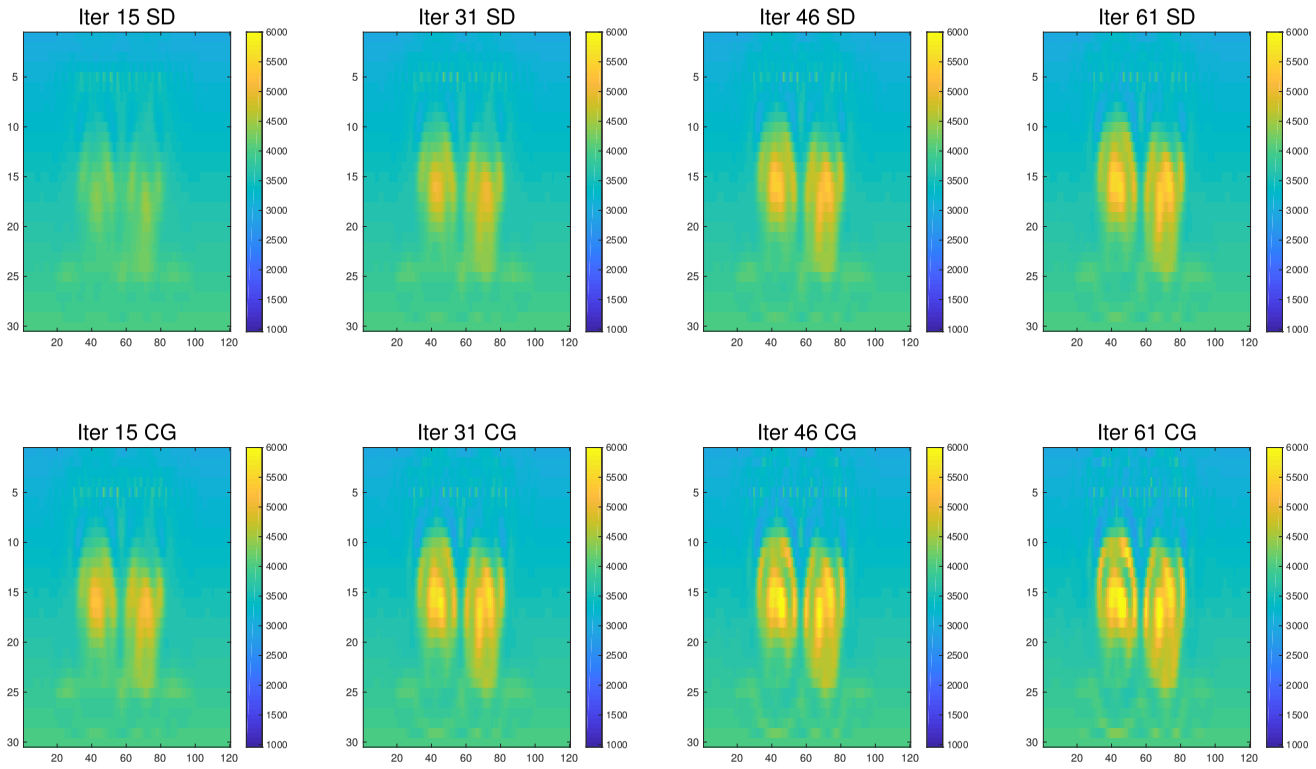


Figure 51: Line search comparison: Top figures show the value of V_p using steepest descent at iterations 13, 25, 38 and 50 from left to right; the bottom figures show the result using conjugate gradients at same steps.

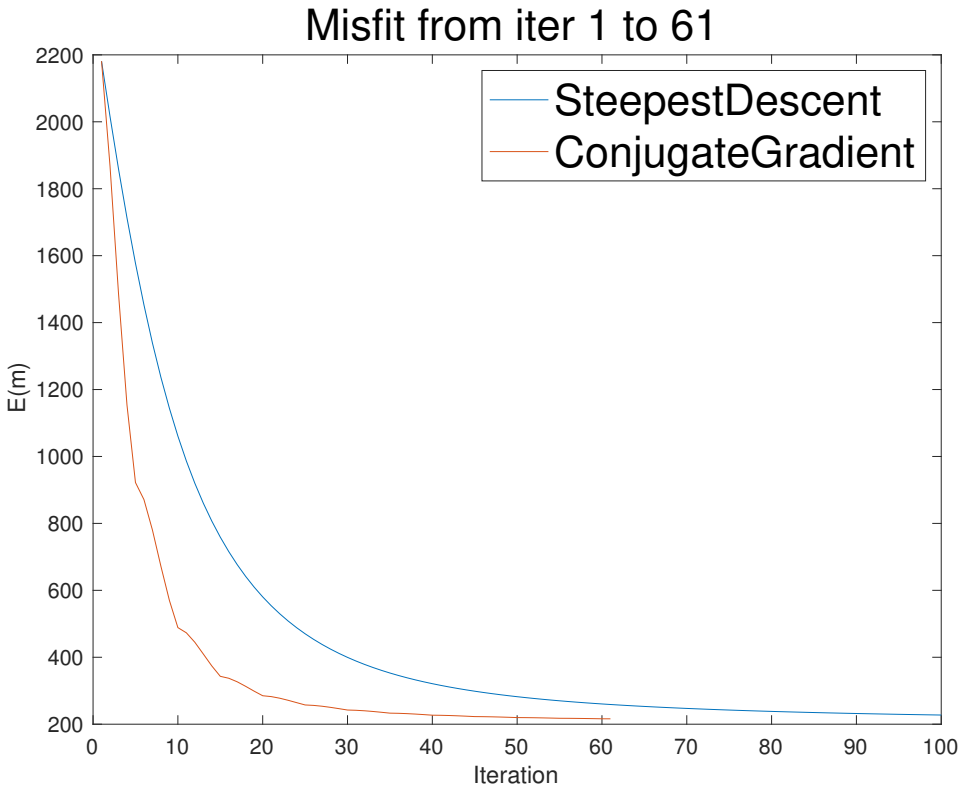


Figure 52: The data misfit using different line search techniques

Notice that the misfit for conjugate gradients decreases faster than steepest descent. Nevertheless, we will compare the values after 100 steps of each of the techniques, when both appear to become stationary. Our comparison demonstrates that for both test cases, using the conjugate gradient method shows a faster convergence than steepest descent method, when using the same step length in each iteration.

4.4.4 Cross-talk

From the previous results, we observed a general difficulty of multi-parameter reconstructio. When comparing all the numerical tests above in figures 29, 31 using different gradient based line search methods, we observed that the approximated internal values of V_p, V_s insded the detected objects seem close to the true model, but the value of ρ seems not to provide a reliable approximation for this value; We believe that this phenomenon is due to the phenomenon called 'cross-talk' [50] which tends to guide the multi-parameter reconstruction into lo-

cal minima where different parameters influence the reconstructions of each other. We demonstrate this by selecting in the following figures a cross section at depth $y = 20$ in the test of figures 29, 31, and monitor the parameter values of the final approximations.

In our numerical tests, we noticed that this 'cross-talk' happens in both test cases; and in both of our test model, it seems that the value of ρ is in particular updating in an incorrect way when combined with updates of the other parameters. When only updating ρ with correct and unchanged remaining parameters, on the other hand, updates for this parameter are far more reliable.

Is this a common phenomenon in all pixel based problems in elastic wave inversion? If so, then this might be one crucial limitation of pixel based methods applied in inverse problems using elastic wave equation when recovering all the parameters simultaneously; moreover, we also need to consider the possibility that these three parameters may not share the same topology. Some alternative methods will be introduced in the following chapter, in order to solve the problems above.

4.5 Conclusion

In this chapter, we introduced a numerical implementation for a traditional elastic FWI technique; we applied the k-wave method [71], [11] for elastic wave modelling for the forward and backward propagation, and made some modifications to the original toolbox module so that it is suited for being applied in the case of our symmetric hyperbolic scheme for forward and backward propagation. We designed an algorithm for addressing the pixel-based FWI problem in section 2, and implemented different sets of test cases for different source and sensor patterns and different line search techniques; we discovered some serious drawbacks when using traditional elastic wave inversion techniques.

Certainly, in practical seismic imaging, the acquisition of air-guns (sources) and geo-phones (receivers) is usually restricted to the surface of the earth; this means that the information we obtain comes only from so-called reflected waves, and the information contained in the large proportion of non-reflected waves has to be neglected. This results in a biased representation of sub-earth information in the data, and affects the accuracy of parameter estimation.

In addition, we have to deal with the phenomenon of 'cross-talk', where the

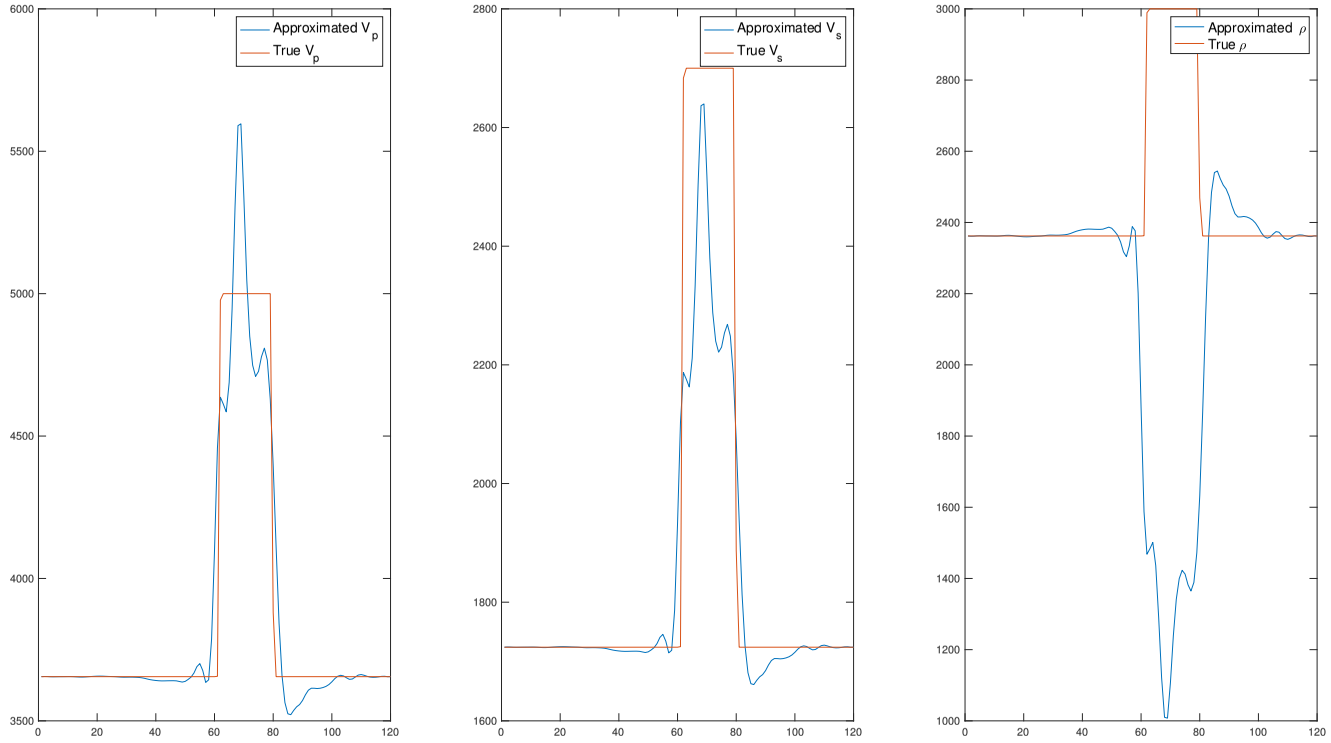


Figure 53: Left to right: cross section of V_p, V_s, ρ at $y = 20$ for test model 1, with sources and sensors at opposite sides

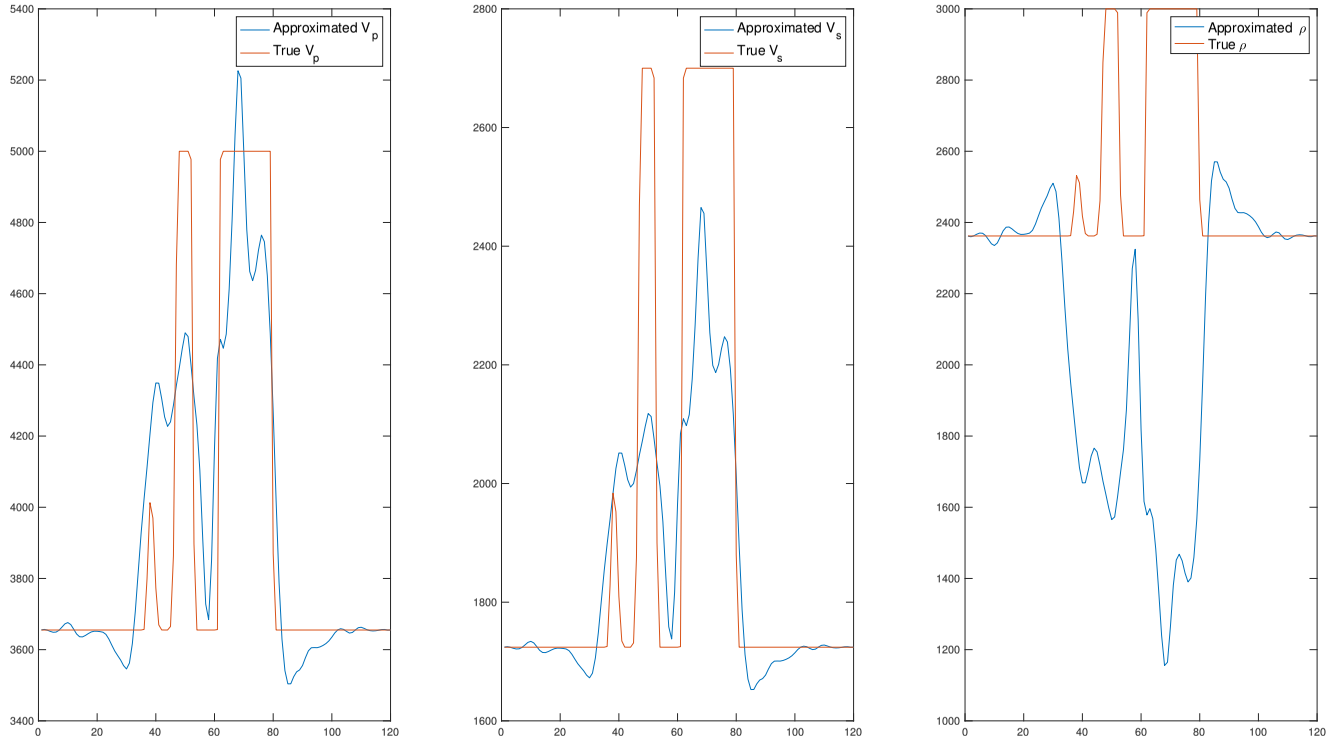


Figure 54: Left to right: cross section of V_p, V_s, ρ at $y = 20$ for test model 2, with sources and sensors at opposite sides

reconstruction of some parameters is highly influenced by updates of the other parameters which yields artificial local minima. This might or might not affect the shapes of the reconstructed objects, but it does definitely affect the estimation of the internal parameter values even in a shape-based inversion approach where all three parameters are assumed to have highly correlated shapes.

In order to exploit further the advantages of shape-based inversion, the next chapter will introduce an alternative method applying the so-called level set technique; we will use this technique to simulate the same seismic reconstruction problem as addressed so far by using the traditional pixel based FWI reconstruction schemes.

5 A level set parameter representation for full waveform inversion

The level set method is a numerical technique applied to the numerical modeling of the propagation of surfaces and shapes; it has been developed in 1988, by S. Osher and J. Sethian [52]; initially, this method has been applied mainly as an alternative for describing topological changes for curves and shapes [51], [62]; traditional methods for the evolution of shapes and interfaces concentrated on the parametrization of edges, curves, or domains; however, when topological changes happen, for example when one object is split into two, those methods have difficulties modeling such changes. The advantage of using level sets is that such changes of topolgies can be computed easily. Therefore level set techniques have been widely used in various fields of interface propagation, as an alternative to other classical techniques of front propagation [61]. In particular when addressing inverse problems where the topology of the unknown objects is unknown a priori, this is of great advantage.

In particular, in our application of FWI the level set method can be applied to solve the problem of estimating shapes of high-contrast objects embedded in more homogeneous backgrounds, such as in salt dome reconstruction [31], [36]. In these cases, pixel based inversion methods have major problems. Firstly, they do not seem to do well due to the high contrast between the salt domes and the background environment, which makes it difficult to obtain a clear shape of the estimated salt dome. This is especially so when the salt domes are scattered into subbodies of irregular shapes. Secondly, in the elastic wave inversion with the need to estimate multiple parameters simultaneously, we have to expect a 'cross-talk' as outlined in Chapter 4. It means that during the iterations updates for some variables will influence the updates of the others at the same or nearby locations. This phenomenon often leads to local minima, due to the complexity of elastic wave model.

The greatest advantage for level set inversion in such applications can be drawn from the possibility of incorporating additional geological prior information on shape structure in the inversion process. For example, it is reasonable to assume that the interfaces of different physical parameters due to salt bodies share a well-defined common boundary towards the background, namely the in-

terface between salt body and background. Therefore, the complex problem of parameter reconstruction can be split into one of finding the correct shape of the salt domes (which applies to all parameters simultaneously), and then of identifying the internal values of the different physical parameters which often even can be considered being constant. Thereby, the dimensionality of the inverse problem is considerably reduced.

In this chapter, we will first introduce the basic idea of level set representations of shapes, the level set evolution and its application in FWI; we develop a parameter estimation approach for such a level set representation in FWI. Next we apply some numerical implementations of this method to the test cases already considered in previous chapters. Special techniques for level set evolution, such as narrow-band, re-initialization and Sobolev regularization, will be developed as well, together with efficient line search techniques. We will then present results of the numerical test cases for level set based FWI with and without incorporating the Sobolev regularization scheme.

5.1 A level set based full waveform inversion

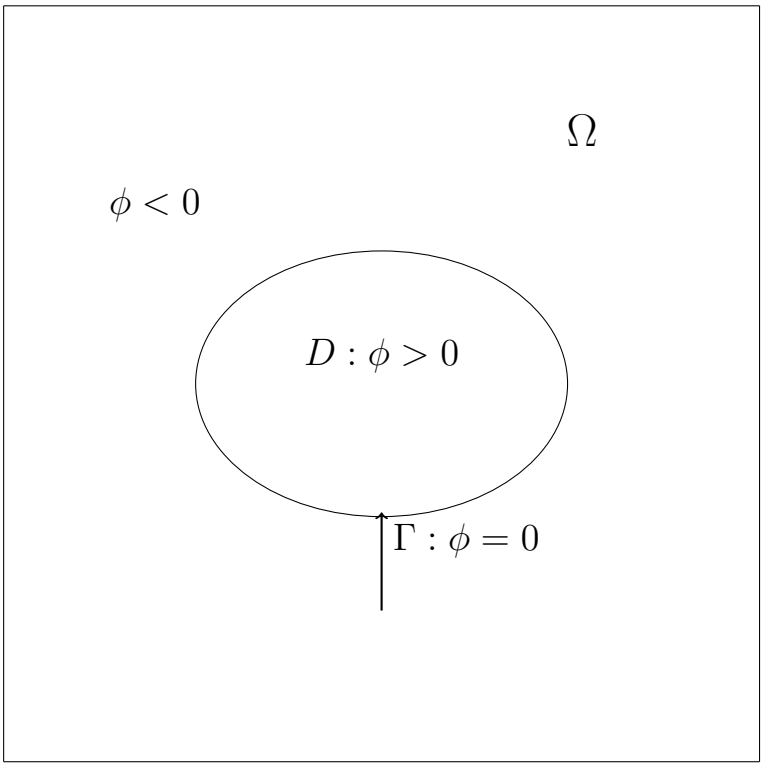
5.1.1 Introduction to level set

We apply a shape based approach for estimating embedded objects in the seismic background from FWI data. As a start, we introduce a level set function for shape representation [52].

Let us consider a given domain $D \subseteq \Omega$, with its boundary $\Gamma = \partial D$; then we define a level set function $\phi(\boldsymbol{x})$, such that

$$\begin{aligned} \phi(\boldsymbol{x}) &> 0 \quad \boldsymbol{x} \in D \\ \phi(\boldsymbol{x}) &= 0 \quad \boldsymbol{x} \in \Gamma \\ \phi(\boldsymbol{x}) &< 0 \quad \boldsymbol{x} \in \Omega \backslash D \end{aligned} \tag{59}$$

where $\phi(\boldsymbol{x}) \geq 0$ describes the interface D.



Compared to traditional front propagation techniques for example parameterizing the moving front or discretizing the interior [52], [61], the level set technique is particularly suited for modelling topological changes, but also to accurately track the interfaces. Therefore it has been widely applied for shape based front propagation problems, especially for complex interfaces.

5.1.2 A Hamilton-Jacobi form of level set evolution

For interface propagation, we normally consider a level set evolution in the configuration space $\phi(\boldsymbol{x}, t)$ with a pseudo-evolution time $t \in [0, \tau]$. By differentiating ϕ, we obtain

$$\mathrm{d}\phi = \nabla\phi \cdot \mathrm{d}\boldsymbol{x} + \frac{\partial\phi}{\partial t}\mathrm{d}t$$

We define a velocity field $V(\boldsymbol{x}, t) = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}$. Consider the evolving boundary of an object where $\phi(\boldsymbol{x}, t) = 0$; then for a small time step, we have that $\mathrm{d}\phi = 0$. Then the above yields that

$$\frac{\partial\phi}{\partial t} + V(\boldsymbol{x}, t) \cdot \nabla\phi = 0 \tag{60}$$

We consider the expression for the outward normal $\boldsymbol{n} = \frac{\nabla\phi}{|\nabla\phi|}$; we apply this to obtain a Hamilton-Jacobi equation for the level set evolution [19], [61], [51];

$$\frac{\partial\phi}{\partial t} + F(\boldsymbol{x}, t)|\nabla\phi| = 0 \tag{61}$$

with $F = V \cdot \boldsymbol{n}$.

The discretization of the Hamilton-Jacobi form of the level set evolution can be viewed as an iteration scheme for the level set function

$$\phi_{n+1} = \phi_n - \alpha F_n|\nabla\phi_n| \tag{62}$$

where α is defined as the pseudo-time step (or simply step-size) and that $\phi_n = \phi(\boldsymbol{x}, t_n)$, $F_n = F(\boldsymbol{x}, t_n)$. We will later calculate a velocity choice of F which guides the evolution into a descent direction of a properly chosen cost functional in order to obtain a good level set evolution for our model.

5.1.3 A steepest descent approach by level set

Now we consider our governing model for FWI (18). Moreover, we consider a situation where the parameter m is comprised by two pieces; the object and the background, and we denote the domain by $D \subseteq \Omega$. The parameter is therefore considered of the form

$$\begin{aligned} m &= m_{obj}(\boldsymbol{x}) \quad \boldsymbol{x} \in D \\ m &= m_{back}(\boldsymbol{x}) \quad \boldsymbol{x} \in \Omega \backslash D \end{aligned}$$

where m_{obj} and m_{back} are defined from the model setup. Consider a characteristic function χ_D, such that $\chi_D(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in D$, and $\chi_D(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in \Omega\backslash D$; Then the model representation can be formulated also as $m = m_{obj}(\boldsymbol{x})\chi_D(\boldsymbol{x}) + m_{back}(\boldsymbol{x})(1 - \chi_D(\boldsymbol{x}))$. Applying the level set definition in (59), we consider the one-dimensional Heaviside functional satisfying $H(\phi(\boldsymbol{x})) = \chi_D(\boldsymbol{x})$. This gives a level set based representation

$$m = m_{obj}(\boldsymbol{x})H(\phi) + m_{back}(\boldsymbol{x})(1 - H(\phi)) \tag{63}$$

augmented by the 'free-surface' condition.

Now we reconsider the governing form based on a level set representation of

m. This yields the following form

$$min_\phi E(m(\phi)) = \frac{1}{2}\sum_s \sum_r \|\boldsymbol{P}_{\mathcal{R}}\boldsymbol{w}^s(\boldsymbol{x},t) - d^s_{obs}(\boldsymbol{x}_r,t)\|^2_U \tag{64}$$

with constraints

$$\mathcal{L}(m(\phi))\boldsymbol{w}^s = \boldsymbol{q}^s$$

$$m = m_{obj}(\boldsymbol{x})H(\phi) + m_{back}(\boldsymbol{x})(1 - H(\phi))$$

For the case of a level set representation, we notice that all parameters contained in $m = (\lambda, \mu, \rho)$ can be written using such a level set representations (63). This indicates that the evolution of ϕ and the estimation of m can be considered equivalent.

We need to derive a suitable level set iteration for ϕ; In [19] a 'Shape reconstruction by steepest descent' method is introduced for level set based inverse problems; In that work a Maxwell equation is applied to electromagnetic imaging, which has some similarities to our elastic setup; we therefore apply the same technique for seismic imaging in our model. We recall the pseudo-time $t \in [0, \tau]$ for the level set evolution; applying then the chain rule for the derivative of the energy functional E with respect to t we obtain

$$\begin{aligned}\frac{\partial E}{\partial t} &= \frac{\delta E}{\delta m}\frac{\partial m}{\partial \phi}\frac{\partial \phi}{\partial t} \\ &= ((m_{obj} - m_{back})\boldsymbol{g}_m \cdot \delta(\phi))\frac{\partial \phi}{\partial t}\end{aligned} \tag{65}$$

where $\boldsymbol{g}_m$ is the parameter gradient defined in (31), and where $\delta(\phi) = H'(\phi)$ is a Dirac delta functional. We assume for simplicity that the level set function ϕ is continuously differentiable on D and that $|\nabla\phi| \neq 0$ on the boundary. Then we use that

$$\delta(\phi) = \frac{\delta_{\partial D}(\boldsymbol{x})}{|\nabla\phi(\boldsymbol{x})|} \tag{66}$$

Now we define the steepest descent direction as

$$\frac{\partial \phi}{\partial t} = -\boldsymbol{g}_m \cdot |\nabla\phi| \tag{67}$$

on the boundary Γ. Theoretically, we still need to find a suitable 'extension velocity' [19] so that the evolution of the level set function it is defined on the

entire domain Ω. We assume a velocity field F_{SD} defined on Ω, such that $F_{SD} = -\boldsymbol{g}_m \cdot (m_{obj} - m_{back})$ on Γ, and arbitrary elsewhere; this gives us a Hamilton-Jacobi equation so that it is identical to steepest descent evolution

$$\frac{\partial \phi}{\partial t} + F_{SD}|\nabla \phi| = 0 \tag{68}$$

Define $\boldsymbol{g}_\phi = -(m_{obj} - m_{back})\boldsymbol{g}_m \cdot |\nabla \phi|$; then we obtain the steepest descent iteration by

$$\phi_{n+1} = \phi_n - \alpha_n \boldsymbol{g}_\phi(\phi_n) \tag{69}$$

which finally defines our steepest descent shape evolution scheme.

5.1.4 Narrow band approach

The above gradient descent formulation has a major problem; the gradient direction for its level set evolution is only defined at the boundary Γ of the current object representation, where the level set function is strictly zero; also, the gradient of level set function is strictly defined non-zero by this standard. However in numerical implementation, the zero contour boundary region would be difficult to detect, and it is difficult to define numerically the outward normal direction. Also, the condition that $|\nabla \phi| \neq 0$ is difficult to guarantee during evolution. Some authors apply the technique of re-initialization [39] to ensure that the level set gradient is approximately smooth during the evolution such that $|\nabla \phi| = 1$; We will instead use a narrowband technique as outlined below which has proven to provide very good results when applied in the solution of inverse problems.

A narrow band approach has been produced by D. Adalsteinsson and J. Sethian [1] as to only compute the level set evolution in a narrow band close to Γ, as to reduce the computational cost, but still keeps the characteristics of the object boundary as to ensure its front propagation still follows the required velocity field. We define the narrow band region $D_{nbd}(\phi, d)$ as follows

$$D_{nbd}(\phi, d) = \{\boldsymbol{x} | \exists \boldsymbol{x}_0, \; such \; that \, |\boldsymbol{x} - \boldsymbol{x}_0| < d, \phi(\boldsymbol{x_0}) = 0\} \tag{70}$$

In an extended narrow band, the assumption of $|\nabla \phi| \neq 0$ might not be easy to achieve. Some additional research applies a re-initialization technique so as to

evolve ϕ such that $|\nabla\phi| \approx 1$, so as to ensure the level set evolution would not experience instability problems; but it would risk breaking the structure of zero contour γ [39]. Therefore we remove the term of $|\nabla\phi|$ in our descent direction, and obtain the following definition as [19]:

$$\frac{\partial\phi}{\partial t} = -\boldsymbol{g}_m \cdot \chi_{D_{nbd}(\phi,d)}(m_{obj} - m_{back}) \tag{71}$$

where χ is the characteristic function. Therefore we arrive at the iteration with narrow-band

$$\phi_{n+1} = \phi_n - \alpha_n \boldsymbol{g}_\phi(\phi_n) \tag{72}$$

with

$$\boldsymbol{g}_\phi(\phi_n) = -(m_{obj} - m_{back})\boldsymbol{g}_m \cdot \chi_{D_{nbd}(\phi_n,d)} \tag{73}$$

Notice that those modifications conserve the descent property of the shape evolution which is the main objective when deriving our speed functions.

5.1.5 A level set based full waveform inversion algorithm

We apply a smoothing technique using Sobolev regularization method in Chapter 2.7.1, as a smoothing term for the level set gradient; this method is also applied in other research [19], [20]. Different other regularization techniques such as DRLSE, or a Mumford-Shah functional, could be applied as well [39], [45], [19], but we will not consider those in this thesis. We finally obtain the following

algorithm for level set based full waveform inversion:

Algorithm 5: Level set based full-waveform inversion algorithm

Initial m_0,ϕ_0, α, source and receivers, narrow band $D_{nbd}(\phi_0)$ with defined bandwidth d;

for $i = 1, 2, ...$ **do**

- m_i;
- Forward wave propagation;
- Adjoint-state Backward propagation;
- Compute pixel based gradient $\boldsymbol{g}_m^i = \frac{\delta E}{\delta m}(m_i)$;
- Compute level set gradient $\boldsymbol{g}_\phi^i$ (L^2 or Sobolev gradient);
- Define narrow band $D_{nbd}(\phi_i)$;
- Update α_i with backtrack line search technique; Update ϕ^{i+1}, such that $\phi_{i+1} = \phi_i - \alpha_i \boldsymbol{g}_\phi^i$ in narrow band;
- Update related level set representation of $m_{i+1} = m(\phi_{i+1})$;
- **if** *Stopping criterion satisfies* **then**
  - break;
- **end**

end

5.2 Numerical tests

5.2.1 Model set

We introduce a numerical test case for verifying the above algorithm in level set based FWI using Algorithm 5. The following results have been published as a conference paper *'A level set method for shape reconstruction in seismic full waveform inversion using a linear elastic model in 2D'* [76].

We test our algorithm by using two different reference set-ups displayed in figures 56, 58, both addressing the imaging of salt domes buried in the ground with FWI. The left hand column shows a profile where only one salt dome is included in the ground. The right hand column shows a slightly more complicated set-up with two separate salt domes hidden in the ground. The top row of this figure shows the p-wave velocities V_p, the center row the s-wave velocities V_s, and the bottom row the densities ρ for both cases; the model is displayed on a 30×120 domain.

We apply a source distribution of 21 equidistant seismic locations along a horizontal line to the surface, and the source wavelet given in a Ricker-type wavelet; for sensors, we choose a set of 81 equidistant locations also along the horizontal line, as to record the time series of arriving seismic waves. This means that we have only a top view on the domain of interest available for gathering data, which should simulate a practical experiment in seismic surveys. Specially, we apply a 5% of Gaussian noise in the misfit data, in order to avoid the inverse crime.

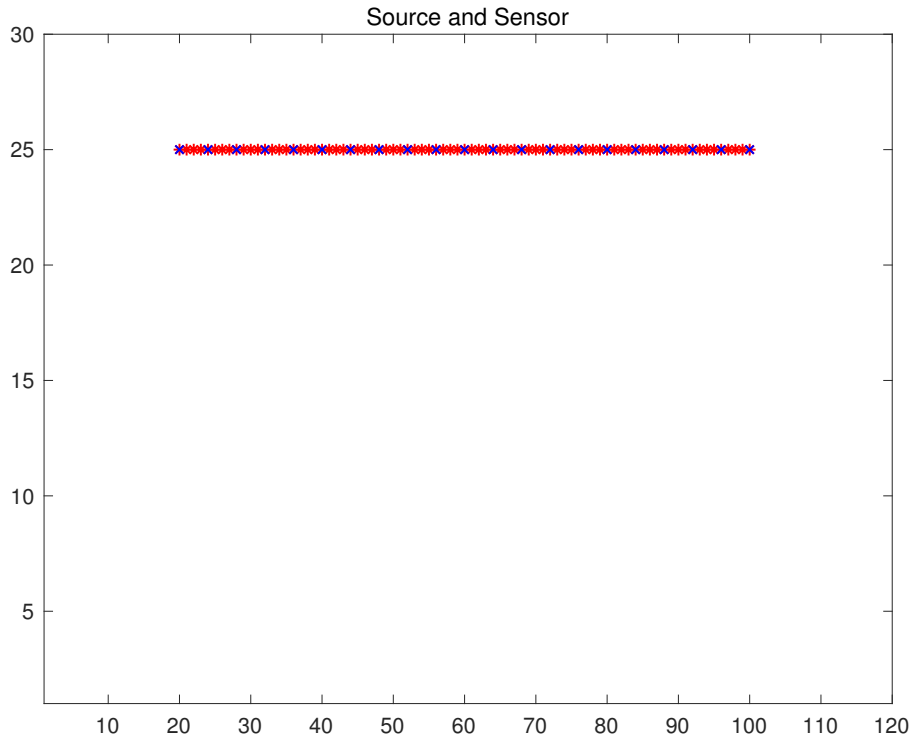


Figure 55: Source and sensor: the blue crosses denotes the source positions, and the red dots represent the sensor location

We apply the same object and background parameter as in (58), and obtain the following 3 models with different set up of level set representation describing different salt dome information:

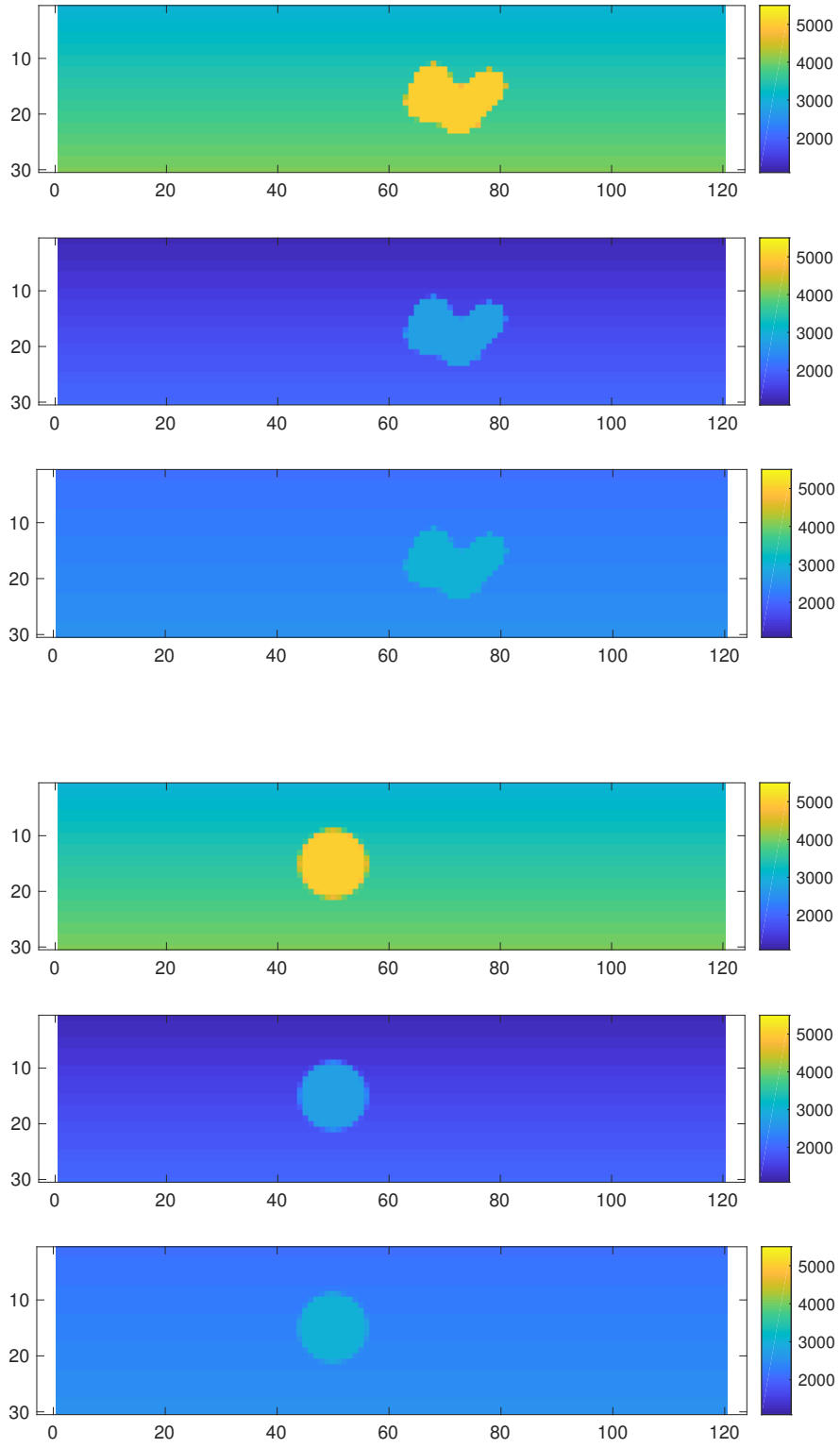


Figure 56: Left: true model for model 1; Right: initial model for model 1. From top to bottom are model values of parameters of V_p, V_s, ρ

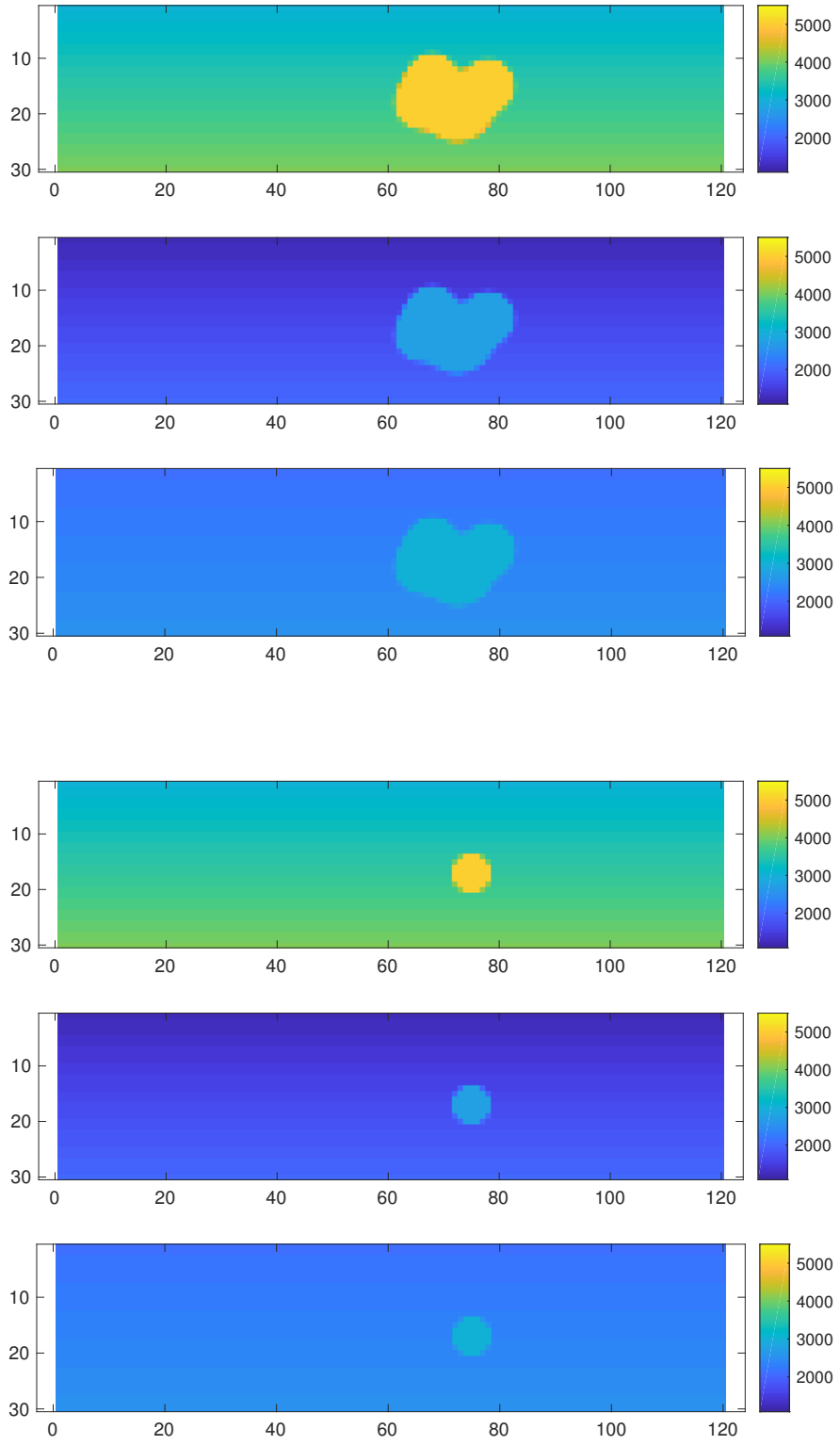


Figure 57: Left: true model for model 2; Right: initial model for model 2. From top to bottom are model values of parameters of V_p, V_s, ρ

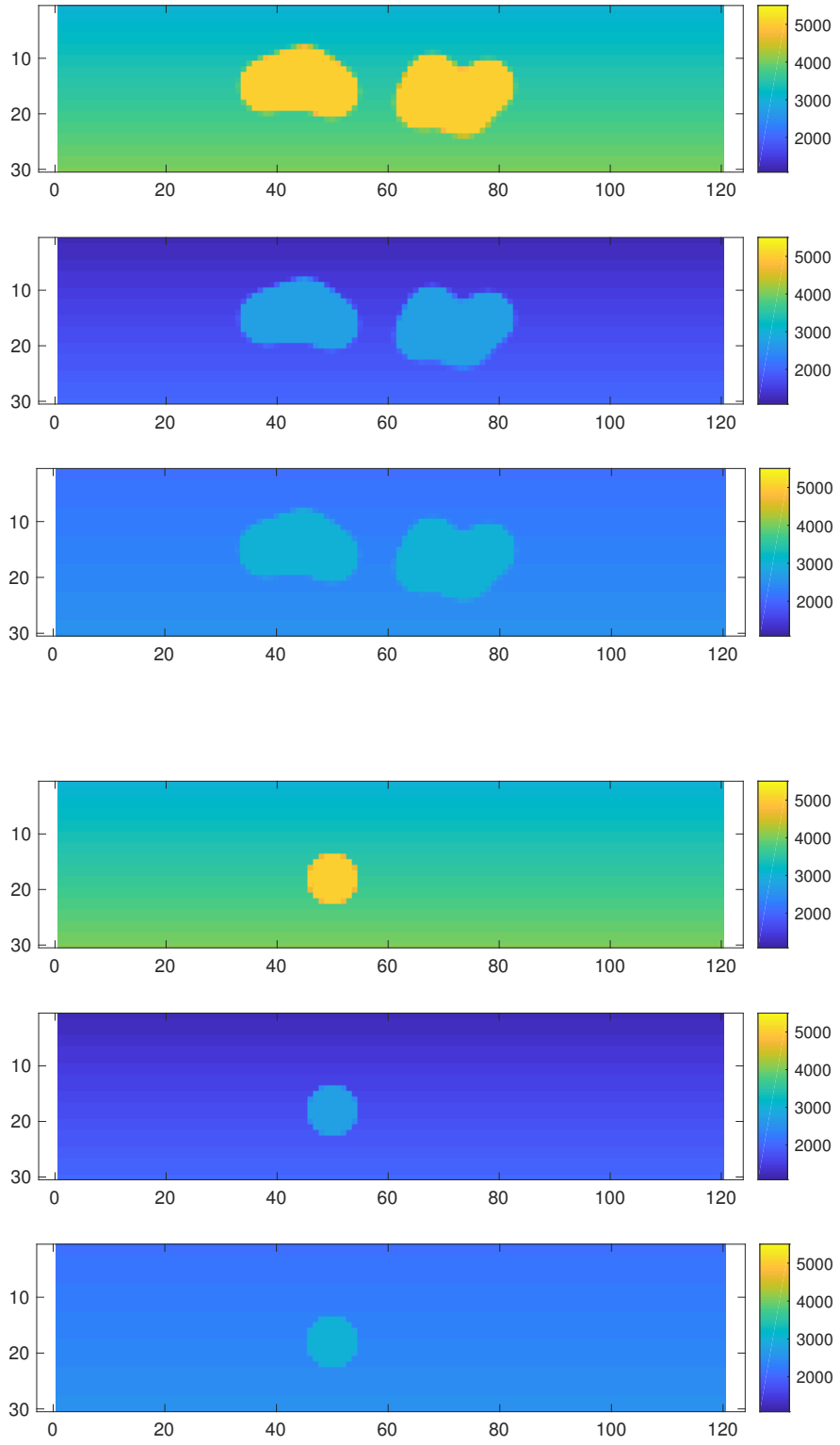


Figure 58: Left: true model for model 3; Right: initial model for model 3. From top to bottom are model values of parameters of V_p, V_s, ρ

5.2.2 Some settings concerning level set evolution

We also need to specify some numerical implementation details concerning the level set evolution.

Level set construction The level set functions ϕ that represent the initial and true models are defined here as a linear expansion of a radial basis functions [3]

$$\phi = \sum_i \alpha_i \psi(\|\beta_i(\boldsymbol{x} - \boldsymbol{x}_i)\|) - c \tag{74}$$

where α_i, β_i, c are weight scalars, $\boldsymbol{x}_i$ are the center locations of every basis function. We define the radial basis function (RBF) in the form of Gaussians with

$$\psi(\boldsymbol{r}) = \exp(-\boldsymbol{r}^2) \tag{75}$$

In this form we can generate a sufficiently big set of artificial level set functions representing initial and true shapes. During the shape evolution, however, this RBF parameterization is not used here.

Heaviside function The numerical approximation of the Heaviside function is given in by following definition

$$\begin{aligned} H(\phi) &= 0, \ \phi < \epsilon \\ &= \frac{1}{2}(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin\frac{\pi\phi}{\epsilon}), \ -\epsilon < \phi < \epsilon \\ &= 1, \ \phi > \epsilon \end{aligned} \tag{76}$$

with ϵ sufficiently small.

5.2.3 Numerical Results

We apply the above level set evolution scheme for FWI following Algorithm 5; we set the maximum iteration number to be 30 for model 1, model 2, and 50 for model 3 (which is slightly more complex); we set up the appropriate line search step size for different tests, and the stopping criterion is linked to the error bound reaching 5×10^{-6}. Notice that using the level set method significantly increases the sensitivity of steepest descent direction falling into a local minimum; therefore the

choice of the initial line search step size has to be extremely cautious. Therefore we choose the appropriate initial step size for each different test case individually, and the results are given next.

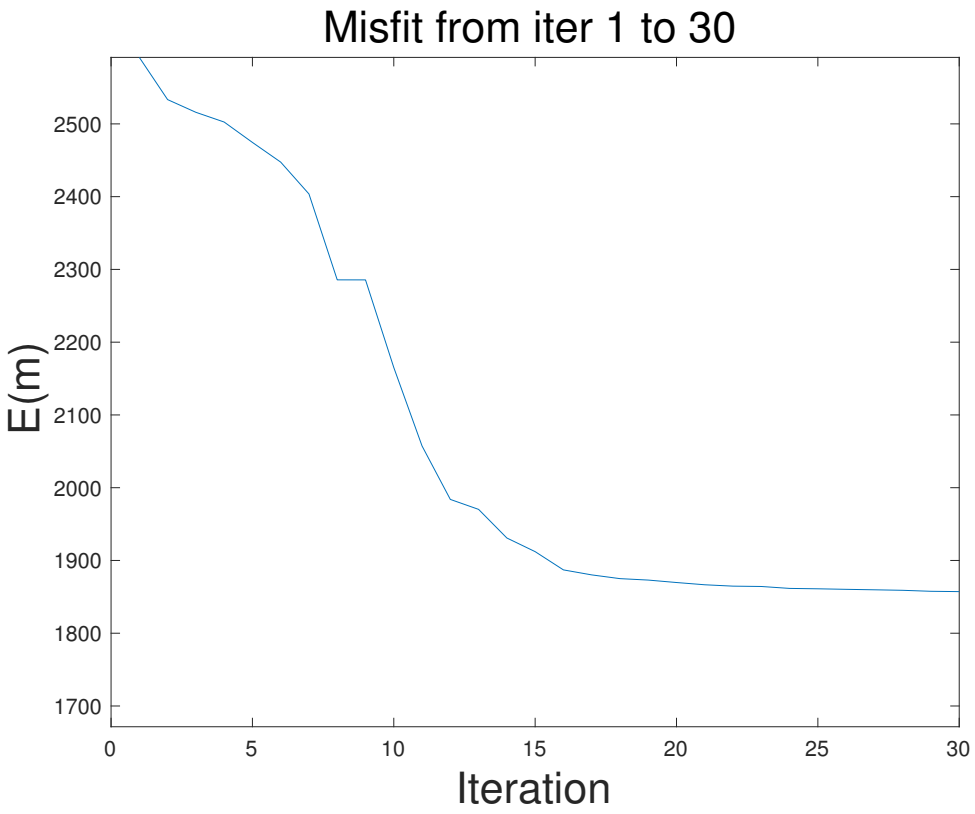


Figure 60: Misfit for model 1

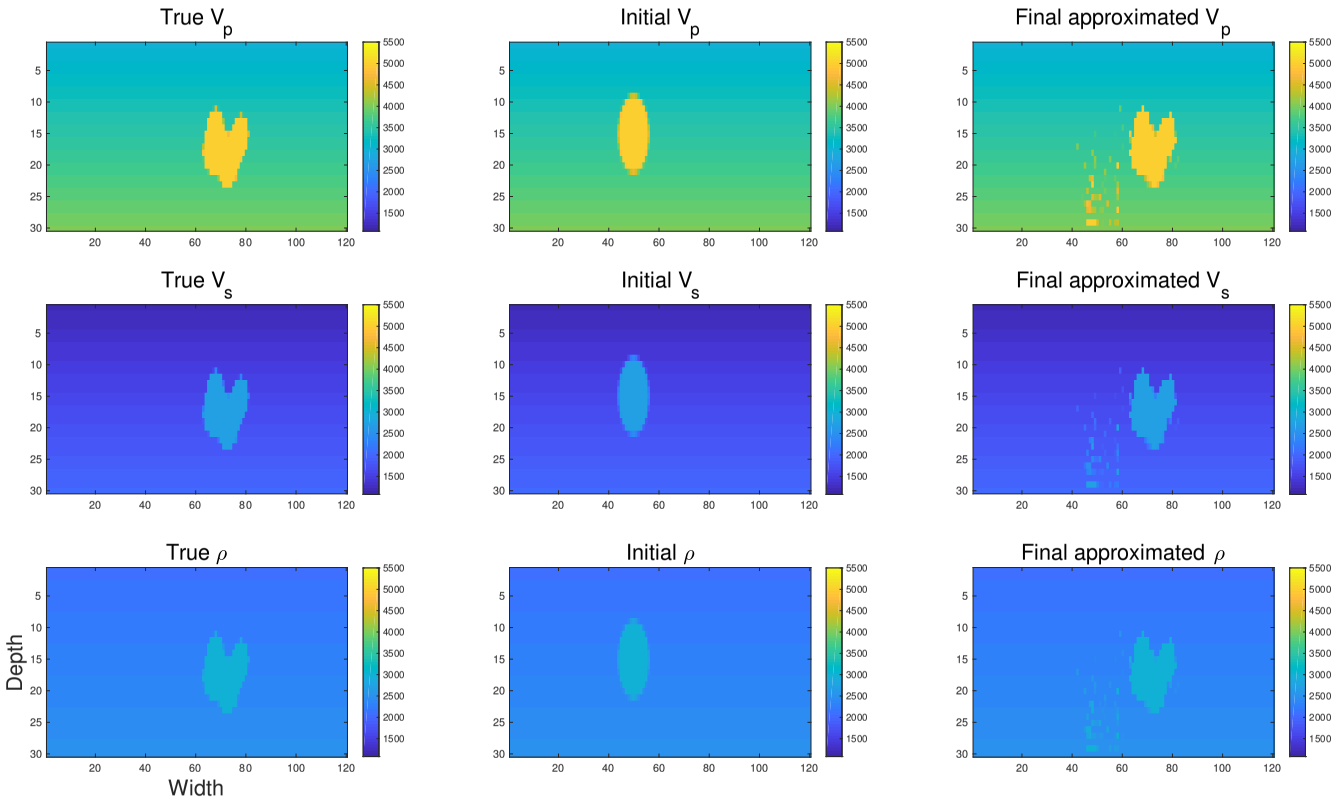


Figure 59: Test result for model 1. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

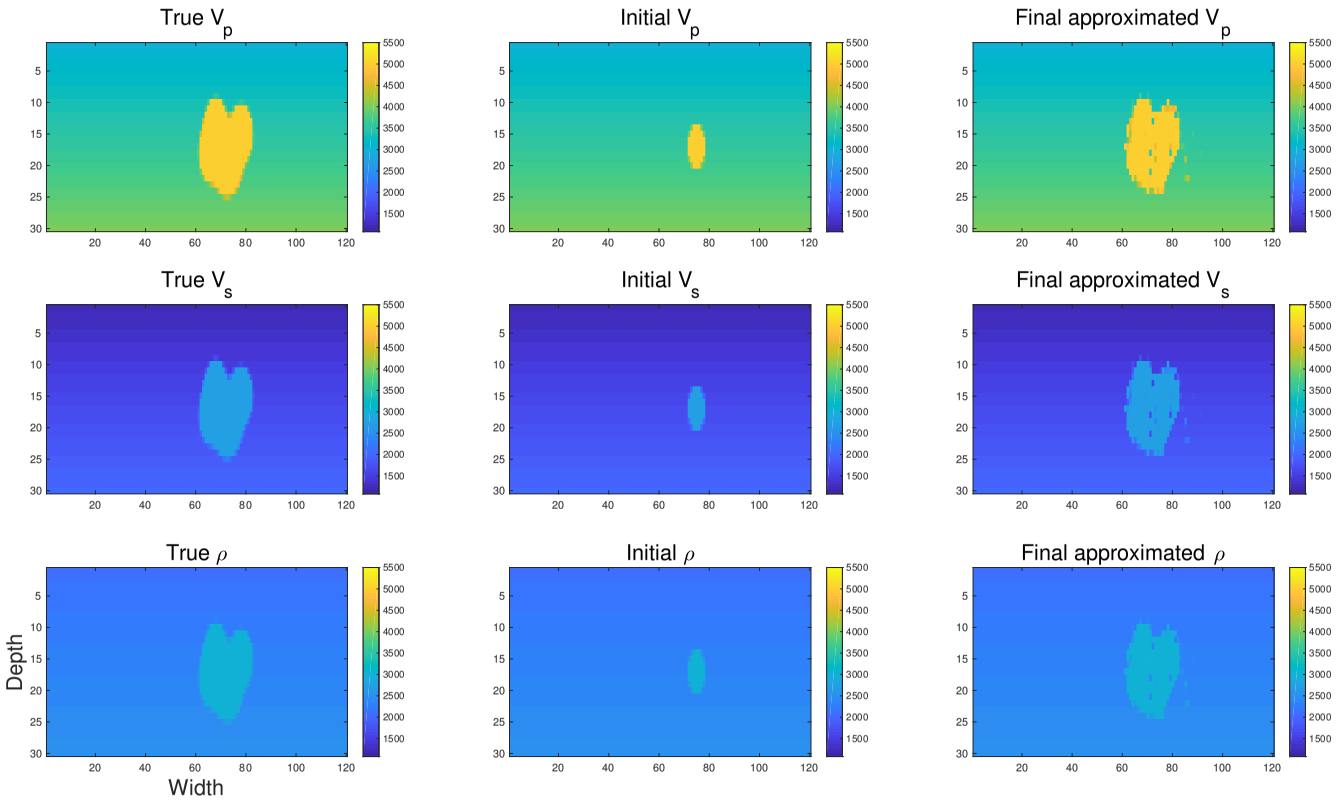


Figure 61: Test result for model 2. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

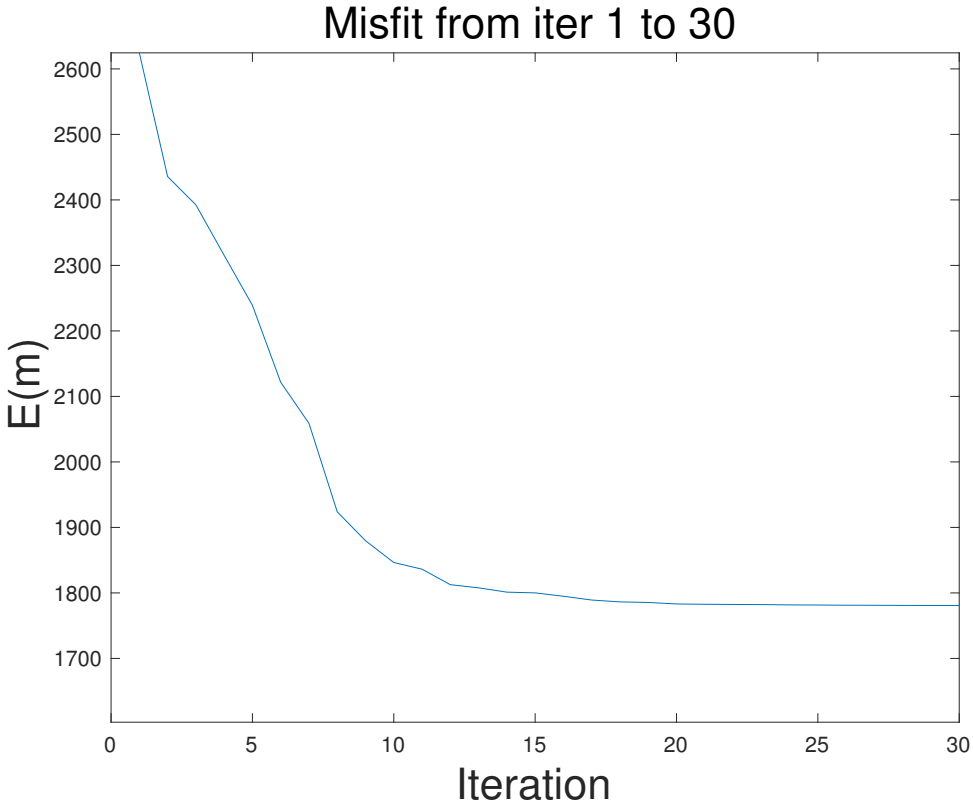


Figure 62: Misfit for model 2

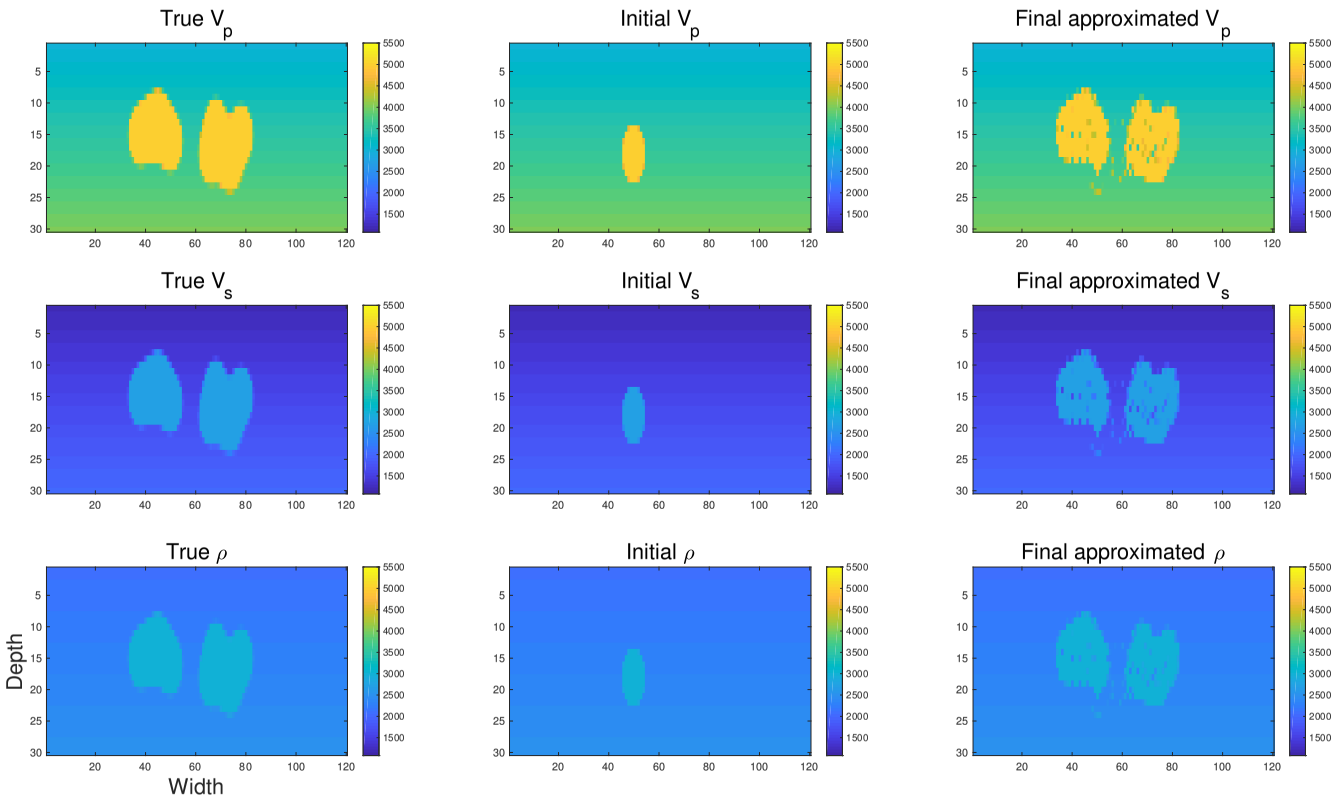


Figure 63: Test result for model 3. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

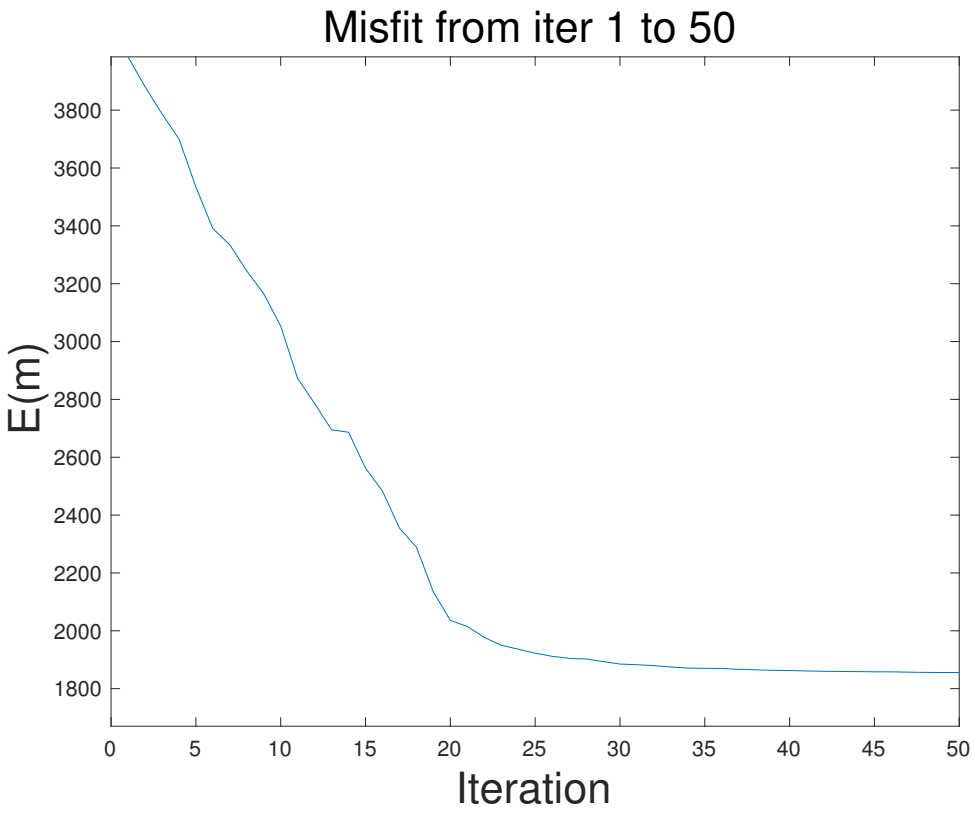


Figure 64: Misfit for model 3

5.2.4 Sobolev gradient in level set technique

The previous test cases show that, as expected, compared to pixel based methods, a level set representation with prescribed parameter values inside the inclusions and only updating the common shape avoids the problem of 'cross-talk'. The final approximation is improved, but boundaries tend to be fractured and small holes are visible inside the salt bodies which are most likely not justified by the data but need to be considered artifacts of the reconstruction scheme. In order to address this shortcoming, we apply now the Sobolev gradient method to smooth the level set function before applying the updates; the weight parameter is set to $\gamma = 0.2$ for model 1, and $\gamma = 1$ for model 2 and model 3. Notice that increasing γ will certainly smooth the level set function and the object boundary, but it comes with the risk of losing other details of the shapes or of blocking some topological changes (splitting) necessary to reach the final solution; we choose the above value for smoothing as a trade-off between those different aspects.

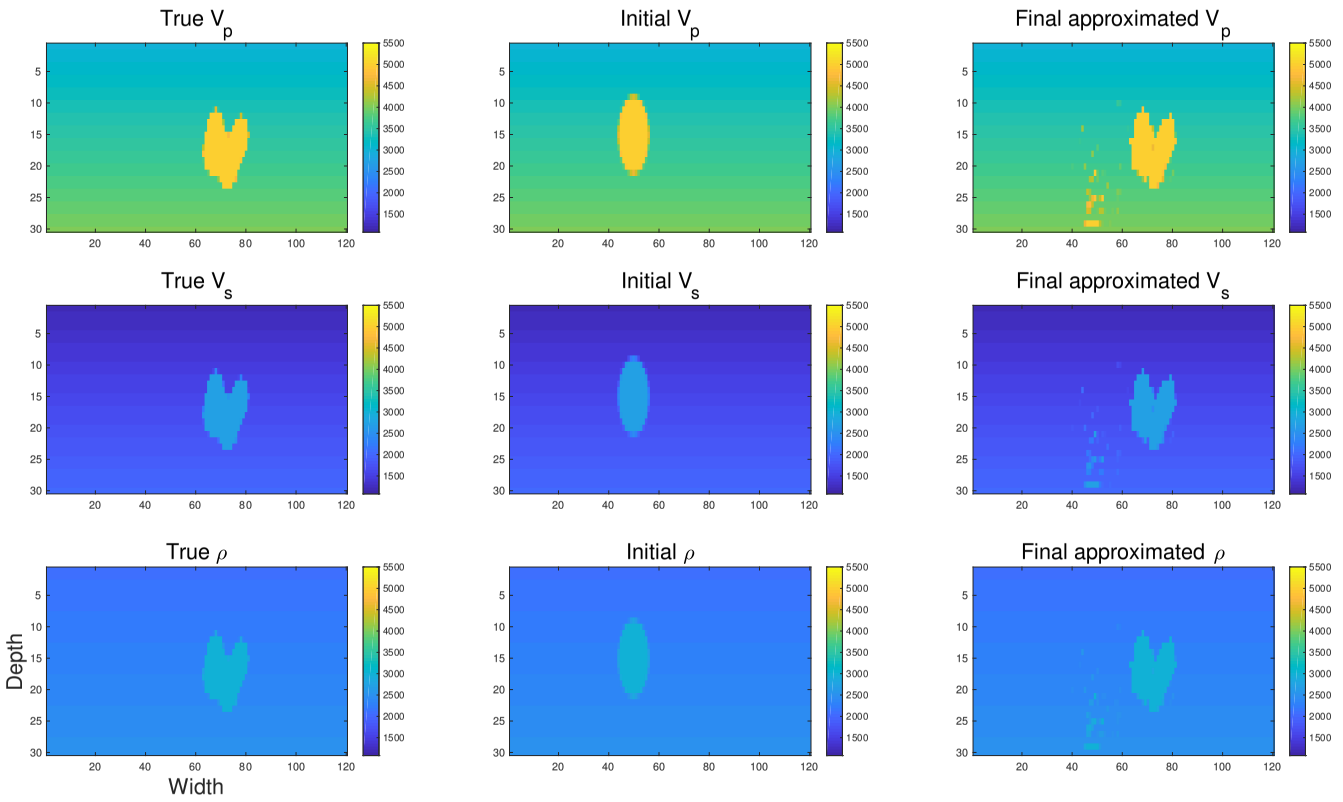


Figure 65: Test result for model 1: Sobolev gradient where $\gamma = 0.2$. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

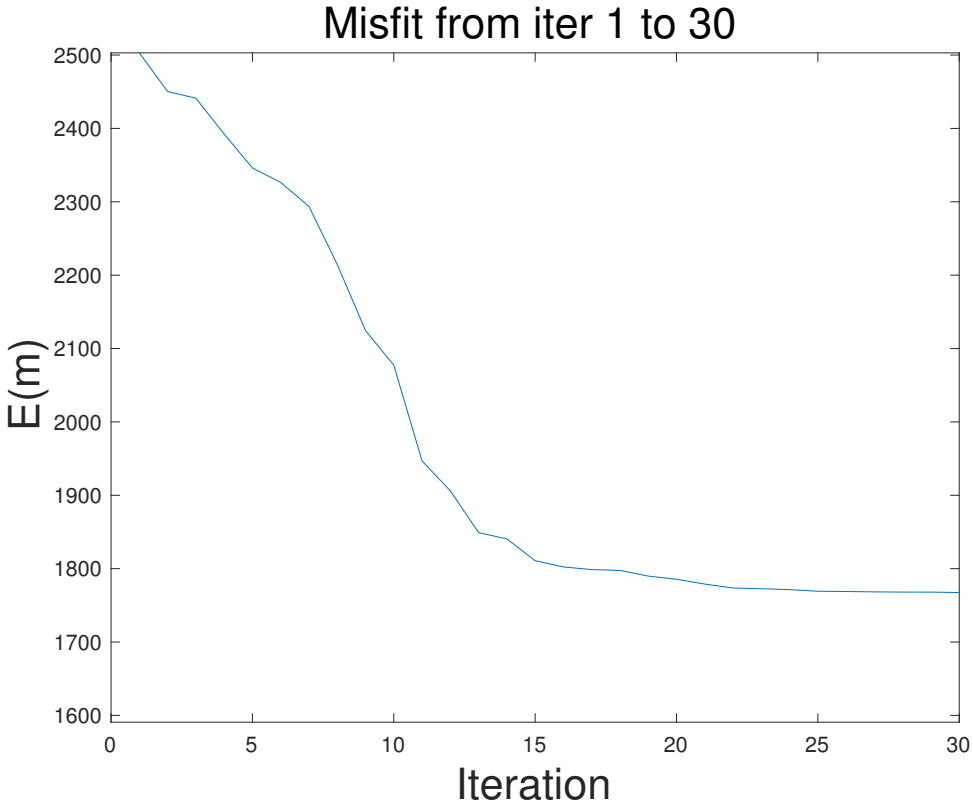


Figure 66: Misfit for model 1: Sobolev gradient where $\gamma = 0.2$

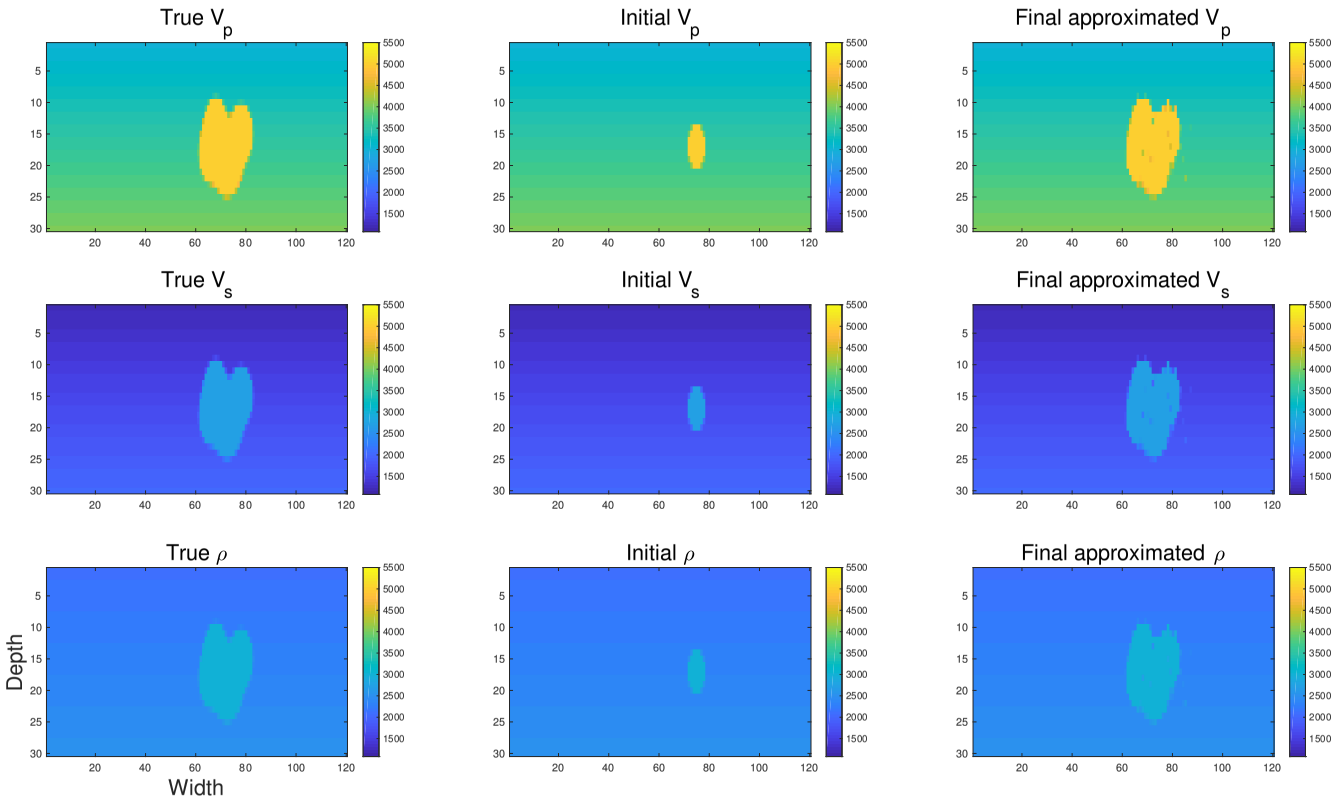


Figure 67: Test result for model 2: Sobolev gradient where $\gamma = 1$. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

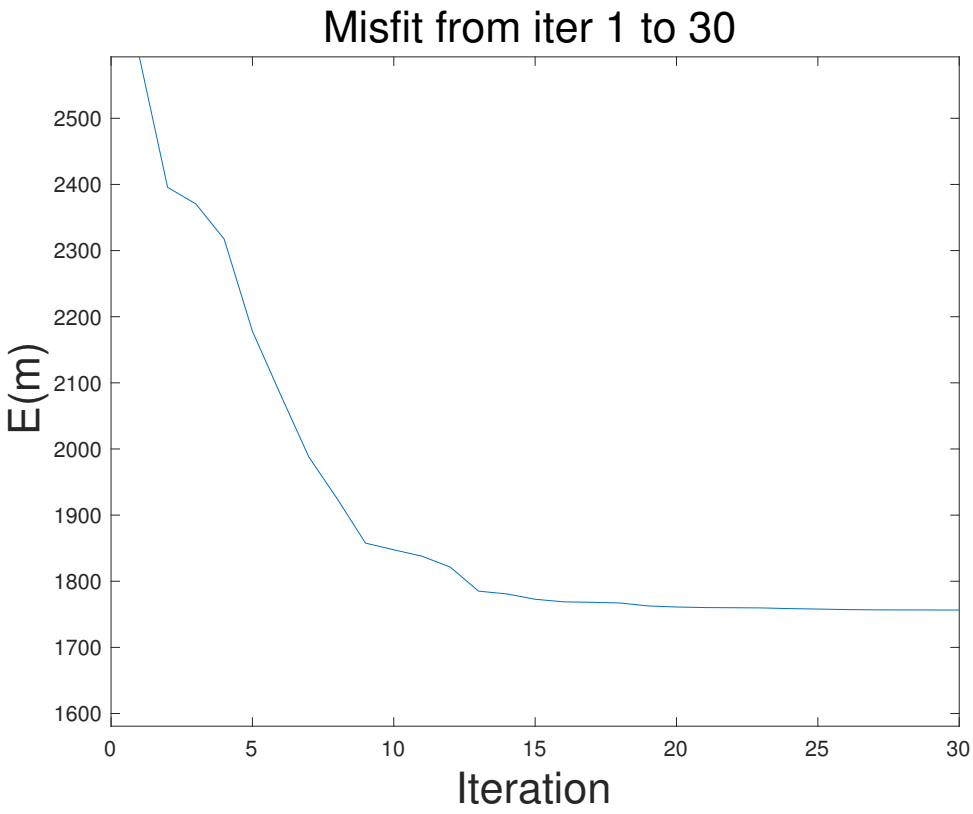


Figure 68: Misfit for model 2: Sobolev gradient where $\gamma = 1$

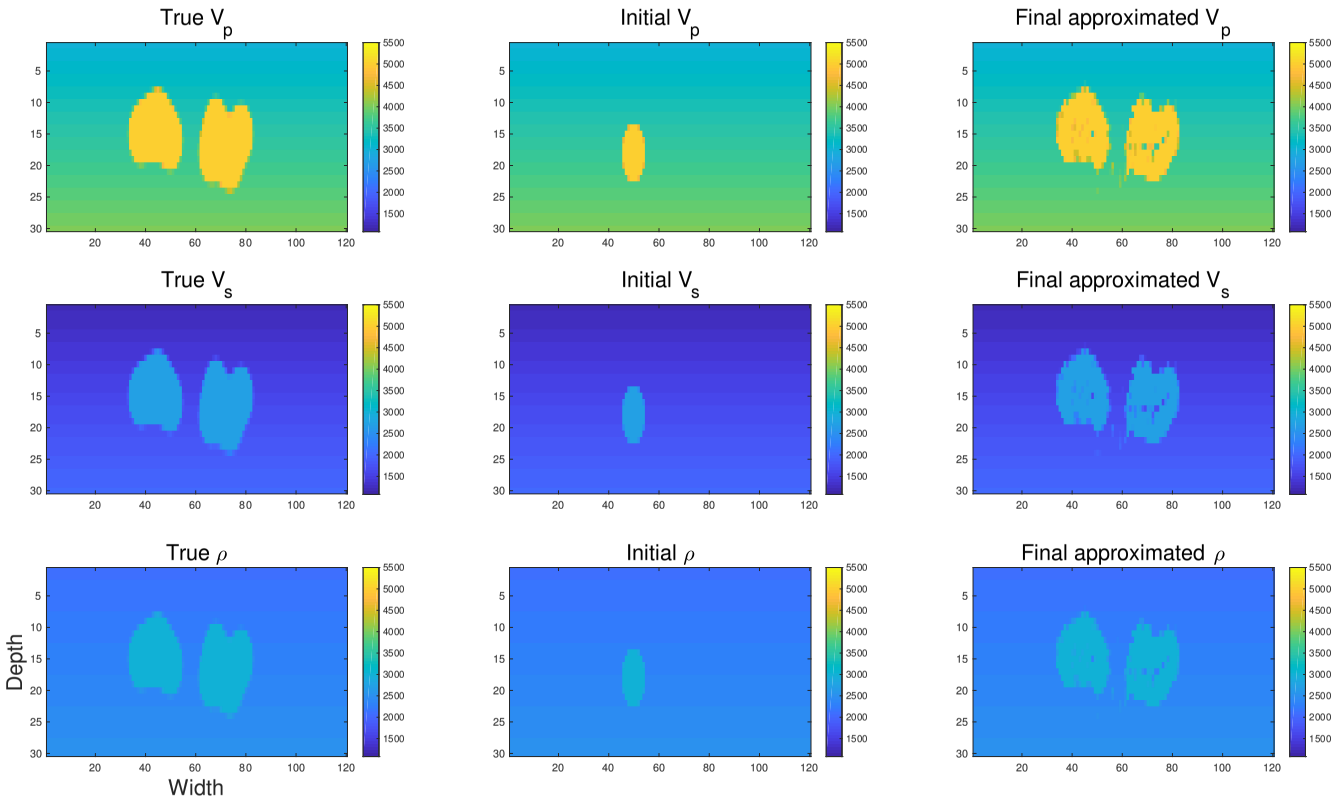


Figure 69: Test result for model 3: Sobolev gradient where $\gamma = 1$. The subfigures from left to right show the true model, initial model and final approximation; from top to bottom they represent V_p, V_s and ρ

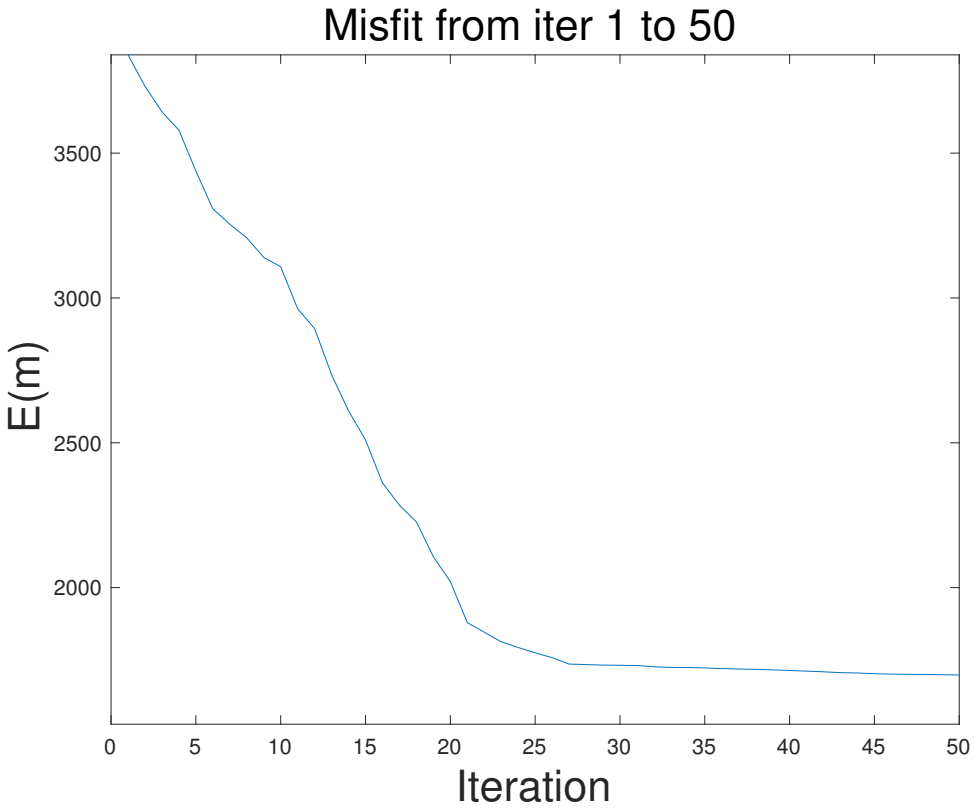


Figure 70: Misfit for model 3: Sobolev gradient where $\gamma = 1$

In the following figures we compare the numerical results for V_p with and without Sobolev gradient.

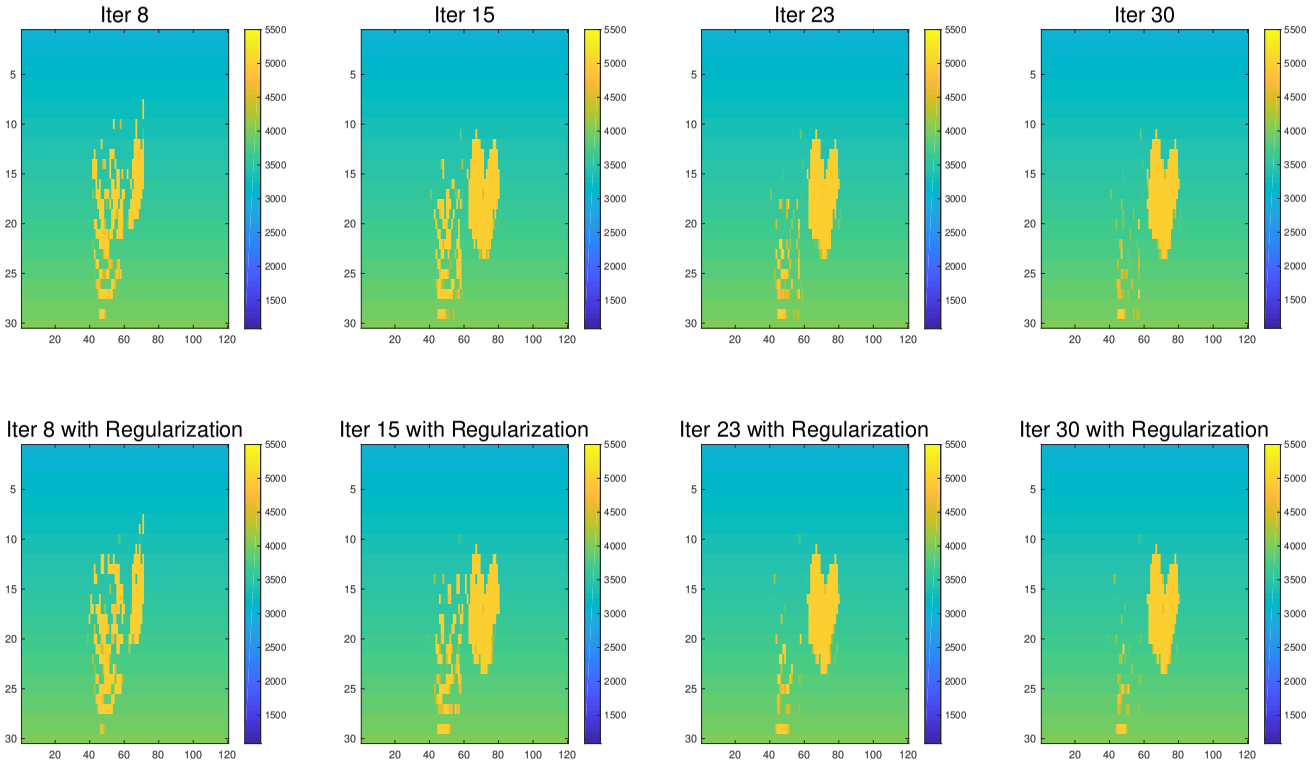


Figure 71: V_p comparison for model 1. Top and bottom: Level set iteration with L^2 gradient and Sobolev gradient ($\gamma = 1$). Left to right: V_p at iteration 8, 15, 23, 30

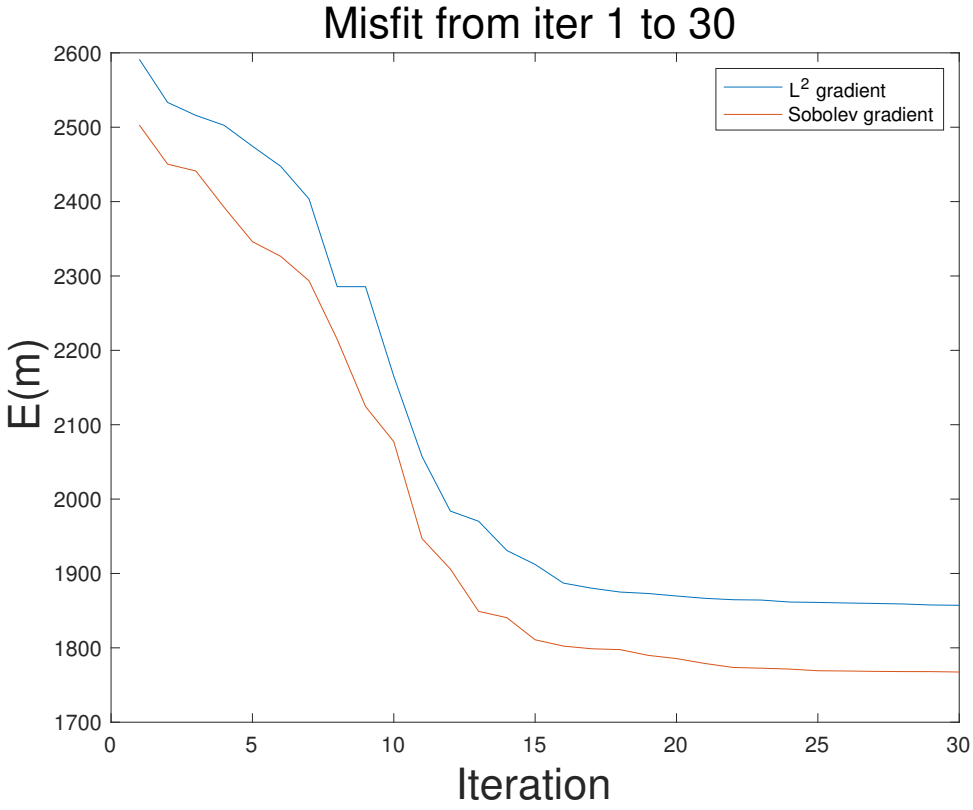


Figure 72: V_p comparison for model 1: Misfit evolution

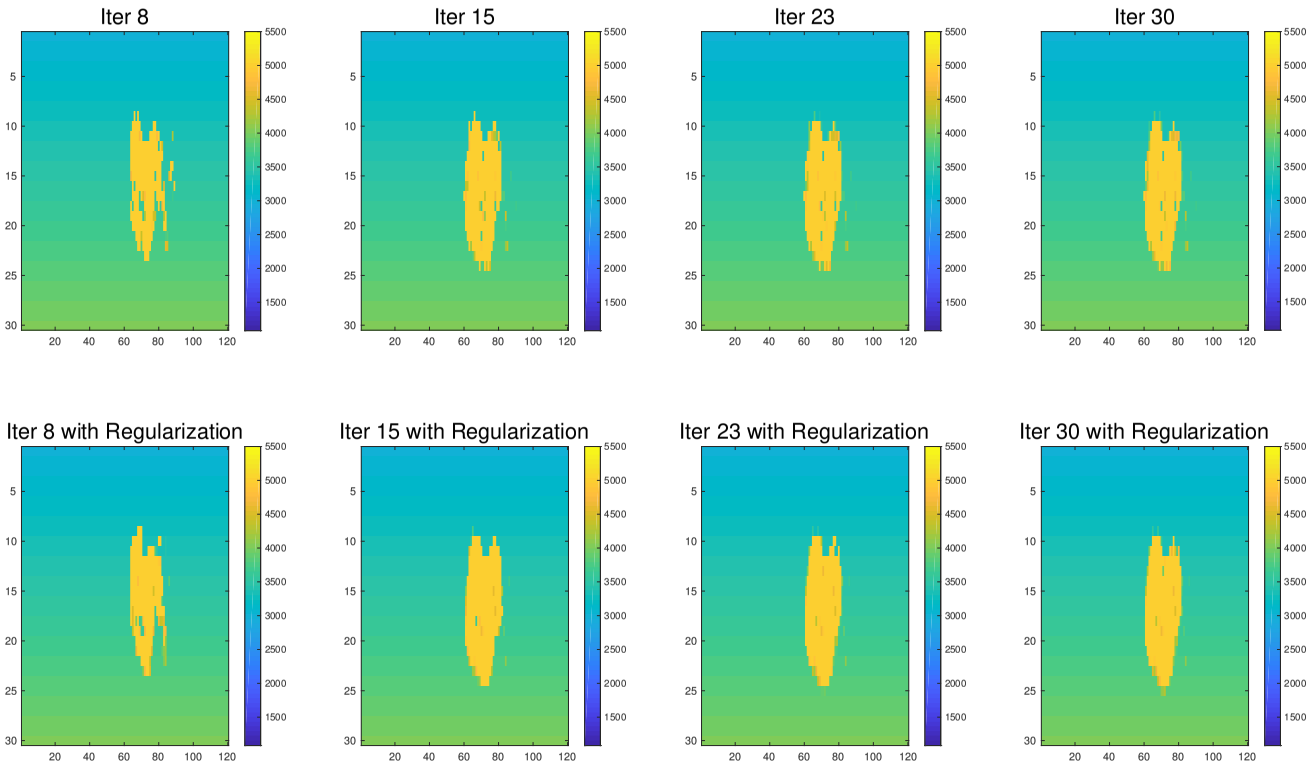


Figure 73: V_p comparison for model 2. Top and bottom: Level set iteration with L^2 gradient and Sobolev gradient ($\gamma = 1$). Left to right: V_p at Iteration 8, 15, 23, 30

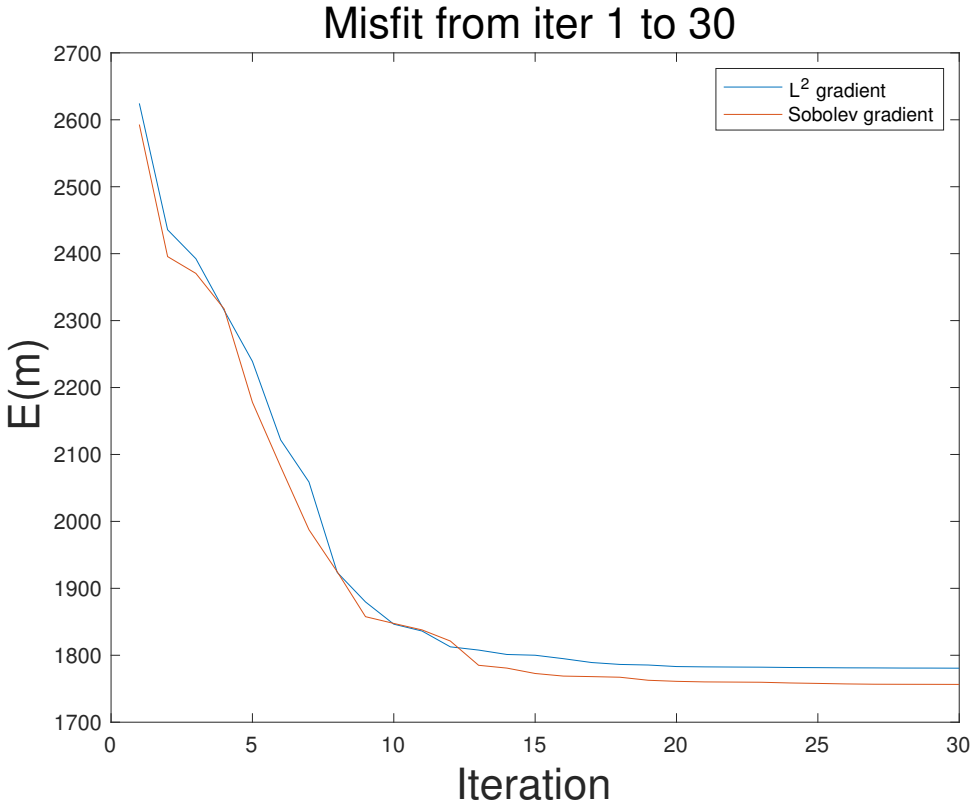


Figure 74: V_p comparison for model 2: Misfit evolution

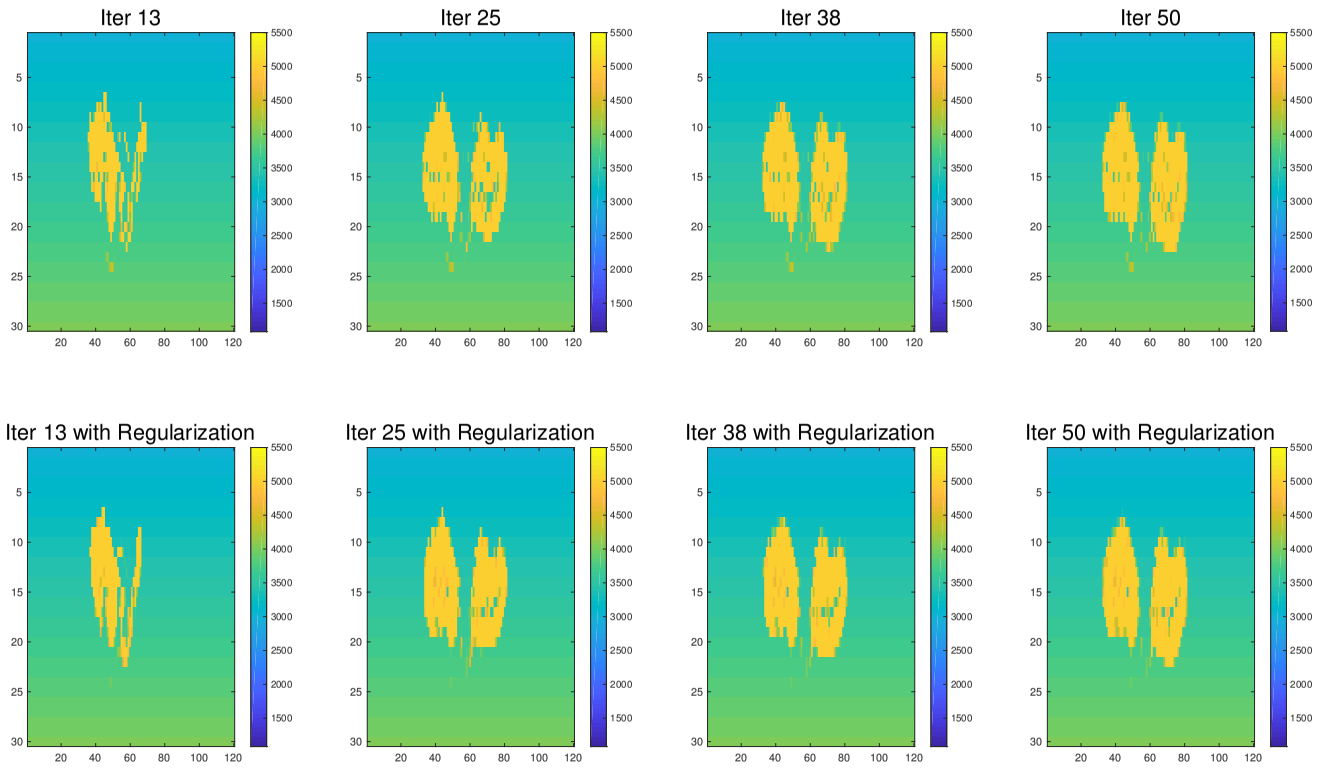


Figure 75: V_p comparison for model 3. Top and bottom: Level set iteration with L^2 gradient and Sobolev gradient ($\gamma = 1$). Left to right: V_p at Iteration 13, 25, 38, 50

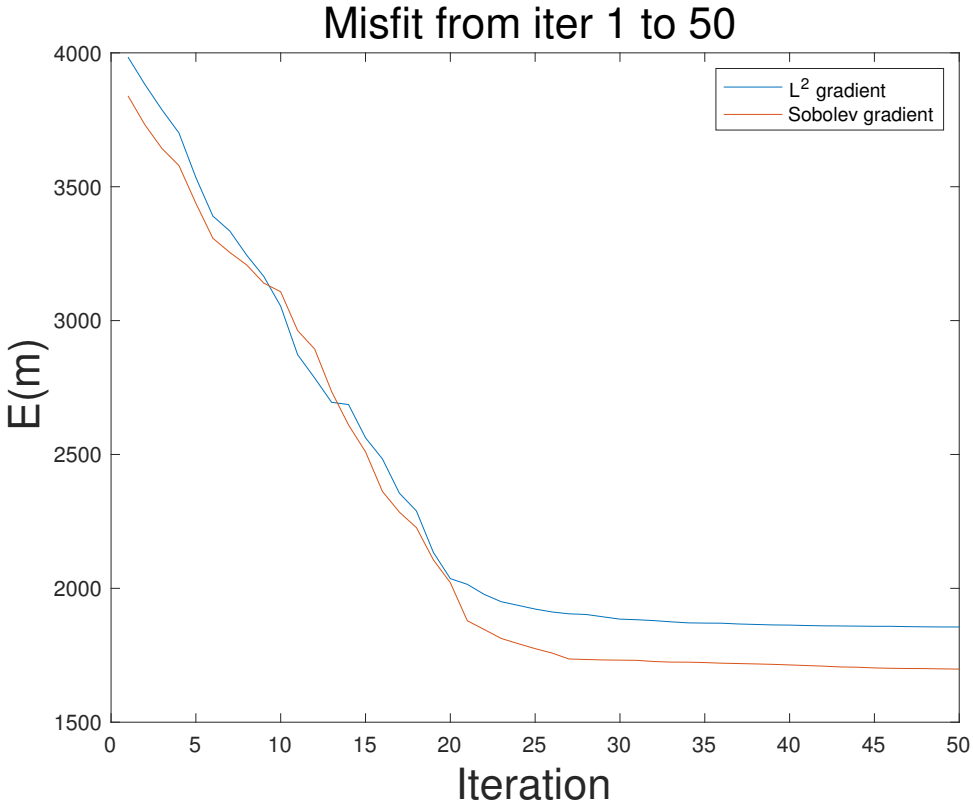


Figure 76: V_p comparison for model 3: Misfit evolution

5.3 Conclusion

The numerical tests presented above show that applying a Sobolev gradient has a vital impact to the smoothing of the boundaries in the level set evolution, which is significantly different compared to the pixel based method. How can we explain this difference? Compared to the pixel based method in 4.4.2, our model test for the level set representation divides the seismic survey into domains of object and background, which creates a clear sharp boundary. This is strictly defined by the level set function. Therefore, irregularities around the object boundary are diminished by the incorporation of a Sobolev gradient.

Compared to pixel based methods, using a level set method seems to avoid the inaccuracies caused by the insufficient information from the limited data problem; it helps solving the problem of 'cross-talk', since the object value and background information can be assumed approximately known for salt domes, and therefore can be better controlled. For those reasons, the level set based inversion scheme

shows a clear advantage over pixel based inversion, especially when sharp boundaries are present and for irregular shape reconstruction problems such as salt dome characterization. There might be some irregular distributions with traditional L^2 gradient, but the application of Sobolev gradient solves that generally well.

The application of the Sobolev gradient also comes up with a side effect, that the smoothing technique may damage the seismic information provided in the previous update; therefore a good choice for weight parameter γ for Sobolev gradient remains crucial. However, numerical results also show that using traditional line search techniques in a level set evolution can be extremely time consuming, especially when initial guesses are poorly chosen; Also, considering that the line search applied for reducing the data misfit also requires massive computation, the choice of initial line step should be done carefully.

Note that O. Dorn and A. Hiles [20] provide an alternative line search technique using the idea of Landweber-Kaczmarz type. See also [38] in a different field of level set based imaging; this single step idea updates the level set function by using gradients for subsets of data, for example corresponding to individual source terms. Notice that this single step method is also very similar to the stochastic gradient method discussed frequently in the literature nowadays for big data problems, where its idea has also been applied for full waveform inversion [42].

In the next chapter we will apply a very similar technique to our level set evolution for our 2D model. This will serve as a starting point for future application in 3D full waveform inversion which represent a big data problem.

6 A Stochastic gradient technique for 2D level set based full waveform inversion

The stochastic gradient method (SGD) is an iterative method applied to solve optimization problems based on unbiased stochastic gradient computation; it is orginigated by the work 'A Stochastic Approximation Method' by H. Robbins and S. Monro in 1951 [56]; now it has been widely applied in large-scale problems, especially in machine learning [9]. Its basic idea is by randomly taking gradients from independent sources from data sets as descent direction, instead of traditional methods which compute the standard gradient from all sources. While one gradient descent direction may not represent a standard gradient information, the complete iteration should generally follow the standard descent direction from a bird-eye perspective. In large-scale FWI, this method can be expected to be much more efficient than computing standard gradients. T. van Leeuwen [42] introduces a stochastic optimization method in full waveform inversion; however, the application of SGD method in full waveform inversion is still at an early stage.

O. Dorn and A. Hiles [20] presented a nonlinear Landweber-Kaczmarz type method applied in level set based electromagnetic imaging, which shows high similarity to the stochastic gradient descent method; we will compare a variant of the SGD to Landweber-Kaczmarz type algorithm to find out their relations; then we will apply a stochastic gradient descent algorithm in level set based full waveform inversion for our model. We test our model reconstruction with a relatively large scale set of sources and sensors, and apply a holdout method for cross validation to measure our data estimation.

6.1 Stochastic gradient descent

We briefly review the generalised stochastic gradient descent method. Consider the optimization problem

$$\min_m F(m) = \frac{1}{N} \sum_{s=1}^{N_s} F_s(m) \tag{77}$$

where $F_s(m)$ is the energy least square functional denoting data misfit based on source s; N_s is the number of sources. The standard gradient descent method gives a steepest descent direction, written as $\nabla_m F$. This could be written as

$$m_{n+1} = m_n - \frac{\alpha_n}{N} \sum_{i=1}^{N_s} \nabla_m F_s(m_n) \tag{78}$$

where α is given as the step length, sometimes called learning rate. For this method, the computation of the gradient with respect to m requires all sources evaluated at F_s; with a large source number of N_s, especially in practical applications where the source number is generally of the order of several thousands, this computation would be extremely expensive.

In SGD, one computes only one single source version of the gradient $\nabla_m F_{\gamma(s)}(m)$, where $\gamma(s)$ is a randomized selection of sources s from $(1, 2, ..., N_s)$ with a probability density $p(s)$. This gives the simple form of SGD iteration written as

$$m_{n+1} = m_n - \alpha_n \nabla_m F_{\gamma(s)}(m_n) \tag{79}$$

This descent direction does not need to represent a descent direction for the full data misfit, but it only represents one for a single source; therefore the computational cost is significantly reduced. This gives SGD a clear advantage in large-scale problems, such as full waveform inversion.

6.1.1 Weighted SGD

Some additional research [46] applies a normalized weight function $w(s)$ as to weight gradient component that

$$F_s^{(w)}(m) = \frac{1}{w(s)} F_s(m) \tag{80}$$

This gives us an equivalent stochastic representation for $F(m)$

$$F(m) = \mathbf{E}(F_s(m)) = \mathbf{E}(F_s^{(w)}(m))/\mathbf{E}(w(s)) \tag{81}$$

where $\mathbf{E}$ represents the expectation with respect to $s \in S$, and $\mathbf{E}(w(s)) = 1$. With this we weigh the gradient component according to

$$\nabla F_s^{(w)}(m) = \frac{1}{w(s)} \nabla F_s(m). \tag{82}$$

Therefore we obtain a weighted SGD from (79), (82) with

$$m_{n+1} = m_n - \alpha_n \frac{1}{w(\gamma(s))} \nabla_m F_{\gamma(s)}(m_n) \tag{83}$$

6.1.2 Relation to randomized Landweber-Kaczmarz method

From the above idea, we can derive that the idea presented in [20] is actually a variant of a stochastic gradient descent method. So, what is the relationship between SGD and Landweber-Kaczmarz method?

The Kaczmarz method is applied in the nonlinear Algebraic Reconstruction Technique (ART) [29], extending the linear ART technique to nonlinear problems. It has originally be developed as an iterative model for solving linear systems arising in tomography, but lately it has also been considered as a promising technique for solving non-linear ill-posed problems [20], [38], [12].

A generalised linear least square problem is given to solve the linear system $\boldsymbol{Ax} = \boldsymbol{b}$ such that

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) = \frac{1}{2} \|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 \tag{84}$$

where $\boldsymbol{A}$ is an $m \times n$ matrix, and $\boldsymbol{x}$, $\boldsymbol{b}$ are corresponding vectors. Consider $\boldsymbol{A}$ with rows $\boldsymbol{a}_i$ such that $i = 1, 2, ..., m$, which provides the linear system $\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle = b_i$ with $\boldsymbol{b} = (b_i)$. The randomized Kaczmarz method for solving such a system therefore proceeds as [46], [77], [66], [32]:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x_k} - \alpha \frac{\langle \boldsymbol{a_i}, \boldsymbol{x_k} \rangle - b_i}{\|\boldsymbol{a}_i\|_2^2} \boldsymbol{a_i} \tag{85}$$

where i is the randomized selection from $1, 2, ..., m$. In [46], a randomized Kaczmarz method is considered as a specialised case for weighted SGD method (83),

where the weight function is defined as:

$$w(i) = \frac{m\|\boldsymbol{a}_i\|_2^2}{\sum_{i=1}^m \|\boldsymbol{a}_i\|_2^2} \tag{86}$$

As for nonlinear Landweber-Kaczmarz method mentioned in [20], [38], [12], this approach follows the same formulation of (79) or (83), such that it can also be considered as an application of the SGD method.

6.2 A level set based SGD algorithm for full waveform inversion

6.2.1 SGD gradient

In the following part, we apply a SGD method for full waveform inversion based on level set; for simplicity, we set the weight function to be that $w(s) = 1$, assuming that the information from all sources are unbiased. Recall the data misfit of the level set based FWI $E(\phi)$ (64), and the level set gradient $\boldsymbol{g}_\phi$ based on steepest descent. We can rewrite them as the sum of s sources such that $E(\phi) = \sum_s E_s(\phi)$, and $\boldsymbol{g}_\phi = \sum_s \boldsymbol{g}_\phi^s$. Consider the chain rule for level set gradient

$$\boldsymbol{g}_\phi = \boldsymbol{g}_m \frac{\partial m}{\partial \phi} \tag{87}$$

and the gradient form of (31)

$$\boldsymbol{g}_m = -\sum_{s\in\mathcal{S}} \langle \boldsymbol{v}^s, \frac{\partial \mathcal{L}}{\partial m}\boldsymbol{w}^s\rangle_{\boldsymbol{\tau}} \tag{88}$$

This gives us the s component of gradient information

$$\boldsymbol{g}_\phi^s = \langle \boldsymbol{v}^s, \frac{\partial \mathcal{L}}{\partial m}\boldsymbol{w}^s\rangle_{\boldsymbol{\tau}} \frac{\partial m}{\partial \phi} \tag{89}$$

We consider the random selection of $\gamma(s)$, and apply the SGD gradient from (79) as to obtain the algorithm; we apply the narrow band technique for extending the level set evolution as in 5.1.4. However, the estimation of step length α can be a challenge. we will discussing the determination of the step length in the following part, and then construct our algorithm.

6.2.2 Estimating step length

In standard gradient problems, the search for a suitable step length (learning rate) α is normally controlled by traditional conditions such as an Armijo, Wolfe or strong Wolfe condition, as to ensure that the step length is suitable for line search. Stochastic gradient descent directions, however, as well as Kaczmarz descent directions, do not contain the full gradient information in one search; therefore a large step length (also known as learning rate) would provide misleading results. Most researchers tend to take a sufficiently small step length, but then the convergence speed can be relatively slow. This happens most commonly with the artificial selection of step lengths, such as the adaptive learning rate method; alternative methods such as Adagrad or Adadelta try to provide a monotonically decreasing learning rate; but for nonlinear problems, its computation would be too time-consuming.

Another alternative idea is to apply a residual based method to simulate the data control of the step length; this can give a good control for generalised problems, but it also has high risk of over-fitting the model, especially for noisy data. The above considerations outline the difficulty of finding an optimum estimation for step length.

Pixel count control In level set reconstructions, specifically, we could apply a pixel-count based inexact backtracking line search technique as proposed in [20]. This technique controls the evolution of shapes to be restricted by relatively small amounts. in addition, any additional forward or backward solves are avoided in the step length selection, in order not to increase the computational cost; Certainly, without additional forward solves it is difficult to monitor the evolution of the data misfit accurately.

In our approach following that of [20], we record the number N_j of pixels that changes value in each update, where j is defined as the source index; such change occurs when the level set value at certain pixels evolves changes sign (either direction). The goal is to identify a step length which delivers a shape evolution where the number of pixels that change value is restricted between certain upper and lower bounds.

We define an upper and lower bound N_{max}, N_{min} for the number of pixels that change value in each update. Then for any update for a source j inside a

given sweep, we make sure that $N_j \in [N_{min}, N_{max}]$. For the step size α, we apply a backtracking line search where we monitor the pixel change N_j (instead of the data misfit as in traditional backtracking schemes) which avoids time-consuming data misfit calculation; we assume that τ_1, τ_2 are the step length decrease and increase ratio such that $0 \leq \tau_1 \leq 1$, and $\tau_2 \geq 1$. For a more detailed description of the technique, together with further numerical evidence of its performance, we refer the reader to [20].

6.2.3 Validation method

In machine learning, validation is a process of evaluating the behaviour of model reconstruction. We should introduce some of the basic concepts of machine learning, and derive their equivalent explanations in FWI, before we go into the details of validation methods.

Training set Training set indicates the data sets that are used to run the model. For example, in FWI the training set is the source set for running the wave propagation simulation.

Test set Test set indicates the data sets that are used to validate the model. In FWI, the test set refers to the source set that is used for measuring the data from receivers.

In machine learning, one of the most common model error is known as over-fitting. In statistics, over-fitting refers to the result that the model fits the training data too well, such that even some of the residual variation (noise, for example) also fits into the data estimation, which reduces its accuracy for future observation prediction.

Over-fitting commonly happens in data mining process; in linear regression, for example, over-fitting mostly happens when one fits noisy linear data by a high-order polynomial function; this regression could fit very well for the given noisy data, even better than the linear estimation; but it has lost its value in prediction, therefore estimating a new set of data would most likely cause a very large error. In full waveform inversion, data measurements normally come with a certain proportion of noise. Therefore it is very likely that for a certain model,

one obtains a reconstruction where the modelled data fit the noisy observed data extremely well, but the result fails to approximate the true model. In validation techniques, we try try to apply a different set of sources to detect and measure this error.

Most of the over-fitting happens when the iteration procedure starts to 'memorize' the data fitting process rather than to 'learn' the right way of iteration. In FWI, each individual step of the SGD does not represent the full gradient; therefore the case of over-fitting happens frequently. Over-fitting also happens sometimes if we fail to separate the test set from the training set during the simulation; then the model will try to 'memorize' the test set data in the future steps. This happens more commonly with residual evaluation, meaning that the model updates with a criterion involving data residual measurement.

How do we prevent over-fitting? The most common idea is to apply a validation method; validation method is a numerical model validation technique to test how good the numerical results actually are from a group of validation sets. In order to avoid over-fitting, we need to separate the training set from the test set. Different methods of validation are applied in all sorts of data mining problems; we will in the following introduce three of the most common methods, and analyse their pros and cons in suitability for FWI.

Hold-out method Holdout method is considered as the simplest validation form; its basic idea is to separate the original data set S into training set S_1 and test set S_2; the set up of S_1 and S_2 is somehow artificial, and fixed for all model tests. To avoid over-fitting, it is normally defined that the size of test set S_2 should be sufficiently smaller than the size of the training set S_1. The advantage of the hold-out method is its simplicity and consistency in data evaluation; but the evaluation can also turn out to be highly dependent on the division of training set and test set. In FWI, considering a large scale source set, data information from different source terms vary significantly. Therefore the selection of training sources and test sources should be done very carefully.

K-fold validation K-cross validation is considered an improvement to the hold-out method. Its basic idea is to separate the data set S into k different subsets $S_1, S_2, ..., S_k$. One of the k subsets is defined as test sets, and the other $k-1$ are training sets. We should repeat the model evaluation k times, so that

each subset of S are used both for training and test set. The advantage of K-fold validation is that, each observation is used for training and testing for repeated tests; in a large sample, the data variation would be reduced as k increases. However, it requires at least k times of model tests, therefore it requires k times more computation to validate our model result. In FWI, considering that one model test would normally require a significant amount of computation, the application of k-fold validation might become very time-consuming.

Leave-p-out cross-validation Leaving p-out cross validation is considered to using p observations for validation among the n data sets; these p validations are test sets and other $n - p$ are training sets. This validation requires C_n^p times of model tests; specifically, $p = 1$ gives an equivalent form of k-fold validation where $k = |S|$ is the observation number of data set. Leave-p-out validation generally requires more model tests.

6.2.4 SGD algorithm

In FWI, we apply a randomized source evaluation process; one iteration should sample over all sources in a randomized order. We repeat the iteration until the

model update suits the stopping criterion. The algorithm is given next.

Algorithm 6: SGD level set inversion algorithm

Initial model m_0, initial level set ϕ_0;
Iteration $i = 1, 2, ...N$; Source set S, training set S_1, test set S_2;
Source $s = 1, 2, ..., |S_1|$;
Initial set up for step length search: pixel change bound N_{min}, N_{max}, initial step length α_{start}, the line step decrease and increase ratio τ_1, τ_2;
for $i = 1 : N$ **do**
  for $j = 1 : |S_1|$ **do**
    Randomized order $\gamma_i(j)$;
    Forward and backward propagation on source $\gamma_i(j)$;
    Compute corresponding level set gradient: $\boldsymbol{g}_\phi^{\gamma_i(j)}$;
    Initialize α_{start};
    Find step length α_j such that pixel change lies in N_{min}, N_{max};
    if $j \leq |S_1| - 1$ **then**
      $\phi_i^{(j+1)} = \phi_i^{(j)} - \alpha_j \boldsymbol{g}_\phi^{\gamma_i(j)}$;
    end
    if $j = |S_1|$ **then**
      $\phi_{i+1}^{(1)} = \phi_i^{(j)} - \alpha_j \boldsymbol{g}_\phi^{\gamma_i(j)}$;
    end
    Update m; Validation on S_2;
    Verify stopping criterion;
  end
end

Notice that in the validation procedure the test set is independent from the training set.

6.3 Numerical results

6.3.1 Hold-out validation tests

For our model, we will apply the SGD algorithm above for our model update; we will also implement our numerical tests on the models considered in Chapter 5. Notice the SGD method is expected to show its real power when moving to large scale 3D problems where many more source and receiver locations are usu-

ally considered and each individual computation requires considerable computing time. This makes full gradient based methods extremely slow, and even cyclic Kaczmarz type methods might take a long time to even complete one single sweep (addressing the entire data set).

For source and sensors, we apply in our 2D test case a larger set of sources as to better describe the application of SGD. In particular, we use 101 sources, defined as our training set. The sensors are set up in the same location as in Chapter 5, with 81 equidistant sensors at the same level near the surface.

We apply a hold-out method for validation to monitor the model reconstruction, so as to add to its simplicity in validation and reduce computation cost. Our test set is provided be an independent set of sources; In particular, we set up 5 different test receivers located 1 grid cell beneath the line of source locations, equality distributed beneath the sources 1, 26, 51, 76, 101; so that this test data should be independent from observed data, but could also give a relatively fair validation on the model reconstruction. This is sufficient for our test case, but in practical applications any other selection might be possible. The distribution of sources in the test sets are given in the following figure.

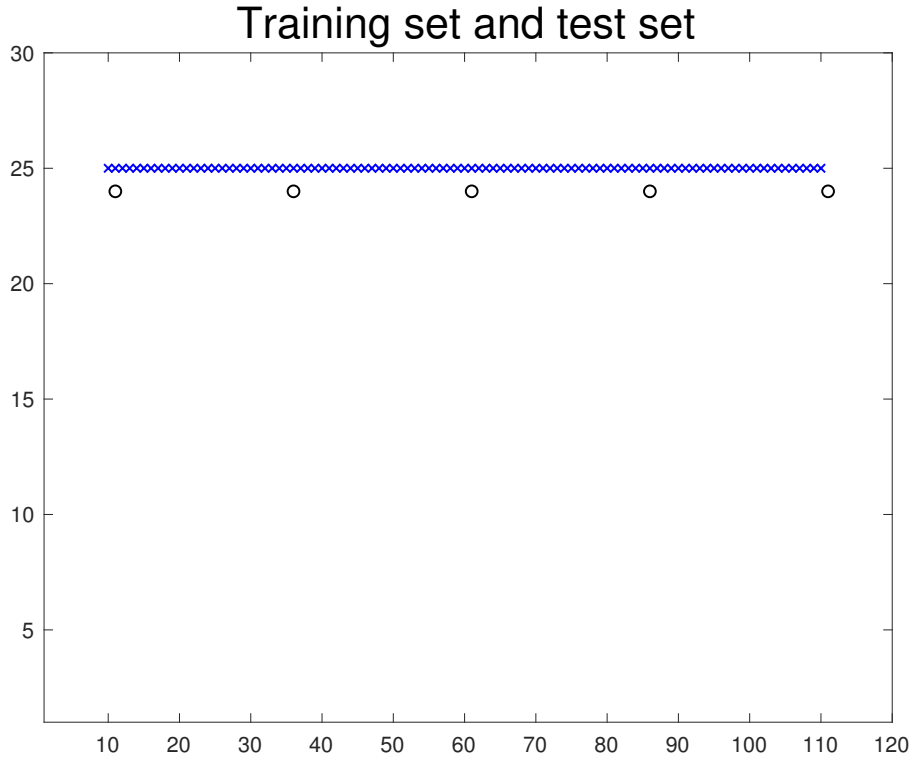


Figure 77: Training and test set. Blue cross: Training set distributed at 101 source locations. Black circle: Test set located below source locations 1, 26, 51, 76, 101

We set up the total iteration number to be 10 for model 1, and 5 for models 2, model 3 from 5.2.1; For each iteration, we do 101 sweeps of data estimation as for 101 sources. As step length control, we consider a technique as described above, counting pixels that change between inside and outside the objects and restricting their number to lie inside an interval with upper and lower bounds as $[0, 10]$. Moreover, the ratio number for reducing and increasing the step length to be $\tau_1 = 0.5, \tau_2 = 1.5$, with initial step lengths $\alpha = 2 \times 10^7$ for model 1, 5×10^6 for model 2 and model 3. Considering that the descent direction is not controlled by the full gradient, and the entire data set is never really considered as a whole, we cannot set up the stopping criterion related to the total data misfit, which actually proves to be more time saving than numerical test in Chapter 4 and Chapter 5. Instead, we apply an error-bound on the norm of source based gradient $\|\boldsymbol{g}_\phi^s\|$.

We will start our validation after the model test ends; in validation, we add in a 5% noise in both the training data and test data; we need to test the data residual of our approximated model and denoised test data to find out whether the evaluation avoids over-fitting successfully.

Our numerical results are presented in the following figures.

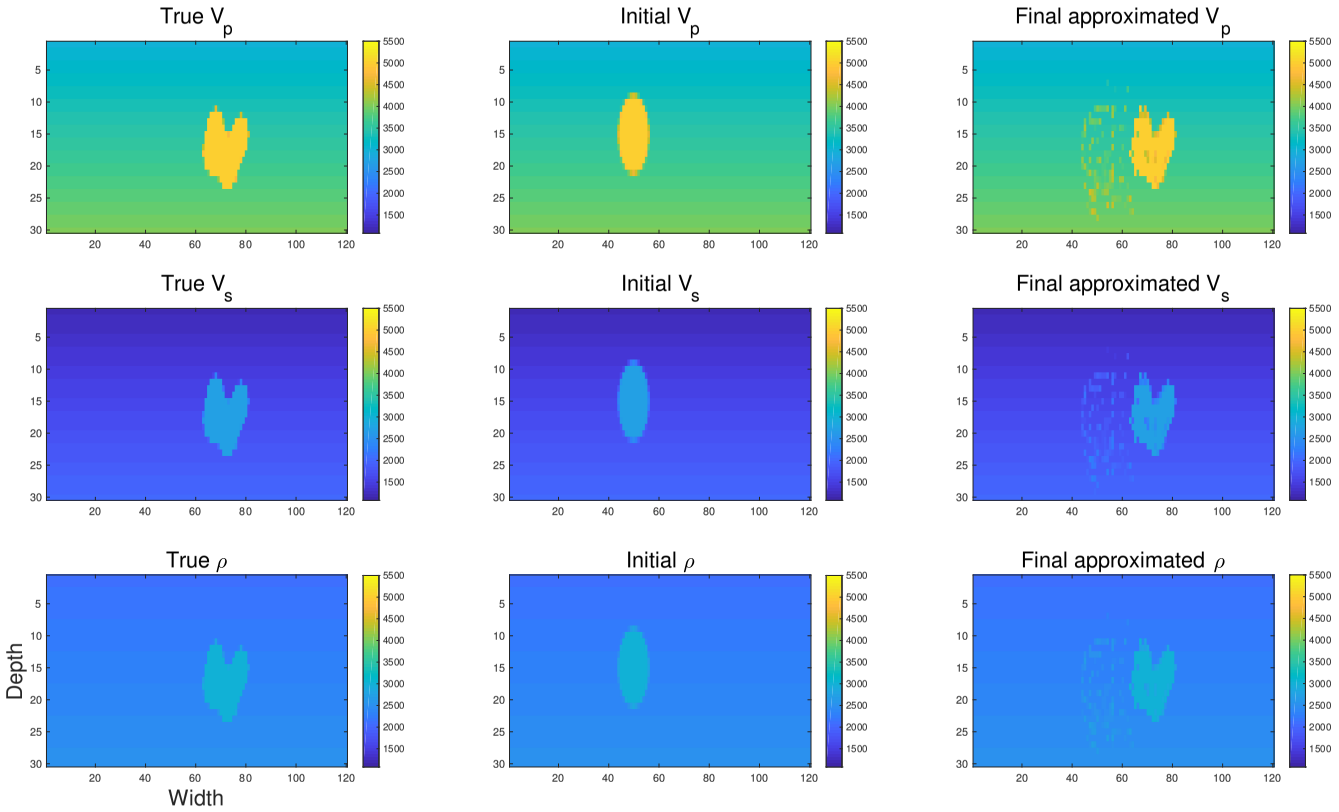


Figure 78: Left to right: True model, initial model and final approximation for model 1, with the parameter from top to bottom V_p, V_s, ρ

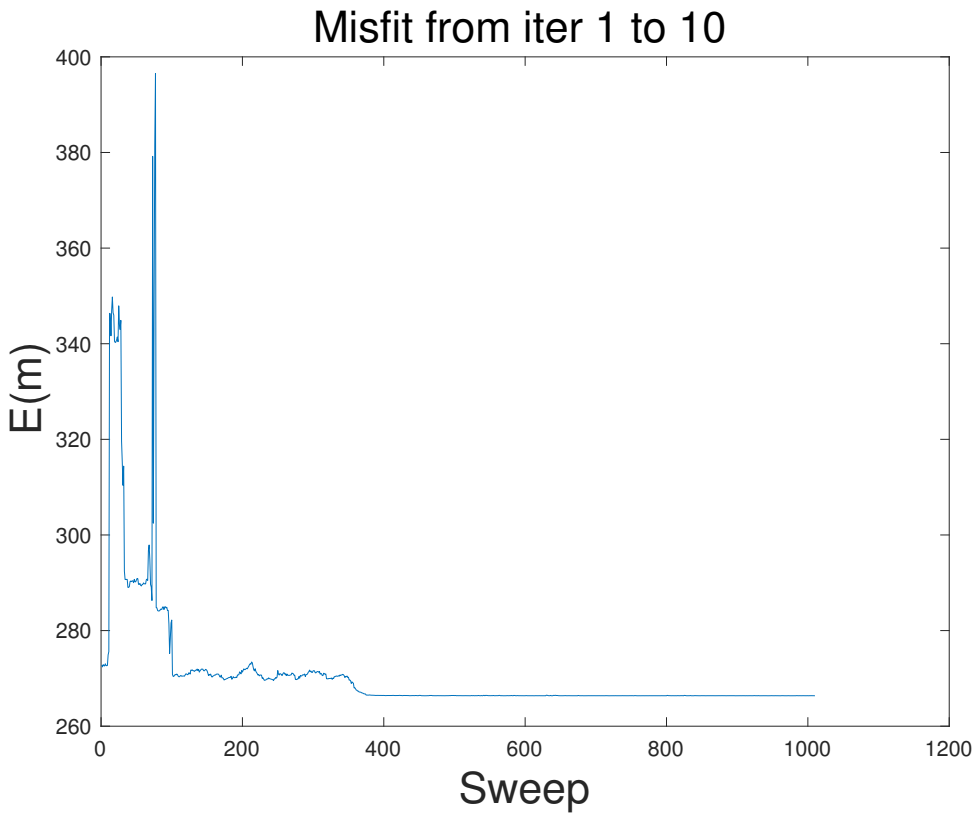


Figure 79: Validation misfit from test sets on Model1: iteration 1 to 10

We can clearly see some rather sharp oscillations of the data misfit, especially in the starting sweeps; this is due to random selection of gradient descent directions. But after a certain burn-in stage, the evolution of the data misfit becomes more stable. We should make a diagnosis of this result, to ensure that our approximation successfully avoids over-fitting. Considering that our data misfit for the validation set includes noisy data, we apply a random sample of generation of data misfit between measured noisy data and validation on the true model estimation; we compute the 'average noise misfit' for a sufficiently large sample. In our test, we select the sample length to be 20. Therefore we compare the 'average noise misfit' to our validated data misfit. Over-fitting happens when validation of data misfit is significantly smaller, which indicates that the model simulation is severely affected by data noise.

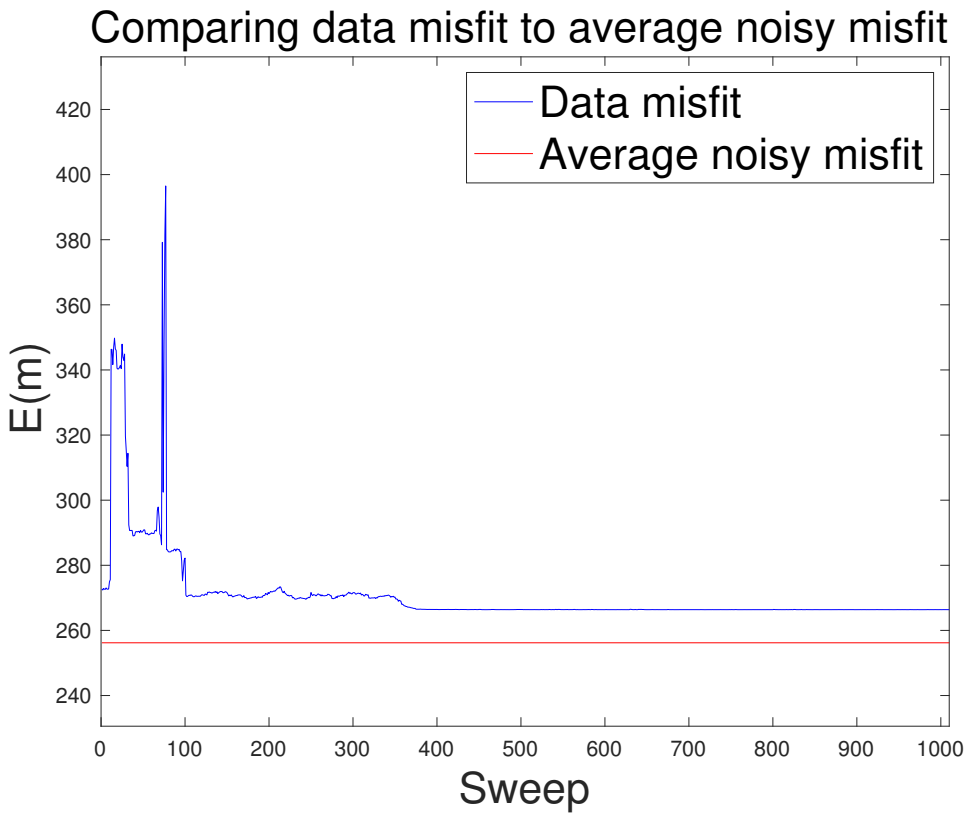


Figure 80: Model1: Comparison of validation data misfit and 'Average' noisy misfit

The result is that during the model estimation, the data misfit is generally close to noisy data misfit; this indicates that our model reconstruction is not affected by data noise. Therefore it indicates that our model simulation successfully avoids the result of over-fitting. For model 2 and model 3, the results show high similarity.

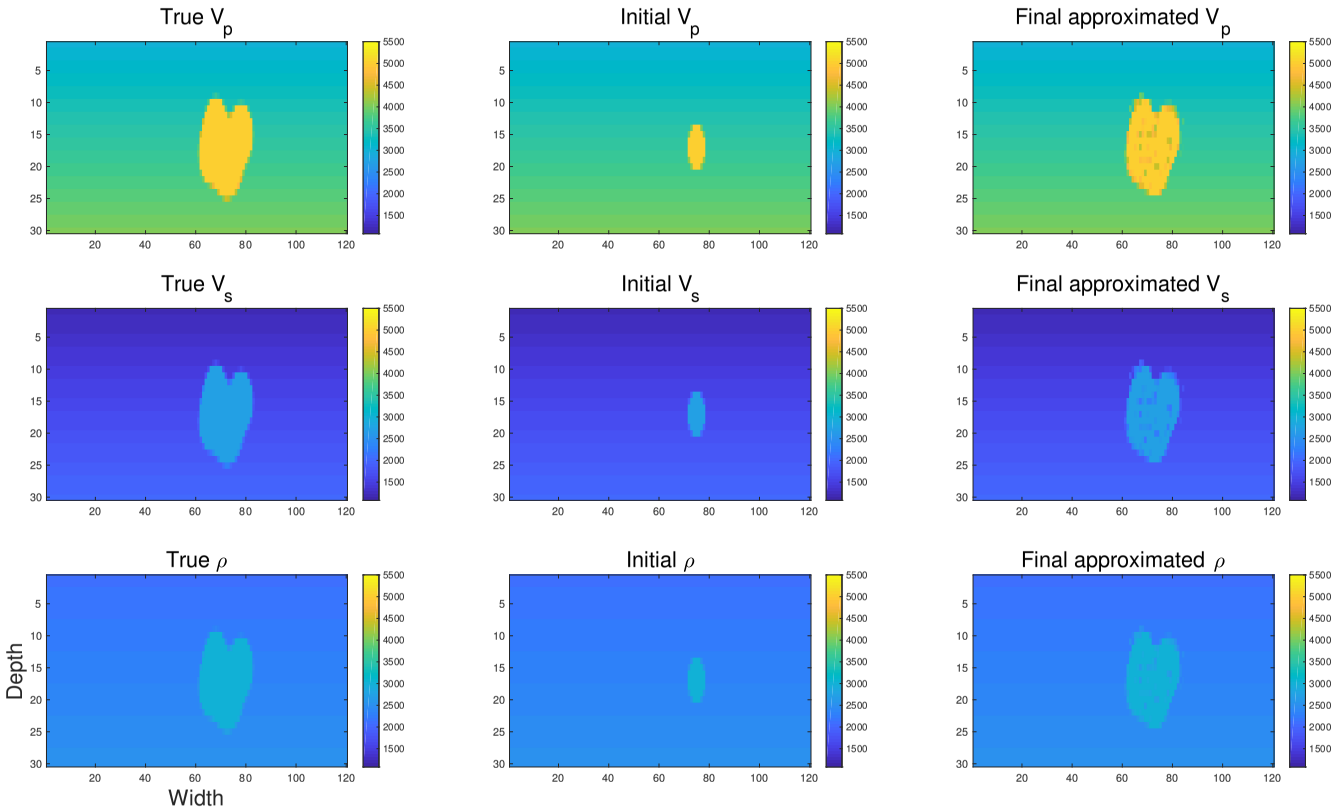


Figure 81: Left to right: True model, initial model and final approximation for model 2, with the parameter from top to bottom V_p, V_s, ρ

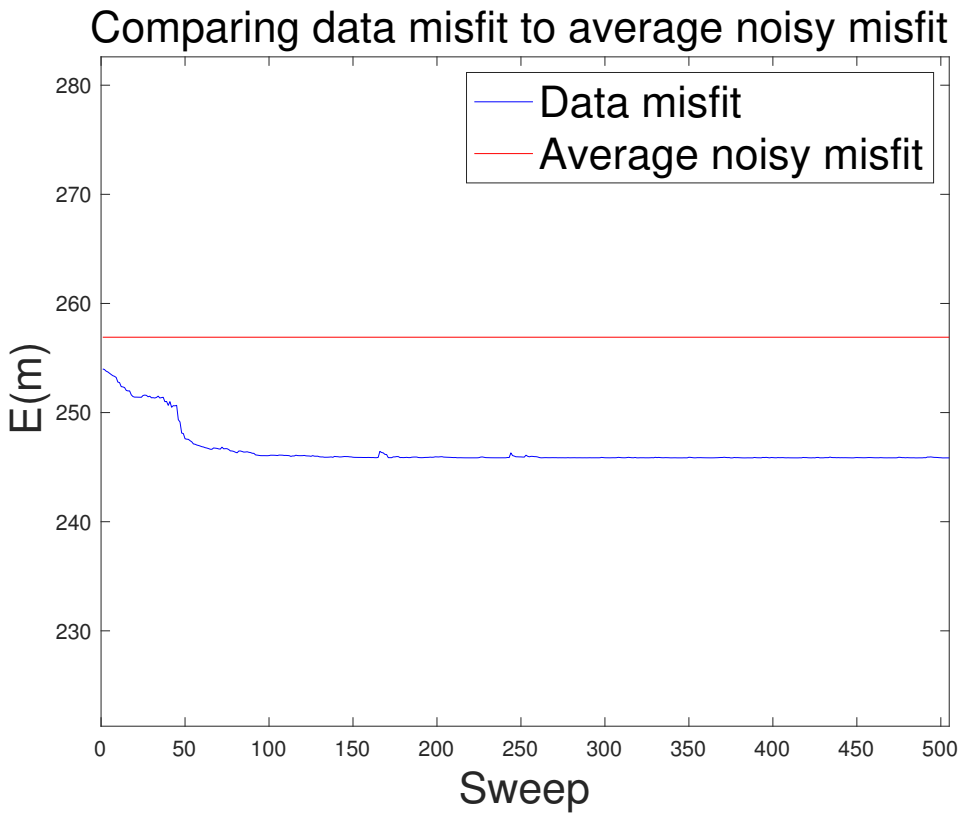


Figure 82: Model2: Comparison of validation data misfit and 'Average' noisy misfit

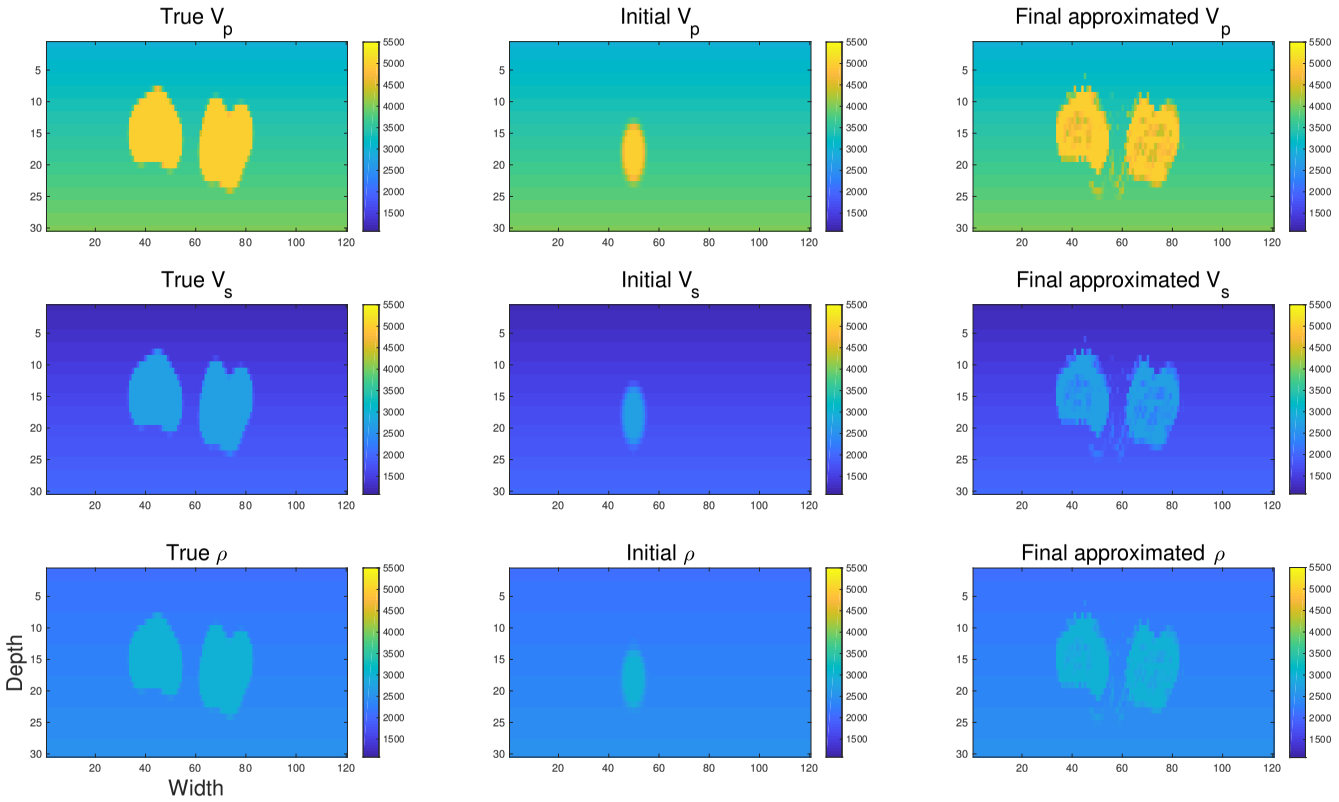


Figure 83: Left to right: True model, initial model and final approximation for model 3, with the parameter from top to bottom V_p, V_s, ρ

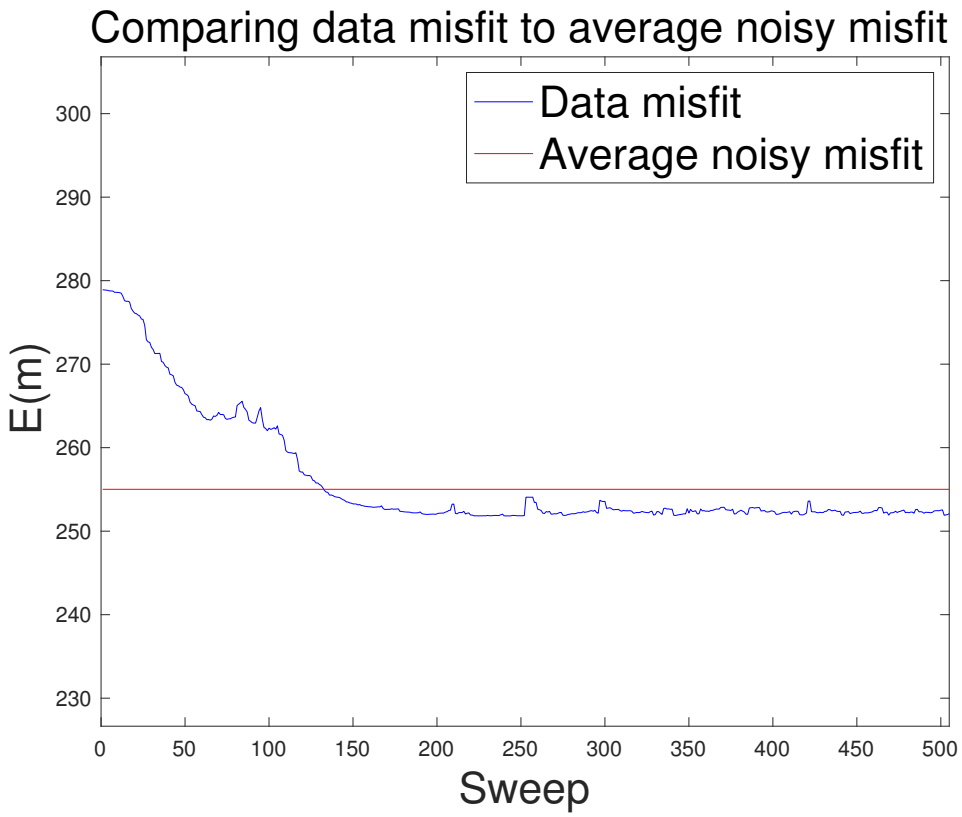


Figure 84: Model3: Comparison of validation data misfit and 'Average' noisy misfit

For these 2 model sets, considering that the initial model is relatively better and the starting step length relatively small, the possible overshooting in the starting sweeps does not occur; the parameter iteration is relatively stable and provides a good approximation.

Our conclusion is that: based on three model tests, we are able to verify that applying the stochastic gradient descent (SGD) method generally delivers a good and stable approximation for the model estimation for a level set based problem, with a large scale of source distribution where traditional line search technique tend to be too time-consuming. Also, our hold-out cross validation approach shows that the data misfit keeps relatively close to 'average noisy misfit' during the model estimation; this indicates that our model test successfully avoids over-fitting.

6.4 Conclusion

In this chapter, we have introduced the basic idea of stochastic gradient method (SGD) and its application to our FWI problem; notice that for the SGD method, we are not required to compute the full gradient in each step of gradient computation and backtracking line search compared to traditional methods. This indicates that the gradient computation potentially is much cheaper than standard gradient methods; therefore we could expect that applying SGD in level set based FWI would be much more time-saving, especially for large-data mining problems for 3D extensions of our technique. However, considering that our problem of waveform inversion often applies with the reconstruction of noisy data, sometimes the data reconstruction will obtain a too precise approximation that also includes the information of noisy data; this comes with the risk of over-fitting. To avoid this, we apply a hold-out cross validation method to monitor our model reconstruction using SGD. We construct a 2D model reconstruction test case with the same model as in Chapter 5, but considering a larger set of 101 equally distributed sources near the top of our survey; we apply a 5% of noise to the data and apply a hold-out method for validation. Our numerical results show that applying SGD in the above models provide a very good approximation overall, and it also successfully avoids over-fitting. From the above test results, we expect that especially in potential future applications to 3D situations, SGD methods appear extremely promising for being applied to large scale FWI problems.

7 Internal value reconstruction

In the level set approaches presented so far we have assumed that the value inside the object is approximately known a priori. However for salt dome reconstruction, there might be some cases where the parameters inside are not quite accurate, indicating that more detailed information needs to be explored. In those cases, we also need to update the parameters inside of the salt domes in order to obtain a reliable estimate.

Therefore, in this section, we will explore the method of internal parameter value reconstruction; for this purpose, we consider that the estimated internal value of the objects are incorrectly chosen in the starting guess. This implies that we need to also estimate this value in our shape reconstruction algorithm.

7.1 Reconstructing internal value

We recall that the level set representation in (63), jointly with the internal parameter, are defined as λ_{obj}, μ_{obj}, ρ_{obj}; we assume also here that the internal parameters are constant, but unknown.

In this chapter, we consider the situation where also λ_{obj}, μ_{obj}, ρ_{obj} are scalar variables that need to be estimated. We recall the parameter gradient $\boldsymbol{g}_m$ (31) and its component with respect to a given source s; using the chain rule and the level set representation (63), we obtain the gradient on internal parameter (neglecting s in the notation) as follows.

$$\begin{aligned}
\boldsymbol{g}_{\lambda_{obs}} &= \boldsymbol{g} \cdot \frac{\partial m}{\partial \lambda_{obs}} = \boldsymbol{g}_\lambda \cdot H(\phi) \\
\boldsymbol{g}_{\mu_{obs}} &= \boldsymbol{g} \cdot \frac{\partial m}{\partial \mu_{obs}} = \boldsymbol{g}_\mu \cdot H(\phi) \\
\boldsymbol{g}_{\rho_{obs}} &= \boldsymbol{g} \cdot \frac{\partial m}{\partial \rho_{obs}} = \boldsymbol{g}_\rho \cdot H(\phi)
\end{aligned}$$

We also apply a stochastic gradient descent type algorithm for the determination of these internal parameters; we define θ to be the step length, and the initial internal values $\lambda^0_{obs}, \mu^0_{obs}, \rho^0_{obs}$ follow the same formulation as in (79). Then the internal update for iteration n is formulated as

$$\begin{aligned}\lambda_{obs}^{n+1} &= \lambda_{obs}^{n} - \theta_n \boldsymbol{g}_{\lambda_{obs}}(\phi_n) \\ \mu_{obs}^{n+1} &= \mu_{obs}^{n} - \theta_n \boldsymbol{g}_{\mu_{obs}}(\phi_n) \\ \rho_{obs}^{n+1} &= \rho_{obs}^{n} - \theta_n \boldsymbol{g}_{\rho_{obs}}(\phi_n)\end{aligned} \tag{90}$$

The above update will be applied inside each sweep after the update of level set function in Algorithm 6. Considering parameter updates, a straight-forward idea is to update the three parameters simultaneously; but the selection of step length θ will be difficult, considering the complexity of the different gradient information and the possibility of 'cross-talk'. Since the value of λ_{obs}, μ_{obs} and ρ_{obs} are scalar, we could use a separate update technique in a bilinear SGD-type case. We apply the implementation that for 3 consecutive updates, we update one parameter each time in a sweep; this way after 3 updates, all of the 3 parameters have been updated once. We need to keep the step length θ small enough to ensure that the information from previous updates are conserved. We also need to choose a good line step for each internal parameter. Notice that the data misfit $E_s(m)$ corresponding to source s can be considered as a quadratic form of λ_{obs}, μ_{obs}, ρ_{obs}; therefore if we update only one parameters of three at a time, this corresponds to a coordinate descent scheme. Also, we could obtain a general formulation of second-order function that $E_s(m_{obs}) = am_{obs}^2 + bm_{obs} + c$, for any $m \in (\lambda, \mu, \rho)$.

One possible practical approach for calculating updates is to use 2nd order interpolation in order to obtain a quadratic regression of the function, and then choose the minimum of this local quadratic approximation to be the next update; however, forming the exact quadratic is still time-consuming. Therefore, we apply a simpler two-way type iteration by comparing the misfit from two directions: for sufficiently small θ, we only need to select the direction where the data misfit is

reduced. Therefore we practically consider the following procedure for update:

Algorithm 7: Internal value update

Initialize θ, parameter $0 < \tau < 1$;
for $i = 0, 1, ...,\ s = 1 : S$ **do**
  Denote sweep s, internal parameter $\lambda_{obs}^{i,s}, \mu_{obs}^{i,s}, \rho_{obs}^{i,s}$, and source based data misfit E_s;
  Choose one parameter m_{temp} from three internal parameters for the update, and the corresponding weight function $w(m_{temp})$;
  Update $m_{temp}^1 = m_{temp} - \theta w(m_{temp}), m_{temp}^2 = m_{temp} + \theta w(m_{temp})$;
  Compute the corresponding data misfit on source s $E_s^1(m_{temp})$, $E_s^2(m_{temp})$; compare with E_s;
  if E_s *is the smallest* **then**
    $\theta = \tau\theta$;
  else
    Accept the minimal of $E_s^1(m_{temp})$, $E_s^2(m_{temp})$, and accept the corresponding update m_{temp}^1 or m_{temp}^2;
  end
  Keep the other two parameters the same;
  Update $\lambda_{obs}^{i,s+1}, \mu_{obs}^{i,s+1}, \rho_{obs}^{i,s+1}$ ($\lambda_{obs}^{i+1,1}, \mu_{obs}^{i+1,1}, \rho_{obs}^{i+1,1}$ if $s = S$);
end

From the above, we obtain the following algorithm for level set based inversion

with internal value update:

Algorithm 8: Level set inversion with internal value update

Initial level set ϕ_0, $m_0 = (\lambda^0_{obs}, \mu^0_{obs}, \rho^0_{obs})$, internal value step length θ, SGD step length α, training set S_1 and test set S_2 ;

for $i = 0 : N$, $s = 1 : |S_1|$ **do**

- Shuffle $\gamma(s)$;
- Forward propagation;
- Backward propagation;
- Compute $\boldsymbol{g}_\phi^{i,\gamma(s)}$, $\boldsymbol{g}_{\lambda_{obs}}^{i,\gamma(s)}(\phi_i), \boldsymbol{g}_{\mu_{obs}}^{i,\gamma(s)}(\phi_i), \boldsymbol{g}_{\rho_{obs}}^{i,\gamma(s)}(\phi_i)$;
- For 3 consecutive terms;
- Select one of three parameters to update from $\lambda^{i,s}_{obs}, \mu^{i,s}_{obs}, \rho^{i,s}_{obs}$; all three parameters should be updated once for each 3 consecutive terms;
- Keep the other two parameter values;
- Choose line search θ_i^s and corresponding descent direction;
- Update $m_{obs} = (\lambda^{i,s+1}_{obs}, \mu^{i,s+1}_{obs}, \rho^{i,s+1}_{obs})$ ($\lambda^{i+1,1}_{obs}, \mu^{i+1,1}_{obs}, \rho^{i+1,1}_{obs}$ if $s = |S_1|$) with Algorithm 7;
- Search step length α_s;
- Update $\phi^{i,s+1} = \phi^{i,s} - \alpha_s \boldsymbol{g}_\phi^{i,\gamma(s)}$;
- (if $s = |S_1|$, $\phi^{i+1,1} = \phi^{i,s} - \alpha_s \boldsymbol{g}_\phi^{i,\gamma(s)}$);
- Update $m_i^{s+1} = m_{back} H(\phi^{i,s+1}) + m_{obs}(1 - H(\phi^{i,s+1})$;
- Verifying stopping criterion;

end

We will apply the above algorithm for internal value reconstruction with a level set evolution applying SGD method.

7.2 Numerical tests

	V_p	V_s	ρ
Background	[3000,4000]	[1200,2000]	[2100,2500]
True internal	5000	2700	3000
Initial internal	5500	3000	2700

When starting the evolution we need to take into account that, during early iterations where the shapes are still far away from the correct one, the internal

value might be updating towards the wrong direction to compensate the misfit caused by inaccurate shape; therefore we set up the maximum iteration number of the total scheme to be 10, but the iteration for the internal parameter value starts not earlier than at iteration 5; We measure the internal value of V_p, V_s and ρ, based on the computed parameterization λ, μ and ρ. Also here, the data are polluted artificially by 5% noise. We use as step size $\Theta = 0.1$, and the weight parameter $w_{V_p} = 500$, $w_{V_s} = 300$, $w_\rho = 300$, defined as the absolute difference of initial and true internal value of 3 parameters. We apply the Sobolev gradient where $\lambda = 1$ (Sobolev gradient coefficient) as to smooth the gradient, so as to obtain a smoother and more stable model reconstruction.

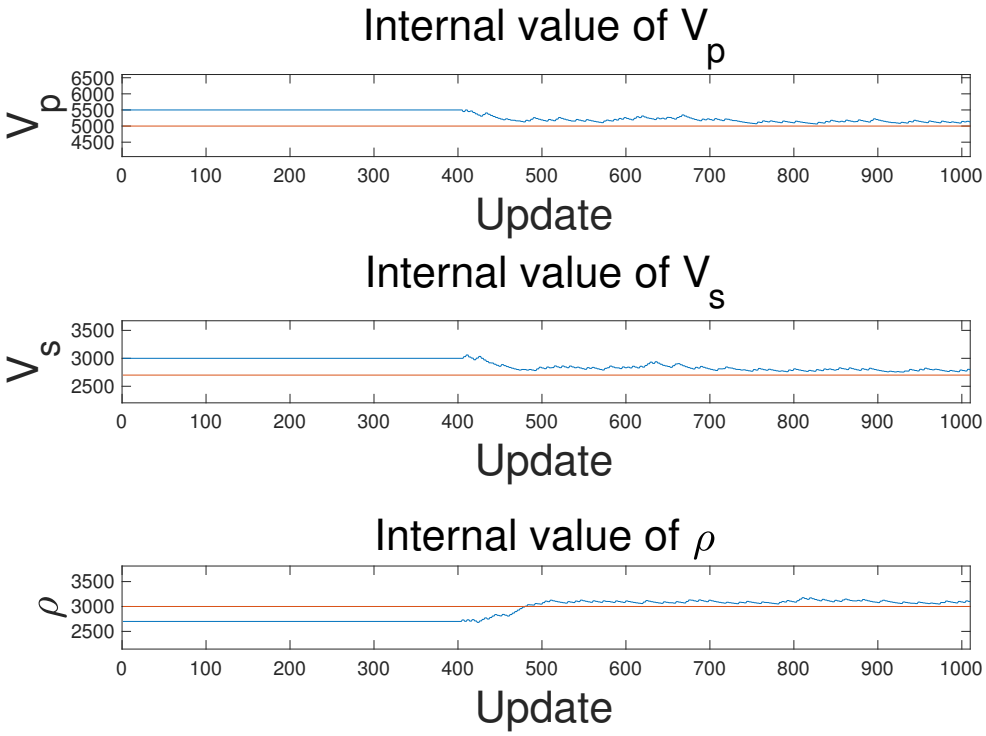


Figure 86: The internal value reconstruction. Top to bottom: the internal value record of V_p, V_s and ρ. Red line records the true value, blue line records the approximated internal value

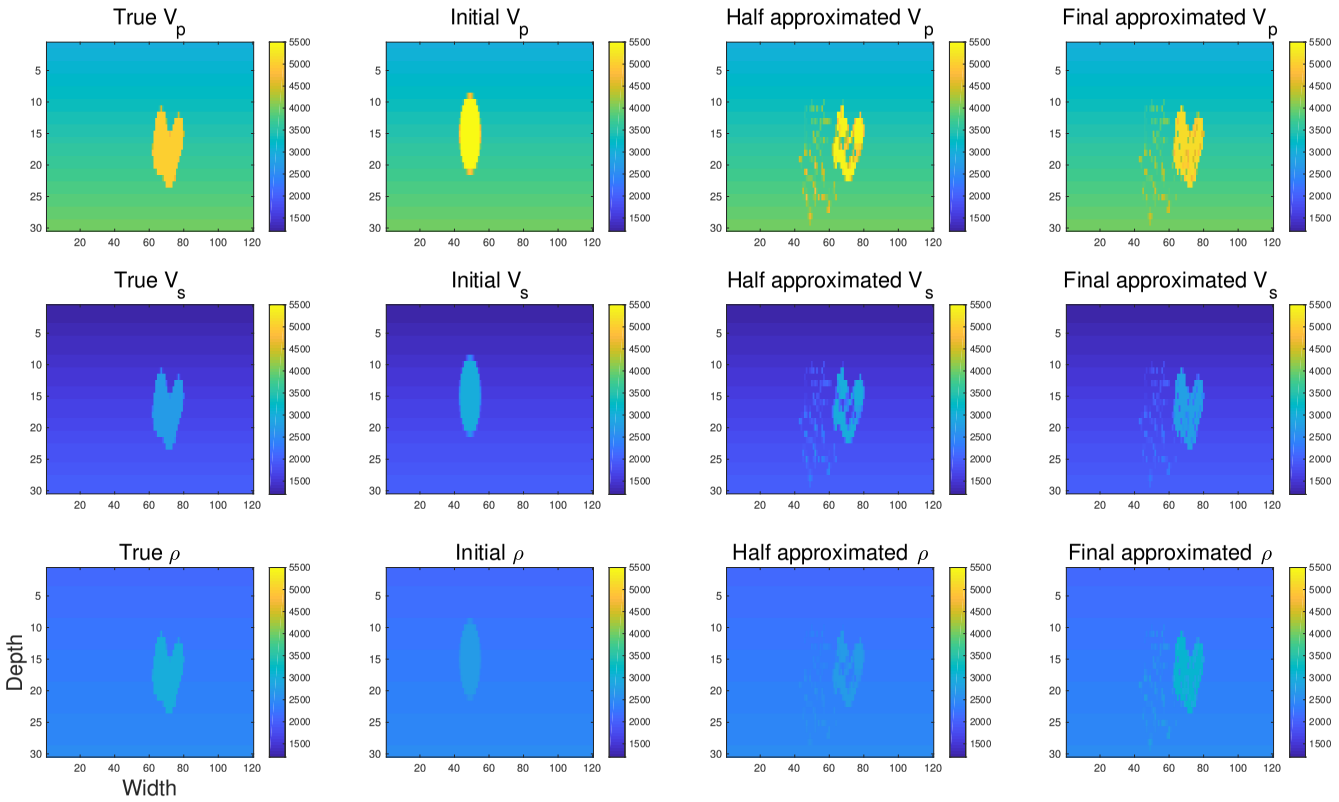


Figure 85: Numerical result for model 1. Top to bottom: parameter V_p, V_s, ρ. Left to right: true model, initial model, iteration 15 where initial value reconstruction starts, final approximation

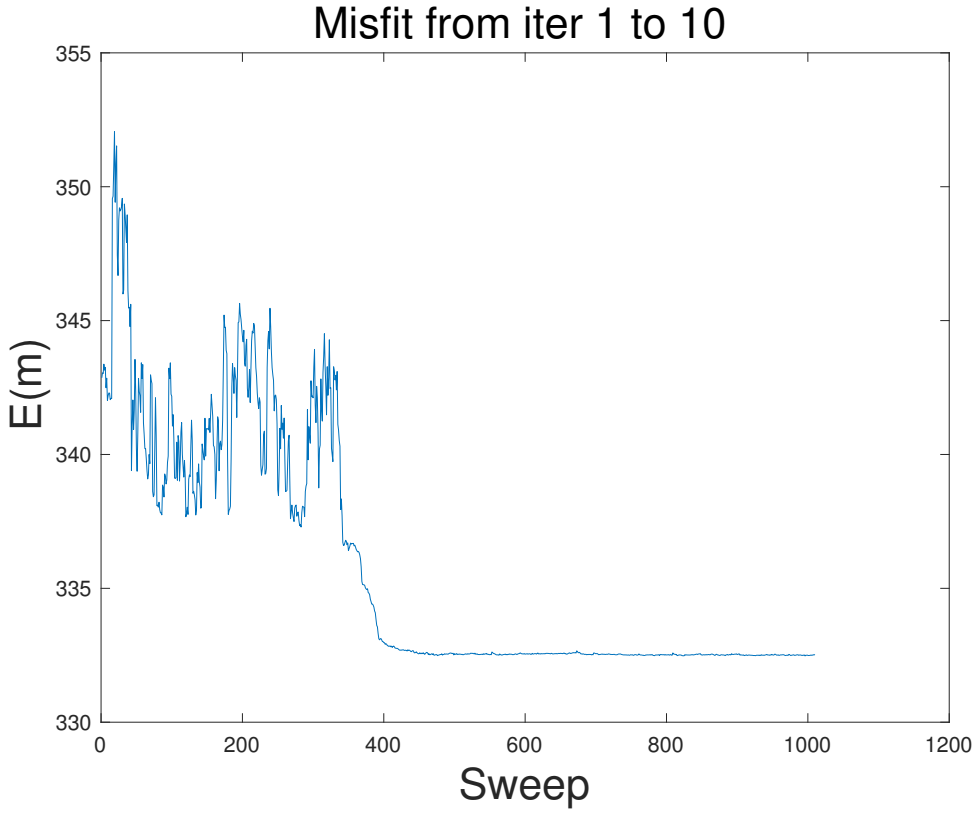


Figure 87: The validation misfit for model 1

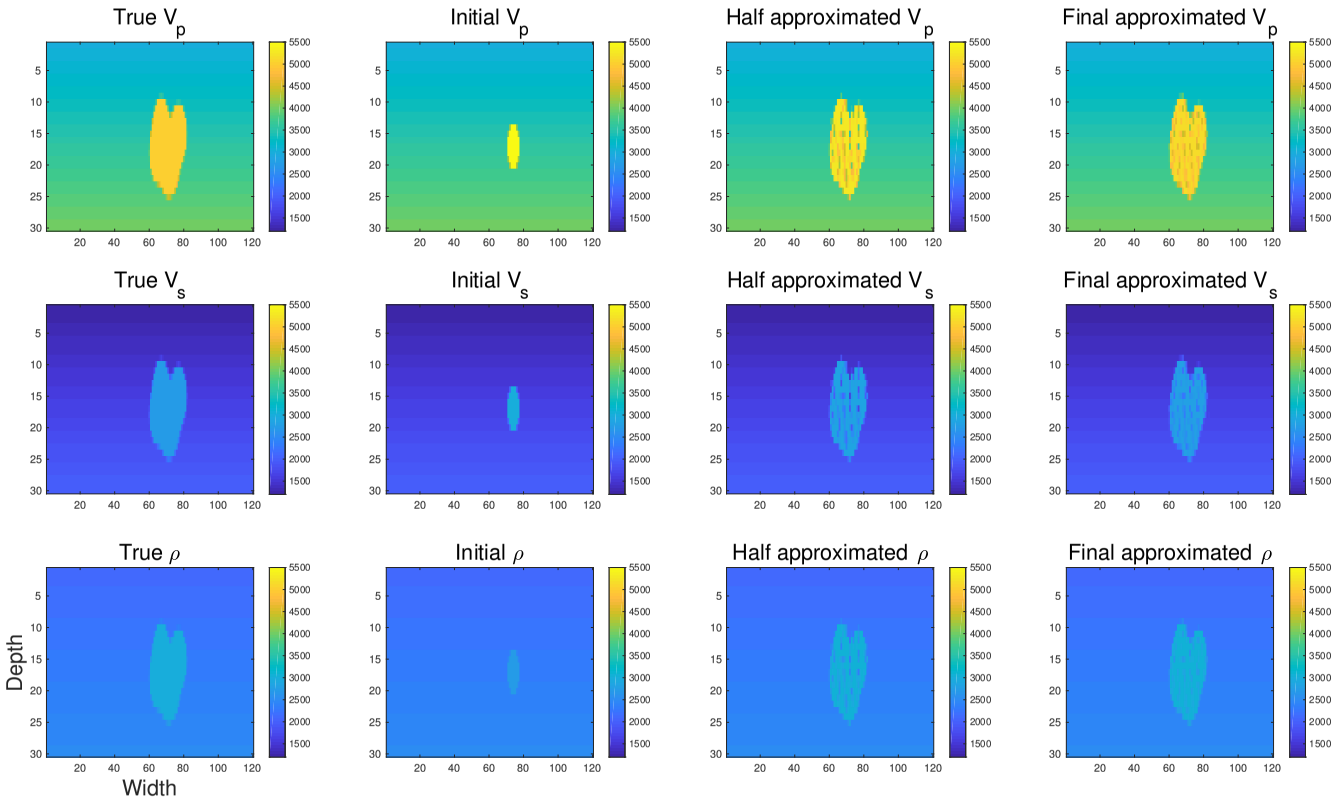


Figure 88: Numerical result for model 2. Top to bottom: parameter V_p, V_s, ρ. Left to right: true model, initial model, iteration 15 where initial value reconstruction starts, final approximation

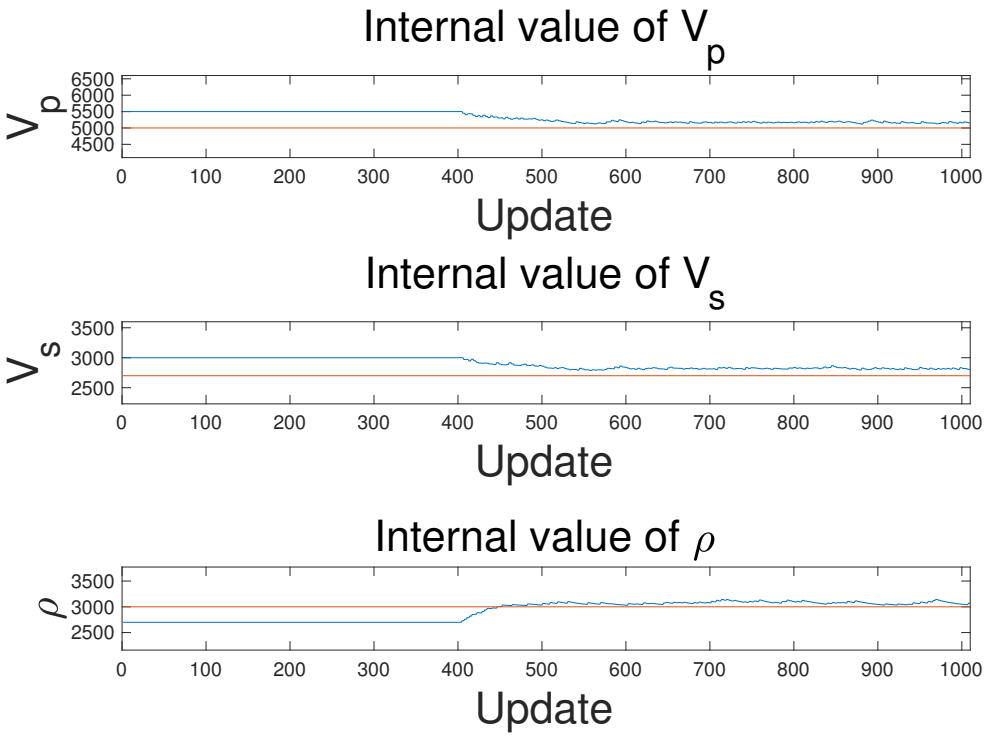


Figure 89: The internal value reconstruction. Top to bottom: the internal value record of V_p, V_s and ρ. Red line records the true value, blue line records the approximated internal value

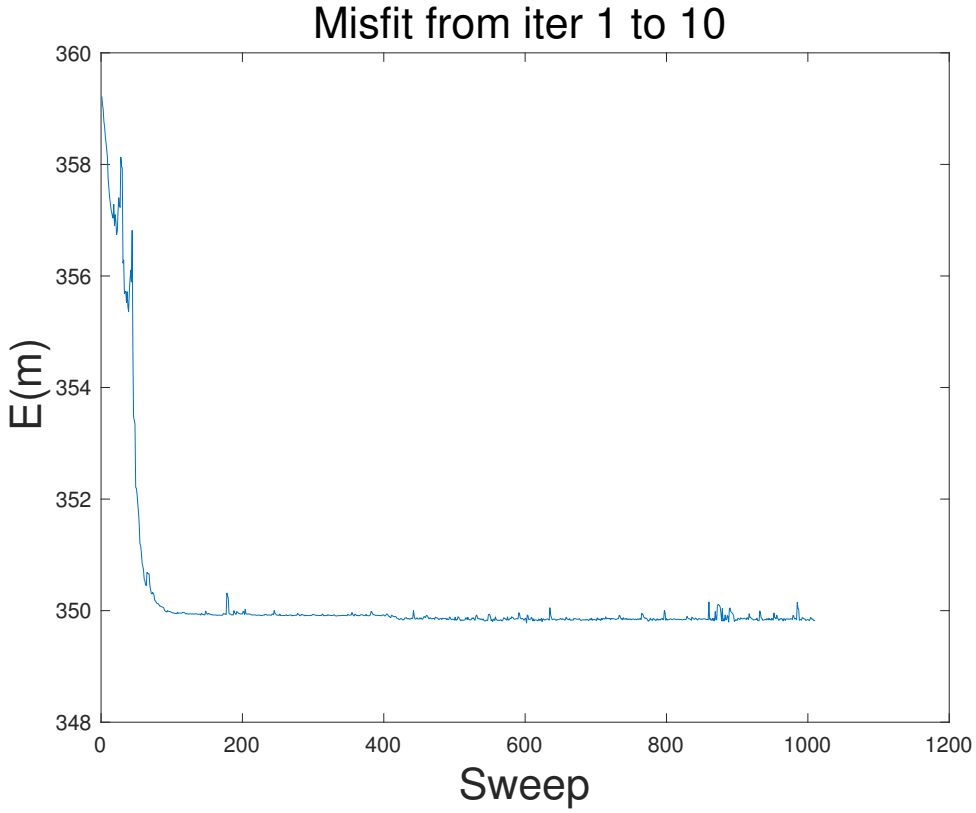


Figure 90: The validation misfit for model 2

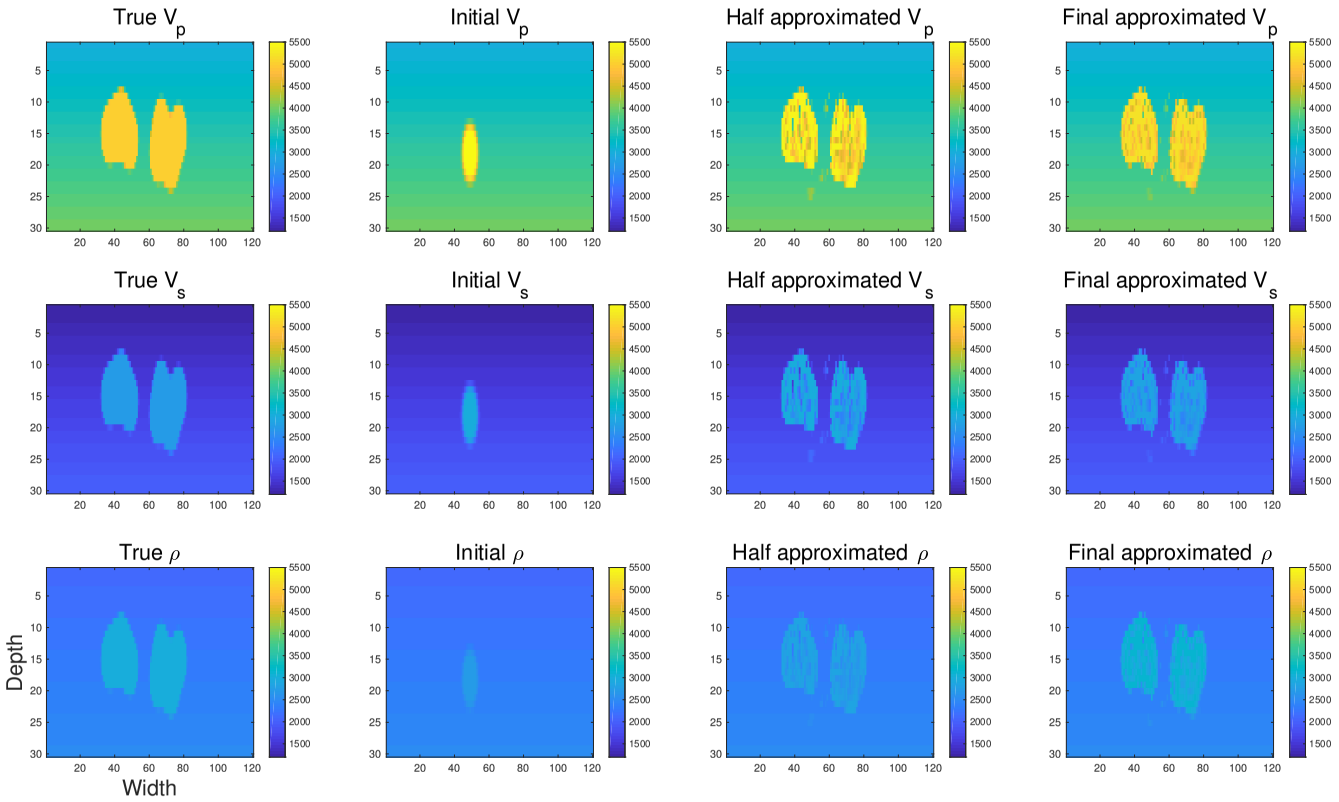


Figure 91: Numerical result for model 3. Top to bottom: parameter V_p, V_s, ρ. Left to right: true model, initial model, iteration 15 where initial value reconstruction starts, final approximation

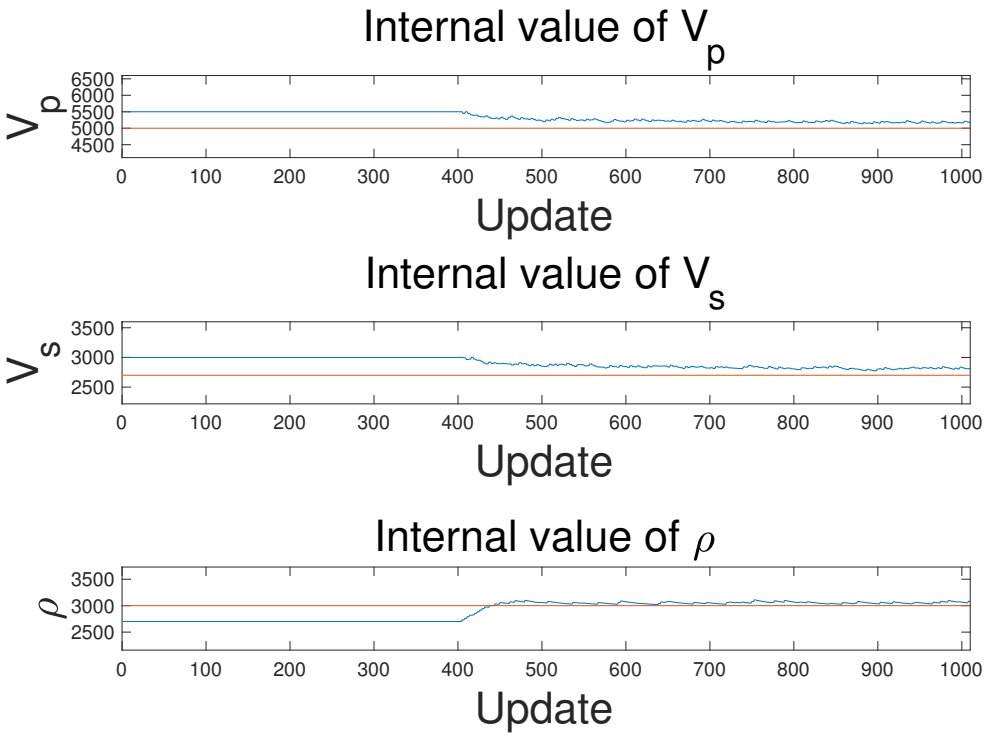


Figure 92: The internal value reconstruction. Top to bottom: the internal value record of V_p, V_s and ρ. Red line records the true value, blue line records the approximated internal value

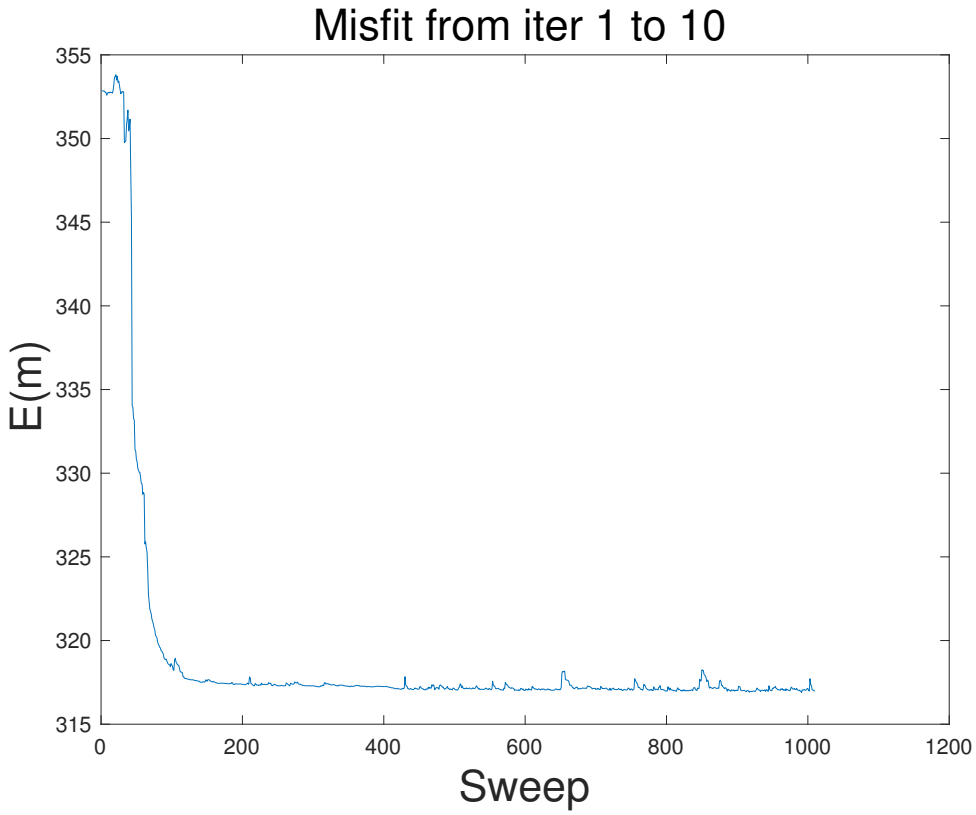


Figure 93: The validation misfit for model 3

We choose the total iteration count to be 10, for a large scale of source terms in Chapter 6; for all tests above, the starting iteration count for internal value reconstruction is set to be iteration 5 which ensures that the shape approximation has obtained an acceptable accuracy when this part of the optimization starts, so that the internal value updates do not try to compensate for the shape error.

The reconstruction show that the SGD type internal value reconstruction produces good results for our test problems. Certainly, our initial guess for the internal value is relatively close to the true internal parameter which however is justified for the intended application of salt dome reconstruction. In theory, we have not yet considered the case where the initial guess is relatively far away from the true object parameters; this may cause a change in waveform structure, which might give rise to local minima or to erratic convergence behaviours. Such local minima are a general difficulty in FWI and are usually avoided by starting with an initial guess constructed by using all available prior information.

7.3 Conclusion

The construction of the internal values shows a good approximation to the true value for all three different tests, regardless of some oscillation during early iterations; as expected, the accuracy of shape reconstruction is not seriously affected by this additional search. This is mainly because that we start evolving the internal value at half time, when the level set evolution has already converged suffiicently and the guess for the shape is already relatively close to the true model; this avoids the possibility that the internal value updates might try to compensate for the misfit caused by incorrect shapes. However, the also means that the internal value reconstruction is expected to work only if the level set reconstruction is successful as well.

Overall, this chapter introduced a complimentary test for internal parameter reconstruction based by SGD in level set FWI adding to the techniques already considered in Chapter 6.

8 Summary and future prospect

In this thesis, we have developed novel shape based techniques for elastic FWI with a special emphasis on seismic imaging; we apply a symmetric hyperbolic scheme for the forward modelling which contains explicitly the pressure term often used for data measurement; in a first-order scheme, our calculation suggests that the adjoint-state operator is skew-symmetric and represents a time-reversal back propagation feature, which adds to its simplicity in its numerical implementation and physical interpretation of the gradient. We apply an adjoint-state method [55] for the gradient computation to obtain a least square solver for our governing model. Next, we have implemented a numerical method called pseudo-spectral method based on a similar scheme used in the open source k-wave package; we modified that code so that it is suited to deal with our specific elastic forward and backward propagation model. We also develop a level set representation for special types of seismic imaging problems such as salt reconstruction, in order to compensate for the limitations that come with more standard pixel based full waveform inversion. Then we introduced a level set based method for parameter representation, so as to ensure that all parameters are topologically identical; we also introduced a narrow-band technique for numerical implementation. We applied a Sobolev gradient as a regularization term to refine our model results. Then, we investigated numerical techniques such as stochastic gradient descent method in order to reduce computational expenses in large scale inversion problems. Instead of calculating the full gradient, it randomly computes the gradient on individual source components for each iteration; our test examples in 2D indicated that this method is expected to be much less time-consuming for large problems, which makes it well suited for future work of large-scale inversion in 3D. In addition, we considered the problem of internal parameter value estimation in order to solve the inverse problem where the internal values of the initial objects are not accurately known;

Most research of full waveform inversion up to now concentrates on simpler waveforms such as acoustic or Helmholtz equation; both of them fail to consider the elasticity of seismic environments in their model, and therefore the corresponding inverse model cannot accurately reconstruct a seismic survey using the generated data. One of the original idea of our thesis is apply a first-order symmetric hyperbolic system, which is suggested from [57], [18], such that it contains

a pressure term that could be directly linked to practical seismic measurements, which is an advantage compared to traditional formulations of second-order wave displacement relationships [26], [37], and also could be linked to other first-order elastic wave propagation research such as [11]. We generated an adjoint-state method following general ideas outlined in R. Plessix [55] to obtain the adjoint-state equation and calculated the gradient involving back propagation; for symmetric hyperbolic systems, the back propagation operator is skew-symmetric to the forward operator, and is applied with a time-reversal initial condition in a 'free-surface' condition. We then computed the gradient specifically for the symmetric hyperbolic form. We constructed a Sobolev regularization technique for smoothing the gradient computation; then we computed the gradient based on the adjoint-state form [55], and then compared the different line search techniques on steepest descent and conjugate gradient techniques for full waveform inversion.

For numerical implementation for elastic wave inversion, we applied a modification of the k-wave toolbox [71] which originally was designed for acoustic wave modeling, but later was adjusted to incorporate an elastic wave scheme; We apply a staggered grid and a pseudo-spectral technique for the spatial derivative of all the variables, which is proved to be numerically more accurate; also, we need to add a PML at the boundary of our domain, so that it simulates a free-surface environment just like in practical applications; our numerical tests show the significant difference of boundary wave performance with and without PML. Furthermore, we need to consider the influence of CFL and grid dispersion that may reflect the convergence and smoothness of wave propagation; we also designed different numerical tests that show the negative affect of grid dispersion when we do not define the source wavelet properly. A numerical test was implemented following an example in a previous elastic forward model package of SOFI2D [8].

Then we designed two different test models simulating a simple seismic environment with objective such as salt domes; we firstly considered the original algorithm, which is pixel based. The challenge in seismic imaging is that, the acquisition of receivers is usually only at the surface of the earth, which means that the wave information we gather from the data only contains the reflection of seismic wave. However, how does such limitation affect the reconstruction?

We designed two different cases for receiver locations, and compared the result in final approximation; there was clear evidence that when we set the receivers at the bottom (which is impossible in practice), the final approximation shows a much better approximation to the true seismic, indicating that most of the wave information is contained at the bottom of the survey, meaning that there is a clear limitation for real world seismic reconstruction. Moreover, applying elastic wave forward model means that we need to estimate three parameters V_p, V_s and ρ instead of V_p in previous research; this may result in a 'cross-talk' phenomenon, meaning that three of the parameters may compensate for each other to reduce the data misfit error. Our numerical implementation also indicated this problem in both test examples, showing that this type of update happens mainly in ρ.

The results of Chapter 4 show the clear disadvantage of pixel based full waveform inversion applied in certain problems such as salt dome reconstruction; [30], [31] provide an alternative method of using level set representations of salt domes to solve the problems with high-contrast, irregular shaped full waveform inversion. Considering level sets, we introduced a level set representation to divide the seismic environment into two layers that describe background and object information.

In Chapter 5, our numerical results show that the level set representation avoids the 'cross-talk' phenomenon, and is able to obtain a more accurate shape reconstruction; we also apply an application of Sobolev gradient which helps us to smooth the boundaries in level set iteration, and therefore shows some good refinement in our test models. Considering that the internal parameter values of the objects are known, the parameters are topologically identical defined by the level set evolution; however, the computation of level set inversion is quite time-consuming, due to the computation of the full gradient, repeated wave propagation implementation in line search, for calculation of data misfit; this makes it extremely difficult to be applied for a larger scale problem, especially in 3D.

Therefore, we considered a stochastic gradient descent (SGD) method in Chapter 6, as a representative line search technique, which avoids the expensive calculation of the full gradient, and the data misfit computation during the traditional backtrack line search technique. However, this also came with the challenge that finding the suitable line step would be challenging; instead we

considered a pixel-count procedure, monitoring the pixel change in object domains before and after one iteration, so that level set updates for each source would not deteriorate the previously obtained shape information.

We designed our 2D example test based on the same model test in Chapter 5, and chose a relatively large source set of 101 sources equally distributed on the surface line, comparing to the numerical test with 21 sources in Chapter 4 and Chapter 5. Our results demonstrated that the numerical implementation of SGD was much less time-consuming than using traditional line search techniques, and also able to obtain a good accuracy; moreover, we applied a hold-out validation method as to compare the validation data to noisy data, so as to confirm that our test model successfully avoids over-fitting. The results indicated that our test of SGD method was considered a successful method for level set based full waveform inversion, and could be extended to a 3D application for addressing large-scale problems.

In Chapter 7, we introduced a complimentary test with internal parameter value reconstruction, where our initial guess for the internal value was not far away from the true value; our approach applied an internal value evolution and level set evolution together. To solve the problem where the internal value might compensate for inaccurate level set evolution, we started our internal value reconstruction relatively late, to make sure that there would be no side affect from compensation. Our test results showed that due to the small difference in initial and true value, the internal value reconstruction worked fairly well; this also helped us to expand our model application into a more general case of tracking the correct parameter value of salt dome.

As possible future research pathway, we would like to mention expanding our elastic waveform inversion algorithm to a more realistic scenario. Our research introduces a relatively novel elastic waveform inversion technique to the community that is applied in salt dome reconstruction; we apply level set representation, Sobolev gradient, and stochastic gradient descent as numerical approaches to solve the 'cross-talk' problem, and overcome the limit in reflective wave information, reduce the time-cost for large-scale approximation. Numerical results of our model demonstrate that our algorithm provides a good final approximation; but so far numerical results have been obtained only in a 2D domain. We should consider applying our model to a more complicated 3D environment, with larger

scales and more complicated background and object information. In addition, we may want to investigate in more details different regularization techniques. In our case, we only consider the Sobolev gradient as regularization, but there are also other regularization techniques such as DRLSE [39], Mumford-Shah [45] functionals that have been applied successfully in other level set applications of image segmentation; but could they be applied in elastic wave inversion?

Finally, we have observed that level set techniques for seismic imaging have become more and more popular these days; and elastic full wave inversion, although in comparison less investigated due to its complexity to acoustic and Helmholtz equation, is widely known to be more accurate in describing seismic wave information. This thesis is only a starting point of level set techniques in elastic wave inversion, but the obtained and presented results in 2D look promising and show a good example of combining the ideas of mathematical contributions from different fields, therefore it could be extended to a 3D case for real application. Let us look forward to see more work on this in the near future.

9 Appendices

9.1 Appendix 1. Derivation of wave velocity from Lamé parameters

In elastodynamics, the wave displacement $\boldsymbol{u}$ can be decomposed in the form of P and S wave potentials, known as Helmholtz decomposition. We state that [35]

$$\boldsymbol{u} = \nabla\boldsymbol{\phi} + \nabla \times \boldsymbol{\psi} \tag{91}$$

where $\boldsymbol{\phi}$ is the potential for the P-wave, and $\boldsymbol{\psi}$ is the potential for the S-wave. Also, the shear potential satisfies the condition that $\nabla \cdot \boldsymbol{\psi} = 0$. For λ and μ homogeneous, we form a vectorised elastic scheme [35]

$$\rho\partial_{tt}\boldsymbol{u} = (\lambda + 2\mu)\nabla\nabla \cdot \boldsymbol{u} - \mu(\nabla \times \nabla \times \boldsymbol{u}) \tag{92}$$

Therefore, with (91) it follows

$$\begin{aligned}\rho\partial_{tt}(\nabla\phi + \nabla \times \psi) &= (\lambda + 2\mu)\nabla\nabla \cdot (\nabla\phi + \nabla \times \psi) - \mu(\nabla \times \nabla \times (\nabla\phi + \nabla \times \psi)) \\ &= (\lambda + 2\mu)\nabla(\Delta\phi + \nabla \cdot \nabla \times \psi) - \mu(\nabla \times (\nabla \times \nabla\phi + \nabla \times \nabla \times \psi))\end{aligned}$$

We consider that $\Delta\phi = \nabla \cdot \nabla\phi$ is the Laplacian operator. Applying the properties of gradient, divergence and curl, we obtain that

$$\begin{aligned}\nabla \times (\nabla\phi) &= 0 \\ \nabla \cdot (\nabla \times \psi) &= 0 \\ \nabla \times \nabla \times \psi &= \nabla(\nabla \cdot \psi) - \Delta\psi \\ &= -\Delta\psi\end{aligned}$$

Substituting the above into (92), we obtain two separate equations on P and S potential, namely

$$\begin{aligned}\partial_{tt}\boldsymbol{\phi} &= V_p^2\Delta\boldsymbol{\phi} \\ \partial_{tt}\boldsymbol{\psi} &= V_s^2\Delta\boldsymbol{\psi}\end{aligned}$$

Therefore P and S wave velocities are computed as

$$\begin{aligned} V_p &= \sqrt{\frac{\lambda + 2\mu}{\rho}} \\ V_s &= \sqrt{\frac{\mu}{\rho}} \end{aligned} \tag{93}$$

This is the derivation of the wave velocities V_p, V_s from the Lamé parameters.

9.2 Appendix 2: A heat-kernel solution

Noted that solving the Sobolev regularization equation (42) requires an inversion of a Laplacian-related operator; we will introduce a heat kernel solution that approximates the above solution. We assume that Sobolev gradient method is a time-depending procedure in pseudo evolution time, and define the initial function Φ, and the expected approximation Ψ that satisfies $\Psi = (I - \gamma\Delta)^{-1}(\Phi)$. We consider a heat kernel where we implement the function ψ on the pseudo-time line $[0, \tau]$, where the heat-type equation states

$$\begin{aligned} \psi_t + (I - \gamma\Delta)\psi - \Phi = 0, \quad t \in [0, \tau] \\ \psi(0) = \Phi \end{aligned} \tag{94}$$

The above equation asks that $\psi(\infty) = \Psi$, therefore we could use a finite-difference scheme to approximate the expected approximation. Numerical discretization provides that

$$\begin{aligned} \psi^0 &= \Phi \\ \psi^{n+1} &= \psi^n - \theta(\psi^n - \gamma\Delta\psi^n - \Phi) \end{aligned} \tag{95}$$

where θ is considered the pesudo-time step equivalent to Δt. Recall the gradient and Sobolev gradient $\boldsymbol{g}_m$, $\boldsymbol{g}^{H_1,\gamma}$ in (42); we can apply the same technique by defining that $\Phi = \boldsymbol{g}_m$, and $\Psi = \boldsymbol{g}^{H_1,\gamma}$. We obtain the practical algorithm for

Sobolev gradient as

Algorithm 9: Sobolev transform algorithm

Initialize gradient $\boldsymbol{g}_m$, γ, ψ_0, pesudo-time step length θ, pseudo iteration number N;

for $i = 0 : N$ **do**

  $\psi^{n+1} = \psi^n - \theta(\psi^n - \gamma\Delta\psi^n - \boldsymbol{g}_m)$;

end

Accept Sobolev gradient that $\boldsymbol{g}^{H_1,\gamma} = \psi^N$;

This should apply a quick numerical scheme to approximate Sobolev gradient.

References

[1] D. Adalsteinsson, J. Sethian, *A Fast Level Set Method for Propagating Interfaces*, Journal of Computational Physics, Volume 118, Issue 2, May, 1995, Page(s) 269-277

[2] R. Adams, J. Fournier, *Sobolev Spaces*, Volume 140, 2nd Edition, Academic Press, June, 2003

[3] A. Aghasi, M. Kilmer, and E. Miller, *Parametric Level Set Methods for Inverse Problems*, SIAM Journal on Imaging Sciences, Volume 4, Issue 2, June, 2011, Page(s) 618-650

[4] A. Asnaashari, R. Brossier, S. Garambois, F. Audebert, P. Thore and J. Virieux, *Regularized seismic full waveform inversion with prior model information*, Geophysics, Volume 78, Issue 2, March, 2013, R25-R36

[5] A. Bedford, D. Drumheller, *Introduction to Elastic Wave Propagation*, 1st Edition, John Wiler & Sons, January, 1994

[6] J. Berenger, *A Perfectly Matched Layer for the Absorption of Electromagnetic Waves*, Journal of Computational Physics, Volume 114, Issue 2, October, 1994, Page(s) 185-200

[7] K. Blazek, C. Stolk, and W. Symes, *A mathematical framework for inverse wave problems in heterogeneous media*, Inverse Problems, Volume 29, No. 6, April, 2013, 065001

[8] T. Bohlen, D. De Nil, D. Köhn and S. Jetschny, *SOFI2D: seismic modelling with finite differences 2D elastic and viscoelastic version: User guide*, Karlsruhe Institute of Technology (KIT) , Karlsruhe, November 29, 2016

[9] L. Bottou, and O. Bousquet, *The Tradeoffs of Large Scale Learning*, Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007 Curran Associates Inc. 2007.

[10] M. Burger, *A framework for the construction of level set methods for shape optimization and reconstruction*, Interfaces and Free Boundaries, Volume 5, Issue 3, 2003, Page(s) 301-329

[11] B. Cox, B. Treeby, D, Rohrbach and J. Jaros, *Modelling Elastic Wave Propagation Using the k-wave MATLAB Toolbox*, IEEE International Ultrasonics Symposium, September 3-6, 2014, Page(s) 146-149

[12] A. Cezaro, M. Haltmeier, A. Leitão, and O. Scherzer, *On Steepest-Descent-Kaczmarz methods for regularizing systems of nonlinear ill-posed equations*, Applied Mathematics and Computation, Volume 202, Issue 2, August, 2008, Page(s) 596-607

[13] K. Chen, Q. Li, and J. Liu, *Online learning in optical tomography: a stochastic approach*, Inverse Problems, Volume 34, No. 7, May, 2018, 075010

[14] T. Chan, L. Vese, *Active Contours Without Edges*, IEEE Transactions on Image Processing, Volume 10, Issue 2, February, 2001, Page(s) 266-277

[15] S. David, S. Shwartz, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, July, 2014

[16] O. Dorn, R. Villegas, *A level set method for 3D low frequency electromagnetic imaging with applications in geophysical prospecting*, Preceedings in Applied Mathematics and Mechanics, Volume 7, Issue 1, December, 2007, Page(s): 2150023-2150024

[17] O. Dorn, *Shape reconstruction in 3D low-frequency electromagnetic induction tomography using level sets and adjoint fields*, IEEE, Antennas and Propagation Society International Symposium, Volume 2, June, 2002, Page(s): 777-780

[18] O. Dorn, *Time-reversal and the adjoint imaging method with an application in telecommunication*, Inverse problem and imaging, Volume 1943 of the series Lecture Notes in Mathematics, 2008, Page(s) 135-170

[19] O. Dorn, D. Lesselier, *Level set methods for inverse scattering: Topical review*, Inverse Problems, Volume 22, No. 4, August, 2006, R67

[20] O. Dorn, A. Hiles, *A level set method for magnetic induction tomography of 3D boxes and containers*, IOS press, Volume 43, Electromagnetic Non-Destructive Evaluation(XXI), 2018, Page(s) 33-40

[21] O. Dorn, R. Villegas, *History Matching of Petroleum Reservoirs Using a Level Set Technique*, Inverse Problems, Volume 24, No. 3, June, 2008, 035015

[22] E. Engel, R. Dreizler, *Density Functional Theory: An Advanced Course*, Springer, Berlin, Heidelberg, 2011

[23] E. Esser, L. Guasch, T.van Leeuwen, A. Aravkin, and F. Hermann, *Total-variation regularization strategies in full-waveform inversion*, SIAM Journal on Imaging Sciences, Volume 11, No. 1, February, 2018, Page(s) 376-406

[24] E. Faccioli, F. Maggio, R. Paolucci, and A. Quarteroni, *2D and 3D elastic wave propagation by a pseudo-spectral domain decomposition method*, Journal of Seismology, Volume 1, Issue 3, November, 1997, Page(s) 237-251

[25] K. Firouzi, B. Cox, B. Treeby, and N. Saffari, *A first-order k-space model for elastic wave propagation in heterogeneous media*, The Journal of Acoustical Society of America, Volume 132, Issue 3, September, 2012, Page(s): 1271-1283

[26] A. Fitchner, *Full Seismic Waveform Modelling and Inversion*, Springer, Advances in Geophysical and Environmental Mechanics and Mathematics, 2011

[27] B. Fornberg, *The pseudospectral method: Comparisons with finite differences for the elastic wave equation*, Geophysics, Volume 52, No. 4, April, 1987, Page(s) 483-501

[28] F. Gibou, R. Fedkiw, S. Osher, *A review of level-set methods and some recent applications*, Journal of Computational Physics, Volume 353, January, 2018, Page(s) 82-109

[29] R. Gordon, R. Bendar, and G. Herman, *Algebraic Reconstruction Techniques (ART) for three-dimensional electron microscopy and X-ray photography*, Journal of Theoretical Biology, Volume 29, Issue 3, December, 1970, Page(s) 471-476, IN1-IN2, 477-481

[30] Z. Guo, M. de Hoop, *Shape optimization in full waveform inversion with sparse blocky model representation*, GMIG Report 12-12, Proceedings of the Project Review, Geo-Mathematical Imaging Group (Purdue University, West Lafayette IN), Volume 1, 2012, Page(s) 189-208

[31] Z. Guo, M. de Hoop, *Shape optimization and level set method in full waveform inversion with 3D body reconstruction*, SEG Technical Program Expanded Abstracts 2013, September, 2013, Page(s) 1079-1083

[32] M. Hanke, W. Niethammer, *On the acceleration of Kaczmarz's method for inconsistent linear systems*, Linear Algebra and its Applications, Volume 130, March, 1990, Page(s) 83-98

[33] B. Hustedt, S. Operto and J. Virieux, *Mixed-grid and staggered-grid finite-difference methods for frequency-domain acoustic wave modelling*, Geophysical Journal International, Volume 157, Issue 3, June, 2004, Page(s) 1269-1296

[34] T. Hughes, J. Marsden, *Classical elastodynamics as a linear symmetric hyperbolic system*, Journal of Elasticity, Volume 8, No. 1, January, 1978, Page(s) 97-110

[35] H. Igel, Lecture, *The elastic wave equation*, `https://www.geophysik.uni-muenchen.de/~igel/Lectures/Sedi/sedi_weq.pdf`

[36] A. Kadu, T. Van Leeuwen, and W. Mulder, *A parametric level-set approach for seismic full-waveform inversion*, SEG Technical Program Expanded Abstracts, October, 2016, Page(s) 1146-1150

[37] D. Köhn, *Time Domain 2D Elastic Full Waveform Tomography*, PhD thesis, University of Kiel, March, 2011

[38] A. Leitão, B. Svaite, *On projective Landweber-Kaczmarz methods for solving systems of nonlinear ill-posed equations*, Inverse Problems, Volume 32, No. 2, February, 2016

[39] C. Li, C. Xu, C. Gui, and M. Fox, *Distance Regularized Level Set Evolution and Its Application to Image Segmentation*, IEEE transactions on image processing, Volume 19, No. 12, December, 2010, Page(s) 3243-3254

[40] E. Luo, H. Kreiss, *Pseudospectral vs. Finite Difference Methods for Initial Value Problems with Discontinuous Coefficient*, SIAM Journal on Scientific Computing, Volume 20, Issue 1, August, 1998, Page(s) 148-163

[41] T. van Leeuwen, F. Hermann, *A penalty method for PDE-constrained optimization in inverse problems*, Inverse Problems, Volume 32, No. 1, January, 2016, 015007

[42] T. van Leeuwen, A. Aravkin, and F. Herrmann, *Seismic Waveform Inversion by Stochastic Optimization*, International Journal of Geophysics, Vol 2011, Article ID 689041, 18 pages, 2011

[43] W. Lewis, B. Starr, and D. Vigh, *A level set approach to salt geometry inversion in full waveform inversion*, SEG Technical Program Expanded Abstracts 2012, November, 2012, Page(s) 1-5

[44] L. Métivier, R. Brossier, S. Operto and J. Virieux, *Full Waveform Inversion and the Truncated Newton Method*, SIAM review, Volume 59, Issue 1, 2017, Page(s) 153-195.

[45] M. Marcolli and D. Tsao, *Image Segmentation: the Mumford-Shah functional*, `http://www.its.caltech.edu/~matilde/MumfordShahSlides.pdf`, 2017

[46] D. Needell, N. Srebro, and R. Ward, *Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz algorithm*, Mathematical Programming, Springer, Volume 155, Issue 1-2, January, 2016, Page(s) 549-573

[47] J. Nocedal, S. Wright, *Numerical Optimization*, Springer Series in Operations Research, Second edition, August, 2006

[48] J. Neuberger, *Steepest descent and differential equations*, Journal of the Mathematical Society of Japan, Volume 37, No. 2, 1985, Page(s) 187-195

[49] J. Neuberger, *Sobolev Gradients and Differential Equations*, Springer, 1st Edition, 1997

[50] S. Operto , Y. Gholami , V. Prieux , A. Ribodetti , R. Brossier , L. Metivier and J. Virieux, *A guided tour of multiparameter full-waveform inversion with*

multicomponent data: From theory to practice, The Leading Edge, Volume 32, Issue 9, September, 2013, Page(s) 1040-1054

[51] S. Osher, R. Fedkiw, *Level Set Methods: An Overview and Some Recent Results*, Journal of Computational Physics, Volume 169, Issue 2, May, 2001, Page(s) 463-502

[52] S. Osher, J. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, Journal of Computational Physics, Volume 79, Issue 1, November, 1988, Page(s) 12-49

[53] K. Prieto, O. Dorn, *Sparsity and level set regularization for diffuse optical tomography using a transport model in 2D*, Inverse Problems, Volume 33, No. 1, January, 2017, 014001

[54] R. Pratt, C. Shin and G. Hicks, *Gauss Newton and full Newton methods In frequency space seismic waveform inversion*, Geophysical Journal International, Volume 133, Issue 2, May, 1998, Page(s) 341-362

[55] R. Plessix, *A review of the adjoint-state method for computing the gradient of a functional with geophysical applications*, Geophysical Journal International, Volume 167, Issue 2, November 2006, Page(s) 495-503

[56] H. Robbins, S. Monro, *A Stochastic Approximation Method*, The Annals of Mathematical Statistics, Volume 22, No. 3, 1951, Page(s) 400-407

[57] L. Ryzhik, G. Papanicolaou, and J. Keller, *Transport equations for Elastic and Other Waves in Random Media*, Wave Motion, Volume 24, Issue 4, December, 1996, Page(s) 327-370

[58] P. Rodríguez, M. Kindelan, M. Moscoso and O. Dorn, *History matching problem in reservoir engineering using the propagation-backpropagation method*, Inverse Problems, Volume 21, No. 2, April, 2005, 565

[59] R. Renka, *A Simple Explanation of the Sobolev Gradient Method*, `http://www.cse.unt.edu/~renka/papers/sobolev.pdf`, July, 2006

[60] R. Renka, *Image segmentation with a Sobolev gradient method*, Nonlinear anaylsis, Volume 71, Issue 12, December, 2009, Page(s) e774-e780.

[61] J. Sethian, P. Smereka, *Level Set Methods for Fluid Interfaces*, Annual Review of Fluid Mechanics, Volume 35, No. 1, 2003, Page(s) 341-372

[62] J. Sethian, *Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces*, Acta Numerica, Volume 5. January, 1996, Page(s) 309-395

[63] C. Shin, D. Min, T. Ha, and Y. Choi, *Efficient calculation of the steepest descent direction for source-independent seismic waveform inversion: An amplitude approach*, Journal of Computational Physics, Volume 208, Issue 2, September, 2005, Page(s) 455-468

[64] J. Salencon, *Handbook of Continuum Mechanics: General Concepts, Thermoelasticity*, Applied Mechanics Reviews, Volume 55, Issue 3, June, 2002, B43-B44

[65] M. Schweiger, O. Dorn, A. Zacharopoulos, I. Nissilä, and S. Arridge, *3D level set reconstruction of model and experimental data in Diffuse Optical Tomography*, Optics Express, Volume 18, Issue 1, 2010, Page(s) 150-164

[66] T. Strohmer, R. Vershynin, *A Randomized Kaczmarz Algorithm with Exponential Convergence*, Journal of Fourier Analysis and Applications, 15: 262, April, 2009, Page(s) 262-278

[67] W. Symes, *The seismic reflection inverse problem*, Inverse Problems, Volume 25, No. 12, December, 2009, 123008

[68] W. Symes, *Objective functions for full waveform inversion*, `http://trip.rice.edu/downloads/talk_ws05.pdf`

[69] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*, Philadelphia: Society of Industrial and Applied Mathematics, 2005, `https://epubs.siam.org/doi/book/10.1137/1.9780898717921`

[70] A. Tarantola, *Inversion of seismic reflection data in the acoustic approximation*, Geophysics, Volume 49, No. 8, August, 1984. Page(s) 1259-1266

[71] B. Treeby, B. Cox, and J. Jaros, *k-wave: A MATLAB toolbox for the time domain simulation of acoustic wave fields*, Manual Version 1.1, August, 2016

[72] M. Tabei, T. Mast, and R. Waag, *A k-space method for coupled first-order acoustic propagation equations*, The Journal of the Acoustical Society of America, Volume 111, January, 2002, Page(s) 53-63

[73] L. Vese, *Multiphase Object Detection and Image Segmentation*, Geometric Level Set Methods in Imaging, Vision, and Graphics, Springer, New York, NY, 2003, Page(s) 175-194

[74] J. Virieux, S. Operto, *An overview of full-waveform inversion in exploration geophysics*, Geophysics, Volume 74, No. 6, November, 2009

[75] J. Virieux, *P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method*, Geophysics, Volume 51, Issue 4, April, 1986, Page(s) 889-901

[76] Y. Wu, O. Dorn, *A level set method for shape reconstruction in seismic full waveform inversion using a linear elastic model in 2D*, Journal of physics: Conference series, Volume 1131, conference 1, 2018, 012001

[77] A. Zouzias, N. Freris, *Randomized Extended Kaczmarz for Solving Least Squares*, SIAM Journal on Matrix Analysis and Applications, Volume 34, Issue 2, 2013, Page(s) 773-793.