Spectral element and the adjoint method in seismology

- Modelling of discontinuous boundaries

Master thesis

Written by *Niels Fabrin Nymand*

16.08.2021

Supervised by

Klaus Mosegaard

University of Copenhagen

UNIVERSITY OF COPENHAGEN

Name of Institute:	Niels Bohr Institute
Name of Department:	Physics of Ice, Climate and Earth
Author(s):	Niels Fabrin Nymand
Email:	wnp722@alumni.ku.dk

Title:	Spectral element and the adjoint method in seismology - Modelling of discontinuous boundaries
Supervisor(s):	Klaus Mosegaard
Handed in:	16.08.2021

Abstract

In this thesis I join spectral element modelling of seismic waves with the *adjoint method* to try and reproduce discontinuous boundary layers from seismic signals. The adjoint method is a highly efficient way of generating model gradients to be used in steepest descent type inversion algorithms. A series of simple models are made to demonstrate how the adjoint method can be used to model sharp boundaries, such as faults, from seismic data. The results show that the method is able to locate and horizontally delimit sharp contrasts not accounted for by the prior model. The method is also tested on a more complex model, based on the seismic survey of North Viking Graben [Keys et al., 1998], where the success of the method does in part rely upon good prior model estimations.

Spectral element modelling has the advantage of being able to accurately honour complex model geometry with a well designed mesh. Based on a seismic flattening algorithm I present a simple approach to generating a finite element mesh to be used with the spectral element software SPECFEM2D [Komatitsch et al., 2012].

Contents

1 Introduction

The Earth is full of waves, excited by earthquakes, oceans, atmosphere or man-made sources. These waves travel trough the Earth where they encounter obstacles that send echos back to where the wave originated, just like when you hear you own echo in a large room. These echos, or reflections as they are called, can carry information about the physical properties of the Earth. The ultimate goal of seismology is to use these reflections to infer information about what is hiding underneath our feet, and figure out what the Earth looks like on the inside. To answer this question it is important to understand the physics of these waves. From the theory of continuum mechanics it is possible to derive the equations that govern the waves propagation though the Earth. It is however no easy task to find solutions to these equations that can help explain the reflections seen on the seismograms. It is therefor necessary to seek numerical solutions, i.e. simulating the wave propagation, also called full waveform modelling. However, it requires huge amounts of computing power to do realistic wave simulations, and for that reason full waveform modelling is a fairly new discipline within the world of seismology.

Because there in the past 20 years or so has been made great advances in the available computing power, it is now possible to run simple full waveform problems on laptops. Simple 2D problems can easily run on modern laptops, but even a realistic 2D seismic survey with hundreds of source and receiver configurations would require unhuman patience to simulate accurately. It is therefor common to use large computer clusters when a full 3D problem, like in earthquake seismology, is modelled.

There exist different ways of doing full waveform modelling, but the two most common in seismology is the finite difference method and the spectral element method. In this thesis I will present the spectral element method, both the theory behind it and some of the more practical aspects, such as generating a suitable mesh grid. I will also walk though the adjoint method [Tarantola, A. 1984], and how to use it with full waveform modelling to reproduce sharp contrasts and discontinuous reflector geometry. A number

of examples will be given, as well as how the adjoint method handles the introduction of random uncorrelated noise. Lastly, I use the adjoint method on a semi-realistic synthetic problem, where model geometry, velocity logs and source-receiver setups are taken from a real seismic survey of the Viking Graben in the North Sea. The goal is to see the adjoint methods ability to find fault like structures using this more complex setup.

2 Full Waveform Modelling

In this section I will discuss full waveform modelling of seismic waves using spectral element method. In short, full waveform modelling is simulating waves propagating throughout a modelling domain with a predefined set of model parameters like p- and s-wave velocities and density. Analytical solutions to the wave equation is limited, and usually involve highly idealised systems with little root in reality, but can be useful to test the accuracy of modelling algorithms. Numerical solutions to the wave equation is much more flexible with regards to the complexity of the systems, and is mostly limited by the computer power available. Two numerical methods frequently used in seismic wave propagation are finite difference method and spectral element method. Finite difference has the advantage of being easy to understand and implement. One of the problems with finite difference methods is its limitations in dealing with complex geometry. Modelling a seismic wave across a sloping boundary requires a fine grid to accurately represent such boundaries, because the step like structure of the approximated boundary would otherwise cause scattering. In the spectral element method such boundaries can be modelled by creating a mesh grid that accurately represent the geometry present in the geology, without increasing the grid resolution. Spectral element also has the advantage of being able to have varying grid resolution throughout the modelling domain, depending on the model parameters, like the p- and s-wave velocities [Fichtner, A. 2011]. In the following I will go through some of the theory behind the spectral element method.

2.1 Spectral Element

As mentioned above, both finite difference and spectral element has its advantages and disadvantages, but for this thesis I have chosen to use spectral element for modelling. In the following I will give a brief description of the spectral element method.

The spectral element method is a subclass of the finite element method, with properties that makes it suited for wave propagation. In one dimension the derivation is as follows.

Starting with the 1D scalar wave equation

$$\rho(x)\ddot{u}(x,t) - \frac{\partial}{\partial x}[\mu(x)\frac{\partial}{\partial x}u(x,t)] = f(x,t) \tag{2.1}$$

where ρ is the density, u is the displacement, μ is the elastic parameter, and f is external forces (a source term). The model is limited in both time and space, $x \in G = [0, L]$ and $t \in [0, T]$, where L is the length of the modelling domain and T is the modelled time. The displacement field is subject to the Neumann boundary conditions

$$\frac{\partial}{\partial x}u(x,t)|_{x=0} = \frac{\partial}{\partial x}u(x,t)|_{x=L} = 0^{1} \tag{2.2}$$

and the initial conditions

$$u(x,t)|_{t=0} = \dot{u}(x,t)|_{t=0} = 0 \tag{2.3}$$

Equations (2.1)-(2.3) is commonly known as the strong form of the wave equation, and in finite difference this would be all we need. However, in finite element the weak form is used. To derive the weak form, equation 2.1 is multiplied by an arbitrary, time

[1]Note that $\frac{\partial}{\partial x}u(x,t)$ is the strain, which is proportional to the stress. It is therefor also refered to as the stress free boundary condition, and in 2D and 3D it would be $\mathbf{n} \cdot \boldsymbol{\sigma}|_{\mathbf{x}\in\partial G} = 0$, where $\mathbf{n}$ is the direction normal to the boundary ∂G and $\boldsymbol{\sigma}$ is the stress tensor.

independent test function w, and is then integrated over space,

$$\int_G \rho\, w\, \ddot{u}\, dx - \int_G w\, \frac{\partial}{\partial x}\left(\mu\, \frac{\partial}{\partial x} u\right)\, dx = \int_G w\, f\, dx \tag{2.4}$$

Using integration by parts on the second term on the left-hand side,

$$\int_G w\, \frac{\partial}{\partial x}\left(\mu\, \frac{\partial}{\partial x} u\right)\, dx = \cancelto{0}{\left[w\, \mu\, \frac{\partial}{\partial x} u\right]_G} - \int_G \frac{\partial}{\partial x} w\, \mu\, \frac{\partial}{\partial x} u\, dx$$

where the boundary condition 2.2 is used. The weak form of the wave equation can now be written as,

$$\boxed{\int_G \rho\, w\, \ddot{u}\, dx + \int_G \mu\, \frac{\partial}{\partial x} w\, \frac{\partial}{\partial x} u\, dx = \int_G w\, f\, dx} \tag{2.5}$$

with the initial conditions,

$$\int_G \rho\, w\, u|_{t=0}\, dx = \int_G \rho\, w\, \dot{u}|_{t=0}\, dx = 0 \tag{2.6}$$

One of the advantages of the weak form, especially when modelling surface waves, is the implicitly satisfied boundary conditions 2.2.

The first step in solving equations 2.5 and 2.6 is to lower our ambitions, and instead look for solutions to an approximate displacement field $\overline{u}$,

$$u(x,t) \approx \overline{u}(x,t) = \sum_{i=1}^{n} u_i(t)\psi_i(x) \tag{2.7}$$

where ψ_i are space-dependent basis functions and u_i are time-dependent expansion coefficients. The success of this approximation depend strongly on the choice of basis functions ψ_i. The approximate weak form can then be written as,

$$\int_G \rho\, \psi_j\, \ddot{\overline{u}}\, dx + \int_G \mu\, \frac{\partial}{\partial x}\psi_j\, \frac{\partial}{\partial x}\overline{u}\, dx = \int_G \psi_j\, f\, dx \tag{2.8}$$

where $w = \psi_j$. Inserting $\overline{u}$ to get the weak formulation for $u_i(t)$,

$$\begin{aligned}&\sum_{i=1}^{n}\left[\ddot{u}_i(t)\overbrace{\int_G \rho(x)\,\psi_j(x)\,\psi_i(x)\,dx}^{M_{ji}}\right]\\ &+\sum_{i=1}^{n}\left[u_i(t)\overbrace{\int_G \mu(x)\,\frac{\partial}{\partial x}\psi_j(x)\,\frac{\partial}{\partial x}\psi_i(x)\,dx}^{K_{ji}}\right]=\overbrace{\int_G \psi_j(x)\,f(x,t)\,dx}^{f_j(x,t)}\end{aligned} \tag{2.9}$$

for all $j = 1, \dots, n$. This equation can be written as a matrix equation by defining the vectors $\mathbf{u}(t) = [u_1(t), \dots, u_n(t)]^T$, $\mathbf{f}(t) = [f_1(t), \dots, f_n(t)]^T$ and the mass matrix $\mathbf{M} = \{M_{ji}\}$ and stiffness matrix $\mathbf{K} = \{K_{ji}\}$.

$$\mathbf{M} \cdot \ddot{\mathbf{u}}(t) + \mathbf{K} \cdot \mathbf{u}(t) = \mathbf{f}(t) \tag{2.10}$$

It would require very complex basis functions to accurately model the entire wavefield as described in equation 2.10. The next step is therefor to once more lower our ambitions, and divide the domain G into n_e subdomains G_e, each with a local set of basis functions ψ_i^e $(i = 1, \dots, N+1)$. These subdomains is also called elements. The displacement field within each element can then be approximated by,

$$\overline{u}(x,t)|_{x\in G_e} = \sum_{i=1}^{N+1} u_i^e(t)\psi_i^e(x) \tag{2.11}$$

Equation 2.9 now holds for each element,

$$\begin{aligned}&\sum_{i=1}^{N+1}\left[\ddot{u}_i^e(t)\int_{G_e} \rho(x)\,\psi_j^e(x)\,\psi_i^e(x)\,dx\right]\\ &+\sum_{i=1}^{N+1}\left[u_i^e(t)\int_{G_e} \mu(x)\,\frac{\partial}{\partial x}\psi_j^e(x)\,\frac{\partial}{\partial x}\psi_i^e(x)\,dx\right]=\int_{G_e} \psi_j^e(x)\,f(x,t)\,dx\end{aligned} \tag{2.12}$$

or on matrix form,

$$\mathbf{M}^e \cdot \ddot{\mathbf{u}}^e(t) + \mathbf{K}^e \cdot \mathbf{u}^e(t) = \mathbf{f}^e(t) \tag{2.13}$$

To simplify the integrals that constitutes $\mathbf{M}^e$,$\mathbf{K}^e$ and $\mathbf{f}^e$ the element domains G_e are mapped onto a reference interval $[-1, 1]$ via an element transformations F_e,

$$\begin{aligned} &F_e : [-1, 1] \rightarrow G_e, \\ &x = F_e(\gamma) = h_e \frac{\gamma + 1}{2} + x_e, \\ &\gamma = \gamma(x) = F_e^{-1}(x) = 2\frac{x - x_e}{h_e} - 1 \end{aligned} \tag{2.14}$$

where h_e is the width of the e'th element and x_e is the position of the left boundary of the e'th element.

Now we have arrived at the interesting part, what basis functions should we use? What differentiates spectral element from other finite element schemes, is the choice of $N + 1$ Lagrange polynomials of degree N as basis functions with Gauss-Lobatto-Legendre points (GLL points) as collocation points,

$$\psi_i^e \rightarrow \ell_i^{(N)}(\gamma) = \prod_{j \neq i}^{N+1} \frac{\gamma - \gamma_j}{\gamma_i - \gamma_j}, \quad i, j = 1, 2, ..., N + 1 \tag{2.15}$$

where γ_i are fixed points in the reference interval $[-1, 1]$ called collocation points, which in this case is GLL points.

There are a few good reason for this exact choice of basis functions and collocation points that I will not go into detail with here, see [Fichtner, A. 2011] and [Igle, H. 2017]. However, one of the great benefits, from a computational perspective, is the fact that the mass matrix become diagonal. This property comes from the orthogonality of the

Lagrange polynomials,

$$\ell_i^{(N)}(\gamma_j) = \prod_{j \neq i}^{N+1} \frac{\gamma_j - \gamma_j}{\gamma_i - \gamma_j} = \delta_{ij} \tag{2.16}$$

where δ_{ij} is the Kronecker symbol.

Combining the choice of basis functions and the element transformation lead to the following equation,

$$\begin{aligned} &\sum_{i=1}^{N+1} \ddot{u}_i^e(t) \int_{-1}^{1} \rho'(\gamma) \ell_j^{(N)}(\gamma) \ell_i^{(N)}(\gamma) \frac{dx}{d\gamma} \, d\gamma \\ \cdot+ &\sum_{i=1}^{N+1} u_i^e(t) \int_{-1}^{1} \mu'(\gamma) \left(\frac{\partial}{\partial \gamma} \ell_j^{(N)}(\gamma) \right) \left(\frac{\partial}{\partial \gamma} \ell_i^{(N)}(\gamma) \right) \left(\frac{d\gamma}{dx} \right)^2 \frac{dx}{d\gamma} \, d\gamma \\ = &\int_{-1}^{1} \ell_j^{(N)}(\gamma) f'(\gamma, t) \frac{dx}{d\gamma} \, d\gamma \end{aligned} \tag{2.17}$$

where ρ', μ' and $'$ are the transformed ρ, μ and f. A benefit of using the GLL points is that it makes it possible to apply the GLL quadrature formulas to accurately approximate the integrals in equation 2.17. The integrals can be exchanged with a weighted sum of the integrands evaluated at the GLL points. If γ_k is a GLL point and ω_k is the corresponding integration weight, equation 2.17 can then be approximated as,

$$\begin{aligned} &\sum_{i=1}^{N+1} \ddot{u}_i^e(t) \sum_{k=1}^{N+1} \omega_k \rho'(\gamma) \ell_j^{(N)}(\gamma) \ell_i^{(N)}(\gamma) \frac{dx}{d\gamma} \bigg|_{\gamma=\gamma_k} \\ + &\sum_{i=1}^{N+1} u_i^e(t) \sum_{k=1}^{N+1} \omega_k \mu'(\gamma) \left(\frac{\partial}{\partial \gamma} \ell_j^{(N)}(\gamma) \right) \left(\frac{\partial}{\partial \gamma} \ell_i^{(N)}(\gamma) \right) \left(\frac{d\gamma}{dx} \right)^2 \frac{dx}{d\gamma} \bigg|_{\gamma=\gamma_k} \\ = &\sum_{k=1}^{N+1} \omega_k \ell_j^{(N)}(\gamma) f'(\gamma, t) \frac{dx}{d\gamma} \bigg|_{\gamma=\gamma_k} \end{aligned} \tag{2.18}$$

The orthogonality of the Lagrange polynomials comes into play in the first term as,

$$\ell_j^{(N)}(\gamma)\ell_i^{(N)}(\gamma)\Big|_{\gamma=\gamma_k} = \ell_j^{(N)}(\gamma_k)\ell_i^{(N)}(\gamma_k) = \delta_{jk}\delta_{ik} = \delta_{ji}$$

which results in the mass matrix being diagonal,

$$M_{ji}^e = \omega_j \rho'(\gamma)\frac{dx}{d\gamma}\delta_{ji}\Big|_{\gamma=\gamma_j}$$

All that is missing now is to collect it all into a global set of equations that can be solved for the total displacement field. Another ”small” step is to extent all above to 2D and 3D using the full elastic wave equation [Fichtner, A. 2011].

2.2 Modelling software

2.2.1 SPECFEM2D

There exist different software packages that can handle seismic wave propagation using spectral element method. In this thesis I have chosen to use the SPECFEM2D package [Komatitsch et al., 2012]. SPECFEM2D is an open-source software package capable of doing forward and adjoint elastic wave simulations using spectral element. The choice landed on SPECFEM2D because it is a proven piece of software and with relatively good documentation [2]. SPECFEM2D is great for testing smaller problems that can easily be run on a modern laptop, and can then be upscaled to the full 3D problem at a later time using SPECFEM3D together with a more powerful computer.

[2] Documentation for open source software is usually quite sparse, especially when dealing with a niche subject like seismic waveform modelling with spectral element method.

2.3 Mesh generation

Subdividing the modelling domain into elements is called mesh generation, and is in many ways, the most important and difficult part of the spectral element modelling process. As mentioned above, a great advantage of spectral element is the flexibility with regards to geometrical complexity. To fully exploit this feature requires a well designed mesh. In section 3 I will present a semi automatic way of extracting information about the geometry of an area based on a seismic profile. The idea is to use this information to create a mesh that honour internal boundaries of the model domain. To help create the mesh, another piece of software is used, Gmsh.

2.3.1 Gmsh

There exist many different mesh generating tools, such as CUBIT and Gmsh. Following the path of open source I have chosen to use Gmsh [Geuzaine et al., 2009] for mesh generation. Gmsh has good documentation, is relatively easy to use, but most importantly it has the ability to be used together with Python. The connection with Python allows for a semi automatic mesh generation routine.

3 Seismic Flattening

Seismic flattening is the discipline of removing geological structures in a seismic image. The goal is to find a transform that maps the curved reflectors of a seismic profile to flat horizontal reflectors. It can be thought of as finding the transform that arranges the internal layers after age, or as removing the effects of geodynamical processes.

There are more than one way of finding the flattening transform. The most obvious would be to do a manual/automatic tracking of all the major horizons present in the seismic profile, and then do an elastic transformation of the seismic traces mapping the horizons to flat lines. This method will guarantee that the tracked horizons becomes

completely flat, but everything inbetween may not. Tracking horizons manually can also be a time consuming process, especially in the case of large 3D data sets, which require experience to be done well.

Another way to attack the problem would be to view the seismic profile as an 2D/3D image, and use techniques developed in image processing to preform the flattening. In the following I follow the derivation in [Parks, D. 2010]. The goal is to find the transformation that relate the flattened image $g(x, \tau)$ with the input image $f(x, t)$, where τ and t are depth coordinates[3]. Note that the variable τ is constant along seismic events. Each sample in the input image can be shifted up or down to create the flattened image. Let $s(x, \tau)$ be the function describing these shifts. In figure 3.1 an example of such a shift function is visualised.

From figure 3.1 it is easy to see that going from f to g can be done with the transformation $t(x, \tau_0) = \tau_0 - s(x, \tau_0)$, or generally

$$g(x, \tau) = f(x, t(x, \tau)) \tag{3.1}$$

where $t(x, \tau) = \tau - s(x, \tau)$ is the 2D mapping function relating the input image with the flattened image.

A way of relating s with the input image can be found by taking derivatives on both sides of equation 3.1 with respect to x:

$$\frac{\partial g}{\partial x} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial t}\frac{\partial t}{\partial x}$$

where $\frac{\partial g}{\partial x} = 0$ per definition of g, and $\frac{\partial t}{\partial x} = -\frac{\partial s}{\partial x}$

$$0 = \frac{\partial f}{\partial x} - \frac{\partial f}{\partial t}\frac{\partial s}{\partial x}$$

[3]Here depth should be understood as the vertical axis of the image. It does not matter if this is a time (e.g. measure in s) or depth (e.g. measured in m) axis

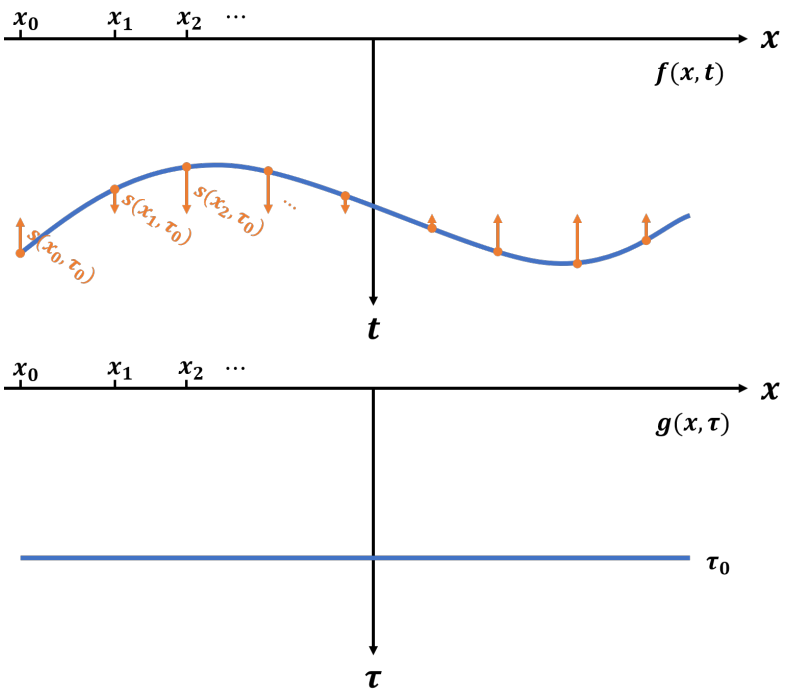


Figure 3.1: Top figure shows an example of an event in $f(x,t)$ with the shift function $s(x,\tau)$ visualised as arrows. Bottom figure show that same event in $g(x,\tau)$.

$$\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial t}} = \frac{\partial s}{\partial x} \tag{3.2}$$

Equation 3.2 is a bit problematic for a few reasons. First, the image almost certainly contains noise, which will cause noisy derivatives. Second, if $\frac{\partial f}{\partial t} = 0$ the fraction is undefined. Third, since s can be found by integrating both sides of equation 3.2, even small errors in the derivatives can accumulate to large errors in s. A different approach to calculating the derivatives is therefor needed, and such an approach could be the structure tensor.

The structure tensor is used to find the direction of greatest change in a window around every sample of the image, making it more robust towards noise. In a seismic image the direction of greatest change is always normal to the events. If $\mathbf{n} = [n_x, n_t]^T$ is normal to an event, then n_x and n_t would be proportional to $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial t}$ respectively.

$$p(x,t) \equiv -\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial t}} = -\frac{n_x(x,t)}{n_t(x,t)} \tag{3.3}$$

Equation 3.2 can then be rewritten as,

$$p(x,t) = -\frac{\partial}{\partial x} s(x,\tau) \tag{3.4}$$

3.1 Seismic flattening as an inverse problem

Equation 3.4 can be solved as a non linear inverse problem, where the non linearity is introduced by the fact that the left side of the equation depend on t and the right side on τ. However by approximately linearizing equation 3.4, the shifts can be solved for using a least-squares solution.

$$\begin{aligned} -\frac{\partial}{\partial x} s^{\{i\}}(x,\tau) &= p(x, \tau - s^{\{i-1\}}(x,\tau)) \\ \epsilon \frac{\partial}{\partial \tau} s^{\{i+1\}}(x,\tau) &= 0 \end{aligned} \tag{3.5}$$

Where the last equation in 3.5 is there to ensure that the vertical variation of s stays under control. However, another formulation could be to have a tolerance k for the maximum allowed vertical variation of s, $|\frac{\partial}{\partial \tau} s| > k$, instead of having the parameter ϵ. The algorithm for solving equation 3.5 is outlined in Algorithm 1. The number of iterations needed, N_{iter}, to obtain a flattened image depend on the problem. An example of how the algorithm performs can be seen in figure 3.2. The black lines in figure 3.2g show the inverse flattening transformation, i.e. horizontal lines that have underwent the flattening process in reverse. These black lines representing the inverse flattening transform can then be used to generate a mesh that honour the internal boundaries, given that the vertical axis is units of depth and not time.

input : Image f
output: Flattened image g

`StructureTensor(`*image*`)`:
Takes an image as input and calculates the direction of greatest change $\mathbf{n} = [n_x, n_t]^T$
`GaussianConvolve(`*image*`)`:
Takes an image as input and convolves it with a Gaussian kernel
`LeastSquaresSolver(`G,d,m_0,*max_iter*`)`:
An iterative least squares solver for the equation $Gm = d$, *with* $m^{\{0\}}$ *as starting guess. Stops when* $|m^{\{j\}} - m^{\{j-1\}}|$ *is small, or* $j =$ *max_iter*
`ImageWarping(`*image*,s`)`:
Warps an image according to the shift function s

$F^{\{0\}}$ = `GaussianConvolve(`f`)`
$s^{\{0\}} = 0$ # *starting guess for* s
for $i = 1, N_{iter}$ **do**
  # *Calculating* n_x *and* n_t *with the structure tensor*
  n_x,n_t = `StructureTensor(`F`)`
  $p = -\frac{n_x}{n_t}$
  # *Least squares solution to* $\frac{\partial}{\partial x}s = p$
  s = `LeastSquaresSolver(`$\frac{\partial}{\partial x}$,$p$,$s^{\{i-1\}}$`)`
  while $|\frac{\partial}{\partial \tau}s| > k$ **do**
    # *Least squares solution to* $\frac{\partial}{\partial \tau}s = 0$
    s_{new} = `LeastSquaresSolver(`$\frac{\partial}{\partial \tau}$,*0*,$s$,*max_iter = 1*`)`
    $s = s_{new}$
  end
  $F^{\{i\}}$ = `ImageWarping(`$F^{\{i-1\}}$,s`)`
  $s^{\{i\}} = s$
end
$g = F^{\{N_{iter}\}}$

Algorithm 1: Flattening algorithm. In image processing a kernel, also known as a convolution matrix, is a small matrix used in convolution with an image. A Gaussian kernel is an often used kernel for removing high frequency noise, or blurring.

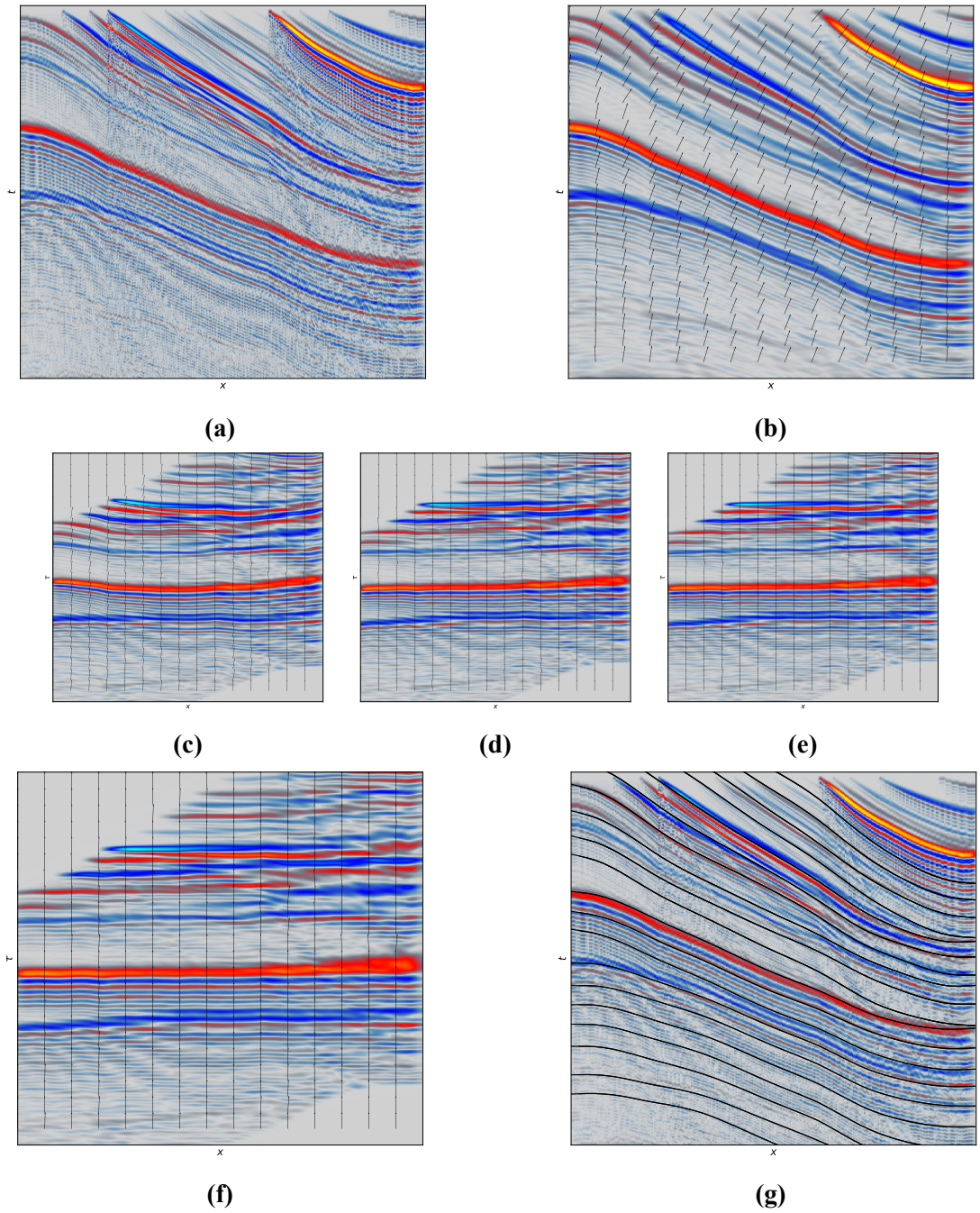


Figure 3.2: Seismic flattening preformed on a synthetic seismic profile. (a): Input image, (b): Input image convolved with gaussian kernel to remove high frequency noise, (c): First iteration of flattening process, (d): Second iteration, (e): Third iteration, (d): Fourth and final iteration, (g): Input image with inverse flattening transformation overlay. Black arrows point in the direction of greatest change, $\mathbf{n} = [n_x, n_t]^T$

4 The Data: Viking Graben

When collecting seismic data it is common to have a single source and multiple receivers, either on both sides of the source or only on one side. Data is then collected throughout the investigated area at multiple source and receiver configurations. This type of data is called shot records. The ultimate goal of full waveform modelling is to help map the subsurface by matching the modelled seismic to the observed seismic, also called full waveform inversion, but more on this later. Because there exist no processing techniques that can create new data, the most useful data should therefor be as raw as possible, i.e. the shot records. In addition to the seismic data it is necessary to know something about the physical properties of the area, which can come from well logs, and preferably at multiple locations spread throughout the investigated area. Lastly it is also important to know the source time function, wavelet, used when collecting the data. The wavelet can be estimate with the use of the well logs.

In reality all of this information is rarely available. It turned out that readily available pre-stack seismic data paired with well log data and wavelet information is a rarity. However, in 1994 the Society of Exploration Geophysicists Research Commmittee held a workshop titled ”Comparison of Seismic Inversion Methods on a single Real Data Set” [Keys et al., 1998]. The goal of this workshop was to test seismic inversion methods on the same data set, which included marine pre-stack seismic data and petrophysical measurements collected at two wells that intersected the seismic line. In addition the data set also includes a measured source pulse from the air gun array, which saves many hours of wavelet estimation. The area of investigation, North Viking Graben, is located in the North Sea.

Due to time restrictions it will not be possible to perform full waveform inversion using this data set. What I will do instead is to use information about the geometry of the area and velocity and density logs to create a semi realistic model that is based on real data. I will also use the source and receiver setups in the modelling process. I will

therefor briefly discuss the data available in the Viking Graben data set.

4.1 Seismic

The seismic data consists of 1001 shot records, where each shot was recorded on 120 receivers for six seconds sampled every $4ms$. Receivers are spaced $25m$ apart, and the distance from source to the near offset receiver is $262m$. Receivers and source are located $10m$ and $6m$ below the surface respectively. An example of a shot record is shown in figure 4.1.

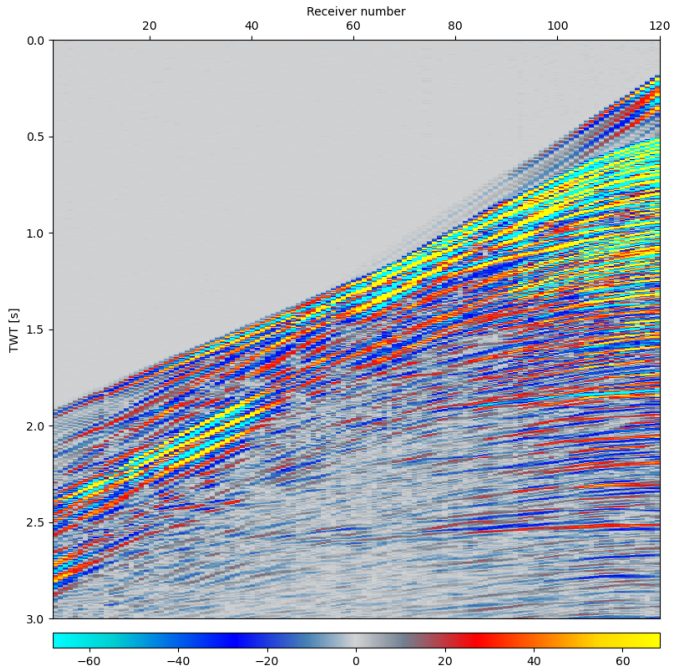


Figure 4.1: Shot record 832 of the Viking Graben data set cut off at 3 seconds. The data units are not specified, but I assume it is in some units of pressure.

To get some information about the geometry of the area it is very useful to have a depth migrated seismic section. It is unfortunately not directly included in the data set, but in an attached pdf document is a scanned plot of a depth migrated seismic section covering the area. Because it is only used in the flattening algorithm, a plot should be okay as the algorithm is based on image processing. The scanned plot can not be used

without a bit of work, because it is taken from a slightly curved page of a book. Via a projection transform the picture is converted to an image with straight boundaries that can hopefully be used in the flattening algorithm. In figure 4.2 the result of the projection transform is shown. The data is in no way ideal, as it shows seismic wiggles. This means that there is no way to distinguish zero amplitude from negative amplitudes, positive amplitudes all have the same colour and the image is full of vertical lines. Because the flattening algorithm look at structure in the image, it might still be possible to extract information about the geometry.

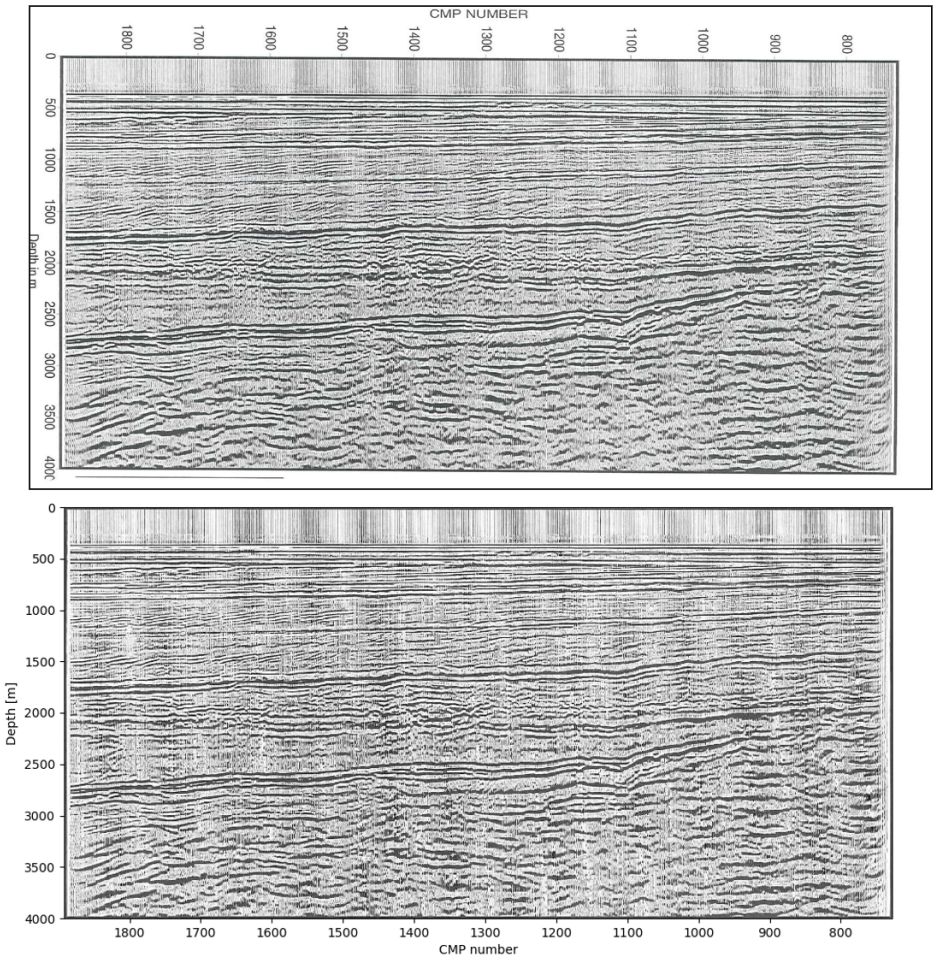


Figure 4.2: Top show a picture of a depth migrated seismic section of the Viking Graben area. After a projection transform this picture has been converted into a data set, bottom plot.

4.2 Well Logs

As mentioned, the data set include petrophysical measurments at two well locations. The most important measurement is the compressional- and shear wave velocity logs. These measurments, together with the density logs, are used to create a prior model, which I discuss in more detail in section 7.

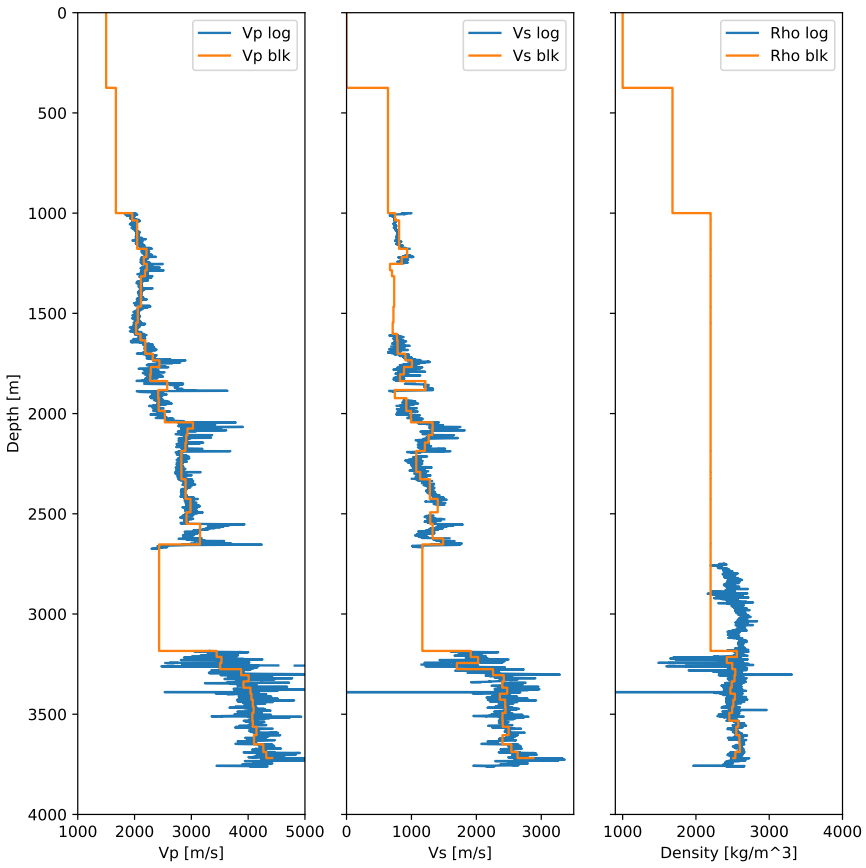


Figure 4.3: Well B log measurements versus blocked (blk) log interpretations. The blocked log is an attempt to recreate the geological layers found in the well and their respective V_p, V_s, ρ.

4.3 Wavelet: Source time function

Except for the prior model, the most important thing when doing full waveform modelling is to get the correct source time function, or wavelet. The Viking Graben data

set actually does include a source time function that has been obtained by positioning a hydrophone $250m$ below the air-gun array, only separated by water. The wavelet can be seen in figure 4.4. Due to reasons I will discuss in section 7 I will not be using this wavelet as source time function.

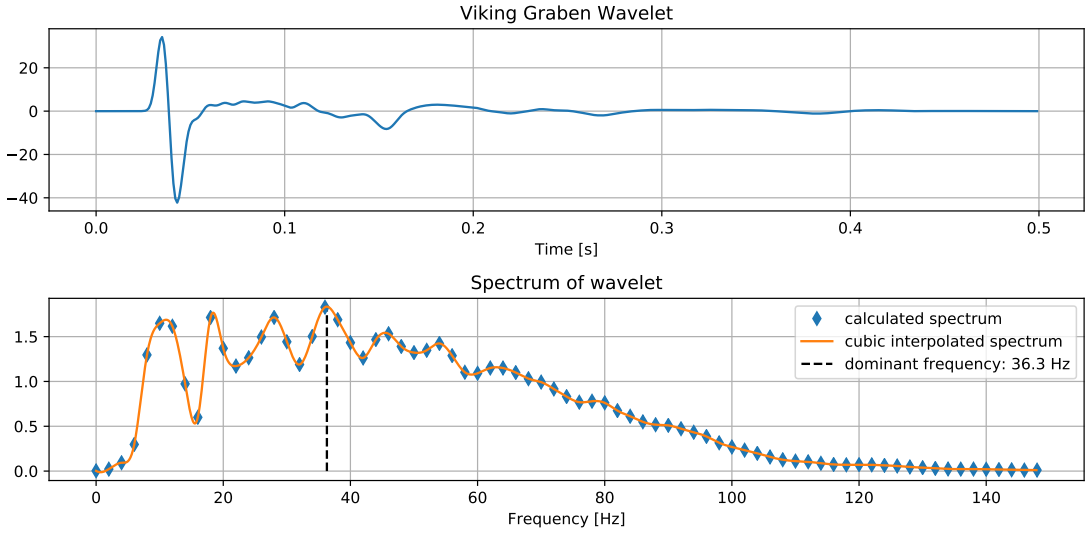


Figure 4.4: Wavelet included in the Viking Graben data set together with its spectrum. The dominant frequency shown is defined as the frequency with the maximum power.

5 Full Waveform Inversion

In physics an inversion is the process of inferring information about a model based on indirect measurements. It could for example be to extract information about the shape of a drum based on the sound it makes [Gordon, C. 1996]. In this case the shape of the drum would be the model we wish to estimate, and the sound of the drum, in the form of an acoustic signal, would be the available data. The goal is then to setup a so called forward model that relates the unknown model parameters, shape of the drum, to the data, the sound it makes. Even this relatively simple problem, where everything other than the shape is held constant, turns out not to have a unique solution. It turns out that almost all inverse problems are haunted by the fact that multiple models can give the exact same measurements. One such example could be to try and figure out the density distribution of the earth based on a single gravity measurement at the surface. The forward model is well know, and the relation between model and data is even linear, but the task is still impossible, due to an infinite number of density distributions giving the exact same gravity measurement at the surface. What makes the problem solvable is a long set of assumptions and prior information. If one knows the radius of the Earth as well as the assumption that the density on average increases with depth, it can be used to constrain the solution. There may still exist an infinite set of possible solutions, but some might violate physical principals, and the solutions can be further reduced.

Just like the examples mentioned above, full waveform inversion is simply the attempt to use measurements, like seismograms, to infer information about an unknown model, for example subsurface wave-velocity and density distributions. In full waveform inversion the forward model is full waveform modelling. Full waveform inversion is a relatively new discipline, as it is only in the past 20 years or so that computers have become powerful enough for it to be feasible. However, when doing inversion one seeks

a model gradient, a direction that minimises a misfit function. To calculate this gradient using a finite difference approach would require a forward calculation for each model parameter, which in many cases might be millions. Even with todays computers, this would be very impractical. It is therefor almost impossible to study full waveform inversion without coming across the *adjoint method*. Before I discuss the adjoint method further I would like to first comment on one of the most important aspects of inversion, the prior model.

5.1 Prior Model

The ambiguity of the general inverse problems means that many different models are equally valid from a mathematical perspective. In the probabilistic formulation of the inverse problem [Tarantola, A. 2005] the solution is a probability distribution over the model space. Such a distribution can be complex with many local maxima. In a deterministic approach, using a steepest descent algorithm for example, will try to find such a maxima. Which maxima it finds depend on where in the model space it starts. The starting point is called the prior model, or the initial guess. Many things can help guide such a guess like well logs, physical laws, geological knowledge of an area, intuition.

To estimate a model that satisfies data is usually easy, to estimate a good approximation to the true model depend greatly on the prior model.

5.2 The Adjoint Method

As mentioned above, the saviour of the full waveform inversion is the adjoint method. One of the first people to use this method in seismology was Tarantola [Tarantola, A. 1984], where he showed that by only doing two simulation you can calculate a model update. Before it would be necessary to do one simulation pr model parameter, which could be millions, to calculate a gradient for a steepest descent algorithm for example.

In the following I will give a brief description of the adjoint method. Starting with the acoustic wave equation,

$$\frac{1}{K(\mathbf{x})}\ddot{p}(\mathbf{x},t) - \nabla \cdot \left[\frac{1}{\rho(\mathbf{x})}\nabla p(\mathbf{x},t)\right] = s(\mathbf{x},t) \tag{5.1}$$

where $K(\mathbf{x})$ is the bulk modulus, $\rho(\mathbf{x})$ is the density, $s(\mathbf{x},t)$ is a source field and $p(\mathbf{x},t)$ is the pressure field. For the sake of simplicity, let $\mathbf{K}, \boldsymbol{\rho}, \mathbf{s}, \mathbf{p}$ represent the functions mentioned above[4]. The solution to equation 5.1, or the forward problem, can be written as,

$$\mathbf{p} = \mathbf{f}(\mathbf{K}, \boldsymbol{\rho}, \mathbf{s}) \tag{5.2}$$

where $\mathbf{f}$ represent a forward operator. The exact nature of $\mathbf{f}$ does not matter for now, but it could for example be a full waveform modelling algorithm like SPECFEM2D. In the real world the pressure field are measured at specific receiver location, $\mathbf{x}_r$, from a source located at $\mathbf{x}_s$. Let the source function be treated as a point source $s(\mathbf{x},t) = \delta(\mathbf{x}-\mathbf{x}_s)S(t)$, where $S(t)$ is a source time function.

The hope, when doing inversion, is to find a model update that improves the difference between the observed pressurefield $\mathbf{p}_0$ and the calculated pressure field from equation 5.2. Let $\mathbf{m}$ represent a model and $\mathbf{m}_0$ the prior model,

$$\mathbf{m} = \begin{pmatrix} \mathbf{K} \\ \boldsymbol{\rho} \\ \mathbf{S} \end{pmatrix}, \quad \mathbf{m}_0 = \begin{pmatrix} \mathbf{K}_0 \\ \boldsymbol{\rho}_0 \\ \mathbf{S}_0 \end{pmatrix}$$

The quality of the estimated model can be summarised by one number, the misfit. The goal is then to minimise this misfit. Using a least squares norm, the misfit for $\mathbf{m}$ can be

[4]For example: In 1D the model domain could be descritized into N points, $x = x_1, x_2, \dots, x_N$. In that case $\mathbf{K}$ could be a vector specifying K at each point, $\mathbf{K} = [K(x_1), K(x_2), \dots, K(x_N)]^T$.

formulated as,

$$2\mathcal{L}(\mathbf{m}) = [\mathbf{p}_0 - \mathbf{f}(\mathbf{m})]^T \mathbf{C}_p^{-1} [\mathbf{p}_0 - \mathbf{f}(\mathbf{m})] + [\mathbf{m}_0 - \mathbf{m}]^T \mathbf{C}_m^{-1} [\mathbf{m}_0 - \mathbf{m}] \tag{5.3}$$

where $\mathbf{C}_p$ and $\mathbf{C}_m$ are the covariance operators for the data set and model respectively, these are often diagonal matrices. The second term is to ensure that the solution, obtained by minimizing equation 5.3, does not become a victim of overfitting, also called Tikhonov regularization. A steepest descent inversion algorithm can then be written as

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \alpha_k \nabla_{\mathbf{m}_k} \mathcal{L}(\mathbf{m}_k) \tag{5.4}$$

where k is the iteration number and α_k is a constant, which could be estimated by a simple line-search method. This equation simply states that an update to $\mathbf{m}$ should be found in the direction that reduces the value of $\mathcal{L}$. To calculate the gradient of $\mathcal{L}$ I will first define a operator F,

$$F = \frac{\mathbf{f}(\mathbf{m} + \delta\mathbf{m}) - \mathbf{f}(\mathbf{m})}{\delta\mathbf{m}},$$

the approximate first derivative of $\mathbf{f}$ at $\mathbf{m}$ in direction $\delta\mathbf{m}$. The gradient of $\mathcal{L}$ can then be written as [Tarantola, 1984],

$$-\nabla_{\mathbf{m}} \mathcal{L}(\mathbf{m}) = C_m F^T C_p^{-1} [\mathbf{p}_0 - \mathbf{f}(\mathbf{m})] + (\mathbf{m}_0 - \mathbf{m}) \tag{5.5}$$

Insert this into equation 5.4,

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha \left[C_m F_k^T C_p^{-1} [\mathbf{p}_0 - \mathbf{f}(\mathbf{m}_k)] + (\mathbf{m}_0 - \mathbf{m}_k) \right] \tag{5.6}$$

The model parameters, $\mathbf{m}$, consist of three different sets of model parameters, $\mathbf{K}$, $\boldsymbol{\rho}$, $\mathbf{S}$. The derivative operator F can therefor also be split into three components $F = (F_K, F_\rho, F_S)$, and equation 5.6 can be written in component form,

$$\begin{pmatrix} \mathbf{K}_{k+1} \\ \boldsymbol{\rho}_{k+1} \\ \mathbf{S}_{k+1} \end{pmatrix} + \alpha \left[\begin{pmatrix} C_{KK}\delta\tilde{\mathbf{K}}_k \\ C_{\rho\rho}\delta\tilde{\boldsymbol{\rho}}_k \\ C_{SS}\delta\tilde{\mathbf{S}}_k \end{pmatrix} + \begin{pmatrix} \mathbf{K}_0 - \mathbf{K}_k \\ \boldsymbol{\rho}_0 - \boldsymbol{\rho}_k \\ \mathbf{S}_0 - \mathbf{S}_k \end{pmatrix} \right], \tag{5.7}$$

where C_m is assumed diagonal and $C_{KK}, C_{\rho\rho}, C_{SS}$ are the diagonal elements,

$$\begin{aligned} \delta\tilde{\mathbf{K}}_k &= F_K^T \delta\tilde{\mathbf{p}}_k \\ \delta\tilde{\boldsymbol{\rho}}_k &= F_\rho^T \delta\tilde{\mathbf{p}}_k \\ \delta\tilde{\mathbf{S}}_k &= F_S^T \delta\tilde{\mathbf{p}}_k \\ \delta\tilde{\mathbf{p}}_k &= C_p^{-1}[\mathbf{p}_0 - \mathbf{f}(\mathbf{K}_k, \boldsymbol{\rho}_k, \mathbf{S}_k)] \end{aligned} \tag{5.8}$$

$\delta\tilde{\mathbf{p}}_k$ is the residuals of the measured seismograms and the synthetics from the forward model, and can easily be calculated. However, to calculate $\delta\tilde{\mathbf{K}}_k, \delta\tilde{\boldsymbol{\rho}}_k$ and $\delta\tilde{\mathbf{S}}_k$ requires two operations. These two operations is really what the adjoint method is all about.

Firstly, Tarantola shows in the appendix of [Tarantola, 1984] that we have to compute the pressure field that corresponds to,

$$\begin{aligned} p'_k(\mathbf{x}, t; \mathbf{x}_s) &= \int d\mathbf{x}' g(\mathbf{x}, 0; \mathbf{x}', t) * \delta s(\mathbf{x}', t; \mathbf{x}_s), \\ \delta s(\mathbf{x}, t; \mathbf{x}_s) &= \sum_r \delta(\mathbf{x} - \mathbf{x}_r)\delta\tilde{p}_k(\mathbf{x}_r, t; \mathbf{x}_s) \end{aligned} \tag{5.9}$$

where $*$ refers to time convolution and g is the Green's function defined via equation 5.1 as,

$$\left[\frac{1}{K(\mathbf{x})} \frac{\partial^2}{\partial t^2} - \nabla \cdot \frac{1}{\rho(\mathbf{x})} \nabla \right] g(\mathbf{x}, t; \mathbf{x}', t') = \delta(\mathbf{x} - \mathbf{x}')\delta(t - t') \tag{5.10}$$

resulting in a solution to equation 5.1 as follows,

$$p(\mathbf{x}, t) = \int d\mathbf{x}' g(\mathbf{x}, t; \mathbf{x}', 0) * \Big(\delta(\mathbf{x} - \mathbf{x}_s) S(t) \Big) \tag{5.11}$$

The difference between equation 5.9 and 5.11, is the source term used and the time reversal of the Green's function. The source term of p'_n is interesting, because it treats the receivers as sources and uses the residual of measured and computed wavefield as the source time function. What the current model does not correctly account for is sent back by the receivers.

Secondly, correlating the two wavefields at all points show areas where the model is inadequate and needs to be modified. The model updates can be written as,

$$\begin{aligned}
\delta\tilde{\mathbf{K}}_k(\mathbf{x}) &= \frac{1}{\mathbf{K}_k(\mathbf{x})^2}\int dt \sum_s \frac{\partial}{\partial t}p'_n(\mathbf{x},t;\mathbf{x}_s)\frac{\partial}{\partial t}p_n(\mathbf{x},t;\mathbf{x}_s) \\
\delta\tilde{\boldsymbol{\rho}}_k(\mathbf{x}) &= \frac{1}{\boldsymbol{\rho}_k(\mathbf{x})^2}\int dt \sum_s \nabla_{\mathbf{x}}p'_n(\mathbf{x},t;\mathbf{x}_s)\cdot\nabla_{\mathbf{x}}p_n(\mathbf{x},t;\mathbf{x}_s) \\
\delta\tilde{\mathbf{S}}_k(\mathbf{x}) &= \sum_s p'_n(\mathbf{x},t;\mathbf{x}_s)
\end{aligned} \tag{5.12}$$

where $\nabla_{\mathbf{x}}$ is the spatial gradient operator. The equation for density and bulk modulus simply state that the model update is non zero in areas where the adjoint wavefield p'_n and the forward wavefield p_n correlate.

What is so remarkable about the results presented in [Tarantola, 1984] is that the model gradient can be computed in three steps,

1. Compute the forward wavefield, p_n
2. Compute the adjoint wavefield, p'_n, by using receivers as sources with the residuals of modelled and measured seismograms as source time functions
3. Correlate the two wavefields to calculate model updates, $\delta\tilde{\mathbf{K}}_k(\mathbf{x})$ and $\delta\tilde{\boldsymbol{\rho}}_k(\mathbf{x})$

Only two simulations are needed per iteration of the inversion algorithm. The adjoint method is therefor a huge time saver when the forward model is slow, like in the case of full waveform modelling.

In the following section I will demonstrate a small example where the adjoint method is used.

5.3 Test Example

I have set up a very simple example with the purpose of demonstrating the adjoint method together with full waveform modelling. The true model and the prior model can be seen in figure 5.1. The data consists of four different shot records, each recorded at 120 different receivers, see figure 5.1 for layout. The seismograms from shot record 0 is shown in figure 5.2. All seismic data shown is the vertical component of the displacement[5]. Like mentioned above, the adjoint sources, see figure 5.2c, are the residuals of the measured and synthetic seismograms. The receivers, marked with blue on figure 5.1, are then used as source points, where the adjoint sources are the source time functions. In appendix A I have included a plot of the mesh grid, figure A.1 right, used together with SPECFEM2D for the spectral element modelling. Right, left and bottom boundary are absorbing and the top boundary is free, see equation 2.2.

Both the prior and the true model, $\mathbf{m}_0$ and $\mathbf{m}_{true}$ respectivly, have a constant density, $\rho_0(x,z) = \rho_{true}(x,z) = 1600\frac{kg}{m^3}$, and zero shear modulus, $V_{s,0}(x,z) = V_{s,true}(x,z) = 0$, where V_s is the shear velocity. The inversion algorithm used is just a simple steepest descent algorithm,

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha\delta\tilde{\mathbf{m}}_k \tag{5.13}$$

where $\delta\tilde{\mathbf{m}}_k$ is the model gradient, the direction that reduces the value of the misfit function,

$$\mathcal{L}(\mathbf{m}_k) = [\mathbf{p}_0 - \mathbf{f}(\mathbf{m}_k)]^T[\mathbf{p}_0 - \mathbf{f}(\mathbf{m}_k)] \tag{5.14}$$

[5]Pressure and vertical displacement are not interchangeable in the above equations, but everything done here should not depend on whether the data is pressure or displacement. The choice to use displacement is purely practical due to the use of preexisting software that was not setup for pressure data.

which is a simplified version of the one presented in [Tarantola, 1984]. α is determined by a line search, see algorithm 2. To help organise the inversion process I have made use of an open source Python based package that can work together with SPECFEM2D to perform full waveform inversion. The package is called SeisFlow, and for more information see [Modrak et al., 2018].

The only thing missing before the inversion can proceed is the source time function. I have made use of a Ricker wavelet, second derivative of a Gaussian, with a dominant frequency of $12Hz$, see figure 5.3.

input : Initial model, $\mathbf{m}_k$ and observed seismograms, $\mathbf{p}_0$
output: Updated model, $\mathbf{m}_{k+1}$

`Forward(`$\mathbf{m}_k$,$\mathbf{x}_r$,$\mathbf{S}$`)`:
returns seismograms recorded at receiver locations $\mathbf{x}_r$
`Adjoint(`$\mathbf{m}_k$,$\mathbf{x}_r$,$\delta\tilde{\mathbf{p}}_k$`)`:
Returns gradient, $\delta\tilde{\mathbf{m}}_k$

$\mathbf{p}_k =$ `Forward(`$\mathbf{m}_k$,$\mathbf{x}_r$,S`)`
 # Calculate the adjoint sources
$\delta\tilde{\mathbf{p}}_k = \mathbf{p}_0 - \mathbf{p}_k$
$\delta\tilde{\mathbf{m}}_k =$ `Adjoint(`$\mathbf{m}_k$,$\mathbf{x}_r$,$\delta\tilde{\mathbf{p}}_k$`)`
 # Pick an initial step length $\alpha = \alpha_0$
$\mathbf{m}_{k+1,0} = \mathbf{m}_k + \alpha_0\delta\tilde{\mathbf{m}}_k$

 # Line search algorithm to determine step length
$i = 0$
while $\mathcal{L}(\mathbf{m}_{k+1,i+1}) < \mathcal{L}(\mathbf{m}_{k+1,i})$ **do**
  $\alpha_{i+1} = \alpha_i + \delta\alpha_i$
  $\mathbf{m}_{k+1,i+1} = \mathbf{m}_k + \alpha_{i+1}\delta\tilde{\mathbf{m}}_k$
  $i = i + 1$
end
$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_i\delta\tilde{\mathbf{m}}_k$

Algorithm 2: Simple inversion algorithm. Note that every iteration of the line search algorithm requires a forward calculation to evaluate equation 5.14. In the case of multiple source and receiver combinations, the gradient calculations (forward and adjoint simulations) are run in parallel for each source and receiver combination and merged to one gradient before the line search.

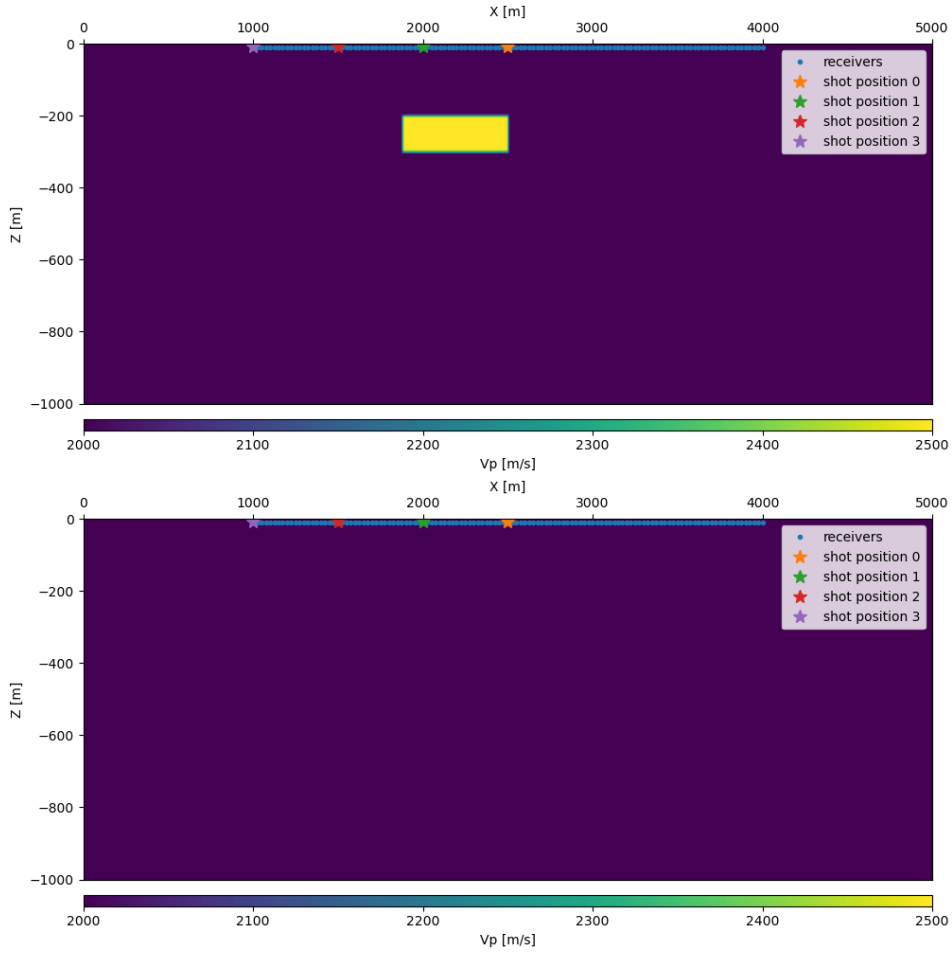
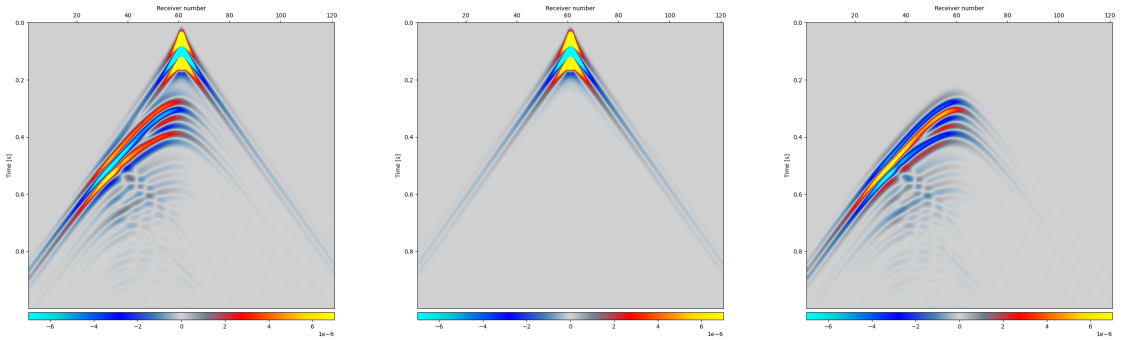


Figure 5.1: Top plot show V_p for the true model. Bottom plot show V_p for the prior model. Density is constant at $\rho = 1600 \frac{kg}{m^3}$, and shear velocity is zero, $V_s = 0 \frac{m}{s}$

5.3.1 Inversion results and discussion

The result of the inversion can be seen in figure 5.4 and figure 5.5. The inversion successfully finds the location of the box anomaly in the true model. The left and right border of the anomaly in the inverted model are well defined, and match the true model. The top and lower boundary are blurred, and have similar characteristics to the Ricker wavelet.

In [Fichtner et al., 2006] they present a similar example with a box anomaly. In their example they calculate the model gradient for two different wavelengths, one that is



(a) (b) (c)

Figure 5.2: Shot record 0 (source located at shot position 0). (a): The observed data from the true model, (b): Synthetic data from the prior model, (c): The adjoint sources.

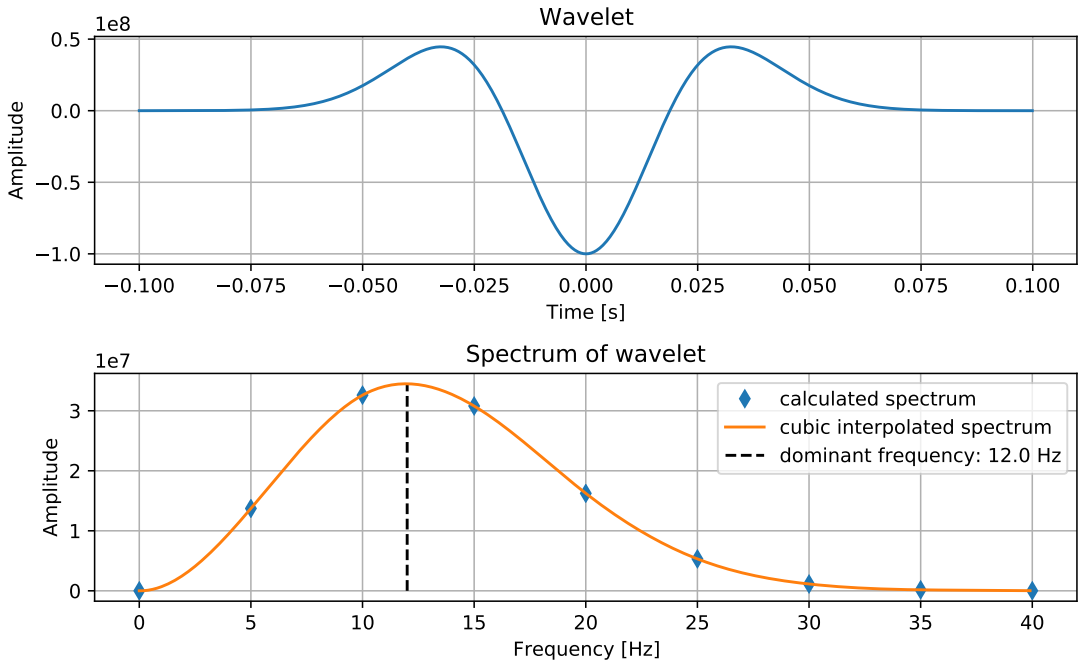


Figure 5.3: Ricker wavelet with dominant frequency of $12 Hz$.

comparable to the size of the anomaly, and one that is significantly larger. The gradient using the shorter wavelength have two significant signals, one from the top boundary and one from the bottom boundary of the box. The gradient using the longer wavelength are much better at mapping the entire box, but the left and right boundary is not as well

defined. For the shorter wavelength the top and lower boundary of the box anomaly act as two separate reflectors. The wavelength in my example is comparable to the size of the anomaly [6], which explain the wavelet like characteristics of the inversion result.

Whether or not the inversion result can be deemed successful or not, depend on what the goal is. If the goal is to reproduce the structure of the true model, then the result seen in figure 5.4 is not very useful. In [Fichtner et al., 2006] they conclude that the determination of structure require a more sophisticated iterative procedure that include higher order derivatives. However, if the goal is to highlight and outline anomalies, as is seen in figure 5.4, then the results of the inversion is very promising.

[6] The wavelength is $\lambda \approx 166m$ and the box anomaly measure $520m \times 100m$

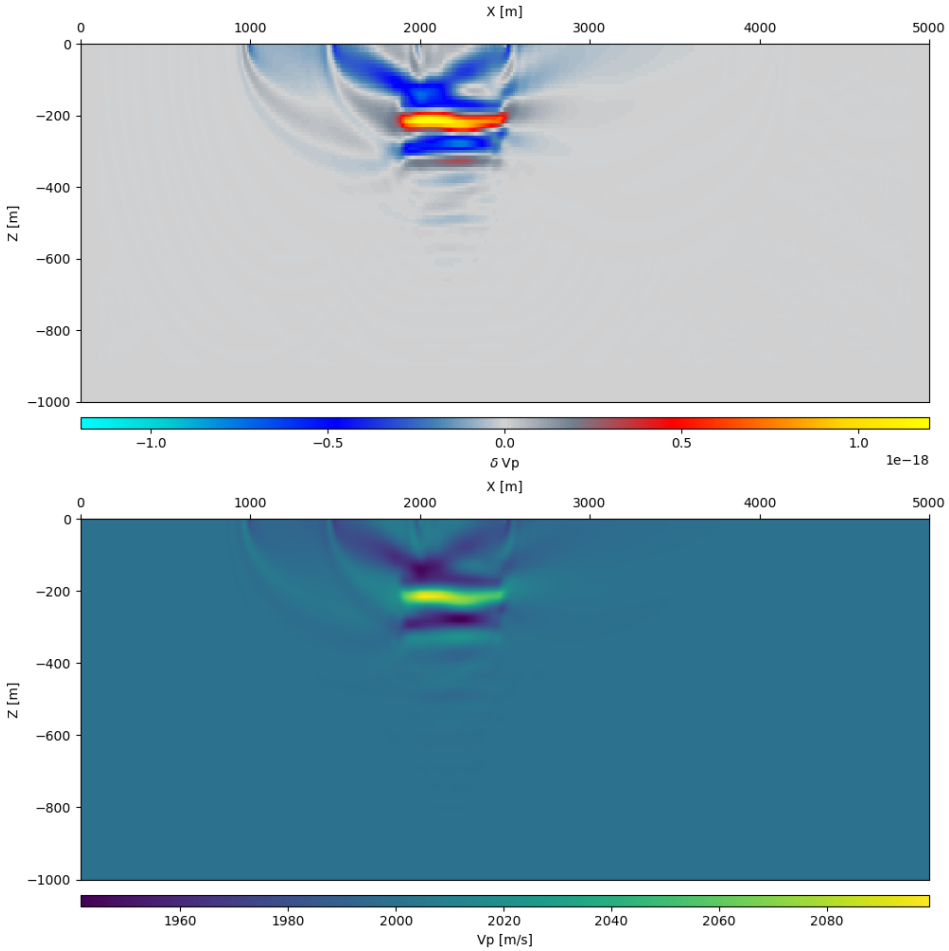
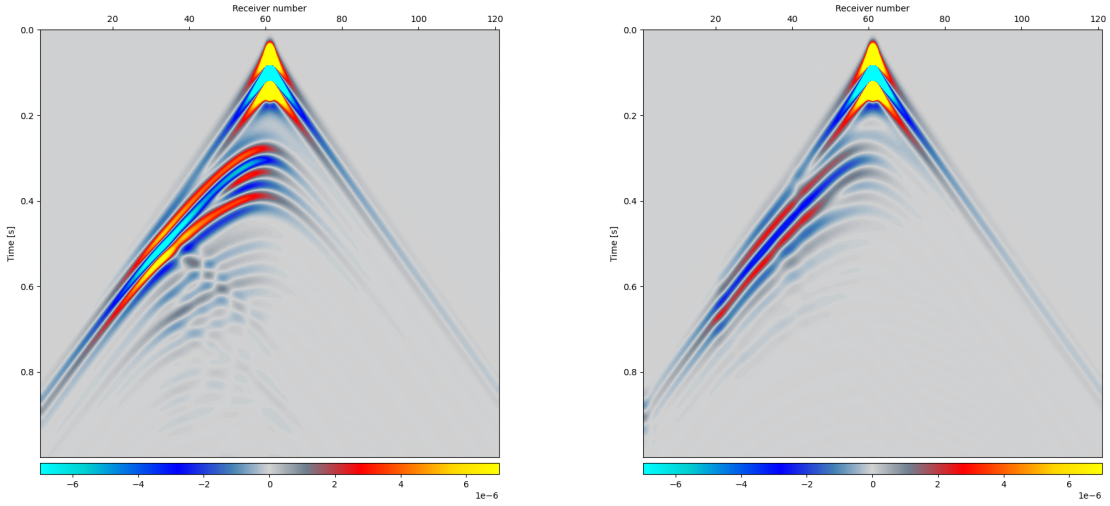


Figure 5.4: Top: The model gradient $\delta\tilde{\mathbf{m}}_k$. Bottom: The inverted model



(a) **(b)**

Figure 5.5: Shot record 0 (source located at shot position 0). (a): The observed data from the true model, (b): Synthetic data from the inverted model.

6 Fault Assessment

The ability to detect faults in a seismic survey can be useful in predicting the potential fluid migration, which would be of great interest in the field of CO_2 storage for example [Lubrano-Lavadera et al., 2018]. In the following I will present a number of simple models with the goal of demonstrating the ability of the adjoint method to reproduce discontinuities in seismic models. At the end of this section I will explore the effects of random noise on the results.

6.1 Two box model

This first model is almost a complete copy of the test example discussed in the previous section, just with a little added complexity in form of another box anomaly. The initial model is identical to the one used in the test example, as is the wavelet. The true model, together with the observed seismic and adjoint sources, can be seen in figure 6.1 and 6.2. The objective is to see if the gradient from the adjoint method is able the reproduce the step like structure of the two boxes. Looking at the seismic in figure 6.2 it looks very similar to the test example in the previous section, figure 5.2. However, there is a small visible kink at $0.4s$ between receiver number 40 and 50, which is not present in the one box example.

In figure 6.3 the result of the inversion is shown. Just like in the single box problem the inversion is able to capture the horizontal extension of the anomaly quite accurately, but the bottom is less well defined. The step like structure is clearly visible in the gradient for the top part of the anomaly. The step on the bottom become more of a smooth transition between the two boxes in the gradient. Some of the explanation for this can be found in the smoothing process that occur when the gradients are summed together. I will explore this a bit further in the following.

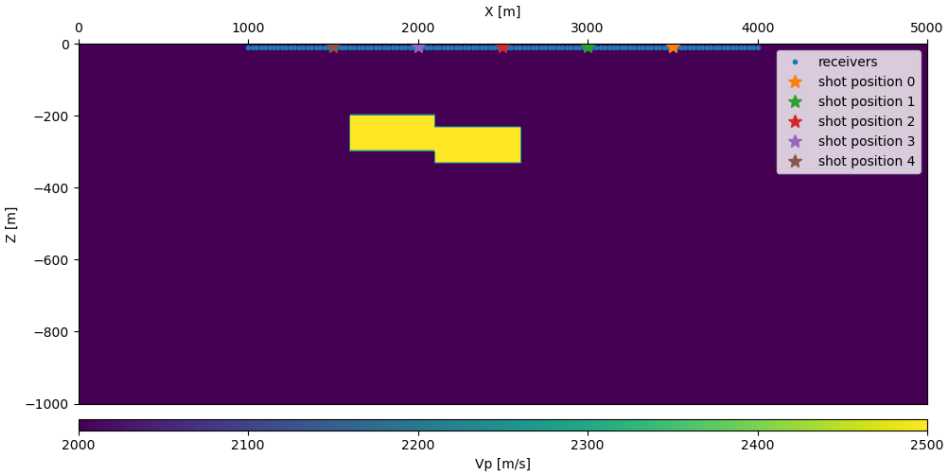


Figure 6.1: V_p for the true model. Density is constant at $\rho = 1600 \frac{kg}{m^3}$, and shear velocity is zero, $V_s = 0 \frac{m}{s}$

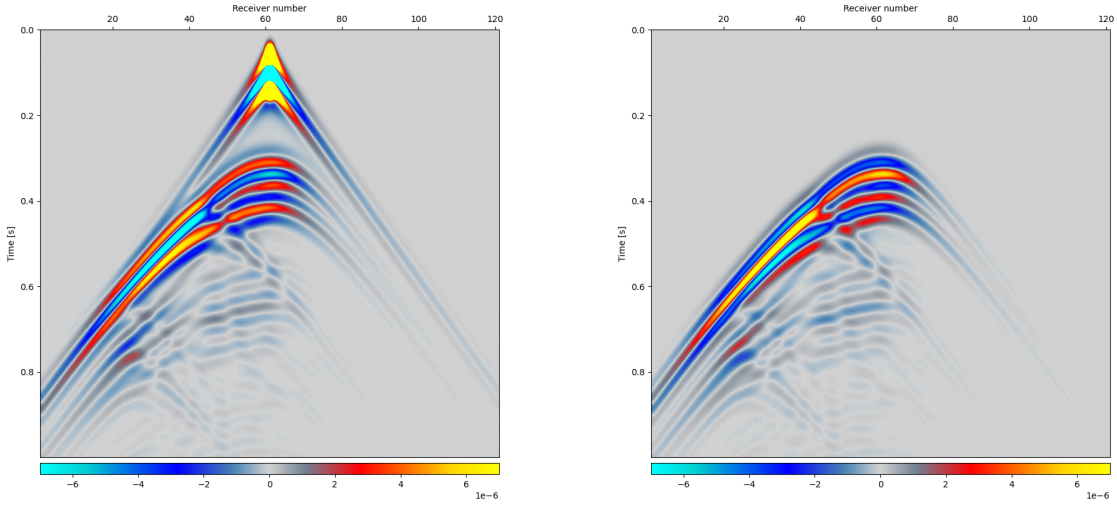


Figure 6.2: Left plot show the observed seismic, and right show the adjoint sources for the true model shown in figure 6.1 and prior model shown in figure 5.1

6.2 Gradient summation

The combined gradient, as shown in figure 6.3, has the advantage of being able to illuminate the box from different directions, creating a more complete picture. There is a downside however, and it comes from the smoothing process that is involved in combining the gradients. To illustrate this I have included all the gradients, one from each

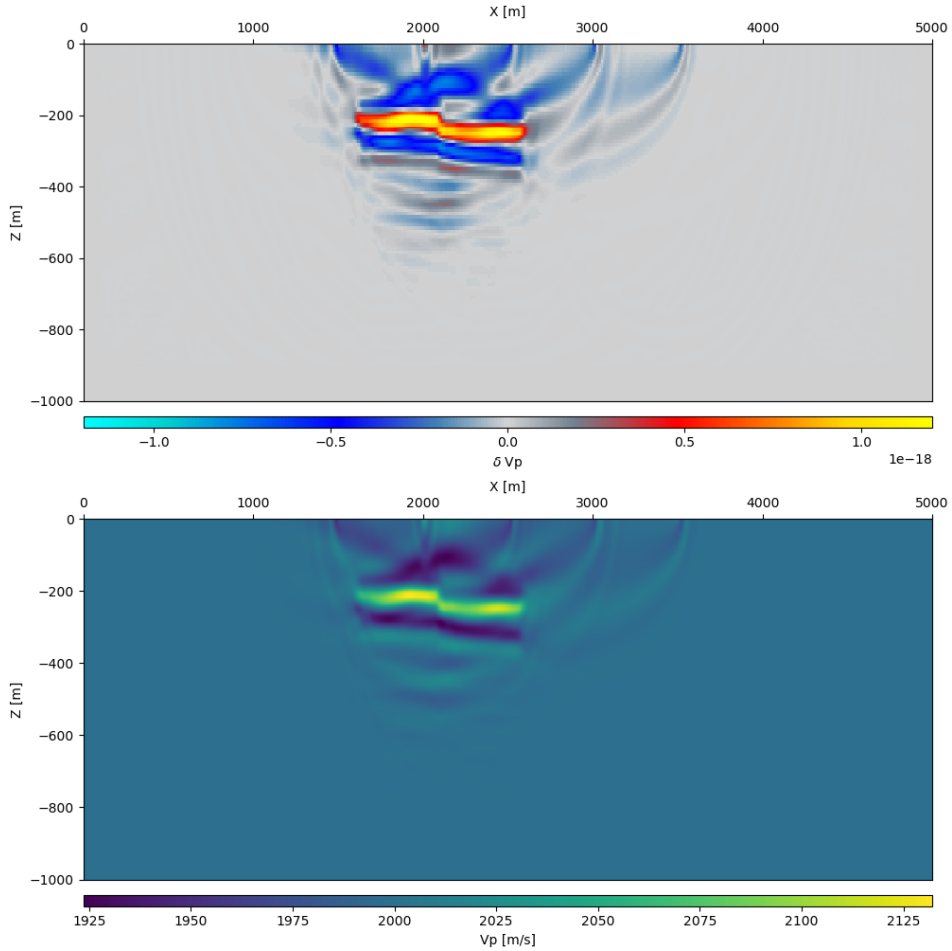


Figure 6.3: Inversion result for the two box model. Top: The model gradient $\delta\tilde{\mathbf{m}}_k$. Bottom: The inverted model

shot position, in figure 6.4. The ability to see the step structure is actually not possible in almost all the individual gradients, except for shot position 3. These plots illustrate very well why multiple shot positions are needed, but also how combining them might remove vital information. Looking at the gradient from shot position 3 the step is very well illuminated, as the shot is almost right above the contact point of the two boxes.

As one would expect, the information contained in the gradient depend on the distance from the source to the area of interest. This is also visible from the amplitude of the gradients, which naturally makes the combined gradient a linear combination weighted

by some measure of the inverse offset.

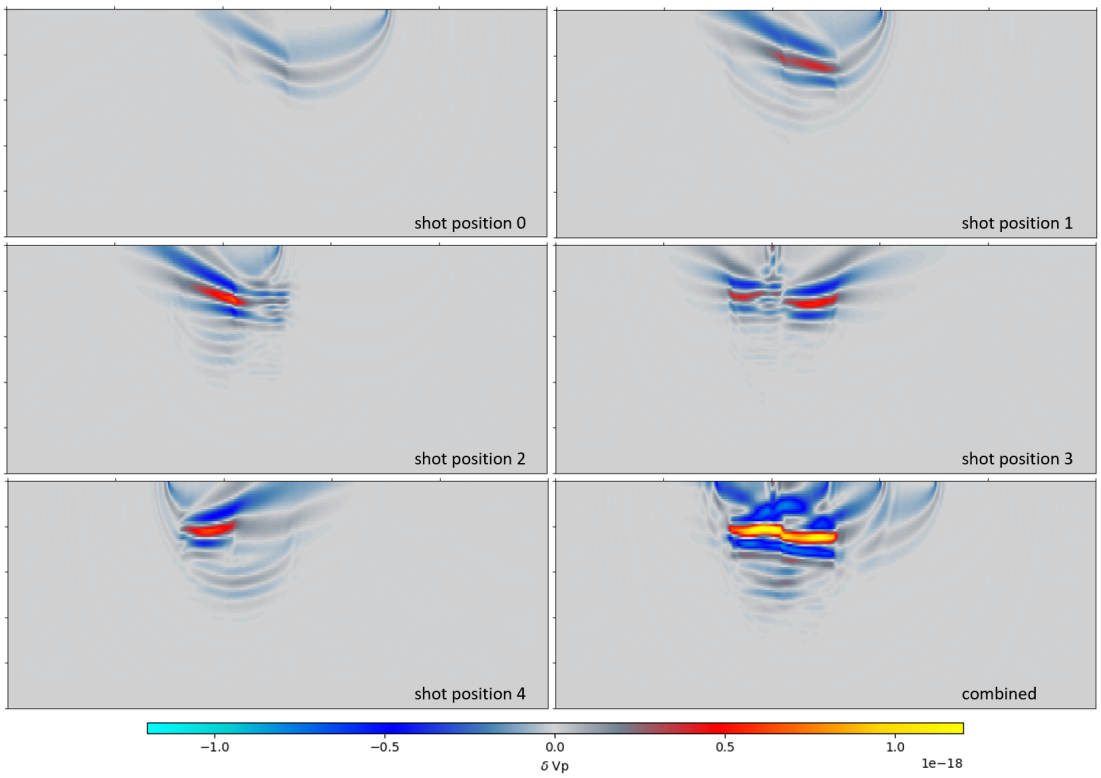


Figure 6.4: Model gradients for the two box model. Individual gradients from all 5 shot positions, as well as the combined gradient in the lower right corner.

6.3 Horizontal reflector with fault

From the two box model we learned that discontinuous horizontal variations are possible to detect. In this section I will lower the complexity and look at a horizontal reflector with a vertical fault of $30m$ in height. The hope is that the fault location and size will be easily detectable in the model gradient, as well as seeing how the method handles a reflector that stretches the entire model domain. The prior model is again completely uniform with $V_p = 2000\frac{m}{s}$, $\rho = 1600\frac{kg}{m^3}$ and $V_s = 0$, see figure 5.1. The true model can be found in figure 6.5. The inverted model can be found in figure 6.6. I have not included the model gradient, as the inverted model and model gradient are proportional

for one iteration.

The reflector is well mapped in the illuminated areas, and the fault is not to be mistaken. The discontinuous nature of the true model is represented, not just in location, but also in size. As mentioned earlier, the size of the dominant wavelength relative to the structure you wish to resolve is key. It is however possible to resolve structure much smaller that the dominant wavelength, as proven here. The dominant wavelength of the source is $\lambda \approx 166m$, making the fault size to dominant wavelength ratio $\approx \frac{2}{11}$.

The reflector in this example is perfectly flat, as is the mesh grid used in the spectral element modelling, but what happens when the reflector is not flat? I have previously mentioned that one of the advantages of the finite element method is the ability to create grids that honour the sloping boundaries of the model. In the following I will therefor show two models, where the only difference between them is the mesh grid used.

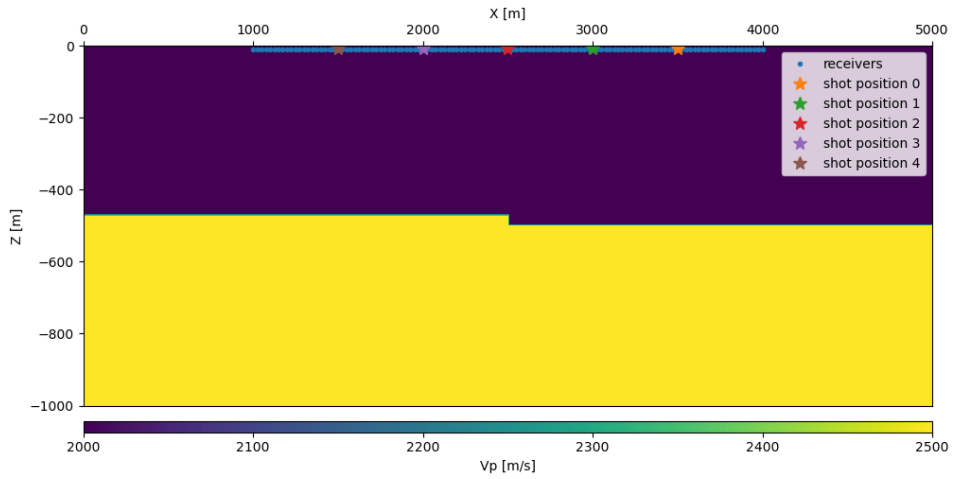


Figure 6.5: V_p for the true horizontal reflector with fault model. Density is constant at $\rho = 1600\frac{kg}{m^3}$, and shear velocity is zero, $V_s = 0\frac{m}{s}$

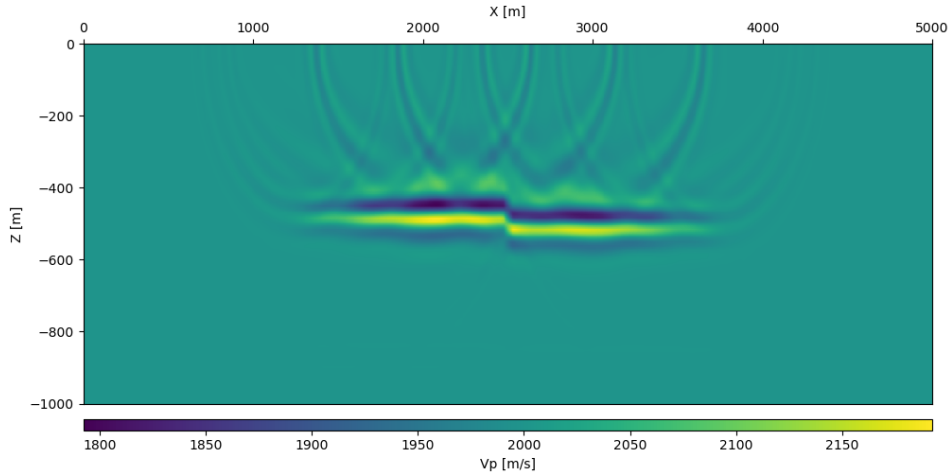


Figure 6.6: Inversion result for the horizontal reflector with fault model after one iteration.

6.4 The importance of a well designed mesh

One of the ideas behind a mesh that honour sloping boundaries, is to save time in the simulation. Any type of smooth boundary could be modelled by a perfectly flat grid with no spatial variation, by increasing the resolution of the grid. This is the strategy that is needed in typical finite difference modelling, but with finite element modelling the grid can simply follow this boundary, and potentially save significant computing power.

I have made one velocity model and two mesh grids. There are two ways to define a velocity model in SPECFEM2D,

1. define areas of the mesh when creating the mesh and assign each area specific values,
2. create a unstructured (x, z, V_p) file that then gets interpolated onto the given mesh.

For more info see the SPECFEM manual [Komatitsch et al., 2020]. I have made use of option 2. Option 1 can potentially give the most accurate model representations, but it makes the mesh generating process much more complicated and time consuming.

Looking at figure 6.7 it is easy to see the difference between the velocity model interpolated onto the square grid (see figure A.1 right), and the grid that try to honour the reflector geometry (see figure A.1 left). Both grids have the same number of elements,

the only difference is the internal shape. The question is now, does it actually make any difference in the modelled seismic? To answer this I have made a forward calculation for each of the two models, see figure 6.8. The difference between the two seismograms is significant. The error introduced by the interpolation onto a square mesh have a similar effect to adding small faults along the reflector, which introduce modelling noise. The main reflector is still well represented, but scattering from the small faults justifies the use of a mesh that follow reflector geometry. [7]

6.5 Curved reflector with fault

In this section I want to explore how the inversion performs when the reflector is curved, and the fault is on a slope. Using the model seen in 6.7c as the true model. The prior model is still completely uniform with $V_p = 2000\frac{m}{s}, \rho = 1600\frac{kg}{m^3}, V_s = 0$, but using the mesh seen in figure A.1 left. The inversion result can be seen in figure 6.9. The curved reflector is well represented in the inversion, and the fault is still visible but not as dominating as in the horizontal reflector problem. Part of the explanation comes from the fact that the fault is not as well illuminated, there is no shot position to the left of the fault. Another explanation can be that the slope might soften the effects of a vertical fault [8]. I say that the fault is still visible, even though it could easily be mistaken for a smooth transition when looking at the inversion. The smooth nature of the inverted model comes from the combining of the different shot record gradient estimations. Figure B.1 top left (no noise) show the gradient estimation from shot record 4, the one closest to the fault. There it is easy to see that the discontinuous nature of the fault leave a clear imprint in the model gradient.

[7] Another benefit of using a mesh that follow reflector geometry is the possibility of using layer dependent element spacing. The maximum allowed size of the elements in the mesh depend on the wavelength of the propagating waves. It is therefor common practice to vary the size of the elements depending on the propagation velocity of a given layer.

[8] in the limit where the reflector slope approaches vertical the vertical fault signal would approach zero.

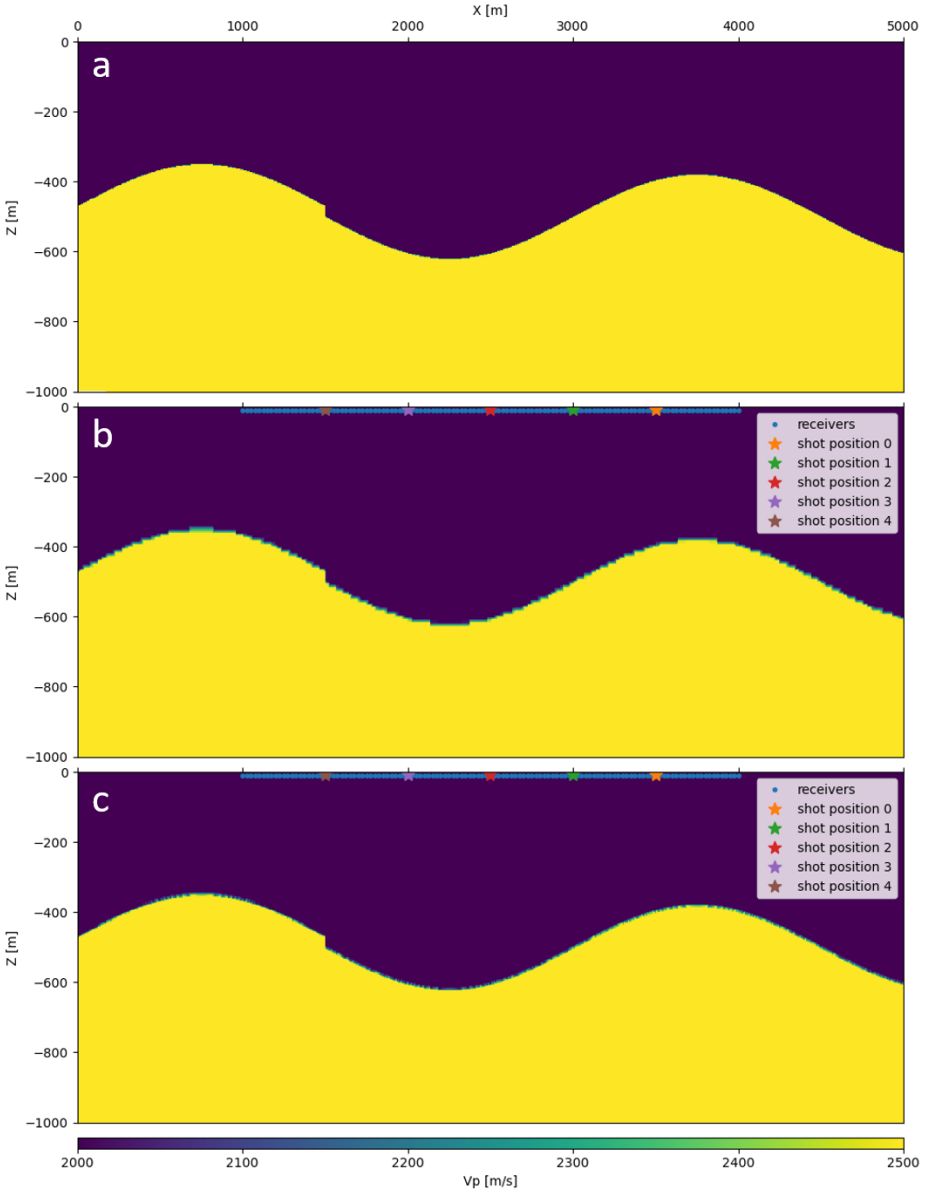


Figure 6.7: Comparison between velocity model interpolated onto a square mesh (b) vs onto an adapted mesh (c). (a) show the true velocity model, (b) shows (a) interpolated onto a square grid (figure A.1 right), and (c) shows (a) interpolated onto an adapted grid (figure A.1 left)

The big question when dealing with any kind of inversion routine is: how does it handle noise? I will explore this in the following subsection.

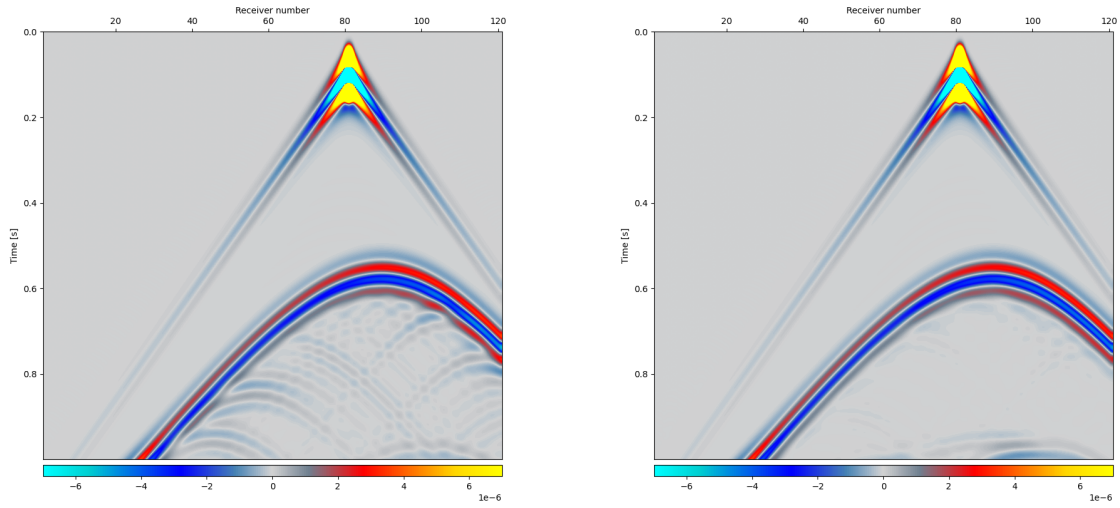


Figure 6.8: Comparison of forward calculation between two different models, for shot position 1. Left is from model (b) in figure 6.7 and right is from model (c).

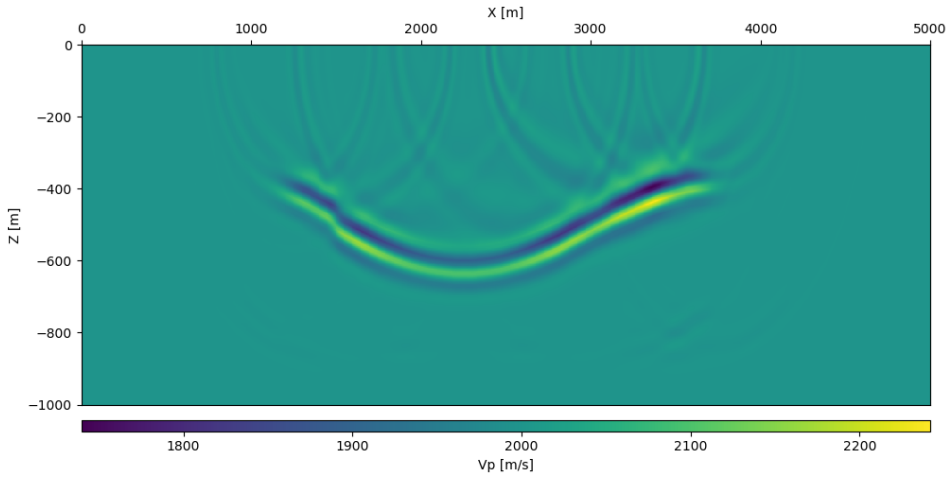


Figure 6.9: Inversion result for the curved reflector with fault after one iteration.

6.6 The adjoint method and random noise

One of the classical problems within inversion theory is how to deal with noise. Any physical measurement is associated with noise, and the perfect inversion method would only fit the part of the data that is not noise, such a method sadly does not exist. What I will explore here is how is the inversion result of the curved reflector with fault affected by the introduction of random noise. Let $p_0(\mathbf{x}_r, t; \mathbf{x}_s)$ be the observed seismogram at

receiver location $\mathbf{x}_r$ and source position $\mathbf{x}_s$, $\eta(t;\mathbf{x}_r)$ be a random realisation of uniformly distributed white noise and $S(t)$ be the source time function used to generate p_0, then the noisy seismogram can be expressed as,

$$\widehat{p}_0(\mathbf{x}_r,t;\mathbf{x}_s) := p_0(\mathbf{x}_r,t;\mathbf{x}_s) + \varepsilon\Big(\eta(t;\mathbf{x}_r) * S(t)\Big) \tag{6.1}$$

where $*$ denotes convolution and ε is a constant used to scale the noise levels. The reason for convolving the white noise with the source time function is to get noise with a wavelength like the real signal. This also helps the simulation from becoming unstable. I have included noise of three different ε. The three ε has been chosen as follows:

1. $\left|\varepsilon\Big(\eta(t;\mathbf{x}_r) * S(t)\Big)\right| \leq \frac{2}{3}\max(|\delta\mathbf{p}|)$
2. $\left|\varepsilon\Big(\eta(t;\mathbf{x}_r) * S(t)\Big)\right| \leq \max(|\delta\mathbf{p}|)$
3. $\left|\varepsilon\Big(\eta(t;\mathbf{x}_r) * S(t)\Big)\right| \leq 2\max(|\delta\mathbf{p}|)$

where $\delta\mathbf{p}$ is the observed seismic with the direct wave subtracted. The noise is scaled with respect to the signal from the reflector and not the direct wave. In figure 6.10 the three different noise levels together with the resulting gradients can be found.

The worst case noise scenario, case 3., the reflector is still clearly visible, but the noise does certainly have an effect on the gradient. The noise mostly manifests itself at the top of the model gradient which might be due to the large correlation between the direct wave and noise signals. Even though the noise does not hide the true reflector, it does hinder the inversions ability to reproduce the seismic signal from the reflector. In figure 6.11 it is clear that the simulated data from the inverted model does actually succeed in removing most noise, but the amplitude of the reflector almost disappears with increasing noise levels. The amplitude of the reflector could be increased by increasing α, but that would also increase the strong noise induced anomalies of the model

gradient. It might be possible to help the inversion by introducing some post processing of the model gradient, like weighing the values by depth.

The fault is not very visible from the combined gradient, just like in the example with no noise, however in figure B.1 the fault still leave a clear imprint in the gradient for all three noise levels.

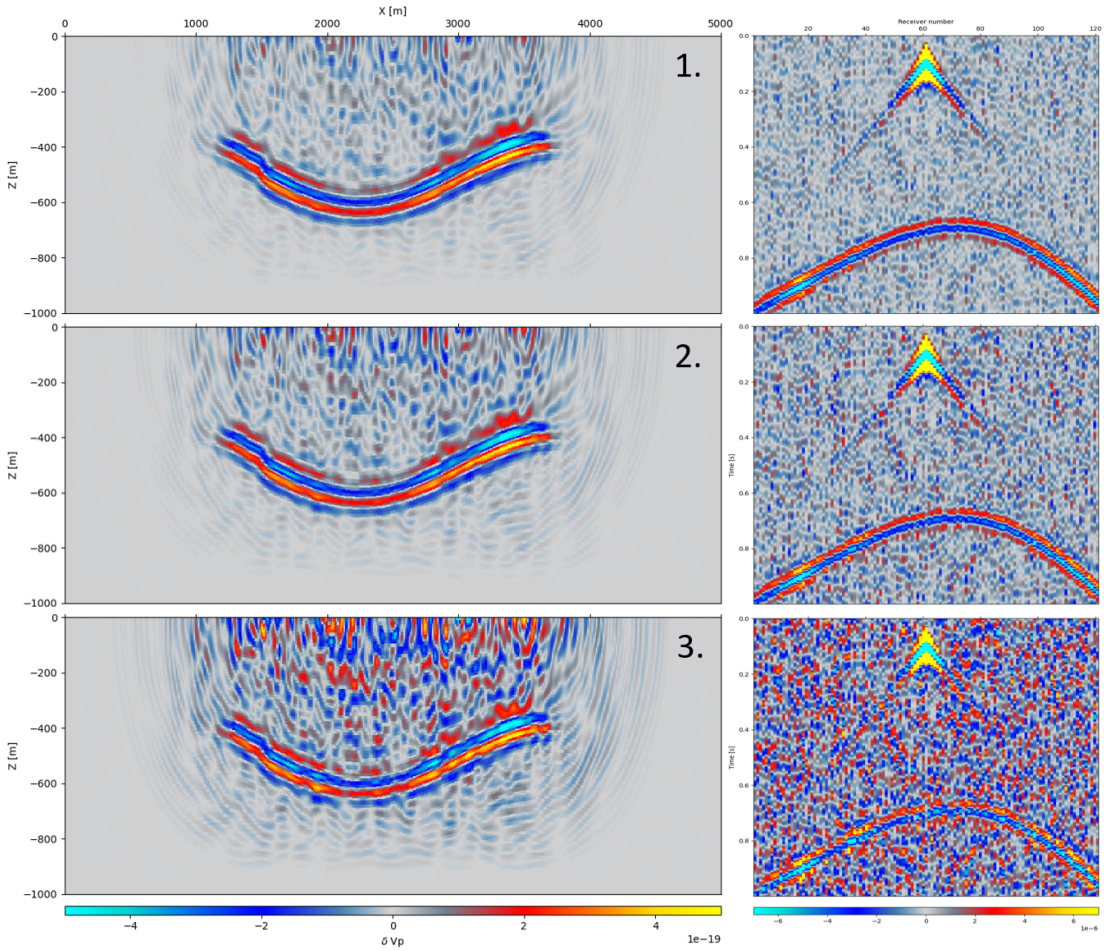


Figure 6.10: The observed data from shot position 2 with three different levels of noise (right) with the corresponding gradient calculation from the adjoint method (left). The numbering correspond to the three different noise levels listed above.

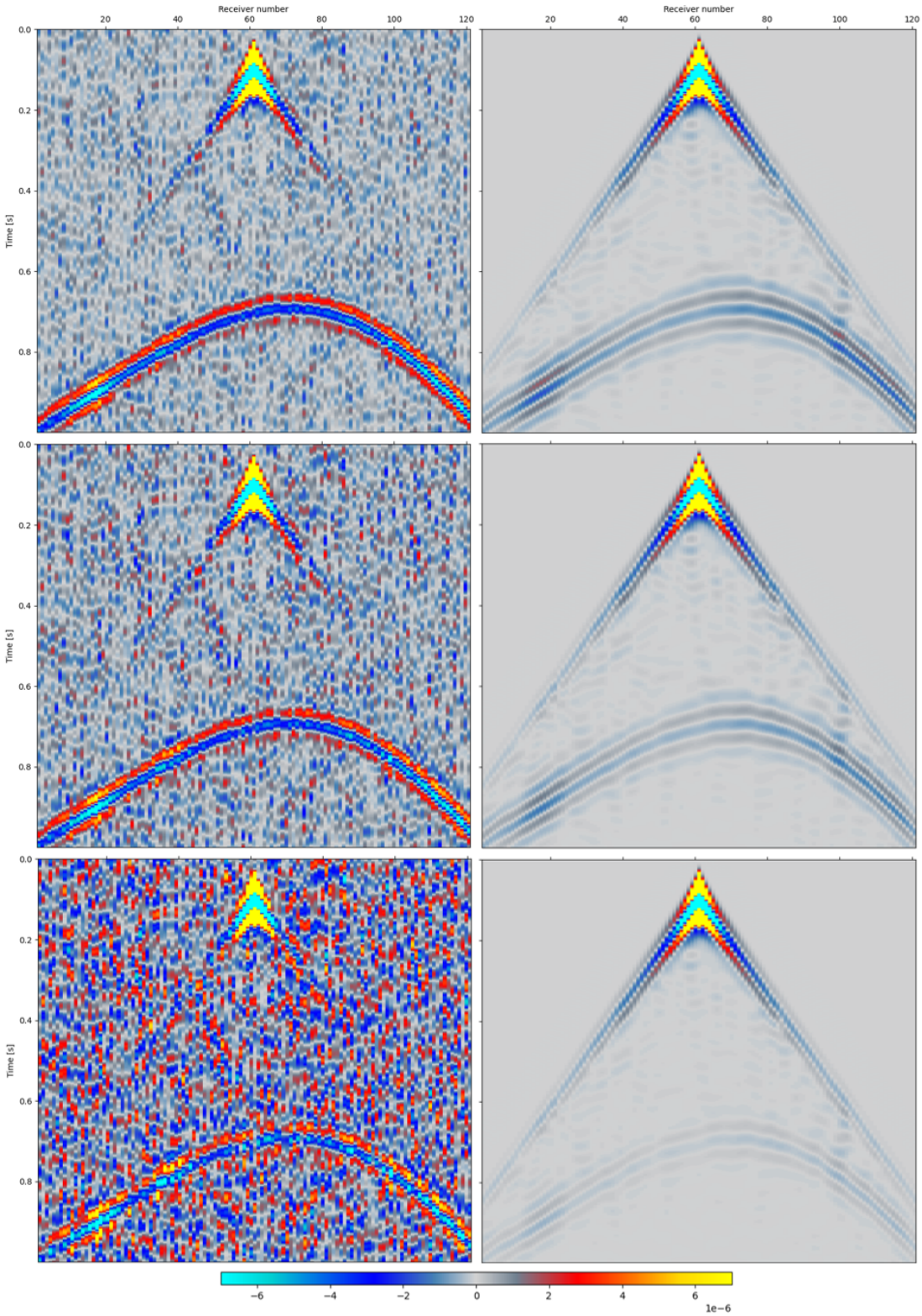


Figure 6.11: Observed data from shot position 2 with three different noise levels (left) and simulated data after inversion (right).

7 A semi realistic problem

In this section I want to demonstrate how the flattening algorithm can be used to create a mesh as well as a prior model. I then want to test how the adjoint method reacts to faults that are not included into the prior model. At the end I introduce a form of *model noise* to my prior model.

7.1 Flattening and mesh generation

In section 4.1 I convert a picture of a depth migrated seismic section into a data set. The purpose of this data is to extract information about the geometry of the area, and to do this I use the flattening algorithm discussed in section 3. The result of the flattening can be seen in figure 7.1. The flattening process works, almost. The strong reflectors are for the most part flattened correctly, but some areas, especially the bottom, do suffer from the data quality. The data set is practically binary, which means any event, no matter amplitude, are weighted equally when calculating the structure tensor. It helps to convolve the image with a 2D Gaussian kernel, but it is a problem. The black and orange lines seen in figure 7.1 represent the inverse flattening transformation. The orange lines are specifically picked to design the mesh and prior model.

The orange lines are then used to specify layers in the mesh. Note that the lines are used to define the geometry, not the layer boundaries. As mentioned in section 4 there are two wells covering the area, and to limit the size of the model I only focus on the area close to well B (CMP number 1572). The orange lines are imported into Gmsh to create the finite element mesh, see figure 7.2.

7.2 Designing the prior model

Here I will show a very simple, and perhaps naive, way to design a prior model that involves the inverse flattening transform. At well B the vertical velocity and density vari-

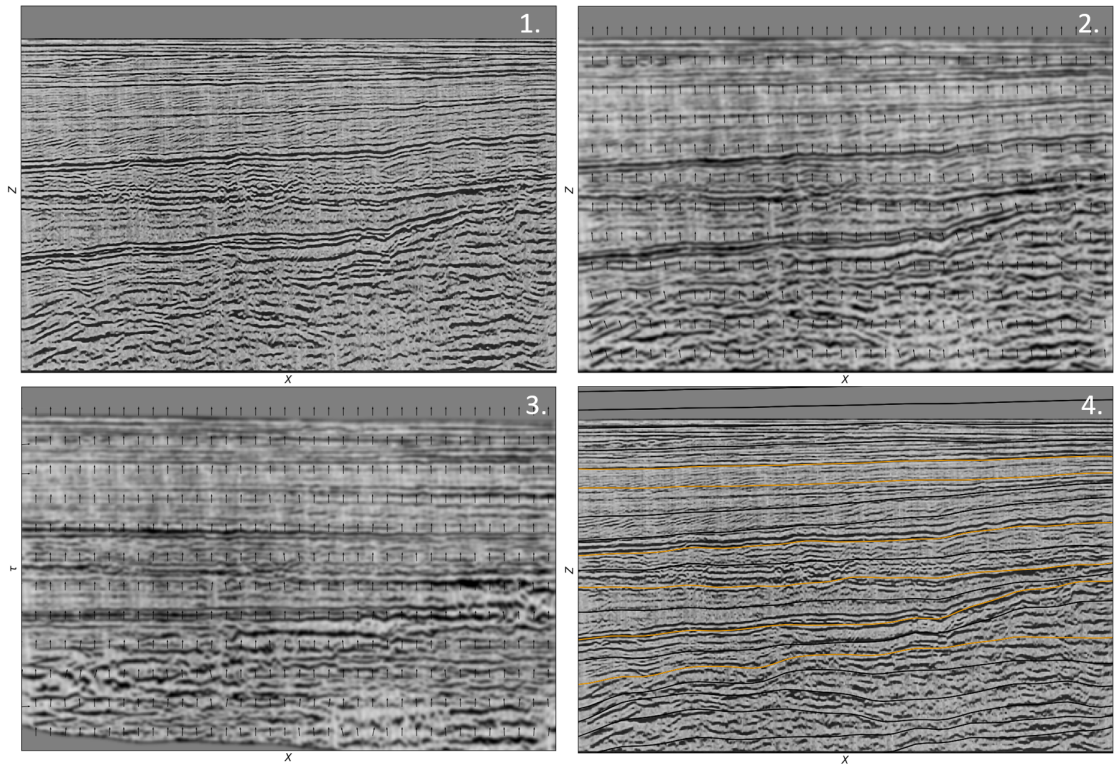


Figure 7.1: Flattening process of the depth migrated seismic section for the Viking Graben data set. 1. The input image; 2. Input image convolved with gaussian kernel with structure tensor overlay (black arrows); 3. Warped image after 5 iterations of the flattening algorithm with structure tensor overlay; 4. Input image with inverse flattening transform overlay (black and orange lines).

ation are measured, and assuming that the measurements and interpretations are correct, it is reasonable to expect that the immediate area around the well share these characteristic. I take this assumption further and assume that the entire 2D profile share these velocity and density variations. Horizontally extending the log interpretations over the entire model domain and using the inverse flattening transform to reshape into the correct geometry, see figure 7.3. Figure 7.4 show the velocity model of the actual prior model used, which is a cutout of the full model shown in figure 7.3[9]

[9]Note that the X axis point opposite directions in the two figures

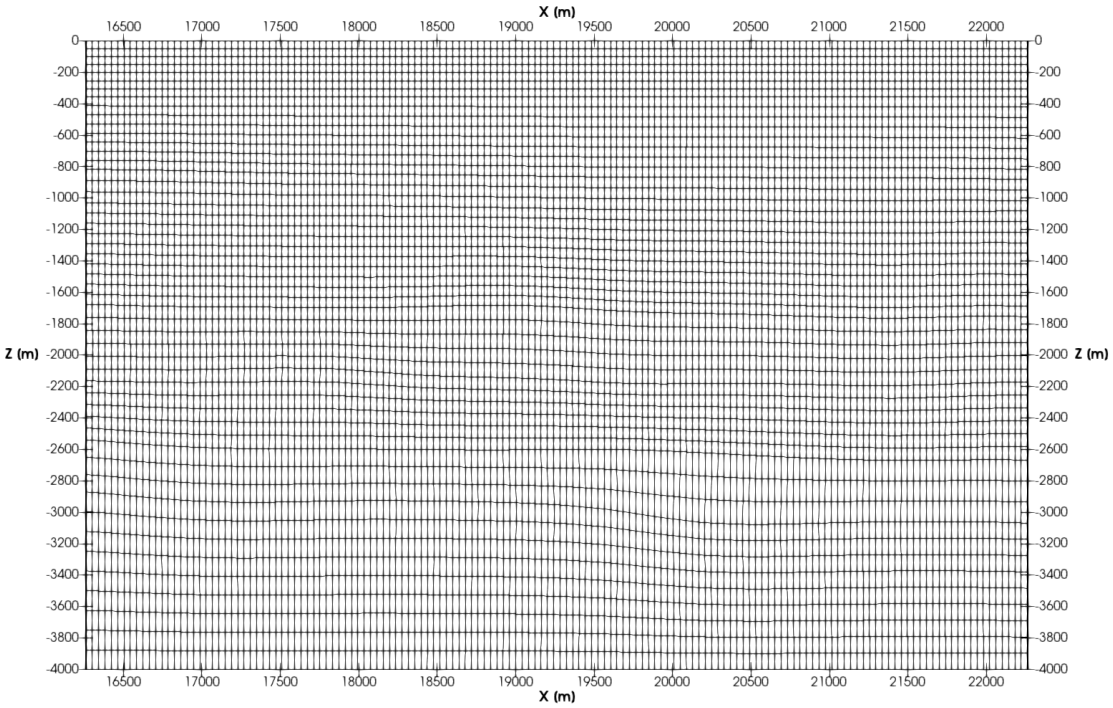


Figure 7.2: Mesh used for the Viking Graben model near well B. The element size is not to scale, the real mesh used in SPECFEM2D is much finer yielding it impractical to show. The X coordinate is in a reference coordinate system following the seismic segy file. Well B is located at X=21262m

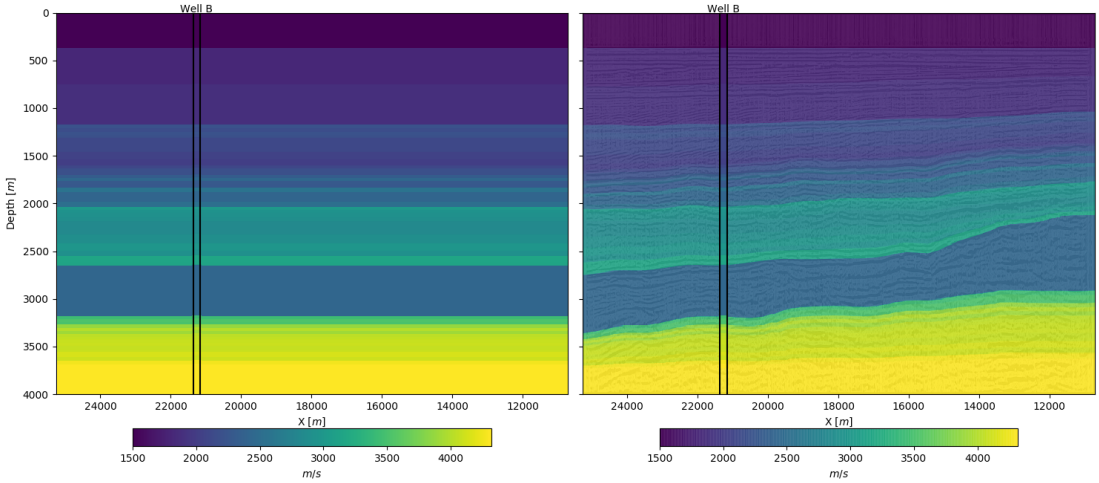


Figure 7.3: Horizontally extended V_p log (left) transformed into the prior model using the inverse flattening transform. The velocity model on the right is plotted on top of the depth migrated seismic.

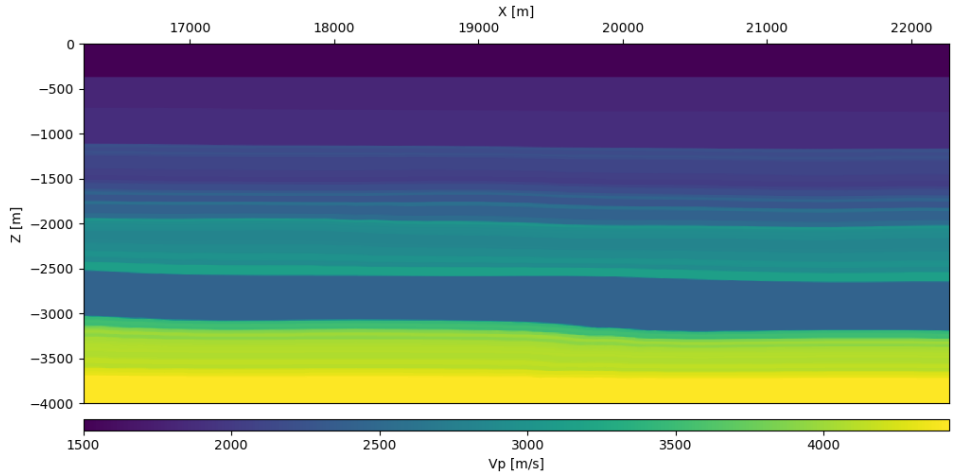


Figure 7.4: The prior V_p model for the Viking Graben area near well B. Well B lie at the right of the model at X=21262m

7.3 Designing the true model

Because this is a synthetic problem the true model also has to be designed, and for this I start with the prior model and add small layer extrusions. The true model can be seen in figure 7.5. The only thing differentiating the true model from the prior mode is the layer extrusions highlighted by white boxes. I use two different shot positions with corresponding receiver setups. Shot position 0 and corresponding receiver positions can be seen in figure 7.2, and shot position 1 is exactly the same setup but everything is translated 1000 m to the left. Figure 7.6 show the ”observed” shot records (modelled with the true model) together with the adjoint sources. From this it is clear that the extrusions, or at least some of them, are visible on the seismograms. This means that the adjoint method should be able to highlight these on the model gradient.

I mentioned in section 4 that I would not be using the wavelet shown in figure 4.4. The reason for this is because it containes a lot of high frequencies, which demands a fine mesh to be able to resolve those shorter wavelengths, which in turn increases the computing time needed. Instead I use a Ricker wavelet with a dominant frequency of 20 Hz.

To further reduce the computing time needed the models are acoustic, shear velocities are set to zero everywhere. Eventhough the reduced computing time is a major reason for only doing acoustic models, it is not the real reason why I have chosen to set the shear velocities to zero. The first model I used had a top layer of around 350 m with water, $V_p \approx 1500\frac{m}{s}$ and $V_s = 0$, followed by solid layers, $V_s \neq 0$. I wanted to include this fluid to solid interface, to keep the problem as realistic as possible, but the SPECFEM2D adjoint simulation could for some reason not handle this transition, and would only propagate waves from the adjoint sources in the fluid region. To avoid having to figure out where in the many thousands lines of code the problem originated and coming up with a solution, I set the shear velocities to zero everywhere.

The boundary conditions used for the modelling are absorbing everywhere.

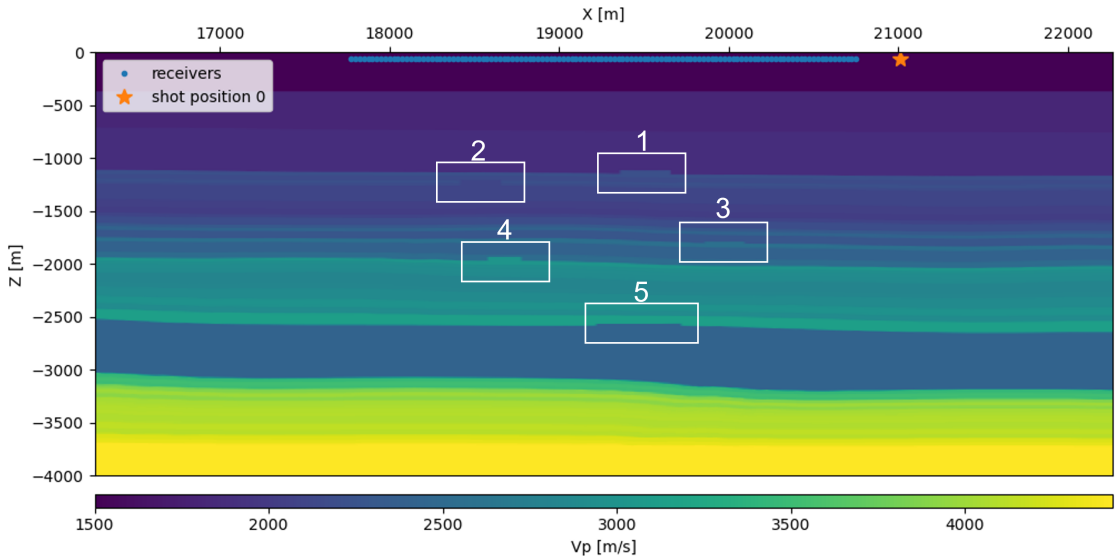


Figure 7.5: The true V_p model. The white boxes highlight the layer extrusions that differentiate the true model from the prior model. The orange star and blue dots show the locations of shot position 0 and the corresponding receivers respectively.

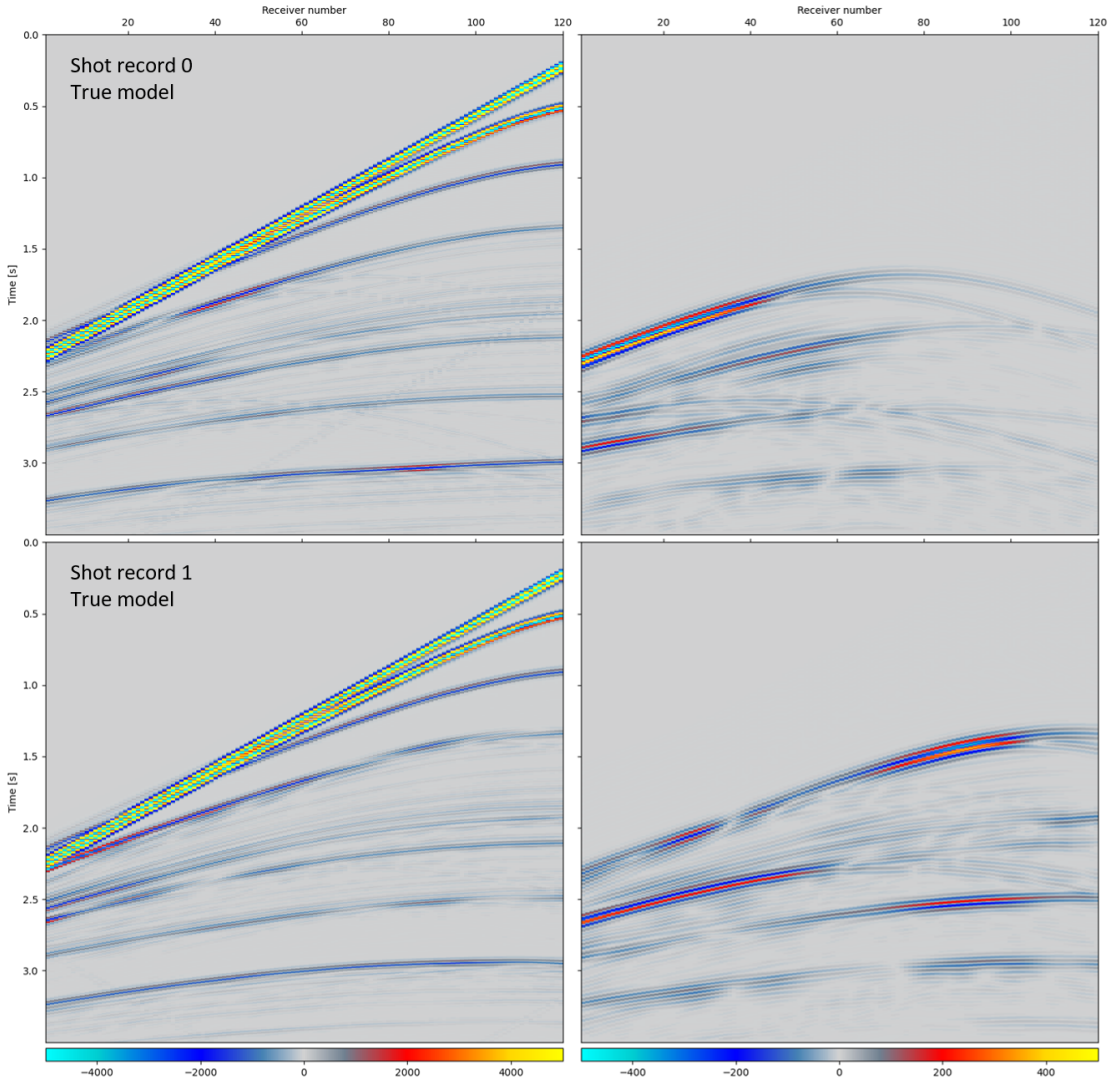


Figure 7.6: Observed shot records (left), modelled with the true model (see figure 7.5), and the adjoint sources (right), residuals of the observed and synthetic seismograms. Note the different colorbars for left and right.

7.4 Fault Assessment

Before I start showing the results of the adjoint method I want to establish a bit of notation to keep track of it all. Let $F_0(X, Z)$ and $F_1(X, Z)$ represent the model gradients at point

(X,Z) of shot position 0 and 1 respectively. Let D represent a depth weighing function,

$$D : F(X, Z) \rightarrow F(X, Z) \cdot Z$$

and let H represent a simple highpass filter[10].

Figure 7.7 show the resulting model gradient. The model gradient without any processing suffer from two things: amplitude decrease rapidly with depth and long wavelength signals. The long wavelength signals are dealt with by applying H and the amplitude is helped by applying D. The good news is that the resulting gradient $D[H(F_0 + F_1)]$ do show strong signal at all of the fault locations. The bad news is that it also show fault like signals at locations where no faults exist in the true model.

7.4.1 Model noise

In section 6 I demonstrated the effects of uncorrelated random noise in the seismograms on the resulting model gradients. Here I want to show what happens when the prior model is slightly altered. To create the prior model the well log and inverse flattening transform was used, but what happens if noise is added to the V_p log? All the layers of the prior model would have slightly different velocities compared to the true model. Figure 7.8 show the new velocity log used to create a new prior model, called *well noise model*, by following the procedure explained in 7.2. The model gradient when using the *well noise model* as prior model can be found in figure 7.9.

7.4.2 Discussion and conclusion

The adjoint method estimates a model gradient for the prior model without noise, see figure 7.7, that show strong signals at the locations of the five missing faults. It also

[10] In practice H is purely a image processing technique that return the difference between the input and the input convolved with a Gaussian kernel.

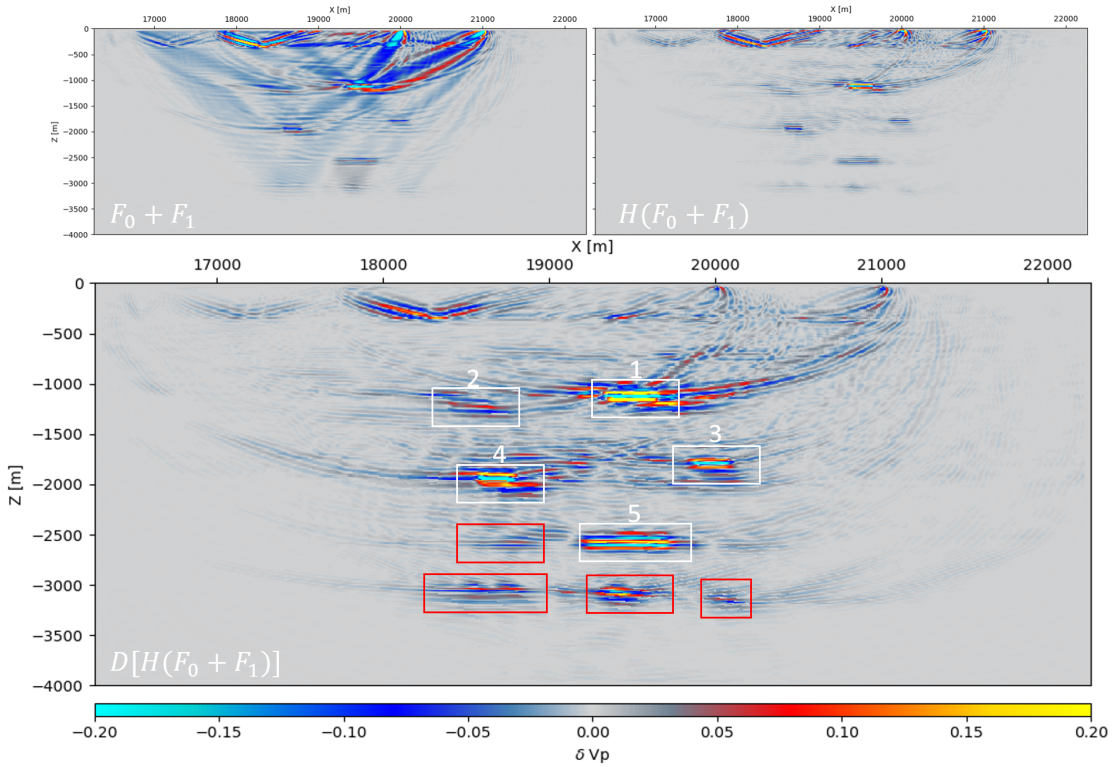


Figure 7.7: Model gradient from the adjoint method. Top left is the sum of the two gradients, one for each shot and receiver setup. Top right show is a highpass filtered version of top left. Bottom is a highpass filtered and depth weighted version of top left. Every gradient is normalized before plotted. The white boxes are located at the fault locations of the true model, and the red boxes are located a fault like signals where no fault exist.

show signals similar to the fault signatures at locations that are perfectly represented by the prior model. In figure 7.7 I highlight four such signatures, and what they all have in common is that they lie such that a ray from the source to the false fault location will intersect a true fault location. Another thing they have in common is that they lie on a strong reflector, see 7.4. Because the rays traveltime is slightly altered by the faults (layer extrusions), the locations of reflectors under these faults are moved slightly up or down, in travletime, compared to the reflectors of the prior model. The false faults are just ”echos” of the true faults.

The estimated model gradient for the *well noise model* is more complicated. Because

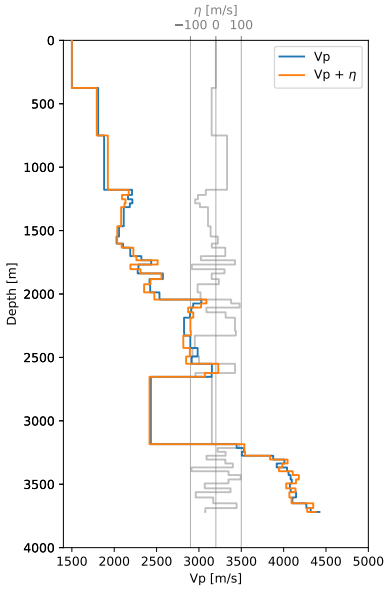


Figure 7.8: Blue show the original V_p log found in figure 4.3 and orange show the new velocity log. The orange log is used to create the prior model: *well noise model*

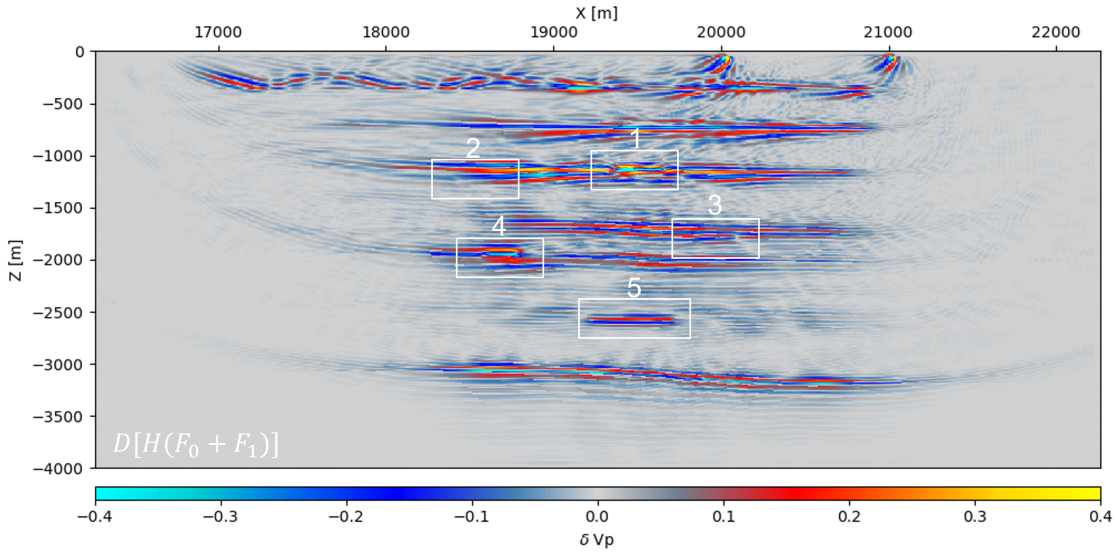


Figure 7.9: Model gradient from the adjoint method using *well noise model* as prior model. The white boxes are located at the fault locations of the true model

all the layer velocities are slightly off compared to the true model, the reflectors are also going to be located at the wrong travletime. This means that the observed and synthetic

seismograms no longer match and the layer boundary reflections no longer cancel each other in the residual. The adjoint method will therefor highlight all the layer boundaries that have a wrong average velocities above. This makes the model gradient far more complicated, see figure 7.9, but not hopeless. Fault nr. 1,4 and 5 (see numbering of the white boxes) still leave signatures that make them distinguishable. Fault nr. 1 manifests itself as a discontinuity of the strong horizontal reflector located around Z = -1250 m. Fault nr. 4 and 5 are both easy to spot. Fault number 2 is not possible to see, and also the weakest in terms of velocity contrast in the true model. Fault number 3 does have some of the characteristics of fault nr. 1, but would be difficult to identify without knowing where to look. Something important to note here is that I only use two source and receiver setups. If this was an inversion of the real Viking Graben prospect, there would be a source and receiver setup for every 25m. I chose to limit the problem to only have two source and receiver setups, to save computing time. I believe more shot positions would improve the results, because the shot position seems to have a big impact in the methods ability to reproduce discontinuities, as seen in figure 6.4.

I have made a second type of model noise that I have included in appendix C. I will not discuss it here as the effects are similar to the *well noise model*.

8 Conclusion

In this thesis I have presented a way of extracting geometric information from a depth migrated seismic profile to be used in generating a mesh and prior model. A simple flattening algorithm is used for this purpose, where the inverse flattening transform can be used to create a mesh for spectral element modelling that honour the sloping boundaries of the model. A practical example is shown where the inverse flattening transform is used to create the mesh as well as the prior model from well log data.

The adjoint method has been used together with simple 2D models to demonstrate the methods ability to reproduce sharp contrasts and discontinuous velocity structures from multiple shot records. The method is able to locate and horizontally delimit anomalies not accounted for by the prior model. The illumination of the anomaly by the shot and receiver locations play an important role in the methods ability to highlight sharp boundaries. However, the process of combining these shot and receiver locations into one gradient estimate, can also result in a smearing that hides the sharp contrast. Shot positions located close to the fault lead to model gradients that show discontinuous reflector geometry that can help identify sharp contrasts not accounted for in the prior model.

I have also demonstrated how the gradient estimations by the adjoint method is affected by different levels of uncorrelated random noise of wavelet frequency. The method seems to be robust towards this type of noise which mostly manifests it self as large anomalies near receiver/source locations, resulting in weak velocity contrasts of the inverted model, but the reflector geometry is still preserved.

Lastly, I have used the adjoint method to try and locate small fault like structures in a semi-realistic synthetic model, based on geometry and well log data from the Viking Graben area in the North Sea. The methods ability to locate and partly reproduce the small faults not included in the prior model, depend greatly upon the quality of the prior model. By introducing small errors in the prior velocity model, it is possible to partly or

entirely hide the signature of the faults. Because these results are only based on the use of two shot positions, it is very possible that the method would be able to distinguish the faults from the model noise by including more shot locations.

Acknowledgements

I want to thank Klaus Mosegaard for his inspiration and guidance throughout the process of this thesis.

I would also like to thank the people behind SPECFEM2D, Gmsh and SeisFlows for making their many thousands hours of work available to the public. A study like this would not be possible without their work.

References

– [Tarantola, A. 1984] Tarantola, A. 1984: Inversion of seismic reflection data in the acoustic approximation *Society of Exploration Geophysicists; GEOPHYSICS, VOL. 49, NO. 8, P. 1259-1266*

– [Fichtner, A. 2011] Fichtner, A. 2011: Full Seismic Waveform Modelling and Inversion. *Springer*

– [Igle, H. 2017] Igle, H. 2017: Computational Seismology - A Practical Introduction. *Oxford University Press*

– [Komatitsch et al., 2012] Komatitsch, D.; Vilotte, J.-P.; Cristini, P.; Labarta, J.; Le Goff, N.; Le Loher, P.; Liu, Q.; Martin, R.; Matzen, R.; Morency, C.; Peter, D.; Tape, C.;Tromp, J.; Xie, Z. (2012), SPECFEM2D v7.0.0 [software], *Computational Infrastructure for Geodynamics*, doi: NoDOI, url: https://geodynamics.org/cig/software/specfem2d/

– [Geuzaine et al., 2009] Geuzaine, C.; Remacle, J.-F., 2009: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331, 2009.*

– [Parks, D. 2010] Parks, D. 2010: Seismic image flattening as a linear inverse problem. *Center for Wave Phenomena, Colorado School of Mines*

– [Keys et al., 1998] Keys, R.G.; Foster, D.J. 1998: Comparison of Seismic Inversion Methods on a Single Real Data Set. *Society of Exploration Geophysicists*

– [Gordon, C. 1996] Gordon, C.; Webb, D. 1996: You Can't Hear the Shape of a Drum *American Scientist Vol. 84, No. 1, pp. 46-55*

- [Tarantola, A. 2005] Tarantola, A. 2005: Inverse Problem Theory - and Methods for Model Parameter Estimation. *Society for Industrial and Applied Mathematics*
- [Modrak et al., 2018] Modrak, R.T.; Borisov, D.; Lefebvre M.; Tromp J., 2018: SeisFlows - Flexible waveform inversion software. *Computers and Geosciences 115 (2018) 88-95*, https://doi.org/10.1016/j.cageo.2018.02.004
- [Fichtner et al., 2006] Fichtner, A.; Bunge, H.-P.; Igel, H., 2006: The adjoint method in seismology - II. Applications: traveltimes and seisitivity functionals. *Physics of the Earth and Planetary Interiors 157 (2006) 105-123*.
- [Lubrano-Lavadera et al., 2018] Lubrano-Lavadera, P.; Senger, K.; Lecomte, I.; Mulrooney, M.J.; Kühn, D., 2018: Seismic modelling of metre-scale normal faults at a reservoir-cap rock interface in Central Spitsbergen, Svalbard: implications for CO_2 storage. *Norwegian Journal of Geology 65-83.*
- [Komatitsch et al., 2020] Komatitsch, D., Vilotte, J.P., et al. 2020: SPECFEM2D - User Manual (Version 7.0). Computational Infrastructure for Geodynamics
- [Komatitsch et al., 2020] Komatitsch, D. et al. 2020: SPECFEM3D Cartesian - User Manual (Version 3.0). *Computational Infrastructure for Geodynamics*

Appendix A Mesh

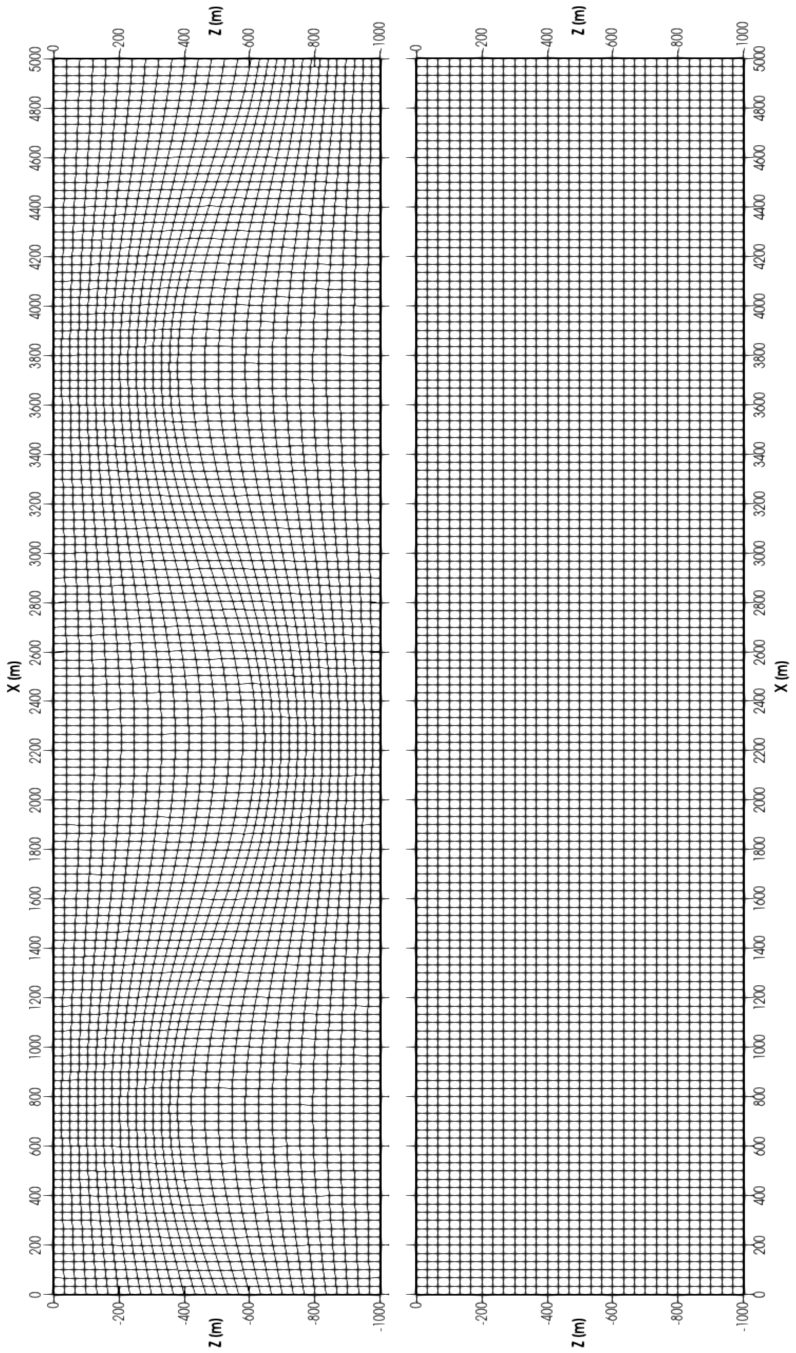


Figure A.1: Left: Simple mesh that follow the spatial variation of the velocity model. Right: Simple mesh with no spatial variation. Created in Gmsh

Appendix B Curved reflector and random noise

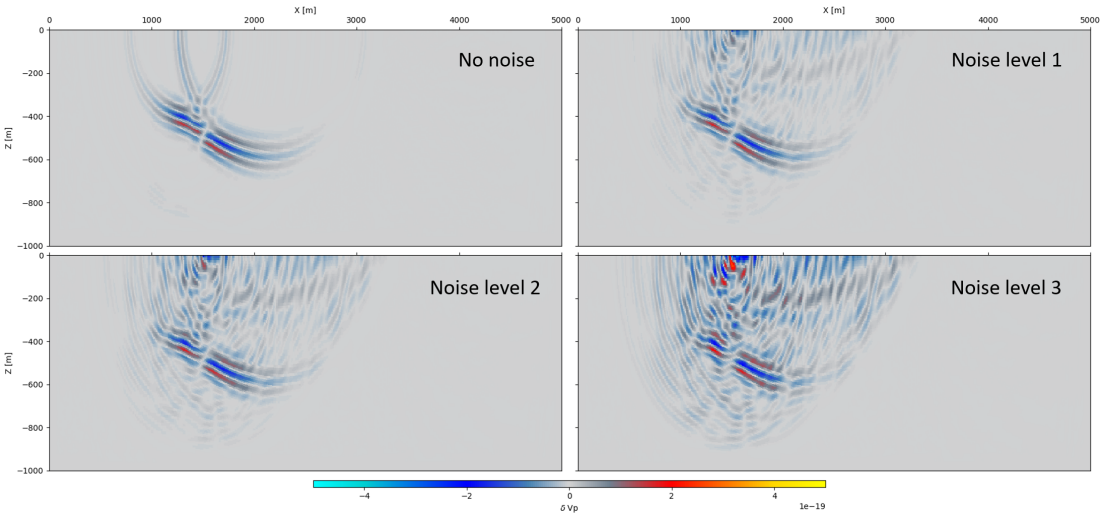


Figure B.1: Model gradient estimation from shot position 4 for *the curved reflector with fault* with and without noise. Noise level 1,2 and 3 are explained in section 6.6.

Appendix C Low frequency model noise

In section 7 I show the effects of model noise by changing the propagation velocities of each layer in the prior model, compared to the true model. Here I will show the effects of another type of model noise. I simply add a low frequency velocity perturbation to the prior model shown in figure 7.4. Figure C.1 show the velocity perturbation. Figure C.2 show the model gradient after adding the velocity perturbation.

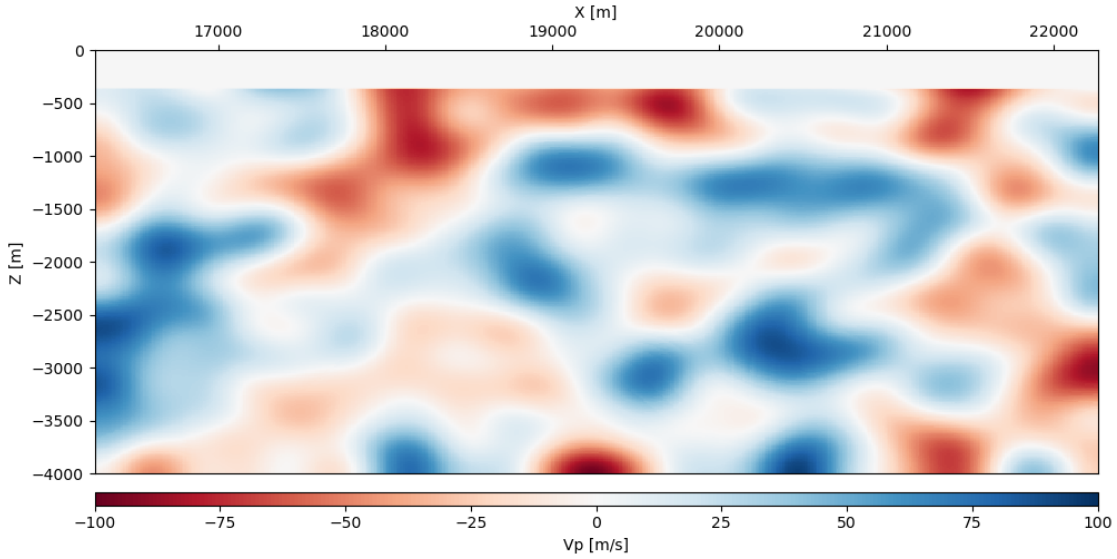


Figure C.1: Low frequency velocity perturbation to be added to the prior model.

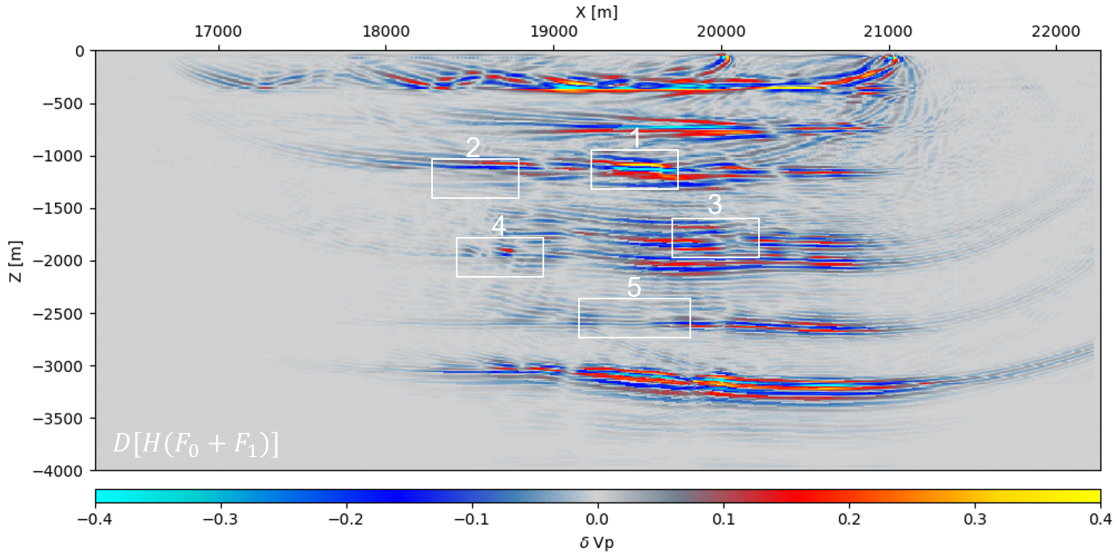


Figure C.2: Model gradient from the adjoint method using low frequency noise model as prior model. The white boxes are located at the fault locations of the true model