

Imperial College London  
Department of Earth Science and Engineering

# 3D Elastic Full-Waveform Inversion

Lluís Guasch  
2011

Supervised by Prof Mike Warner

Submitted in part fulfilment of the requirements for the  
degree of Doctor of Philosophy of Imperial College London  
and the Diploma of Imperial College London



## **Legal disclaimer**

I herewith certify that all material in this dissertation which is not my own work has been duly acknowledged.

Lluís Guasch



# Abstract

Full Waveform Inversion (FWI) is a depth imaging technique that takes advantage of the full information contained in recorded seismic data. FWI provide high resolution images of subsurface properties, usually seismic velocities or related parameters, although in theory it could image any property used to formulate the wave equation. The computational cost of the methodology has historically limited its application to 3D acoustic approximations but recent developments in hardware capabilities have increased computer power to the point that more realistic approximations are viable. In this work the traditional acoustic approximation is extended to include elastic effects by introducing the elastic wave equation as the governing law that describes wave propagation.

I have developed a software based on finite-differences to solve the elastic wave equation in 3D, which I applied in the development of a full-waveform inversion algorithm. The software is fully parallelised for both distributed and shared-memory systems. The first level of parallelisation distributes seismic sources across cluster nodes. Each node solves the 3D elastic wave equation in the whole computational domain. The second level of parallelisation takes advantage of present multi-core computer processor units (CPU) to decompose the computational domain into different volumes that are solved independently by each core. Such parallel design allows the algorithm to handle models of realistic sizes, increasing the computational times only a factor of two compared to those of 3D acoustic full-waveform inversion on the same mesh. I have also implemented a perfectly matched layer absorbing boundary condition to reproduce a semi-infinite model geometry and prevent spurious reflections from the model boundaries from contaminating the modelled wavefields.

The inversion algorithm is based upon the adjoint-state method, which I reformulated for the wave equation that I implemented, which was based on particle-velocities and stresses, providing a comparison and demonstration of equivalence with previous developments.

To examine the performance of the code, I have inverted several synthetic problems of increasing realism. I have principally used only pressure sources and receivers to assess the potential of the method's application to the most common industry surveys: streamer data for offshore and vertical geophones (only one component) for onshore exploration surveys. The results show that the imaged properties increase with the heterogeneity of the models, due to the increase in P-S-P conversions which provides the main source of information to invert shear-wave velocity models from pressure sources and receivers.

It remains to demonstrate the inversion of field datasets and my future research project will focused on achieving this goal.



# Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Prof Mike Warner, who has always encouraged me to trust my own initiatives, and given me the freedom to explore my own ideas. Mike's advice and supervision have taken the level of this thesis far beyond my initial expectations. I am also thankful for the excellent example he has provided as a successful physicist and researcher.

The members of my research group have contributed immensely in the success of the project. I would like to especially acknowledge Adrian Umpleby, who has played a major role in the development of my project, for his patience and effort. I gratefully thank Ivan Stekl for his advice and guidance from the early stages of this research. Nuno Vieira and Nikhil Shah have been the best work mates one could possibly desire; I will always remember both the serious and fun moments we spent together. I also had the pleasure to work with the newest members of our research group Tenice Nangoo, Ahmed Al-Yaqoobi and Sainath Lakshminarayanan.

During my project I had the opportunity to work as an intern at Hess. I record my gratitude to Faqi Liu, who supervised my work with invaluable contributions and introduced me to industry oriented research. I gratefully acknowledge the funding sources that made my Ph.D. work possible. I was funded by BG, BP, CGGVeritas, Chevron, ConocoPhillips, ENI, Maersk, Nexen, Rio Tinto and the Department of Business Enterprise & Regulatory Reform (BERR).

My time at Imperial has given me the opportunity to rediscover the joy of friendship. Jim, Giacomo, Raj, Amer, Manu and Luca have participated more than they know in the success of this work. My sporadic visits to Barcelona have given me the extra boost of energy I needed in the hard moments of these years thanks to my long-life friends, I miss you guys.

To my mum, dad and sister I can only say: *que faria sense vosaltres? gràcies per ensenyar-me a ser qui sóc.*

To my grandma's: *avia i iaia, gràcies per la vostra energia i incondicional devoció, són molt més del que qualsevol nétil pot demanar.*

Muriel, this thesis is dedicated to you; without you none of this would have been possible.



# Contents

<b>1</b>	<b>Introduction</b>	<b>19</b>
1.1	Chapter Overview . . . . .	19
1.1.1	Original work . . . . .	21
1.2	Preamble . . . . .	22
1.3	Seismic modelling . . . . .	24
1.3.1	Finite Differences . . . . .	24
1.3.2	Other numerical methods . . . . .	29
1.3.3	Absorbing boundary conditions . . . . .	30
1.4	Full-waveform inversion . . . . .	32
1.4.1	Travel-time tomography . . . . .	33
1.4.2	History of full-waveform inversion . . . . .	36
1.4.3	Time and other domains . . . . .	38
1.4.4	Inversion of elastic parameters . . . . .	40
1.4.5	Evolution to 3D . . . . .	41
<b>2</b>	<b>Modelling of seismic waves</b>	<b>43</b>
2.1	Theoretical background . . . . .	43
2.1.1	The wave equation . . . . .	43
2.1.2	Alternative formulations of the wave equation . . . . .	48
2.2	Implementation . . . . .	48
2.2.1	Numerical methods . . . . .	49
2.2.2	Finite-difference stencil . . . . .	52
2.2.3	Stability conditions . . . . .	57
2.2.4	Source insertion . . . . .	58
2.2.5	Absorbing boundary condition . . . . .	63
2.3	Optimization . . . . .	66
2.3.1	Adaptive computational domain . . . . .	68
2.3.2	Parallelisation . . . . .	69
2.4	CPU-times and memory requirements . . . . .	70
2.5	Conclusions . . . . .	73
<b>3</b>	<b>Inverse theory</b>	<b>75</b>
3.1	Introduction . . . . .	75
3.2	Linear matrix algebra background . . . . .	79
3.2.1	Vector spaces . . . . .	80
3.2.2	Linear maps and matrices . . . . .	83

3.2.3	Differentiation of vectors and matrices . . . . .	87
3.3	Local methods in inverse problems . . . . .	88
3.3.1	Least squares . . . . .	88
3.3.2	Iterative methods . . . . .	92
3.3.3	Adjoint-state method . . . . .	100
<b>4</b>	<b>Elastic FWI</b>	<b>105</b>
4.1	Matrix algebra formulation . . . . .	106
4.2	Functional analysis derivation . . . . .	117
4.3	Comparison between formulations . . . . .	121
4.4	Implementation . . . . .	124
4.4.1	Code structure . . . . .	124
4.4.2	Strategies to reduce storage of wavefields . . . . .	127
4.5	Conclusions . . . . .	131
<b>5</b>	<b>Pure transmission datasets</b>	<b>133</b>
5.1	Introduction . . . . .	133
5.2	Eight cubes model . . . . .	135
5.3	Two channel model . . . . .	138
5.4	Conclusions . . . . .	139
<b>6</b>	<b>Surface acquisition</b>	<b>141</b>
6.1	Introduction . . . . .	141
6.2	One cube validation tests . . . . .	143
6.3	Two channel surface acquisition . . . . .	150
6.4	Marmousi 3D . . . . .	157
6.5	Objective function evolution . . . . .	171
6.6	Conclusions . . . . .	175
6.6.1	CPU times . . . . .	176
<b>7</b>	<b>Conclusions and further research</b>	<b>179</b>
7.1	Cross-talk between elastic parameters . . . . .	180
7.2	Anisotropic elastic inversion . . . . .	181
7.3	Field data application . . . . .	182
<b>Bibliography</b>		<b>195</b>
<b>Appendix A</b>		<b>198</b>
<b>Appendix B</b>		<b>202</b>

# List of Figures

1.1	Application to the eikonal equation to calculate travel-times (in red) for an experiment with multiple sources and receivers placed on the surface. . . . .	35
1.2	(a) True model used to generate ‘real’ data, note the inversion of velocities in the layer between 700 and 1100 m depth; (b) TT fails to recover the low velocity layer in the middle of the model because the first arrivals never travel through it; (c) FWI, on the other hand, can recover it because the energy has travelled through the low velocity region and its effect is recorded in the full seismogram. . . . .	36
2.1	A volume $V$ delimited by a surface $S$ . $\mathbf{T}(\mathbf{n})$ is the traction applied to $dS$ which has a normal unit vector $\mathbf{n}$ (adaptation of the figure from chapter 2 in [Aki and Richards, 2002]). . . . .	46
2.2	Finite-element mesh for subset of a 2.5D version of the Marmousi model. . . . .	51
2.3	2D FD staggered stencil. The velocities (stresses) used to calculate each stress (velocity) are connected by solid grey lines. . . . .	52
2.4	Unit cell in 3D. . . . .	54
2.5	Stresses are defined at $n\Delta t$ and velocities at $(n + 1/2)\Delta t$ . The set of variables involved in the calculation of the central point on the right figure are highlighted in red. . . . .	54
2.6	The three particle-velocity dipoles simulate a point source centered at the normal stress point. Sources inserted using normal stresses do not need any dipoles because they are all defined at the same point. . . . .	60
2.7	Three perpendicular slices of a cubic homogeneous model with $V_P = 3000m/x$ and $V_S = 1732m/x$ , after 0.8 seconds showing the particle velocities in the $\hat{x}$ direction. In (c) the P-wave front is very weak because the $\hat{x}$ component of the particle velocity is perpendicular to the YZ plane. . . . .	61
2.8	Marmousi 2.5D model snapshots of the velocity $\hat{z}$ component. (a) XY: $z = 1000m$ ; (b) XZ: $y = 2400m$ ; (c) YZ: $x = 2000m$ . The red ellipse highlights the response to a step shaped irregularity in the seabed. . . . .	62
2.9	Seismograms for a shot generated using the acoustic (a) and elastic (b) wave equation. The shear wavefronts are recorded at later times creating all the late events recorded only in the elastic case. . . . .	63
2.10	Imaginary parts of the $x$ coordinate in continuous (a) and discrete (b) domains. In this schematic view, both figures depict a linear damping function although in practice an exponential shape performs better. . . . .	65

2.11	2D slices of 3D pressure wavefields. The model is a cube of size 200x200x200 nodes. The source is a ricker wavelet with peak frequency of 8 Hz and the grid spacing is 20 m. Both $V_P$ and $V_S$ are homogeneous with values 2500 and 1440 m/s respectively. Density is 2000 kg/m <sup>3</sup> everywhere. In the top row no PML was applied and the reflections from the boundaries are clearly visible as the wavefront advance through time. The middle row of snapshots were computed using 10 nodes/face PML and in this case the amplitude of the reflections have decreased considerably. The bottom shows snapshots with 20 nodes/face PML. All the pictures use the same amplitude scale.	67
2.12	Pressure snapshots taken at 0.3, 0.6, 0.9 and 1.2 seconds in a two half-spaces model. The red box encompass the points where the solution is computed.	68
2.13	The blue box reduces its size as the simulation is complete, collapsing around the receivers. Note that the wavefield have been zeroed outside the box to highlight the relevant energy to the receivers.	69
2.14	The master node passes information to all the slaves in the cluster using MPI (red area). Each node executes the instructions given by the master using all the available cores (blue area).	70
2.15	Memory requirement table and graph for different models. The thickness of the PML is 20 cells in all the models.	71
2.16	Effect of the loop order and the optimisation flags on the run-times. Note that the bigger the model the most accentuated the difference between them. The horizontal axis is logarithmic to avoid excessive concentration of points on the left of the plot.	72
2.17	Speed-ups for the tested cubic models using 8 threads.	73
2.18	The independence of the model size is proved by the invariance of the shape of the trend for each model (for the sake of clarity the vertical axis is logarithmic).	74
3.1	Schematic representation of the relation between physical quantities (model parameters) and empirical observations (data). In the depicted image, the model parameters are seismic velocities and density, and the data are pressure seismograms.	76
3.2	(a) mapping of grid-model to vector-model; (b) mapping of traces to vector-data	82
3.3	Solution space for a simple problem with two model parameters $m_1$ and $m_2$ ; (a) example of a local method (steepest-descent): starting at two different points $X_1$ and $X_2$ lead to different solutions; (b) example of a global method (Monte Carlo): the global minimum is found calculating many different values of $F$ and then choosing the model parameters that generate the minimum value of $F$ at $X_m$ . The background figure is from [Sirgue, 2003].	90

3.4	$+$ is the point where the gradient is computed, the red line is the gradient direction in (a) (with the arrow pointing downhill) which have residual values $\mathbf{d} - \mathbf{d}_{obs}$ shown in (b); $+$ is the position of the minimum of $F$ along the direction of the gradient, and it is achieved when the gradient is scaled by a factor $\alpha_0$ . . . . .	96
3.5	For two different functional curves along gradient directions (a) and (b) three different step length guesses are computed, leading to dramatically different results. $gs_1$ , $gs_2$ and $gs_3$ are the three different guessed step lengths leading to optimal step lengths of $\alpha_1$ , $\alpha_2$ and $\alpha_3$ . . . . .	97
3.6	Two different sets of points chosen to fit a parabola are shown in red and blue for two different objective function curves (a) and (b). $gs_{i,1}$ and $gs_{i,2}$ are the two values of the functional used to fit parabolas with minima $\alpha_i$ for $i \in 1, 2$ . Every point calculation costs an extra forward modelling, so this method is, computationally speaking, more expensive than the linear fit, although it may speed up convergence. . . . .	98
3.7	The grey paraboloid is the approximation of the solution space that results from assuming that the objective function is quadratic with respect to the model parameters. The red line indicates the direction of the gradient and the blue line the direction of the gradient scaled with the approximate diagonal Hessian. Note that the real shape of the objective function is absent in the plot. . . . .	100
4.1	Flowchart diagram of the inverse problem for any wave equation formulation.	125
4.2	Interaction between the scheduler node and the slaves. The former is responsible to distribute tasks across the slave nodes, which bear most of the computational load. . . . .	126
4.3	All the snapshots of the wavefield (above) are undersampled to reduce memory usage(below). . . . .	128
4.4	The absorbing boundary regions where the wavefield is not stored is depicted in light grey; the actual model, in dark grey, only is stored at the final time-step; the black region is where the wavefield is stored for all time-steps. The thickness of the black region is of two nodes, as the overlaid grid highlights. . . . .	130
5.1	Spatial distribution of sources and receivers. There are six plane wave sources (red) recorded only in their opposite faces (blue). The grey planes in the model represent the shape of the waves' path, and the arrows its direction. . . . .	134
5.2	2D slice of the model showing the evolution as iterations proceed. The The colourscale is black and white with the minimum and maximum values matching those of the true model. . . . .	135
5.3	Cutaways of the the starting (a), target (b) and recovered models for $V_P$ (c) and $V_S$ (d). . . . .	136
5.4	Transparency views of target (a) and recovered $V_S$ (b). The cutaways show a good match of amplitude in the recovered anomalies in the middle and a smooth transition to the background velocity in the edges. . . . .	137

5.5	Geometrical view of the model features: (a) to highlight the channel structure, transparency is applied to the lower and higher values; (b) 2D vertical slice of the model exhibiting the opposite nature of the channels velocities: the shallower has higher and the deeper has lower velocities than the surrounding background velocity field. . . . .	138
5.6	Spatial distribution of sources and receivers (see figure 5.1 for the description of colours, planes, etc) . . . . .	139
5.7	Horizontal view at a depth of 260 m of: (a) starting model, a 1D velocity gradient; (b) the target model; (c) recovered $V_P$ after five iterations; and (d) recovered $V_S$ after five iterations. Note that the colour scale has been clipped to brighten the channels. . . . .	140
6.1	Source signature and amplitude spectrum. . . . .	142
6.2	Experiment geometry depicting all 25 sources in red and 8100 receivers in blue. The shape of the wavefield in this case is spherical as shown by the grey isosurfaces which correspond to a constant amplitude values of the wavefield for the different shaped source. . . . .	143
6.3	The anomaly coordinates (given in nodes) are: $x \in \{45 \dots 55\}$ , $y \in \{45 \dots 55\}$ and $z \in \{14 \dots 19\}$ . . . . .	144
6.4	Vertical and horizontal slices of the target model for both positive (a) and negative (b) anomalies. Recovered models for the purely acoustic cases after 30 iterations (c) and (d). The $x$ and $y$ directions in the plot are equivalent due to the symmetry of the experiment. . . . .	145
6.5	A positive anomaly in $V_P$ is recovered well (a) and there is no trade-off with $V_S$ (b); a negative anomaly is only recovered partially (c) and there are no leaks to $V_S$ (d); a positive anomaly in $V_S$ , while leaving $V_P$ unchanged (e) as expected, gets stuck in a local minimum and therefore is not recovered (f); the recovered $V_S$ model for a negative anomaly does not converge to the global minimum either (h), although $V_P$ remains correctly unchanged (g). . . . .	148
6.6	Positive anomalies for both seismic velocities are imaged correctly (a,b); when the target has positive $V_P$ and negative $V_S$ anomalies, the former is well recovered with some minor artifacts (c) while the latter is not recovered at all (d); the opposite anomaly configuration leads to similar results although in this case the recovered $V_P$ presents higher amplitude artifacts (e) and the recovered $V_S$ is still completely wrong (f); a target model with both anomalies negative leads to a correct recovered models for both seismic velocities (g,h) . . . . .	149
6.7	Composite sources (red patches) and receivers (blue plane) are located at a depth of 130 m. The shape of the wavefield generated by one of the composite sources is described as a constant amplitude isosurface. . . . .	150
6.8	2D vertical slice of the $V_P/V_S$ ratio for the twoch-elastic model described in table 6.2. The ratio spans from 2.0 in the shallow areas to 1.7 in the deepest parts of the model. . . . .	151
6.9	$XY$ planes of the target and start $V_P$ models (a-f) at different depths given in nodes; same depth slices for the purely acoustic inversion results (g-i). . . . .	152

6.10	Depth slices for the recovered $V_P$ models corresponding to the inversion of twoch-elastic (a-c) and twoch-elconst (d-f). . . . .	153
6.11	$XY$ planes of the target and start $V_S$ models (a-f); inversion results for the twoch-elastic case (g-i). . . . .	154
6.12	$XY$ planes of the target and start $V_S$ models (a-f); inversion results for the twoch-elconst case (g-i). . . . .	155
6.13	Vertical logs for the twoch-elconst and twoch-elastic models at locations: log 1 $x = 1170$ m, $y = 845$ m; log 2 $x = 715$ m, $y = 1105$ m. The grey line is the starting model, the black is the target and the dashed is the recovered model. . . . .	156
6.14	Isosurfaces corresponding to a velocity of 1300 m/s for the target (left) and recovered (right) $V_S$ models. The positions of the logs are indicated with red bars, the front and rear correspond to logs 1 and 2 respectively. . . . .	157
6.15	Cutaways of the 3D Marmousi (left) and the $V_P/V_S$ ratio (right). Note that the models vary in all three directions (not just an extension of a 2D model like the one shown in figures 2.8 and 2.9). . . . .	158
6.16	Normalised 3D shot gathers for a source positioned at $x = 720$ m, $y = 600$ m, $z = 24$ m. The recorded energy corresponding to elastic events becomes apparent in (b) and (c), while in the acoustic case the energy is concentrated around the first arrivals. The amplitude values span from -0.001 to +0.001. . . . .	159
6.17	Depth slices for the target (a-c), start (d-f) and recovered $V_P$ models for the acoustic (g-i) and elastic (j-l) inversions. . . . .	161
6.18	Depth slices for the target (a-c), start (d-f) and recovered (g-i) $V_S$ models. . . . .	162
6.19	Vertical slices of the $V_P$ models at $y = 125$ nodes for the target (a), starting (b) and recovered models from acoustic (c) and elastic (b) inversions. Due to the presence of Scholte waves, the middle shallow region is better imaged in the acoustic (c) than in the elastic (d) inverted models. . . . .	163
6.20	Vertical slices of the $V_P$ models at $x = 75$ nodes for the target (a), starting (b) and recovered models from acoustic (c) and elastic (b) inversions. The same differences than in figure 6.19 between (c) and (d) are observed. . . . .	164
6.21	Vertical slices of the $V_S$ models at $y = 125$ nodes for the target (a), starting (b) and recovered model (c). The shallow middle region does not suffer from the artifacts introduced by the Scholte waves as much as the recovered $V_P$ models. Note that in the deeper regions the amplitude of the high-velocity intrusion layers is overestimated. . . . .	165
6.22	Vertical slices of the $V_S$ models at $y = 125$ nodes for the target (a), starting (b) and recovered model (c). . . . .	166
6.23	Vertical logs for the recovered $V_P$ and $V_S$ velocity models at locations: log 1 $x = 3120$ m, $y = 2160$ m; log 2 $x = 1680$ m, $y = 3480$ m. The grey, black and dashed lines correspond to the starting model, target and recovered models respectively. . . . .	167
6.24	Isosurfaces corresponding to 2000 m/s and 2800 m/s for the target (top), start (middle) and recovered (bottom) $V_P$ models. The positions of the logs are indicated with red bars: log 1: left; log 2: right. . . . .	168

6.25 Isosurfaces corresponding to 1100 m/s and 1800 m/s for the target (top), start (middle) and recovered (bottom) $V_S$ models. As in the previous figure, the positions of the logs are indicated with red bars: log 1: left; log 2: right.	169
6.26 Normalised seismograms corresponding to a line of receivers at $x = 75$ for a shot in $x = 75, y = 75$ (all units are given in nodes)	170
6.27 (a) the functional is reduced at each frequency block for the two acoustic inversions although the absolute values are very different for the positive and negative anomalies; (b) all the elastic functionals have higher values at the end of the inversion, although the successful iterations show some tendency to decrease at early stages (in this case the functional axis is logarithmic to appreciate the changes at each iteration). The increase in the functional is due to the fact that the functional is computed under different circumstances at each iteration.	173
6.28 Objective function evolution for the two channel (a) and Marmousi (b) experiments. Again, the elastic and the acoustic inversions have very different trends.	174
7.1 Acoustic image of the channels at a depth of 200 m.	182
7.2 3D view of the target gas cloud.	183

# List of Tables

6.1	One cube models . . . . .	145
6.2	Two channel inversions . . . . .	151
6.3	Marmousi 3D inversions . . . . .	158
6.4	CPU-times . . . . .	178

“Everybody gets told to write about what they know. The trouble with many of us is that at the earlier stages of life we think we know everything- or to put it more usefully, we are often unaware of the scope and structure of our ignorance.”

—*Thomas Pynchon*

# 1. Introduction

## 1.1 Chapter Overview

This thesis is organised in 5 chapters. In the first one I have tried to present the current state of the topic covered by my thesis and its historical perspective. The number of publications is vast and in no way is an attempt to provide an exhaustive catalogue of all the published works. I selected a number of them that are relevant and have in some way shaped the evolution of the subject. The first part of the chapter is devoted to the numerical solution of the wave equation —finite differences, finite elements, etc. The difficulties of finding appropriate spatial boundary conditions to mimic a semi-infinite medium and some of the proposed solutions are covered in this first part as well. The second part expose the progress of full-waveform inversion from its first theoretical conceptualisations to current three dimensional applications in field data. I would like to emphasise the important growth that this depth imaging technique has experienced in the last few years, grasping the attention of the oil and gas industry which, unlike in other fields of physics, drives the research trends as much as academia.

In the second chapter I present a formal derivation of the elastic isotropic wave equation. The choice of variables used to parameterise the wavefields in the elastic wave equation have a major impact on the conceptual procedure of both the modelling itself and the inversion scheme used. The elastic wavefields can be expressed in terms of particle velocities or displacements. The former leads to explicit time-stepping schemes in the boundary conditions, for example, whilst the latter results in an implicit time-stepping

algorithm in the boundaries due to its higher order time derivatives. One of the main differences during the inversion process is that they have different matrix operators associated that lead to different formulas to solve the inverse problem. The numerical method of choice is finite differences both for time and space approximation of partial derivatives, for reasons exposed in the middle sections of the chapter. We —our research group— have concluded that it outperforms finite elements, although we lack experience in spectral methods. Optimisation of the code and its performance is discussed at the end of the chapter.

The third and fourth chapter comprises the mathematical derivation of the expressions involved in the elastic inversion of seismic data, as well as a short review of the basic algebra concepts used during the process. A brief summary of the most popular local optimisation methods and some basic definitions are given in chapter 3, as well as the well established FWI theory framework. I start with the matrix algebra derivation of the parameter updates —bear in mind that full-waveform inversion is an iterative process which tries to minimise a pre-defined quantity via local optimisation methods— and provide expressions for the particular formulation of the elastic wave equation I use. The next section highlights the differences between the matrix algebra and the functional analysis derivations, to prove that even if the expressions are different they represent the same physical concepts. Then, I present present the parallelisation of the computer algorithm followed by a discussion on how to avoid storing a vast amount of unnecessary information during the inversion process.

The application of the method to several synthetic models is presented in chapters four and five. It begins with the study of the method in optimal situations for limited-sized models and very few sources. This first inversions were used as a proof of concept tests to appraise the validity of the equations established in the previous chapter. The models are a homogeneous cube with a fixed Poisson ratio with 8 cubic anomalies and a more geological model with two channels at different depths and with different polarity in velocity contrast. Plane waves illuminate the models and the signal is recorded in

opposite faces, in an attempt to reproduce a cross-hole experiment in 3D. The reason is that the transmitted portion of energy is much less sensitive to density than the reflected part. In the following chapter, I investigate the behavior of the inversion for different compressional-shear velocity ratios. I conclude by inverting two synthetic models with realistic geometry acquisitions and complex geological features.

In the last chapter, I expose the general conclusions of the thesis at different levels. First, I give an overview of the achieved goals in the project, as well as a discussion on the results I obtained. Then, I discuss my view in the evolution of the subject, trying to propose solutions and alternatives to several problems and limitations that affect my implementation of the elastic inverse problem. Finally, I introduce the group project in which I am involved in the present: the inversion of a field dataset using our 3D elastic FWI implementation.

### 1.1.1 Original work

The evolution of full-waveform inversion, as any other field in scientific research, benefits from the multiple contributions of many scientists that study, propose solutions and expand the knowledge of the discipline. During my research, I have made use of an extensive amount of existing theory and practical applications to achieve my goal: the design and implementation of a 3D elastic full-waveform inversion software capable of handling real-sized problems. This is the reason why the original work presented in this thesis is blended with existing theoretical and practical knowledge. I have tried to state as clearly as I could, my personal contribution to the subject in the beginning of each chapter.

The original work presented in this thesis consists of the derivation of the elastic FWI formulation not based on previous publications, and its implementation as a parallel computer program. Initially, I coded an elastic wave equation numerical solver as a stand-alone application that I later extended to an stand-alone immature inversion code. Due to the nature of my work, which is part of a bigger project, I have collaborated closely with other members in my group to achieve my objectives. Adrian Umpleby and I have worked

together to convert the stand-alone application into the main FWI software developed and maintained by our group. In a collaborative effort, we have developed the inverse formulas for the elastic case, and implemented solutions to the challenges we faced, for example, optimisation techniques to store time-domain wavefields in big models, source insertion, choice of inversion parameterisations or absorbing boundary conditions among others. The final software has been delivered to sponsoring companies, and it forms the basis of the commercial FWI data processing provided by CGGVeritas with only minimal changes from my original in-house code.

## 1.2 Preamble

Since the end of the 19th century, when the first attempts to locate earthquake epicenters by placing seismometers on the Earth’s surface, seismic methods have been used as a tool to image the subsurface studying the propagation of seismic waves through the ground. Seismic exploration has many different applications, ranging from the oil and gas detection to civil engineering. It was even used to locate enemies’ heavy artillery during the first World War) [Nolet, 1987]. Exploration seismology is by far the most important geophysical technique in terms of capital expenditure because its higher accuracy, higher resolution and greater penetration than other methods [Sheriff et al., 1995]. Observation of the Earth’s response to a particular stimulus can be classified according to the physical phenomena under observation. The number of exploration methods is limited by the capacity of acquiring adequate data and the existence of a source that triggers some sort of Earth response. Gravitational methods measure variations in the gravity field due to heterogeneities in density, but although great exploration depths can be explored, the resolution of the recovered physical properties is not optimum. In this case, the sources are the very presence of mass in the planet. Electrical methods suffer more or less from the same problems, but in this case the physical property examined is electrical resistivity; and usually an artificial source must be injected. Magnetic and electromagnetic

methods exploit Maxwell’s equations and are sensitive to fluid presence in the subsurface, which has made them popular in the oil and gas industry, although, again, the degree of spatial resolution of the resulting images is not very high. Finally, borehole geophysics (also known as well logging) generates very high resolution images for most of the above physical parameters studied, but it only provides information in the surroundings of the well, and drilling is a very expensive procedure and therefore it is not cost-effective when imaging of large 3D volumes is required. The above-mentioned reasons explain the preference for seismic methods in the geophysical community, and especially in oil and gas exploration.

Seismic exploration can be broadly divided into two main categories: near-normal incidence (or seismic reflection) and wide-angle (or seismic refraction), although new acquisition geometries, coil shooting for example, are being increasingly used to improve both quantitative and qualitative images of the subsurface. Near-normal incidence is commonly used to obtain information of the geometrical distribution of geological features. The most common technique to achieve this goal is migration: recorded data is used to generate an image of the reflectivity in time or depth. As its name indicates, the angle between the incident and the reflected wavefields is narrow, close to normal, and the exploration depths are larger than the source-receiver offsets. The industry has shown a special preference for these kind of acquisition geometries due to the high degree of resolution provided by migrated images despite the lack of quantitative information they provide. Wide-angle, on the other hand, is much better suited to generate quantitative images of the Earth’s physical properties —seismic velocities most commonly—and explores depths smaller than the source-receiver maximum offset. From an historic point of view, wide-angle acquisitions were designed specifically to perform travel-time tomography.

Full-waveform inversion combines the best of the two and provides high resolution quantitative images. Its first applications used wide-angle data because refracted waves are less sensitive to density than reflections, and therefore the data is more strongly linked

with only one parameter: velocity. Reflections occur in impedance contrast regions, and since impedance depends both on velocity and density, it is harder to estimate their values univocally. In the recent years, due to the success of full-waveform inversion, this trend is changing and the problem of applying it to all available datasets is starting to awaken the geophysical community's interest.

## 1.3 Seismic modelling

The propagation of energy through the Earth obeys a wave equation which has oscillatory solutions of different sorts. The seismic wave equation (as many other wave equations) has analytical solutions for relatively simple models, where the model is a distribution in space of physical properties describing the medium in which the waves travel —the Earth's subsurface or a limited region of it. The condition of existence of the analytical solution is that the model's physical parameters can be represented by a mathematical analytical function which must have some properties that depend on the wave equation media parameterisation (for example, seismic velocities must be positive). But the geometrical distribution of media parameters in the real world cannot be described by analytical functions, and therefore the wave equation cannot be solved analytically in most realistic representations of the Earth at any scale.

It is not possible then, to find the analytical solution of the wave equation in complex heterogeneous media. Thus, in order to model seismic waves, the wave equation must be solved using numerical methods.

### 1.3.1 Finite Differences

Since its first implementations 50 years ago [Altermann and Karal, 1968, Altermann and Abramovici, 1965], the finite-difference (FD) method has been the most popular approach to tackle this problem.

In Altermann and Karal [1968] and Altermann and Abramovici [1965], a homogeneous

wave equation was used, and the boundaries between different layers were treated explicitly. Ten years later, Madariaga published the application of a staggered FD scheme that could treat heterogeneities implicitly [Madariaga, 1976]. The formulation developed in Madariaga [1976] was based on cylindrical coordinates. Many authors used this staggered grid formulation and extended it into Cartesian coordinate systems [Virieux, 1984, Virieux, 1986, Levander, 1988, Luo and Schuster, 1990], making the formulation more suitable for regular grid models. In order to obtain the staggered FD equations, these authors wrote the wave equation as a first order hyperbolic system, i.e. a system of coupled equations: the equations of motion and Hooke's law equations. Splitting the wave equation and staggering the FD operators, solid-fluid interface phenomena —like Rayleigh or ground roll waves, Scholte waves, etc— arise naturally when using staggered grid FD operators, while they have to be explicitly imposed at the boundaries for other FD discretisations. Plus planar free-surface conditions and source insertion are easily implemented under the staggered formulation [Graves, 1996, Kristek et al., 2002].

There is one important difference between Virieux [1984, 1986], Levander [1988], and Luo and Schuster [1990]: while Virieux and Levander used a velocity-stress formulation, Luo and Schuster used a displacement-stress formulation. The implications of these different approaches are more important now than they were when the papers were published, as it will be shown later, because they affect the nature of the time-stepping scheme in the absorbing boundary conditions. The velocity-stress and displacement-stress are shown in equations (1.1) and (1.2) respectively

$$\begin{aligned} \rho \frac{\partial \mathbf{v}}{\partial t} &= \nabla \cdot \boldsymbol{\tau} \\ \frac{\partial \boldsymbol{\tau}}{\partial t} &= \mathbf{c} : \nabla \mathbf{v} \end{aligned} \quad (1.1)$$

$$\begin{aligned} \rho \frac{\partial^2 \mathbf{u}}{\partial t^2} &= \nabla \cdot \boldsymbol{\tau} \\ \boldsymbol{\tau} &= \mathbf{c} : \nabla \cdot \mathbf{u} \end{aligned} \quad (1.2)$$

where  $\mathbf{v}$  and  $\mathbf{u}$  are the particle velocities and displacements respectively,  $\boldsymbol{\tau}$  is the stress tensor,  $\rho$  is the density,  $\mathbf{c}$  is the elastic tensor and  $:$  denotes the Frobenius inner product defined by

$$\mathbf{A} : \mathbf{B} = \text{tr}(\mathbf{A}^T \mathbf{B}) = \text{tr}(\mathbf{A} \mathbf{B}^T) \quad (1.3)$$

where  $\text{tr}$  denotes the trace of the matrix —sum of the diagonal elements. In equation (2.15) the velocity vector and the stress tensor are differentiated once with respect to time, while in equation (2.12) only the displacement vector is differentiated twice with respect to time.

The strategy to code these equations as an efficient computer algorithm will be shown in the next chapter, but it is worth advancing some of the implications of the implementation here. A straightforward implementation of equations 2.15 and 2.12 requires the storage of 9 variables everywhere in the discretised model. Under the particle-velocity formulation, the 9 variables are stored only once due to the first-order time differential. After discretisation in time, the variable arrays are overwritten by the new values after they have been used (see chapter 2 for a complete description).

The same does not hold true for the particle-displacement formulation, though. The second order in the time derivative operator forces the storage of the particle displacement variables twice. Imagine that, at some point of the modelling process the current discretised time-step is  $t$ , then in order to calculate values at  $t + 1$ , the equation for particle displacements would be of the form

$$u_{t+1} = 2u_t - u_{t-1} + f(\tau) \quad (1.4)$$

where  $f(\tau)$  represents the right hand side of the second line in equation 2.12 after discretisation. The memory locations of  $u_{t-1}$  and  $u_{t+1}$  are the same, therefore the former—which will not be needed after this operation—is overwritten by the latter. But the value at  $t$  will still be used when  $u_{t+2}$  is computed, and it needs to be kept in memory.

Luo and Schuster [1990] used a more sophisticated approach in the displacement-stress

formulation which allows them to avoid the storage of the stress vector by computing its values ‘on the fly’ by substitution of the stress equations into the particle displacements. The drawback of not storing stresses is that the number of computations is noticeably increased because at each time step at least two stress-points (or more, depending on the order of the finite-difference operator) need to be calculated for each displacement.

All the publications presented so far focused on 2D because of the computational limitations both in terms of memory and CPU speed. In the mid 1980’s, first attempts to model the wave equation in 3D were performed on vector computers and were based on the acoustic formulation of the wave equation.

In 1993, Yomogida et al. published a staggered FD implementation of the 3D elastic wave equation applied to earthquake modelling in the Los Angeles basin [Yomogida and Etgen, 1993]. Later works, focused on optimising the algorithmic performance of staggered grids [Graves, 1996, Moczo et al., 2000, Moczo et al., 2002]. The extension of the method to nonuniform spacing grids was developed by Pitarka to reduce the computational cost of the modelling by reducing the number of grid points and discretising time with larger time-steps [Pitarka, 1999]. Elastic rotated operators can be used to improve the restrictions in space discretisation.

The extra computational cost of jumping from 2D to 3D can be mitigated making use of parallel computers. In simulations involving only one source, it is more efficient to fragment the model in several subdomains and parallelise the computation of the wavefield in each subdomain [Bohlen, 2002, Minkoff, 2003]. Under spatial parallelisation, an important amount of information needs to be passed across the cluster of computers because the solution overlaps in the edges of the subdomains.

The accuracy of differential operators using finite differences is subject to limitations in both spatial and time discretisations [Levander, 1988, Virieux, 1986, Moczo et al., 2000]. The number of grid points used to sample the maximum frequency in the lowest-velocity region of the model cannot be arbitrarily small because numerical errors accumulate and create non-physical energy in the wavefields. The time discretisation is also restricted by

the number of cells that can be crossed in a time-step. Both these restrictions depend on the size of the FD stencil, or in other words, depend on the number of surrounding points used to approximate the differentials. High-order operators can relax these conditions at the expense of extra computational cost. Stability conditions and dispersion relations will be discussed in chapter 2.

Nowadays, hardware performance allows us to work on real-sized 3D forward modeling problems. As will be shown in chapter 3, a high number of forward modeling problems are required in full-waveform inversion. Today's challenge is to speed-up this process so real-sized models of several hundreds of millions of grid-points can be inverted within reasonable times. In 2D, an efficient method to speed-up multiple solutions of the wave equation is to Fourier transform the time derivatives. The Fourier transform of a derivative is

$$\mathcal{FT} \left\{ \frac{\partial}{\partial t} \right\} = i\omega \quad (1.5)$$

where  $\omega$  is the temporal frequency. Rotated operators can be applied as well in the frequency-domain to relax the stability conditions [Stekl and Pratt, 1998]. The main benefit of the frequency domain formulation is that the solution is precomputed for any given source and expressed as a diagonalised matrix operator. Then, each shot solution reduces to a simple vector substitution increasing the performance of the algorithm in terms of CPU-time. Another advantage of the frequency domain is that only a few frequencies are required to represent the full bandwidth of the source spectrum.

The extension to 3D poses new problems due to the size of the matrix operators involved in the process. Operto et al. [2007] developed a direct solver to mimic the 2D procedure in 3D in the acoustic case. In their work, the massively parallel solver MUMPS is used to precompute solutions of the wave equation [Operto et al., 2007]. A different approach is to use iterative solvers to compute acoustic solutions for each shot, which does not require such large amounts of memory to store matrix operators although the solution needs to be calculated for each shot independently [Warner et al., 2007b]. But

the application in frequency-domain elastic 3D formulations still remains a challenge due to the massive increase in the size of the matrix operators —and consequent increase in memory demands.

### 1.3.2 Other numerical methods

Although FD is probably the most common method due to its simplicity and adaptability, other methods have been developed to improve the accuracy of the numerical solutions. One of the important limitations of regularly spaced discretisations are their incapability to adapt the shape of the cells. For example, models containing strong topographic variations are poorly represented in FD because the topography of the model only can be defined in a blocky cubical grid. Or, the time-step cannot be increased by adapting the size of the cells to its seismic velocity values —relaxing therefore the stability condition by making faster cells bigger.

The next chapter will cover the most common numerical techniques used to discretise and solve the wave equation. In 1983, Johnson used FD in time and Fourier methods in space to maximise computer efficiency in one of the first 3D numerical implementations [Johnson, 1984]. Its worth mentioning that the literature covering this topic is extensive and I have only selected a few publications relevant to 3D seismic modelling. For the finite-element (FE) method and variations see for example Brossier et al. [2008], de la Puente et al. [2008] and Etienne et al. [2010]. Komatitsch, [Komatitsch and Tromp, 1999, Komatitsch and Vilotte, 1998], is one of the most active authors applying the spectral method (SM) to solve the wave equation.

### 1.3.3 Absorbing boundary conditions<sup>1</sup>

The presence of boundaries at the edge of the computational domains causes artificial reflections that mask the real wavefield. In order to prevent these spurious reflections, absorbing boundary conditions should be used to vanish energy beyond the model edges. The simple solution to this problem would be to enlarge the numerical mesh so the reflections at the boundaries are delayed and do not interfere with the wavefield in the region of interest. In practice, this is not an efficient strategy due to the computational overburden.

In 1974, Smith combined the use of Neumann and Dirichlet boundaries to eliminate spurious reflections from the boundaries [Smith, 1974]. The Neumann condition impose the spatial derivative of the particle displacement to be zero at the boundaries. As a result, the reflected wave has the same amplitude sign as the incident wave. On the other hand, the Dirichlet condition sets the value of the particle displacement to zero in the boundaries, which means that the amplitude of the reflected wave is of opposite sign to the incident wave. Combining these two reflected waves cancels the reflections by virtue of the superposition principle. The shortcoming is that the simulation has to be done twice, one for each boundary condition, and hence the computational cost increases.

Clayton and Engquist developed boundary conditions based on paraxial approximations of the wave equation [Clayton and Engquist, 1977]. The paraxial approximation models correctly the energy travelling in one specific direction and discriminates against the energy moving in the opposite direction. At shallow angles of incidence the absorption decreases and some energy is reintroduced in the model. Furthermore, some instabilities arise in presence of Poisson's ratios larger than 0.37 [Randall, 1988].

A simple solution was proposed by Cerjan et al. [1985] by adding extra nodes at the boundary of the model and damping its values. The damping is applied by multiplying

---

<sup>1</sup>I use a non-strict definition for absorbing boundary conditions. Some authors claim that the definition should only include methods where the absorption occurs in the edge of the model (a line in the 2D case and a plane in 3D). Nonetheless I will use the term without distinction referring to absorbing boundaries and absorbing layers (a plane in 2D and a volume in 3D, added at the edge of the models' faces).

the amplitudes by a negative exponential function resulting in a decrease of the boundary reflected wavefield. The values of the damping function range from 1 at the inner boundary of the extra nodes to some smaller value at the outer edge of the model. In this case, at normal incidence the absorption is perfect, but in order to obtain negligible reflections many nodes need to be added at the edges of the model [Cerjan et al., 1985].

Better absorption can be achieved using projection operators and potentials —where the shear and compressional waves are propagated separately as shown in [Lindman, 1975, Randall, 1988, Randall, 1989]. In this case the absorbing boundaries need to be locally homogeneous.

In 1994, Berenger published a paper presenting a new absorbing boundary condition for electromagnetic waves, the so-called perfectly matched layer (PML) [Berenger, 1994, Berenger, 1996]. Its performance was superior to all the above absorbing methods by more than two orders of magnitude —in terms of reflected amplitudes. This method relies on a rather physically meaningless transformation consisting of converting the space coordinates to complex numbers. Then, the normally oscillatory solutions of the wave equations expressed as exponentials with complex arguments, have a real negative exponent term in time. As a consequence, the amplitude of the wavefields in the PML region decays with time at a rate dictated by the real exponent of the solution.

Its name comes from the fact that there are no reflections at all in the interface between the model and the PML, hence they are *perfect*, although this is only valid for analytical solutions. After discretisation, some energy is still reflected because the damping functions are not perfectly smooth (see subsection 2.2.5).

Due to the similarities between Maxwell's equations and the elastic wave equation, the PML method was soon adapted to elastodynamics problems. For example, Hastings et al. [1996] combined a finite-difference time-domain (FDTD) velocity-stress scheme in the model region with a potential formulation in the boundaries. There exist a wide range of publications covering the subject in detail; I refer to [Chew and Liu, 1996, Collino and Tsogka, 1998, Wang and Tang, 2003, Marcinkovich and Olsen, 2003, Komatitsch and

Tromp, 2003, Drossaert and Giannopoulos, 2007, Papazachos, 2006] for a complete review. They cover a range of different variations of the PML method: non-split PML [Papazachos, 2006], CFS-PML (Complex Frequency Shifted PML) [Drossaert and Giannopoulos, 2007] and C-PML (Convolutional PML) [Komatitsch and Tromp, 2003]. I based my PML implementation mostly on these publications: specifically I use the non-split PML.

## 1.4 Full-waveform inversion

Full-waveform inversion<sup>2</sup> (FWI) is a technique that seeks a wiggle-by-wiggle match of real and modelled data [Tarantola, 2005]. Ideally, all the information contained in the data is used to recover physical properties of the subsurface beneath the region of interest. The underlying assumption of the method is that once the data match is perfect —or at least sufficiently good— the model used to calculate synthetic data represents the real properties of the subsurface because both models give rise to the same data. Due to the non-linearity between data and model properties, FWI is an iterative process that provides small improvements of the estimated model at every iteration. An objective function is defined as a criteria to match both datasets, normally based on the difference between them, and then the objective function is iteratively minimised using the so-called adjoint-state method to avoid the difficult task of inverting the wave equation operator as will be shown in later chapters. The adjoint-state method is used to compute efficiently the gradient of the objective function, without explicit building of the Fréchet derivative matrix [Plessix, 2006].

In the early 1980’s, motivated by the successful implementation of medical tomography, geophysicists started using seismic tomography in the oil and gas industry [Lo and Inderwiesen, 1994]. The aim of seismic tomography is to quantify physical properties of the subsurface at different scales. Active or passive sources are used depending on the scale of the problem. Global and regional tomography deal with the Earth structure and

---

<sup>2</sup>Also referred to as wavefield inversion or waveform tomography.

relies on the use of natural sources —earthquakes— due to the huge amount of energy needed to cover distances over hundreds of kilometres. In this case, the control over the experiment’s geometry is poor and it leads to high uncertainties in the results. On the other hand, seismic tomography using active sources is used for smaller-scale problems like oil and gas exploration, civil engineering or crustal studies. In this case, the geometry of the experiment is not only known but designed for specific purposes, leading to much more accurate results compared to global or regional tomography. My research is focused on active-source problems.

### 1.4.1 Travel-time tomography

FWI needs a starting model close enough to the real one to provide reliable results, although high resolution is not necessary in the starting model. Nowadays, travel-time tomography (TT) is widely used to generate such starting models in real data applications. Indeed, FWI is a tomographic technique, very much similar to TT. Whereas in FWI the full traces are included in the objective function, in TT only the arrival times of the wavefronts are considered. Travel-time tomography is based on the inversion of wave travel-times from the source to the receivers, using the assumption that all the frequencies travel at the same speed in the medium [Luo and Schuster, 1991]. In other words, TT inversion is based on the eikonal equation to generate synthetic travel-times, while FWI uses the wave equation to generate synthetic seismograms. To obtain the eikonal equation, the acoustic wave equation in homogeneous density media is Fourier transformed as

$$-\omega^2 p(\mathbf{x}, \omega) = V_P^2 \nabla^2 p(\mathbf{x}, \omega) \quad (1.6)$$

where  $p$  is pressure,  $\omega$  is temporal frequency,  $\mathbf{x}$  is spatial position and  $V_P$  is the propagation velocity of compressional waves. Then, the local shape of the solution is considered

invariant over time and has the following form in the time and frequency domains

$$p(\mathbf{x}, t) = A(\mathbf{x})\delta(t - \tau(\mathbf{x})) \xrightarrow{\mathcal{FT}} A(\mathbf{x})e^{i\omega\tau(\mathbf{x})} \quad (1.7)$$

where  $A(\mathbf{x})$  is the geometric propagation factor —the shape of the wavefront— and  $\tau(\mathbf{x})$  is the time delay at which the wavefront reaches the position  $\mathbf{x}$ . Reinjecting equation 1.7 into 1.6 leads to

$$-Ae^{i\omega\tau} = V_P^2 \left( \frac{1}{\omega^2} \nabla^2 A + \frac{2i}{\omega} \nabla A \nabla \tau + \frac{iA}{\omega} \nabla^2 \tau - A \nabla \tau \nabla \tau \right) e^{i\omega\tau} \quad (1.8)$$

where the arguments of  $A$  and  $\tau$  have been dropped to avoid over-cluttering the equation. The following assumption is now made: all frequencies travel at the same speed. Therefore, at very high frequencies, equation 1.8 reduces to

$$\nabla \tau \nabla \tau = \frac{1}{V_P^2} \quad (1.9)$$

which is the eikonal equation. Consider a vector  $d\vec{r}$  tangent to the ray trajectory which covers a distance  $ds$ , and note that  $V_P \nabla \tau$  is a unitary vector perpendicular to the wavefronts. Then, the travel-time can be computed by integrating over the whole raypath as

$$T = \int_s \frac{1}{V_P(\mathbf{x})} ds \quad (1.10)$$

where  $T$  is the arrival time of a wavefront travelling over the path  $s$ . Equation 1.10 represents the forward problem for TT inversion in the same way that the wave equation does in FWI. Figure 1.1 illustrates the generation of ray-paths using the eikonal equation.

The main limitation of ray-based methods is that physical parameters of the medium must vary slowly over several wavelengths [Gauthier et al., 1986]. The resources needed, in terms of computational time and memory, are significantly lower than in FWI because

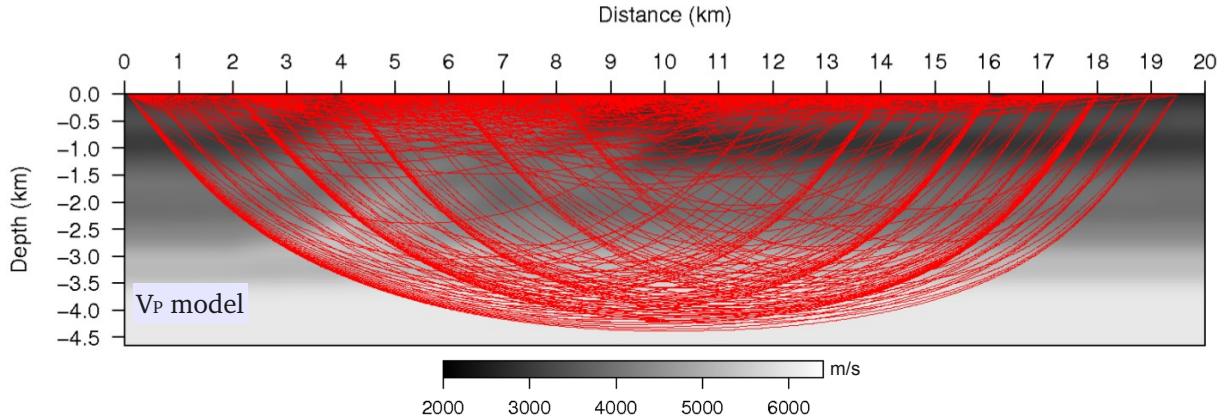


Figure 1.1: Application to the eikonal equation to calculate travel-times (in red) for an experiment with multiple sources and receivers placed on the surface.

the amount of data involved is considerably smaller. One of its main advantages is that sensitivity to data quality is low: a relatively clean arrival in the recorded seismograms provides a set of travel-times that will be used as input data. Travel-time tomography presents some important limitations such as low-resolution image results or insensitivity to velocity inversions (low velocity below high-velocity layers) when only first arrivals are used as shown in figure 1.2.

In 1988 and 1990, Vidale presented a finite-difference method to calculate first arrival travel-times which introduced an efficient numerical method that could handle diffractions and arrivals coming from shadow zones in 2D [Vidale, 1988] and 3D [Vidale, 1990]. Podvin and Lecomte proposed an scheme able to handle strong sharp heterogeneities as well as an extension to calculate different arrivals in [Podvin and Lecomte, 1991]. More sophisticated methods have been used to model travel-times. In [Luo and Schuster, 1991], for example, the travel-times are calculated using the full solution of the wave equation. Some authors have proposed strategies to invert seismic velocities together with other rock properties using joint-inversion schemes; for example gravity and seismic data [Korenaga et al., 2001], joint reflection and refraction arrivals [Korenaga et al., 2000], joint inversion of magnetotelluric, gravity and seismic data [Jegen et al., 2009, Heincke et al., 2006]. All these publications based the relations of different parameters upon some analytical or

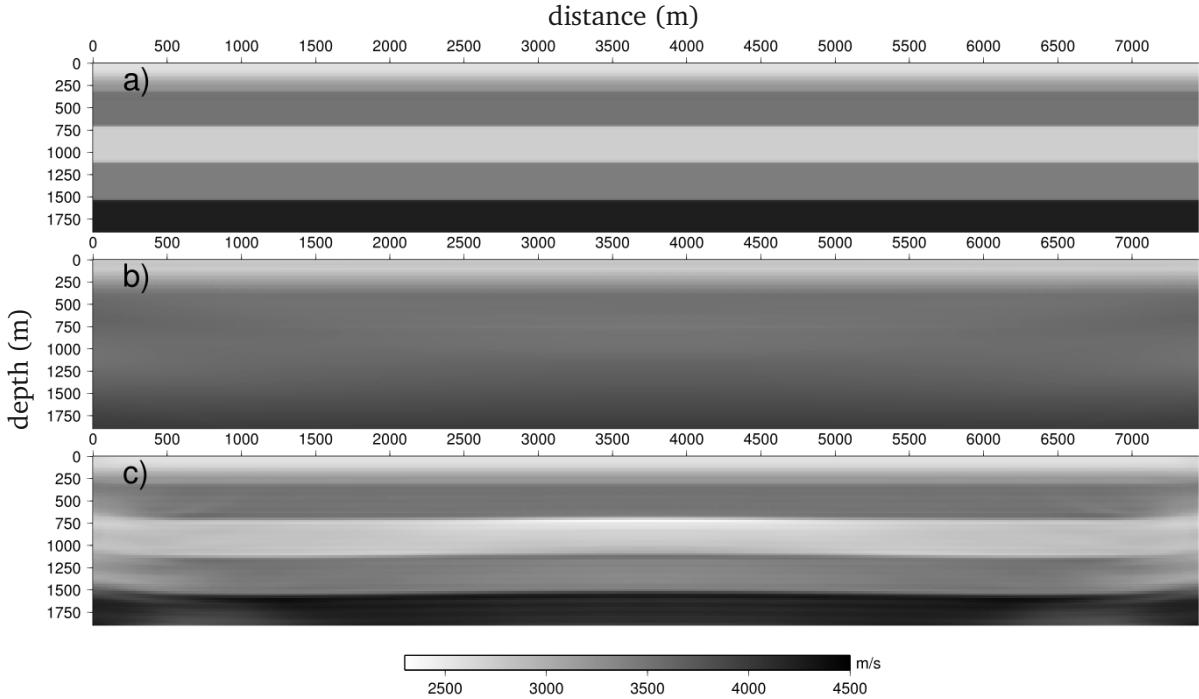


Figure 1.2: (a) True model used to generate ‘real’ data, note the inversion of velocities in the layer between 700 and 1100 m depth; (b) TT fails to recover the low velocity layer in the middle of the model because the first arrivals never travel through it; (c) FWI, on the other hand, can recover it because the energy has travelled through the low velocity region and its effect is recorded in the full seismogram.

empirical relation to constrain the solution of the inverse problem. A complete analysis and review on ray methods applied to seismic problems can be found in the book [Červený et al., 1977].

I would like to mention that there exist some hybrid methods that combine the best characteristics of both FWI and TT [Clayton and Stolt, 1981, Luo and Schuster, 1991], but my research only focuses on the pure waveform method.

#### 1.4.2 History of full-waveform inversion

The emergence of full-waveform inversion started in the mid 1980s with a few publications written by Tarantola, Lailly, Gauthier, Virieux and Mora, but its application to real exploration problems has had to wait for almost two decades due to the computational

cost of its computer implementation and only in the last few years computers have had the capabilities to handle 3D problems.

In 1984, two papers [Tarantola, 1984b, Tarantola, 1984a] and two book chapters [Santosa, 1984, Lailly, 1983] presented a formal formulation of the seismic inverse problem for reflection data, although due to the abovementioned reason, the method was not implemented in a computer program to test its performance. The first test examples came a few years later in a work published by Gauthier et al. [1986] where three different simple 2D models were used to test the performance of what they called inversion of seismic waveforms. First, a homogeneous background model with one diffractor point situated in the middle is illuminated with a different number of sources to determine what geometries give better results. They prove that using only one source leads to some undesirable artifacts, while illuminating the medium with several sources focuses the resulting image. The second model they tested was the same as before with an extra scatterer below the original one. Due to the larger distance of this second scatterer with respect to the original, it is imaged with a lower amplitude, where the amplitude relation between the imaged scatters is roughly proportional to the square root of the distance to the surface (where the sources and receivers are placed). Therefore some sort of preconditioning must be applied in the inversion process in order to recover the correct amplitudes of the velocity field. As will be shown in chapter 3, such preconditioning has the form of a diagonal approximated Hessian matrix. The last model they tested was a circular anomaly with a diameter of approximately half of the model side. The results showed that FWI fails to recover velocity anomalies higher than 10%. We (and by “we”, I refer to the FWI community) think that this limitation is due to cycle skipping of the data which forces the inversion to converge to a local minima, although this is hard to prove and it is mostly based on intuition rather than mathematical proof. I think that this work represents a big advance in FWI because despite the computational limitations at the time of publication, it proves the capacity, effectiveness and potential of the method. The first results on 2D elastic implementations appeared in the next following years [Mora, 1987, Mora,

1988, Crase et al., 1990]. So far, all cited articles used a time-domain approach to the problem.

### 1.4.3 Time and other domains

Over the decade between the late 80s and the late 90s, little activity (in terms of publications) took place in the field, with a few remarkable exceptions. For example, Bunks successfully inverted the Marmousi model using a multi-grid approach where the model was inverted from large scale to low scale to mitigate local minima sensitivity [Bunks et al., 1995].

In 1999 Pratt published two papers presenting a new approach to the inversion problem [Pratt, 1999, Pratt and Shipp, 1999]. The conceptual basis was the same, but the problem was reformulated in the frequency domain. As a consequence, the computational cost of modelling multiple sources using time-stepping schemes is substantially reduced by precomputing the solution for any shot in the frequency domain. Thus, direct vector substitution in a diagonalised system of equations provides solutions —wavefields— in a very efficient manner. Thanks to this new formulation, most of the computer limitations of the time could be overcome and the method implemented to handle real-sized 2D experiments. Another benefit of working in the frequency domain is that only a limited number of frequencies of the source spectrum are necessary, due to wavenumber illumination overlap provided by wide aperture acquisitions [Pratt and Worthington, 1990, Sirgue and Pratt, 2004, Yokota and Matsushima, 2004]. More benefits of working in the frequency domain are: attenuation can be easily introduced by using complex velocities, and the fact that non-linearity can be mitigated by starting at low frequencies and increase them as the inversion proceeds.

Many publications appeared after the frequency-domain formulation of the problem both in field data experiments and methodology. To study offshore crustal structures, ocean bottom seismometers are placed far apart from each other on the seabed to cover large offsets that can capture deep wavefront waves. Classically, travel-time tomography

was used to obtain velocity models of the crustal structure, but with the arrival of FWI the possibility to exploit more information present in the data lead to a number of publications on the topic [Operto et al., 2004, Operto et al., 2006, Brenders and Pratt, 2007].

A new reformulation of the problem in the Laplace domain was presented by Shin et al. in [Shin and Cha, 2008]. The Laplace transform can be interpreted as the Fourier transform for a frequency of 0 Hz when the parameter  $s$  in

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^{\infty} f(t)e^{-st}dt \quad (1.11)$$

is real<sup>3</sup>. If the Fourier transform decomposes a function in vibration modes or frequencies, the Laplace transform decomposes it in moments. Moments are a mathematical concept that can be loosely related to the shape of the function. For example, the first and second moments correspond to the expected value and the width of the function respectively. Laplace domain FWI is very efficient in terms of avoiding local minima because it recovers the low frequency content of the background velocity model, and thus the global minimum of the solution space is broader (the notion of solution space is discussed in chapter 3).

It is not clear yet which domain is best to optimise FWI performance, and different research groups worldwide are currently trying to elucidate it.

FWI is usually focused on inversion of wide-angle datasets because they allow the reconstruction of the large wavelengths from the wide apertures; besides, they allow to image the deeper parts of the volumes explored using transmitted energy information from the data. The application to limited offset datasets presents difficulties because reflexions occur due to seismic velocities and/or density heterogeneities. Even so, Wang and Rao [2009] proposed two different strategies to invert short offset datasets: layer stripping, where the model updates proceed from shallow to deeper areas; and depth-weighted updating based on ray-trace characteristics [Wang and Rao, 2009].

---

<sup>3</sup>Actually, the only differences between the Laplace and the Fourier transform are the integration limits and the fact that the exponent in the integral is purely complex for the Fourier case while Laplace transformations have complex numbers with real and imaginary parts.

#### 1.4.4 Inversion of elastic parameters

Elastic FWI is a multi-parameter joint inversion of both compressional and shear wave propagation velocities. The mathematical development for the elastic formulation of the problem appeared at the same time as the acoustic, but its application has proven to be far more complex than in the acoustic approximation. There are two main reasons for it. The first is the higher number of both model parameters and wavefield variables which have a direct impact on the computational cost, although for 2D cases this is not a limitation anymore. What really explains the lack of publications on the subject is the second reason. By taking a step back and analysing the basic building blocks in acoustic FWI: there is only one data type —pressure— and one model parameter —P-wave velocity— involved and they are univocally related. In the elastic case, on the other hand, there are four data types —three particle displacement or velocity components and pressure— and two model parameters —P- and S-wave velocities. The problem is that there is cross-talk between the solution spaces of  $V_P$  and  $V_S$ , or in other words, the data residuals are not univocally related to the model parameters.

The first applications of the method were restricted to the recovery of only  $V_P$  models using the elastic wave equation to solve the forward problem [Shipp and Singh, 2002]. A few years later, a similar procedure was applied to recover both  $V_P$  and  $V_S$  in a 2D synthetic model [Gélis et al., 2007, Sears et al., 2008], although I believe some of the published works used the wrong expressions to calculate the gradient. As it will be shown in chapter 3, the parameterisation of the wave equation impacts directly on the inversion expressions. Particularly, they missed a time derivative with respect to time of one of the stress wavefields in their calculations.

A rigorous study on different procedures used in elastic inversion was published in 2009 [Brossier et al., 2009]. In this paper, different preprocessing strategies (for example, the effect of time-damping the data), surface conditions (free-surface and absorbing boundary), and local optimisation methods (L-BFGS and PGC) were investigated in the frequency domain. Another paper by the same author explored the best norm to use in

the objective function and concluded that the  $L^1$  was more robust in both onshore and offshore experiments [Brossier et al., 2010]. Both papers were based on a frequency domain of the problem, that is why some of the conclusions may need re-evaluation in other domains. For example, in the time-domain using the  $L^1$  norm leads to a backpropagation of squared signals that create high frequency artifacts in the time domain modelling scheme.

#### 1.4.5 Evolution to 3D

The first publications of 3D applications date from only a few years ago because the computers before then did not have the required capabilities either to handle the size of the model matrices or the number of computations required to implement a three-dimensional FWI algorithm.

The first 3D results were presented by Warner and Stekl in 2007; both papers were based in a frequency-domain formulation of the acoustic problem and used an iterative solver. The first presented a successful recovery of an homogeneous cube with 8 cubic anomalies in it using only two plane-wave sources [Warner et al., 2007a]. The second paper successfully recovered a two-channel model on a perturbed 1D velocity gradient using a 1D model as a starting model [Stekl et al., 2007]. In the following years, the same authors presented applications to 3D streamer data which substantially improved the quality of the migrated images by defining a more accurate velocity model of the subsurface [Warner et al., 2008, Stekl et al., 2008]. The application of 3D acoustic FWI is also possible and has been applied to synthetic datasets to successfully invert acoustic velocities [Ben-Hadj-Ali et al., 2008]. The main benefit of using a direct solver, as mentioned before, is the efficiency in multiple shot simulations, but its drawback is a significant increase in memory. The main model studied was a complex geological overthrust with embedded channels and the quality of the recovered images was remarkable. Thanks to these first 3D implementations, FWI has become a hot topic and has captured the attention of the oil and gas industry, who are starting to apply it in production, principally due to a paper

published by Sirgue in 2008 [Sirgue et al., 2009] showing a very clear image of various shallow channels in the Valhall oil field. Later works by Plessix confirmed the feasibility and potential of 3D acoustic FWI [Plessix, 2009, Plessix and Cao, 2011].

3D elastic FWI, however, still remains a challenge and few works have been published so far. The first publication dates from 2008, where a 3D synthetic model was inverted to recover both seismic velocities using a Newton-CG method [Epanomeritakis et al., 2008]. Since then, only a limited number of studies can be found in the literature. I have found the following conference presentations: [Guasch et al., 2010, Guasch and Warner, 2010, Etienne et al., 2010]. The aim of my PhD project is to fill that gap.

## 2. Modelling of seismic waves

The solution of the wave equation constitutes one of the pillars of full-waveform inversion —as discussed in the previous chapter—and therefore an important part of the effort devoted to examine full-waveform inversion is spent in finding accurate solutions of it. In this chapter the seismic wave equation is derived, and the choice of numerical method used to solve it is presented. Due to the elevated number of solutions required during the inversion process, it is highly desirable to speed up the process of modelling by optimising the algorithms as much as possible.

In this chapter, I used well-established techniques to implement a numerical solution of the elastic wave equation.

### 2.1 Theoretical background

#### 2.1.1 The wave equation

The wave equation is one of the most important equations in physics and it is applied in different forms in a wide range of disciplines as it describes the behaviour of both electromagnetic and mechanical waves. Mathematically it is a partial differential equation which normally involves time and space derivatives. Seismic imaging involves only mechanical waves, and, depending on the assumptions considered, one may obtain different expressions. The simpler case is when the earth is treated as an acoustic isotropic medium, and

in that case the wave equation is written as

$$\frac{\partial^2 \mathbf{p}}{\partial t^2} = \mathbf{V}_P^2 \rho \nabla \cdot \left( \frac{1}{\rho} \nabla \mathbf{p} \right) + \mathbf{s} \quad (2.1)$$

where  $\mathbf{p}$  is the pressure,  $\mathbf{V}_P$  is the propagation velocity of acoustic waves and  $\rho$  is the density of the medium. In this context only compressional waves are taken into consideration. Although this is only an approximation it has been widely used in acoustic full-waveform inversion yielding rather impressive results as it has been shown in the previous chapter.

A more complete description takes into account the fact that the Earth is a solid body and hence it includes elastic phenomena. Linear elasticity is an approximation that considers a linear relation between stress and strain, and is valid for small deformations and forces applied to a medium. Under the linear approximation, a medium is said to be elastic if it reverts to its initial state after any force applied to it is eliminated. The mathematical representation of this concept is the well known Hooke's law which defines a linear relation between stress and strain

$$\tau_{ij} = C_{ijkl} \epsilon_{kl} \quad (2.2)$$

where  $\tau_{ij}$ ,  $C_{ijkl}$  and  $\epsilon_{kl}$  are the stress, stiffness and strain tensors respectively. Since anisotropy is beyond the scope of this thesis, only the isotropic case will be considered. In that case, the stiffness tensor only depends on two elastic parameters. The choice of  $\lambda$  and  $\mu$  (Lamé parameters) is more convenient than any of the multiple alternatives (bulk modulus, Young modulus, Poisson ratio, etc) because the numerical solution is easier to implement as will be shown in the next section. For the isotropic case then, Hooke's law is

$$\tau_{ij} = \lambda \delta_{ij} \epsilon_{kk} + 2\mu \epsilon_{ij} \quad (2.3)$$

where  $\delta_{ij}$  is the Kronecker delta. The definition of the stress tensor is

$$\epsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.4)$$

where  $u_i$  is the particle displacement in the  $i = x, y, z$  direction. Inserting equation 2.4 into 2.3 provides the first of the two coupled equations of the elastic wave equation

$$\tau_{ij} = \lambda \delta_{ij} \frac{\partial u_k}{\partial x_k} + \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.5)$$

The following derivation of the equation of motion follows the one developed by Aki and Richards [Aki and Richards, 2002]. Take an arbitrary volume  $\mathbf{V}$  with surface  $\mathbf{S}$  (see figure 2.1). The total net force applied to its particles must be equal to the change of momentum of such particles

$$\iiint_V \rho \frac{\partial^2 \mathbf{u}}{\partial t^2} dV = \iiint_V \mathbf{f} dV + \iint_S \mathbf{T}(\mathbf{n}) dS \quad (2.6)$$

where  $\mathbf{u} = (u_x, u_y, u_z)$  is the particle-displacement vector,  $\mathbf{f} = (f_x, f_y, f_z)$  is the body force distribution and  $\mathbf{T}(\mathbf{n}) = (T_x, T_y, T_z)$  is the traction (defined positive outwards) applied to  $dS$ . The traction is defined as  $d\mathbf{F}/dS$ , being  $\mathbf{F}$  the external surface forces. The stress tensor is defined by

$$\tau_{ij} = T_j(\hat{\mathbf{x}}_i) \quad (2.7)$$

which is the Cauchy stress theorem. The first index  $i$  in  $\tau_{ij}$  denotes that the stress is applied to a plane perpendicular to  $\hat{\mathbf{x}}_i$ , and the second index  $j$  indicates the direction in which the stress acts. The complete matrix representation of the stress tensor is

$$\tau_{ij} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} \quad (2.8)$$

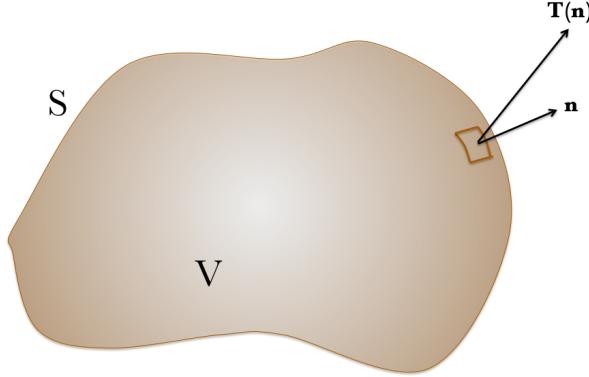


Figure 2.1: A volume  $V$  delimited by a surface  $S$ .  $\mathbf{T}(\mathbf{n})$  is the traction applied to  $dS$  which has a normal unit vector  $\mathbf{n}$  (adaptation of the figure from chapter 2 in [Aki and Richards, 2002]).

In the expression 2.8,  $\tau_{xx}$ ,  $\tau_{yy}$  and  $\tau_{zz}$  are the normal stresses related to the pressure by  $p = (\tau_{xx} + \tau_{yy} + \tau_{zz})/3$ . The  $\tau_{ij}$  components when  $i \neq j$  are known as shear stresses. By inserting the relation 2.7 into the last term of the expression 2.6 and applying Gauss' divergence theorem

$$\iint_S T_i dS = \iint_S \tau_{ij} = \iiint_V \frac{\partial \tau_{ij}}{\partial x_j} dV \quad (2.9)$$

now we can rewrite 2.6 as

$$\iiint_V \left( \rho \frac{\partial^2 \mathbf{u}}{\partial t^2} - \mathbf{f} - \nabla \cdot \boldsymbol{\tau} \right) dV = 0 \quad (2.10)$$

and that must be satisfied in any arbitrary volume, so we can drop the integral from 2.10 and obtain

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \mathbf{f} + \nabla \cdot \boldsymbol{\tau} \quad (2.11)$$

I refer to Aki and Richards for a complete and strict derivation of the equation of motion including the demonstration of the symmetry of the stress tensor and the explanation of why derivatives with respect to Cartesian coordinates  $\mathbf{x}$  (Lagrange approach) are only an approximation to the real solution which contains derivatives with respect to the particle position  $\mathbf{X} = \mathbf{x} + \mathbf{u}$  (Eulerian approach).

Equations 2.5 and 2.11 are a coupled system of equations for stress and displacement which will be the core of the elastic wave propagation numerical solver that will be used for the inversion. For the sake of clarity, and for practical reasons, the coupled set of equations that conforms the elastic wave equation is written again in a more explicit manner by expanding 2.5 and 2.11

$$\begin{aligned}
\rho \partial_{tt} u_x &= \partial_x \tau_{xx} + \partial_y \tau_{xy} + \partial_z \tau_{xz} \\
\rho \partial_{tt} u_y &= \partial_x \tau_{xy} + \partial_y \tau_{yy} + \partial_z \tau_{yz} \\
\rho \partial_{tt} u_z &= \partial_x \tau_{xz} + \partial_y \tau_{yz} + \partial_z \tau_{zz} \\
\tau_{xx} &= (\lambda + 2\mu) \partial_x u_x + \lambda (\partial_y u_y + \partial_z u_z) \\
\tau_{yy} &= (\lambda + 2\mu) \partial_y u_y + \lambda (\partial_x u_x + \partial_z u_z) \\
\tau_{zz} &= (\lambda + 2\mu) \partial_z u_z + \lambda (\partial_x u_x + \partial_y u_y) \\
\tau_{xy} &= \mu (\partial_y u_x + \partial_x u_y) \\
\tau_{xz} &= \mu (\partial_z u_x + \partial_x u_z) \\
\tau_{yz} &= \mu (\partial_z u_y + \partial_y u_z)
\end{aligned} \tag{2.12}$$

Note that a more compact notation for the partial derivatives is used:  $\partial_a b$  ( $\partial_{aa} b$ ) represents the partial derivative of  $b$  with respect to  $a$  (twice). Also the external forces have been dropped, so now the equations represent the response of a medium in absence of sources. The source term can be inserted in a variety of ways as will be shown in the next section 2.2.4, and it will not be discussed in this section.

When the medium is excited using a point source the solutions for the elastic wave equation are known as elastodynamic Green functions [Aki and Richards, 2002], and they satisfy

$$\rho \frac{\partial^2}{\partial t^2} G_{in} = \delta_{in} \delta(x - x_0) \delta(t - t_0) + \frac{\partial}{\partial x_j} \left( c_{ijkl} \frac{\partial}{\partial x_l} G_{km} \right) \tag{2.13}$$

where the first term of the right hand side represents an impulse source located at  $x_0$  occurring at  $t_0$ , and the Green functions are denoted by  $G$ .

### 2.1.2 Alternative formulations of the wave equation

So far, the elastic wave equation has been expressed as a function of particle displacements  $\mathbf{u}$ , stresses  $\boldsymbol{\tau}$  and the Lamé parameters  $\lambda$  and  $\mu$ . In the next section a numerical solution will be introduced, and it will include discretisation of both space and time and introduction of absorbing boundary conditions to limit the region in which the solution is computed. It is more convenient to introduce these boundary conditions in a slightly different set of variables, namely particle velocities instead of displacements. The relation between particle velocities  $\mathbf{v}$  and displacements  $\mathbf{u}$  is, obviously

$$\mathbf{v} = \partial_t \mathbf{u} \quad (2.14)$$

so, by differentiating 2.12 with respect to time leads to

$$\begin{aligned} \rho \partial_t v_x &= \partial_x \tau_{xx} + \partial_y \tau_{xy} + \partial_z \tau_{xz} \\ \rho \partial_t v_y &= \partial_x \tau_{xy} + \partial_y \tau_{yy} + \partial_z \tau_{yz} \\ \rho \partial_t v_z &= \partial_x \tau_{xz} + \partial_y \tau_{yz} + \partial_z \tau_{zz} \\ \partial_t \tau_{xx} &= (\lambda + 2\mu) \partial_x v_x + \lambda (\partial_y v_y + \partial_z v_z) \\ \partial_t \tau_{yy} &= (\lambda + 2\mu) \partial_y v_y + \lambda (\partial_x v_x + \partial_z v_z) \\ \partial_t \tau_{zz} &= (\lambda + 2\mu) \partial_z v_z + \lambda (\partial_x v_x + \partial_y v_y) \\ \partial_t \tau_{xy} &= \mu (\partial_y v_x + \partial_x v_y) \\ \partial_t \tau_{xz} &= \mu (\partial_z v_x + \partial_x v_z) \\ \partial_t \tau_{yz} &= \mu (\partial_z v_y + \partial_y v_z) \end{aligned} \quad (2.15)$$

## 2.2 Implementation

The elastic wave equation presented in the previous section can be solved analytically only in a reduced number of simple cases — for example homogeneous models or models varying linearly with one of the coordinates. Thus, in order to handle arbitrarily complex

models, the differential equations need to make use of numerical methods in order to find an approximate solution.

### 2.2.1 Numerical methods

The most common numerical techniques applied to solve partial differential equations in the geophysics community are:

1. FD: finite differences.
2. FE: finite elements.
3. SM/PS: spectral and pseudo-spectral methods.

where all of them are based on discretisation both in space and time.

Mathematically, the wave equation can be posed as the PDE system

$$\mathcal{L}u = s \quad (2.16)$$

where the operator  $\mathcal{L}$  operates over the wavefield  $u$  produced by a source  $s$ . The aim of numerical methods is to obtain a residual  $R$  in

$$\mathcal{L}u - s = R \quad (2.17)$$

that is small. The wavefield  $u$  is approximated by a combination of trial functions and substituted in equation 2.16 to find a solution. In FD, the trial functions are overlapping polynomials which take nonzero values around every point. The shape of the subdomain in the FD case is constant and the resulting stencil is translated everywhere in the model to calculate the wavefield. Trial functions with subdomain shapes varying in space give rise to the finite-element method (FE), where the solutions are approximated by local functions (polynomials most commonly). Lastly, the spectral method uses trial functions which are defined globally —nonzero throughout the model volume. Common choices are high order polynomials or Chebyshev functions.

The finite-difference method is the simplest and fastest of the three. It is based on the definition of the derivative and the relaxation of its limit for the variable interval  $h$ . In one dimension, the derivative of a function  $f(x)$  at a point  $x_0$  is defined as

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h} \quad (2.18)$$

The approximation of 2.18 is achieved in the FD case by letting  $h$  have a finite value such as  $\Delta x$ , and thus discretising the continuous function  $f(x)$  in  $\Delta x$  intervals. There are 3 main variations of the method: forward, central and backward differences, with the central difference form yielding a more accurate approximation with errors of  $O(\Delta x)^2$  while the other two have errors proportional to  $O(\Delta x)$ . Equation 2.18 is approximated using central differences by

$$f'(x_0) = \frac{f(x_0 + \frac{\Delta x}{2}) - f(x_0 - \frac{\Delta x}{2})}{\Delta x} \quad (2.19)$$

This is our method of choice due to its smaller CPU-times and memory requirements when compared to other methods. Next section —2.2.2— describes the FD stencil used to discretise the elastic wave equation.

The finite-element method discretises the space in non-regular elements (as opposed to the FD constant interval  $\Delta x$ ) which can be arbitrarily complex. In 2D it is more common to define triangles or trapezoids, and in 3D the most common choices are irregular tetrahedra or cuboids. The solution wanted is then approximated by a combination of functions known as basis functions. In each element, the basis function is defined to be non-zero only inside the element. Such functions are usually chosen to be polynomials of any rather low order.

The main advantage of this method is that the size of the elements can be adjusted to minimise the number of elements used to solve a particular problem. This is exemplified in figure 2.2, where a P-wave velocity model is discretised with FE. Where the values of  $V_P$  are higher the elements are larger because the waves travel faster through them

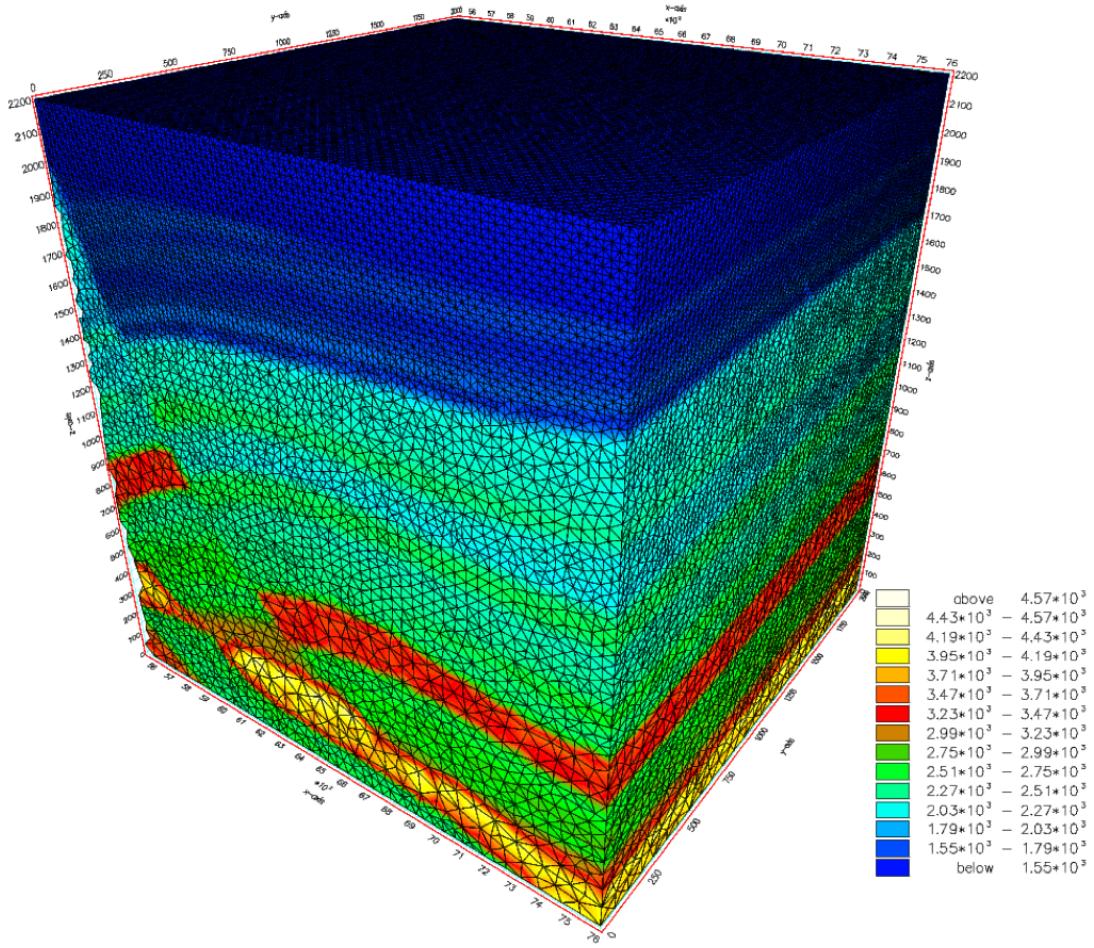


Figure 2.2: Finite-element mesh for subset of a 2.5D version of the Marmousi model.

(i.e., the distance crossed per time-step compared to the total size of the element is kept approximately constant throughout the model). A FE scheme has been implemented and tested by our research group. The cost of getting higher accuracy solutions is that CPU-times and memory are increased compared to the FD case.

Spectral methods exploit the previously exposed property of the Fourier transforms (equation 1.5) to transform the spatial derivatives terms and then solve them in the frequency domain when they are reduced to a simple complex multiplication. Spectral methods have not been tested in our group and it is beyond the scope of this work to enter into the details of it. They are added in the list only for the sake of completeness. Suffice to say that these methods tend to be more expensive than the other two and are not very well

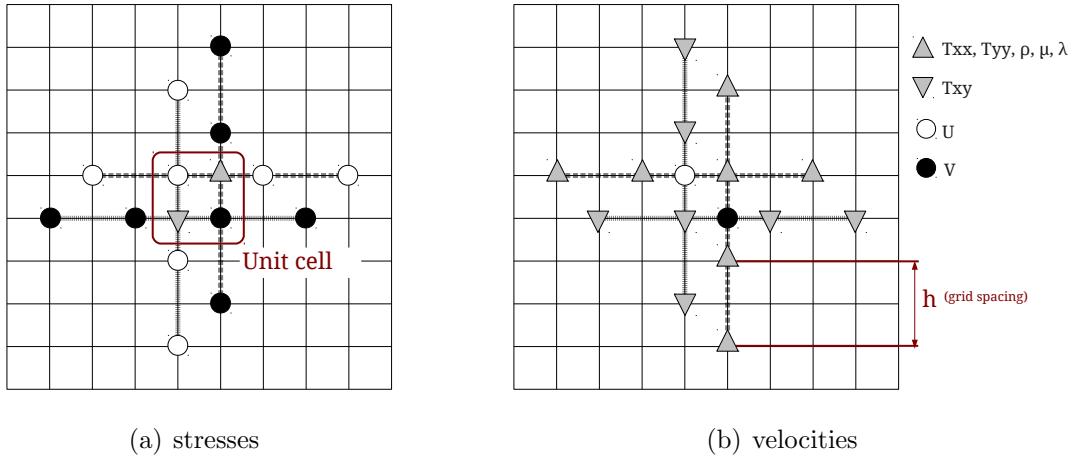


Figure 2.3: 2D FD staggered stencil. The velocities (stresses) used to calculate each stress (velocity) are connected by solid grey lines.

behaved in media with strong heterogeneities. There exists a family of methods based on the spectral principle of decomposing the solution of a PDE in sinusoids (or other functions) over all the computational domain. An important example is the Spectral Element method, where the basis functions are high order polynomials that substitute the original trigonometric series, and the solutions are non-zero only inside subdomains of the model (like in the FE case).

### 2.2.2 Finite-difference stencil

As mentioned in the literature review, Jean Virieux [1986] (using the velocity-stress formulation) and Allan Levander [1988] (using displacement-stress) presented their FD schemes in 2D around around 25 years ago. An extension to the 3D case is presented in this section, describing the accuracy of the stencils and its advantages and limitations. The elastic wave equation, 2.15, is discretised using 4<sup>th</sup> order in space and 2<sup>nd</sup> order in time finite differences, and is staggered in both domains. Virieux [1986] has shown that this implementation is valid for any Poisson's ratio and that liquid-solid interfaces do not need any external boundary imposition. Rotated operators may be used as shown in [Saenger et al., 2000], but I did not use them.

The 2D operators will be introduced first for the sake of simplicity. Figure 2.3 shows the distribution in space of velocities and stresses for the 2D case. The names of the variables have been modified in this new discretised environment:  $U, V$  are the approximate values of  $v_x, v_y$  at the gridpoints and  $T_{ij}$  are the approximate values of the continuous stresses  $\tau_{ij}$ . As shown in the picture, each velocity (stress) is computed using four neighbour stress (velocity) points in each direction, hence the spatial FD operator is of 4<sup>th</sup> order. Another benefit of using staggered grids is that we are effectively using central differences and, thus, the errors are proportional to  $O(\Delta x^2)$ , as mentioned in the previous section. There are three more quantities involved in the solution of the system and they are constant, or more precisely they are treated as constant values, through time: density  $\rho$  and the two Lamé parameters  $\lambda$  and  $\mu$ .

The unit cell is the smallest portion of the stencil that can generate all the grid by repeating its pattern over all the domain (in figure 2.3 the unit cell is highlighted in a red square). The extension to 3D consists of adding an extra spatial dimension and staggering it accordingly to allocate the four new variables that arise:  $W, T_{zz}, T_{xz}$  and  $T_{yz}$ . The unit cell is shown in figure 2.4. Because the operator is of 4<sup>th</sup> order, as in 2D case, each stress (velocity) is computed by combining the values of 4 velocities (stresses) in each direction, so now 12 points are involved in the calculation instead of only 8. The FD operator is staggered in time as well as in space. Time is discretised in time-steps  $\Delta t$ , and the stresses are computed half a time-step away from the particle velocities. Figure 2.5 shows the time positions of both these variables (in 2D for the sake of clarity). At time  $t_{i+1}$ , the stress in the middle of the stencil uses velocities at  $t_{i+1/2}$  (and some more beyond the limits of the figure for higher order in space operators) and the same stress point at  $t_i$ . Once all the stress-points have been computed at  $t_{i+1}$ , the velocities at  $t_{i+1+1/2}$  are computed in a similar way using stresses at  $t_{i+1}$  and velocities at  $t_{i+1/2}$ . This process is repeated until all the desired time-steps have been computed.

Now, all the continuum variables have been discretised and positioned in space and time. The language used to code the algorithm was Fortran90. The main reason for that

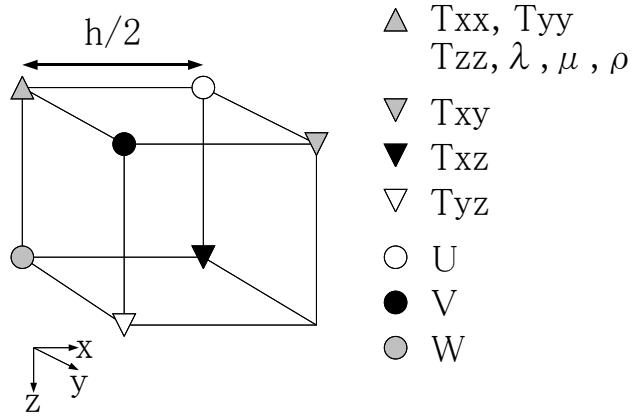
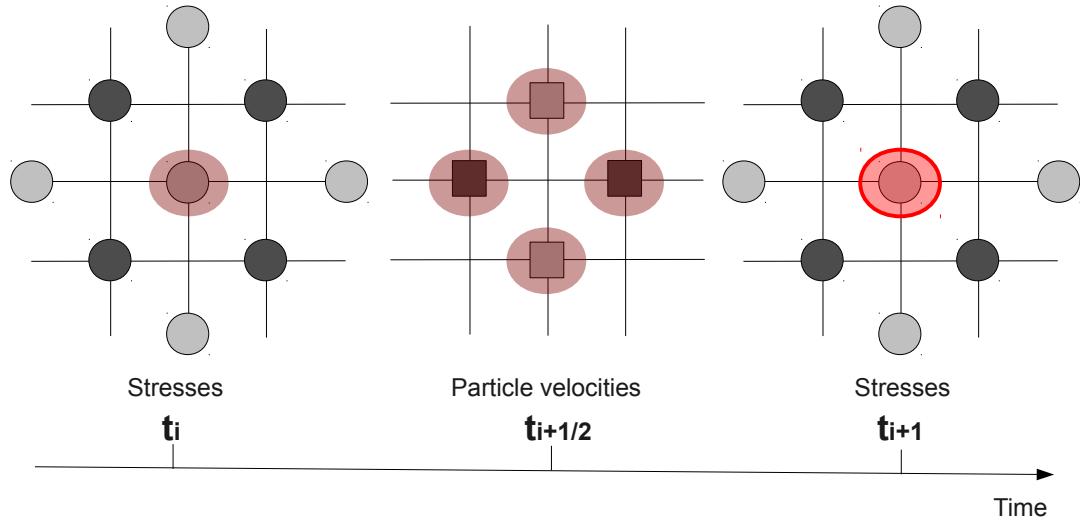


Figure 2.4: Unit cell in 3D.

Figure 2.5: Stresses are defined at  $n\Delta t$  and velocities at  $(n + 1/2)\Delta t$ . The set of variables involved in the calculation of the central point on the right figure are highlighted in red.

choice is that low level languages result in codes that perform much faster when compared to interpreted languages. In Fortran90 it is also easy to parallelise using specialised

APIs (application programming interfaces) such as MPI or OpenMP (section 2.3.2). The complete set of pseudo-code expressions for the FD scheme are in the appendix A, only one component of the particle velocity ( $U$ ) and one stress ( $T_{xy}$ ) are shown here

$$\begin{aligned} U_{i+1/2,j,k}^{t+1/2} = & U_{i+1/2,j,k}^{t-1/2} + \frac{\Delta t}{h\rho} \left[ c_1(T_{i+1,j,k}^{xx,t} - T_{i,j,k}^{xx,t}) - c_2(T_{i+2,j,k}^{xx,t} - T_{i-1,j,k}^{xx,t}) \right. \\ & + c_1(T_{i+1/2,j+1/2,k}^{xy,t} - T_{i+1/2,j-1/2,k}^{xy,t}) - c_2(T_{i+1/2,j+3/2,k}^{xy,t} - T_{i+1/2,j-3/2,k}^{xy,t}) \\ & \left. + c_1(T_{i+1/2,j,k+1/2}^{xz,t} - T_{i+1/2,j,k-1/2}^{xz,t}) - c_2(T_{i+1/2,j,k+3/2}^{xz,t} - T_{i+1/2,j,k-3/2}^{xz,t}) \right] \end{aligned} \quad (2.20)$$

$$\begin{aligned} T_{xy,t+1}^{t+1} = & T_{xy,t}^{t+1} + \frac{\Delta t}{h} \mu \left[ c_1(U_{i+1/2,j+1,k}^{t+1/2} - U_{i+1/2,j,k}^{t+1/2}) - c_2(U_{i+1/2,j+2,k}^{t+1/2} - U_{i+1/2,j-1,k}^{t+1/2}) \right. \\ & \left. + c_1(V_{i+1,j+1/2,k}^{t+1/2} - V_{i,j+1/2,k}^{t+1/2}) - c_2(V_{i+2,j+1/2,k}^{t+1/2} - V_{i-1,j+1/2,k}^{t+1/2}) \right] \end{aligned} \quad (2.21)$$

where  $i, j, k$  correspond to  $x, y, z$  coordinates and  $t$  is the time-step number. As mentioned previously in this chapter, the elastic tensor is symmetric, and therefore  $T^{lm}$  is equal to  $T^{ml}$ , with  $m, l \in \{x, y, z\}$ . The two coefficients  $c_1$  and  $c_2$  take values  $9/8$  and  $1/24$  respectively and control the relative weight of the neighbour and outer values of the  $4^{th}$  order approximation to the first derivative [Levander, 1988]. They are chosen to minimise the error of the solution in homogeneous media by comparing the results with the analytical solution.

Theoretically, the FD scheme shown here is complete and it could be used to solve the elastic wave equation. The only inconvenience is that some of the variables are defined at some points and used for the calculation at others. In equation 2.20, density is involved in the calculation, but as shown in figure 2.4,  $\rho$  is not defined at the same grid-points as  $U$ . A similar situation arises in equation 2.21, where  $\mu$  is not defined at the same points as  $T_{xy}$ . In media with smooth variations the error produced by this misalignment of values is small, but in the presence of strong contrasts they can become a problem. Average values of  $\rho$  and  $\mu$  in the surroundings of the interest point are used. They are known

as effective-media parameters, and they vary depending on the direction or directions in which they are used [Graves, 1996]. Density is involved in the calculation of particle velocities which are staggered only in one direction, and thus the effective  $\rho$  is

$$\begin{aligned}\frac{1}{\rho_{i+\frac{1}{2},j,k}^x} &= \frac{1}{2} \left[ \frac{1}{\rho_{i,j,k}} + \frac{1}{\rho_{i+1,j,k}} \right] \\ \frac{1}{\rho_{i,j+\frac{1}{2},k}^y} &= \frac{1}{2} \left[ \frac{1}{\rho_{i,j,k}} + \frac{1}{\rho_{i,j+1,k}} \right] \\ \frac{1}{\rho_{i,j,k+\frac{1}{2}}^z} &= \frac{1}{2} \left[ \frac{1}{\rho_{i,j,k}} + \frac{1}{\rho_{i,j,k+1}} \right]\end{aligned}\quad (2.22)$$

Analogue expressions for  $\mu$  are

$$\begin{aligned}\mu_{i+\frac{1}{2},j+\frac{1}{2},k}^{xy} &= \left[ \frac{1}{4} \left( \frac{1}{\mu_{i,j,k}} + \frac{1}{\mu_{i+1,j,k}} + \frac{1}{\mu_{i,j+1,k}} + \frac{1}{\mu_{i+1,j+1,k}} \right) \right]^{-1} \\ \mu_{i+\frac{1}{2},j,k+\frac{1}{2}}^{xz} &= \left[ \frac{1}{4} \left( \frac{1}{\mu_{i,j,k}} + \frac{1}{\mu_{i+1,j,k}} + \frac{1}{\mu_{i,j,k+1}} + \frac{1}{\mu_{i+1,j,k+1}} \right) \right]^{-1} \\ \mu_{i,j+\frac{1}{2},k+\frac{1}{2}}^{yz} &= \left[ \frac{1}{4} \left( \frac{1}{\mu_{i,j,k}} + \frac{1}{\mu_{i,j+1,k}} + \frac{1}{\mu_{i,j,k+1}} + \frac{1}{\mu_{i,j+1,k+1}} \right) \right]^{-1}\end{aligned}\quad (2.23)$$

where now, the averaging involves four points instead of two because the shear stresses are staggered in two dimensions. Note that in both cases, the effective parameters only use values in the line or plane of action of the variable (for example,  $\rho^x$  does not use any values along the  $x$  axis).

The fact that the variables are staggered both in space and time may seem confusing because it is not clear how to define an explicit half time-step or half grid-spacing in arrays with integer indices. The time steps do not need to be explicitly halved because the order of the elements in the matrix will define the order of the operations performed, and therefore the staggered nature in time will be concealed in it. For the model parameters, they are defined in a grid with  $\Delta x$  spacing, but the distance between the grid points containing a particular parameter is always  $\Delta x$ . Because array indices in programming languages are independent of any geometry that they may represent, assigning the same array position to different parameters does not create any conflict. For example, equation

2.27 is translated into Fortran90 as:

$$\begin{aligned}
 \text{txx}(i, k) = & \text{txx}(i, k) + dt * ((\text{lam}(i, k) + 2. * \text{mu}(i, k)) * \\
 & (\text{c1} * (\text{u}(i, k) - \text{u}(i - 1, k)) - \text{c2} * (\text{u}(i + 1, k) - \text{u}(i - 2, k))) \\
 & + \text{lam}(i, k) * (\text{c1} * (\text{w}(i, k) - \text{w}(i, k - 1)) - \text{c2} * (\text{w}(i, k + 1) - \text{w}(i, k - 2)))
 \end{aligned} \tag{2.24}$$

where the staggered positions have been moved half a cell according to the unit cell (figure 2.4), and the variable names are adapted to code language (`lam` is  $\lambda$  and so on). The absence of time step indices is because the array is overwritten as the FD stencil goes through all the cells of the model. Note that this is a safe procedure because all the values being overwritten have already been used to compute the wavefield in the previous ‘half’ time-step. For this to be true though, the system must be solved in an alternate order: all the stresses are computed first, and then all the particle velocities (or the other way around).

Free surfaces are not implemented in the current code and they will be introduced in the next stage of the project.

### 2.2.3 Stability conditions

Due to the approximation of the derivatives using FD operators, the solution of the discretised elastic wave equation is only correct if the stability and dispersion conditions are met. These limit the range of values that the time-step interval and the grid spacing can take for a particular model and source.

The dispersion condition imposes spatial sampling of at least 5 gridpoints per wavelength in order to avoid numerical dispersion [Levander, 1988]. Note that in most situations the shortest wavelength appears in the lowest S-wave region of the model, and therefore the shear wave velocity model normally limits the grid spacing  $\Delta h$ .

On the other hand, the time-step interval should obey the stability criterion described

in [Graves, 1996]

$$\Delta t < \frac{h}{V_{Pmax}\sqrt{3}} \left[ \sum_{i=0}^N c_i \right]^{-1} \quad (2.25)$$

where  $V_{Pmax}$  is the maximum velocity in the model —which always correspond to pressure wave velocity,  $h$  is the grid-spacing and  $c_i$  are the FD operator coefficients,  $c_0 = 9/8$  and  $c_1 = 1/24$  [Levander, 1988]. Substituting these values in (2.25) gives

$$\Delta t < \frac{h}{v_{max}\sqrt{3}} \quad (2.26)$$

which means that it is not allowed to cross more than  $1/\sqrt{3}$  in the highest velocity cells. From the above mentioned constraints, it follows that the maximum space sampling is controlled by the lowest velocity in the model, whereas the maximum time sampling is controlled by the highest velocity.

## 2.2.4 Source insertion

The numerical solution presented so far did not include a source term. Obviously, in order to model energy propagating in our medium one has to introduce it first. One of the advantages of the staggered scheme is that the source can be inserted using velocities or stresses. This is convenient because different datasets may demand different ways of inserting sources, especially when we have to backpropagate different kinds of data as will be shown in the next chapter when we introduce the use of the adjoint method in the inversion. In short, there are different types of data that can be inverted: pressure data from hydrophones, 3 component data from ocean bottom cables (OBC), vertical motion geophones from land acquisitions, etc. These different kinds of data, after some modification (subtraction of a modelled wavefield), are used as sources in a way where each trace plays the role of a source signature inserted at the receiver position. Hence, the convenience of being able to use pressure or particle velocities to introduce the source into the model.

In any case, all sources are represented as time series discretised in time. The numerical dispersion and stability criteria that applied to the FD scheme constrain the source characteristics as follows:

1. **Maximum frequency** in the source spectrum must not be higher than the maximum frequency that can be modelled according to dispersion constraints. Higher frequencies in the source lead to numerical dispersion in the modelled wavefields.
2. **Time sampling** of the source should be equal to the time-step  $dt$  used for the modelling —constrained by 2.26.

In the context of seismic exploration, the medium can be excited in two ways: applying pressure in the form of explosions or moving the medium attached to a source in a particular direction. One of the most common sources in seismic surveys is a pressure point-source. Such sources may be, for example, an air-gun shot (offshore), a buried explosive (onshore), a spark source inside a well in a VSP experiment or backpropagated hydrophone data.

The implementation of the source in the stress variables is straightforward, the previously presented relation  $p = (\tau_{xx} + \tau_{yy} + \tau_{zz})/3$  shows a linear relation between pressure and normal stresses. Then, inserting a pressure source only requires an addition to the normal stress variables at each time-step. For example, the source is inserted into the normal stress  $T_{xx}$  as

$$\begin{aligned} T^{xx,t+1} &= T^{xx,t} + \frac{\Delta t}{h} \left\{ \mathbf{src}^{t+1} + (\lambda + 2\mu) \left[ c_1(U_{i+1/2,j,k}^{t+1/2} - U_{i-1/2,j,k}^{t+1/2}) - c_2(U_{i+3/2,j,k}^{t+1/2} - U_{i-3/2,j,k}^{t+1/2}) \right] \right. \\ &\quad + \lambda \left[ c_1(V_{i,j+1/2,k}^{t+1/2} - V_{i,j-1/2,k}^{t+1/2}) - c_2(V_{i,j+3/2,k}^{t+1/2} - V_{i,j-3/2,k}^{t+1/2}) \right. \\ &\quad \left. \left. + c_1(W_{i,j,k+1/2}^{t+1/2} - W_{i,j,k-1/2}^{t+1/2}) - c_2(W_{i,j,k+3/2}^{t+1/2} - W_{i,j,k-3/2}^{t+1/2}) \right] \right\} \end{aligned} \quad (2.27)$$

where  $\mathbf{src}^{t+1}$  is the source sample number  $t + 1$ . The same procedure is followed for the other normal stresses.

A similar procedure may be followed if the source is added to particle velocities. There exists an important difference though, they are not centered with respect to the unit cell.

Normal stresses are defined at the same point in the grid, but the particle displacements are not. A dipole system is used to avoid non-spherical wavefronts due to the asymmetry in the insertion points. Figure 2.6 shows how the source signature is added to the velocity nodes around the origin node of the unit cell. Changing the orientation of the dipoles has

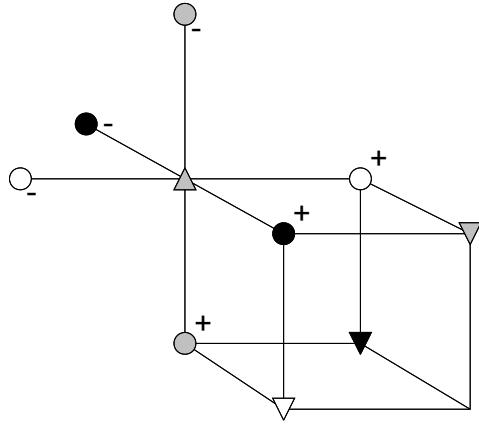


Figure 2.6: The three particle-velocity dipoles simulate a point source centered at the normal stress point. Sources inserted using normal stresses do not need any dipoles because they are all defined at the same point.

the effect of reversing the polarity of the source signature. The corresponding pseudo-code expression for the  $x$  component of the velocity is

$$\begin{aligned}
 U_{i+1/2,j,k}^{t+1/2} = & U_{i+1/2,j,k}^{t-1/2} + \frac{\Delta t}{h\rho} [\mathbf{src}^{t+1/2} + c_1(T_{i+1,j,k}^{xx,t} - T_{i,j,k}^{xx,t}) - c_2(T_{i+2,j,k}^{xx,t} - T_{i-1,j,k}^{xx,t}) \\
 & + c_1(T_{i+1/2,j+1/2,k}^{xy,t} - T_{i+1/2,j-1/2,k}^{xy,t}) - c_2(T_{i+1/2,j+3/2,k}^{xy,t} - T_{i+1/2,j-3/2,k}^{xy,t}) \\
 & + c_1(T_{i+1/2,j,k+1/2}^{xz,t} - T_{i+1/2,j,k-1/2}^{xz,t}) - c_2(T_{i+1/2,j,k+3/2}^{xz,t} - T_{i+1/2,j,k-3/2}^{xz,t})]
 \end{aligned} \tag{2.28}$$

where now, the source sample is at position  $t + 1/2$  instead of  $t + 1$  because it should be on the same space grid as  $U$ . Since pressure is a scalar, the source signature can be scaled to any value that optimises the numerical accuracy of the floats.

Media excited using vertical impacts on the ground or vibroseis need to be introduced in a slightly different way. In this case, we only need to excite the source point along one direction. In this case, the particle velocity of the source itself is the source signature

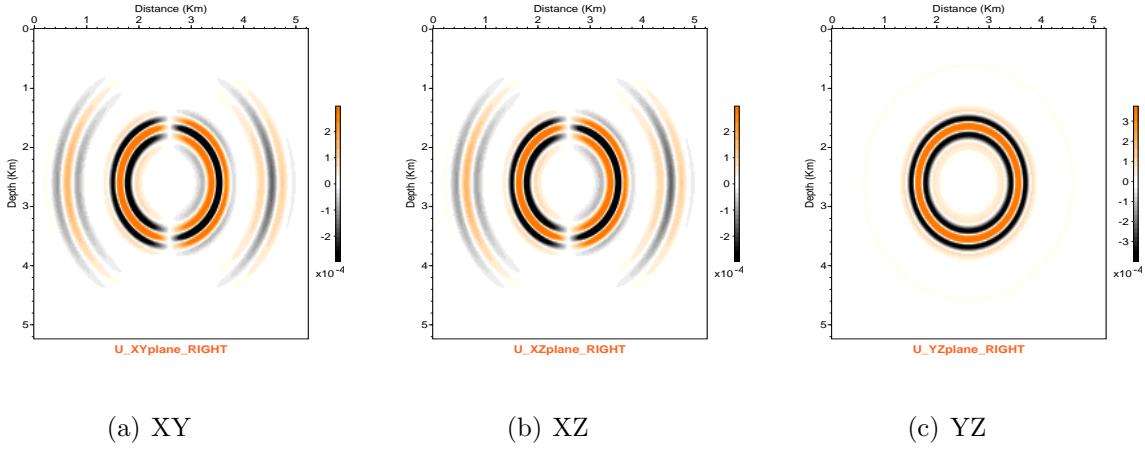


Figure 2.7: Three perpendicular slices of a cubic homogeneous model with  $V_P = 3000\text{m}/\text{x}$  and  $V_S = 1732\text{m}/\text{x}$ , after 0.8 seconds showing the particle velocities in the  $\hat{x}$  direction. In (c) the P-wave front is very weak because the  $\hat{x}$  component of the particle velocity is perpendicular to the YZ plane.

because it is added in to the system velocities as follows

$$\begin{aligned}
 U_{i+1/2,j,k}^{t+1/2} = & U_{i+1/2,j,k}^{t-1/2} + \mathbf{src}^{t+1/2} + \frac{\Delta t}{h\rho} \left[ + c_1(T_{i+1,j,k}^{xx,t} - T_{i,j,k}^{xx,t}) - c_2(T_{i+2,j,k}^{xx,t} - T_{i-1,j,k}^{xx,t}) \right. \\
 & + c_1(T_{i+1/2,j+1/2,k}^{xy,t} - T_{i+1/2,j-1/2,k}^{xy,t}) - c_2(T_{i+1/2,j+3/2,k}^{xy,t} - T_{i+1/2,j-3/2,k}^{xy,t}) \\
 & \left. + c_1(T_{i+1/2,j,k+1/2}^{xz,t} - T_{i+1/2,j,k-1/2}^{xz,t}) - c_2(T_{i+1/2,j,k+3/2}^{xz,t} - T_{i+1/2,j,k-3/2}^{xz,t}) \right]
 \end{aligned} \tag{2.29}$$

Other sources can be constructed using combinations of the two abovementioned methods. For seismological problems where the source can be considerably more complex and spread in space, more elaborate techniques must be used. Slip faults, for example, need shear stress terms to be introduced properly, but they will not be detailed here.

To check the performance of the code, I ran several tests with different velocity models. The first set consists of cubic homogeneous models with the source placed in the middle. The wavefront must be circular when waves travel through an homogeneous velocity medium as shown by figure 2.7. The next test was aimed to check the behaviour of the

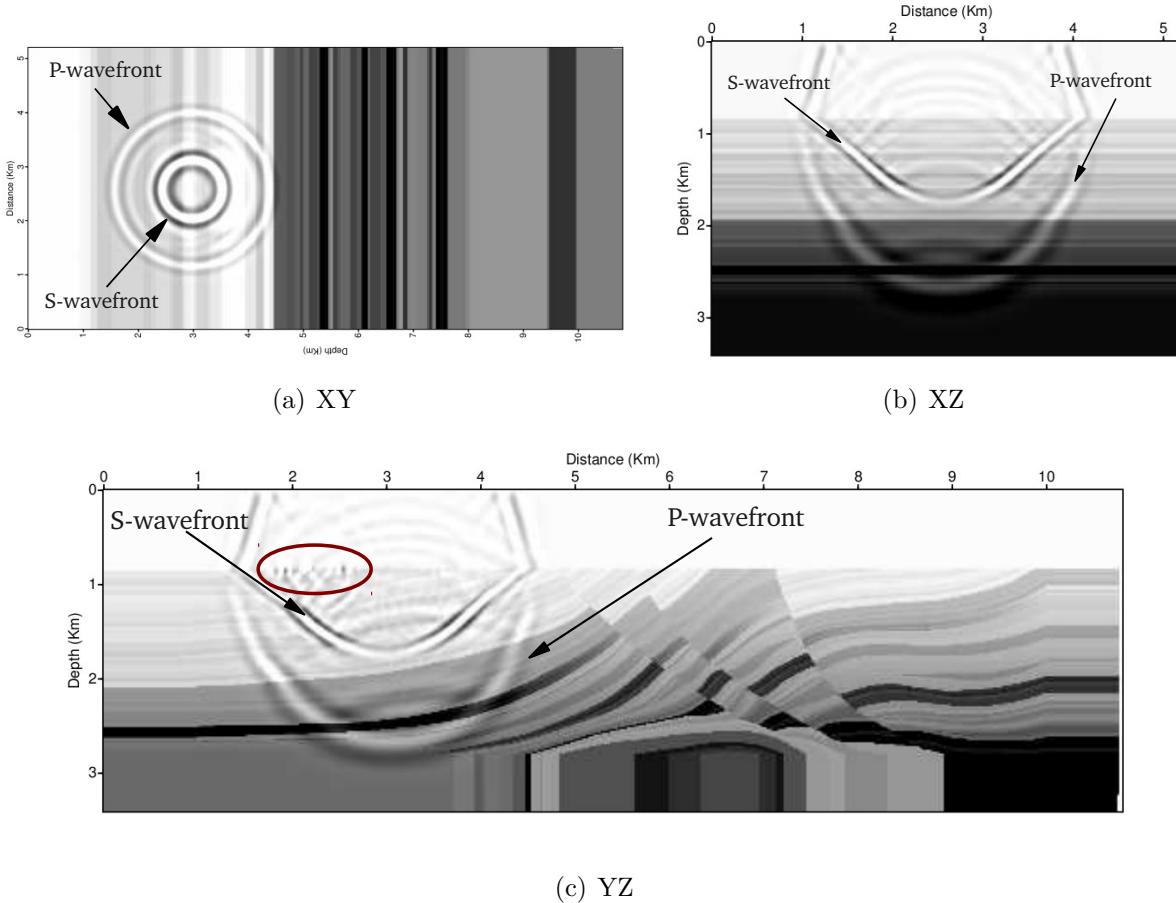


Figure 2.8: Marmousi 2.5D model snapshots of the velocity  $\hat{z}$  component. (a) XY:  $z = 1000\text{m}$ ; (b) XZ:  $y = 2400\text{m}$ ; (c) YZ:  $x = 2000\text{m}$ . The red ellipse highlights the response to a step shaped irregularity in the seabed.

code in the presence of fluid-solid interfaces —where S-wave velocity experiences a sharp contrast. A more complex model was also used to simulate a realistic complex structure. For that purpose, a 2.5D version of the Marmousi model was used. The dimensions of the original model were 481x101 nodes. We extended the model by 201 nodes in the third dimension. The top part of the model —first 11 nodes— corresponds to water. Some snapshots are shown in figure 2.8. There is just one wavefront in the water layer corresponding to compressional energy travelling through the fluid, which separates in two at the interface where conversions P to S occur. In figure 2.8 (c), an irregularity in the sea-bottom level acts as a diffractor at approximately 2.2 km offset. The PMLs extend the model in 1D at the edges of the model, that is why at the edges the picture looks

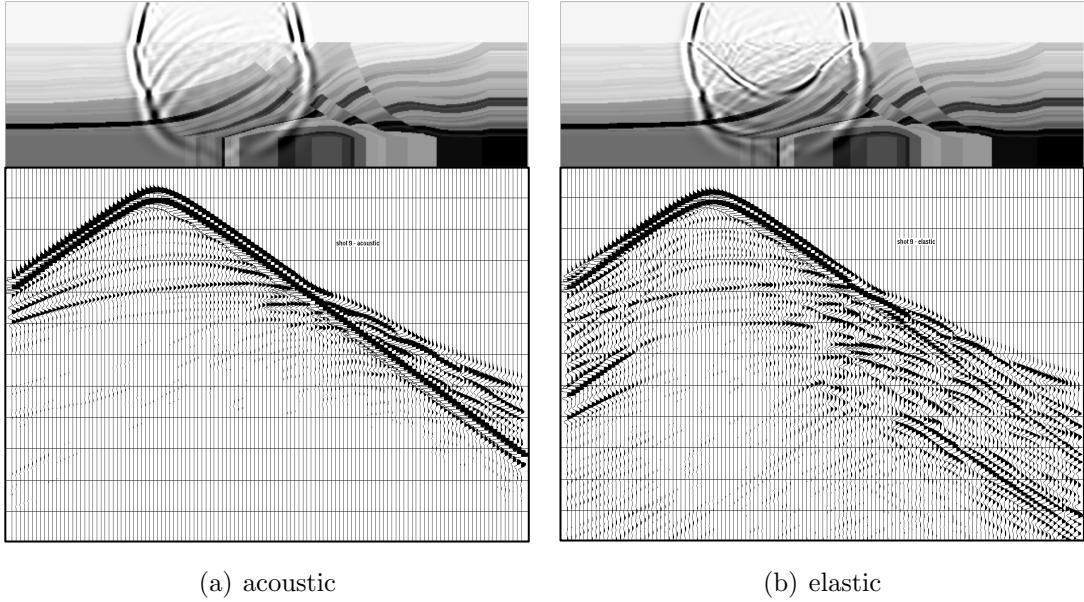


Figure 2.9: Seismograms for a shot generated using the acoustic (a) and elastic (b) wave equation. The shear wavefronts are recorded at later times creating all the late events recorded only in the elastic case.

as a downward continuation of the velocities. It cannot be noticed in the XY and XZ planes because they are already extended to form the 2.5D model. A comparison between acoustic and elastic data is shown in figure 2.9 along with the elastic YZ wavefield snapshot for the acoustic case. Note the amount of energy recorded at later times for the elastic case, demonstrating the need for an elastic formulation of the wave equation.

### 2.2.5 Absorbing boundary condition

Generally, numerical methods applied to solve wave equations in a region of space, face the problem of truncating the medium without creating spurious reflections from the artificially imposed boundaries. A review of the more common methods was presented in the previous chapter, and in this section the focus will be on the method used by me, which is based on the principle of stretching the space to accommodate a complex component in one, two or three dimensions.

The first method applied was the Unsplit Perfectly Matched Layers (U-PML) intro-

duced in the Literature review. In homogeneous, isotropic media, the analytical solution of the wave equation is of the form

$$\mathbf{A}e^{i(\mathbf{k}\cdot\mathbf{x}-\omega t)} \quad (2.30)$$

where  $\mathbf{A}$  is the amplitude,  $\mathbf{k}$  is the wavenumber,  $\omega$  is the frequency and  $t$  is time. The complex term of the exponential solution is telling us that the solution is oscillatory, and more than that, it will propagate indefinitely in the absence of energy absorption media (which is our case). This is true when  $\mathbf{x}$  takes real values. Let the  $x$  coordinate be transformed as

$$x' = x - i\gamma \quad (2.31)$$

where gamma is a parameter that will be discussed later. The new space coordinates are  $(x', y, z)$ . After this modification the solution takes the form

$$\mathbf{A}e^{i(\mathbf{k}\cdot\mathbf{x}-\omega t)-\gamma x} \quad (2.32)$$

The exponent of the solution has now a real term that will decrease the amplitude of the solution in the  $x$  direction and  $\gamma$  controls the strength of this damping.

From an analytical point of view, the parameter  $\gamma$  can take any value. It is even possible to use spatial dimensions which are purely complex [Asvadurov et al., 2004]. Figure 2.10(a) shows the complex-valued spatial dimension  $x$ . The reflection coefficient at the interface between real-valued and complex-valued is exactly zero, so no energy will be reflected here. Nonetheless, this is only true when the medium is continuous. In a discretised space the solution is not exact and the small errors of the solution give rise to some reflections in this interface. A discretised complex valued  $x$  is displayed in figure 2.10(b), where the jumps in  $\gamma$  are clearly appreciated. These spurious reflections can be minimised by choosing an appropriate damping function  $\gamma(x)$ . To simplify the notation, let  $n$  be the node position in a particular direction and  $d(n)$  the discretised  $\gamma$  function.

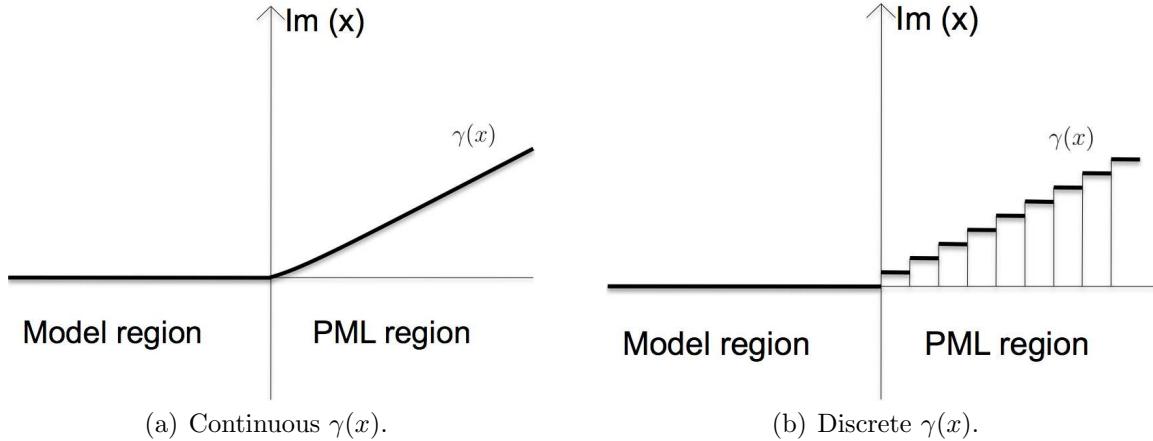


Figure 2.10: Imaginary parts of the  $x$  coordinate in continuous (a) and discrete (b) domains. In this schematic view, both figures depict a linear damping function although in practice an exponential shape performs better.

Collino and Tsogka [Collino and Tsogka, 1998] proposed the expression shown in equation (2.33) for the discrete damping function.

$$d(n) = d_0 \left( \frac{n}{\delta} \right)^2 \quad d_0 = \log \left( \frac{1}{R} \right) \frac{3V_p}{2\delta} \quad (2.33)$$

where  $d(n)$  is the value of the damping function at the  $n$ -node,  $d_0$  is the damping value at the outer edge of the model,  $\delta$  is the thickness of the PML in number of nodes,  $R$  is the theoretical reflection coefficient and  $V_p$  is the P-wave velocity. The values of the damping parameter are quadratic with respect to the position divided by the depth of the PML. The numerical impedance is minimised near the inner parts of the absorbing region because the imaginary part is increasing slowly. As a wave approaches the outer edge of the PML, the numerical impedance is increased but the amplitudes have decreased due to the damping. When the waves reach the outer edge, where a zero Dirichlet boundary condition applies, the wave is perfectly reflected and damped again on its way back into the model. Despite this reflected energy in discrete models, PMLs are still a much better choice than other absorbing boundaries [Collino and Tsogka, 1998, Komatitsch and Tromp,

2003, Marcinkovich and Olsen, 2003].

From equation (2.33) one can see that increasing the thickness of the PML results in a better absorption, and a decreasing of numerical impedance, because the shape of the damping curve is smoother and obviously the damping region is larger and the waves stay there longer (see figure 2.11).

The main drawback of such absorbing boundaries is that the physical memory needed to run the simulations increases substantially, due to two reasons. First, PMLs need extra nodes surrounding the model and the total model size is increased. Second, some integral terms appear in the wave equation after introducing the coordinate stretching. In order to calculate these integrals, some extra variables need to be stored. In the present implementation, the total number of variables is 18 times the number of nodes within the PML. That means, that in a model of size  $400 \times 400 \times 400$  cells, the introduction of an absorbing boundary of 20 nodes/face, for example, the memory increases 1.5 GB (see figure 2.15).

The complete derivation of the PML formulation is shown in the Appendix B. The wave equation is solved inside the PML using  $2^{nd}$  —instead of  $4^{th}$ — order in space FD. Merging different order FD equations does not affect the accuracy or the stability of the results [Papazachos, 2006] and simplifies the formulation.

## 2.3 Optimization

Full-waveform inversion is an iterative method and at each iteration at least three forward problems need to be computed. Typically, the number of iterations required lay between a few tens and a few hundreds, leading to three times the number of numerical solutions of the wave equation. It is crucial then, to optimise the performance of the algorithm as much as possible in order to reduce both memory usage and CPU-time. To that end, the algorithm is parallelised both for shared and for distributed systems. Besides, the computational domain is carefully chosen to avoid regions of the model which are not

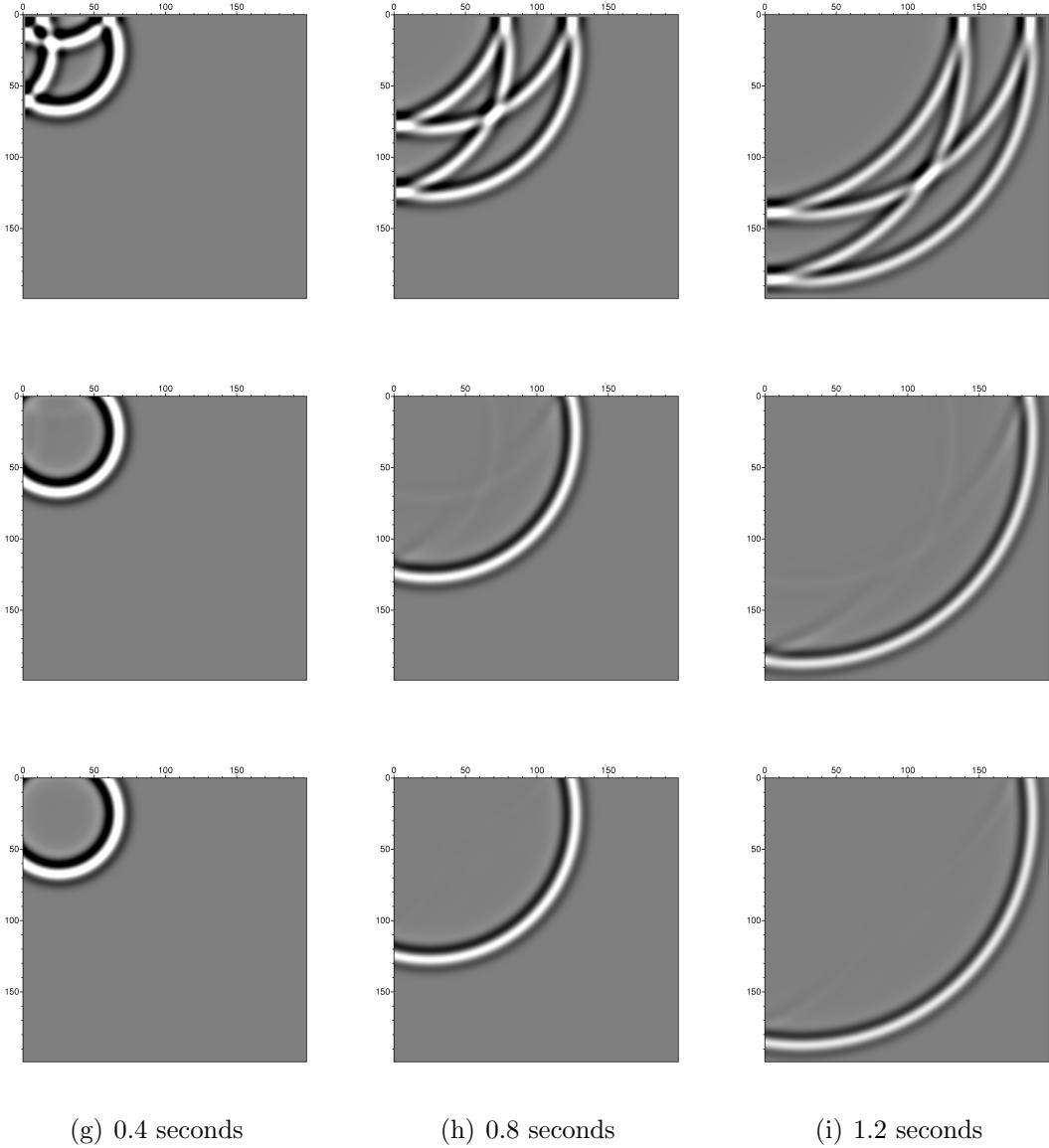


Figure 2.11: 2D slices of 3D pressure wavefields. The model is a cube of size 200x200x200 nodes. The source is a ricker wavelet with peak frequency of 8 Hz and the grid spacing is 20 m. Both  $V_P$  and  $V_S$  are homogeneous with values 2500 and 1440 m/s respectively. Density is 2000 kg/m<sup>3</sup> everywhere. In the top row no PML was applied and the reflections from the boundaries are clearly visible as the wavefront advance through time. The middle row of snapshots were computed using 10 nodes/face PML and in this case the amplitude of the reflections have decreased considerably. The bottom shows snapshots with 20 nodes/face PML. All the pictures use the same amplitude scale.

relevant to the recorded data at the receivers.

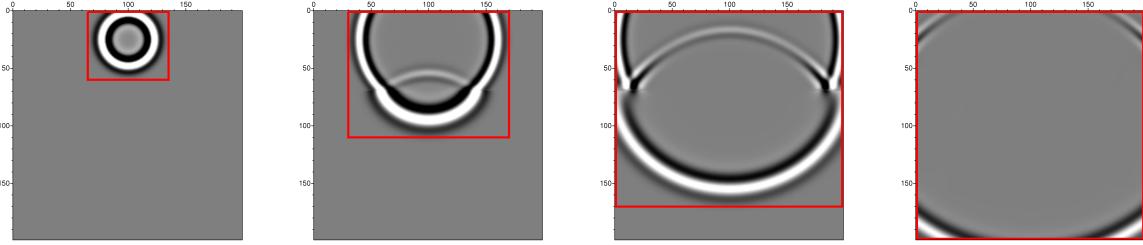


Figure 2.12: Pressure snapshots taken at 0.3, 0.6, 0.9 and 1.2 seconds in a two half-spaces model. The red box encompass the points where the solution is computed.

### 2.3.1 Adaptive computational domain

A seismic exploration survey consists of sources and receivers spread in space to cover the target area. Individual sources excite the medium and its response is registered in the receiver positions, which record a time series of length  $t_{total}$ . The source only covers a small region of a few nodes in the discretised model, and the energy inserted needs time to propagate through all the domain. As shown before by the stability condition, the time-step must be small enough so that energy does not cross more than roughly half a cell for the fastest value of  $V_P$ , and that implies that during the first time-steps, the wavefield is non-zero only in a region around the source point. An efficient way to speed-up the algorithm is to restrict the computational domain to these subspaces where energy have had time to arrive. The size of the subdomain will increase with time until it matches the model size as illustrated in figure 2.12. The starting domain covers a box of 8x8x8 because the FD operator uses a few neighbouring points.

But the same principle remains true applied to the receivers because as the current time-step approaches  $t_{total}$ , only energy in certain regions in the vicinity of the receivers will be recorded. Therefore, the regions further from the receivers are not computed at later stages of the modelling. Suppose that  $t_{total} = 1.2\text{s}$  in the previous example, and receivers are deployed at the same depth as the source. Then, the computational domain will shrink until it collapses in a thin layer of volume containing the receivers and a few adjacent cells as shown in figure 2.13. The wave equation is then solved in the time-dependent region defined by the intersection of these two boxes.

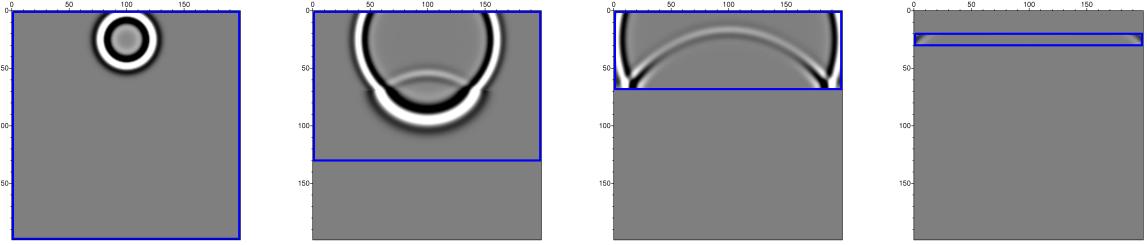


Figure 2.13: The blue box reduces its size as the simulation is complete, collapsing around the receivers. Note that the wavefield have been zeroed outside the box to highlight the relevant energy to the receivers.

It is easier and more efficient to implement a cuboid shaped box because the grid is rectilinear and only 6 integers are required to define its shape. More complex shapes such as spheres or ellipsoids are not easy parametrised and require significant extra memory and/or calculations to be applied. The highest velocity point at each face of the box (for the source) defines the expansion rate at each time step. The shrinking box (for the receivers) is obtained by the same procedure as for the growing box, but backwards in time, and considering the receiver gather as a composite source, so the final shrunk box collapses to a rectangle in the XY plane —if the receivers are all at the same depth.

We have observed an average saving in CPU time of around 25–35% using model sizes and times representative of real seismic acquisitions. This speed-up, though, can vary significantly with the size of the model, the position of sources and receivers and the total time of the simulation.

### 2.3.2 Parallelisation

Modern computers have CPU clock rates and random-access memory (RAM) that cannot cope with the demands of real sized FWI problems, mainly due to the elevated number of sources present in the data. The use of computer clusters can mitigate this problem by distributing the work-load. In fact, parallel computation is the main reason why 3D waveform inversion is feasible on a reasonable time-scale (less than a month) nowadays.

The algorithm has been parallelised at two different levels to optimise efficiency. First

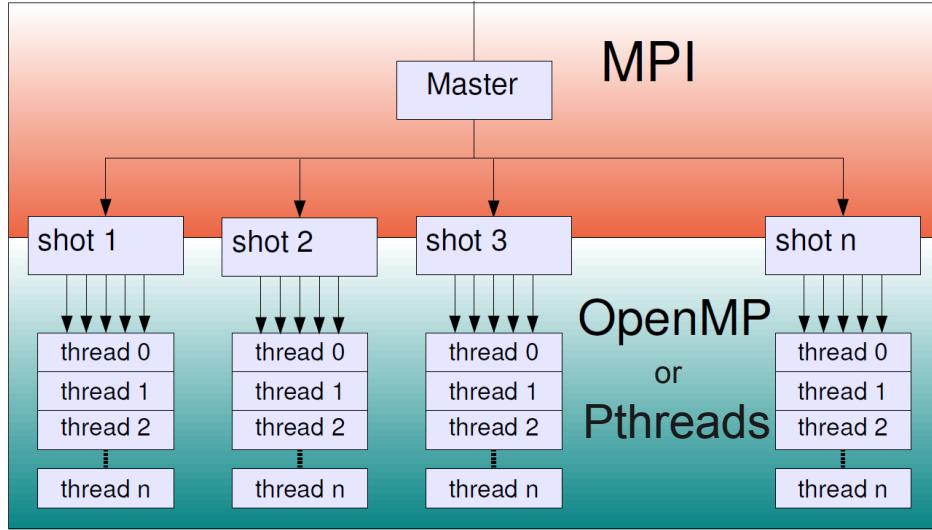


Figure 2.14: The master node passes information to all the slaves in the cluster using MPI (red area). Each node executes the instructions given by the master using all the available cores (blue area).

and most important, MPI —Message Passing Interface, an API used to communicate between processes in distributed memory systems— is used to distribute different shots across different nodes in distributed memory system of computers commonly known as clusters. Second, on each node of the cluster, each shot is computed taking advantage of modern multi-core architectures by breaking the model in small subdomains that are sent to each core to be solved. For that purpose, both OpenMP and POSIX threads have been tested, and we concluded that Pthreads outperform OpenMP shared memory parallelisations by approximately 20%. Figure 2.14 illustrates the two different levels of parallelisation applied. I have not implemented domain decomposition and therefore the memory in the nodes limits the maximum size of the models.

## 2.4 CPU-times and memory requirements

The impact in memory due to the absorbing boundary conditions is illustrated in figure 2.15. On average, the use of PMLs represents an increase of between 30 and 50% for real-sized models (more than 5 millions cells), however the shape of the model drives the

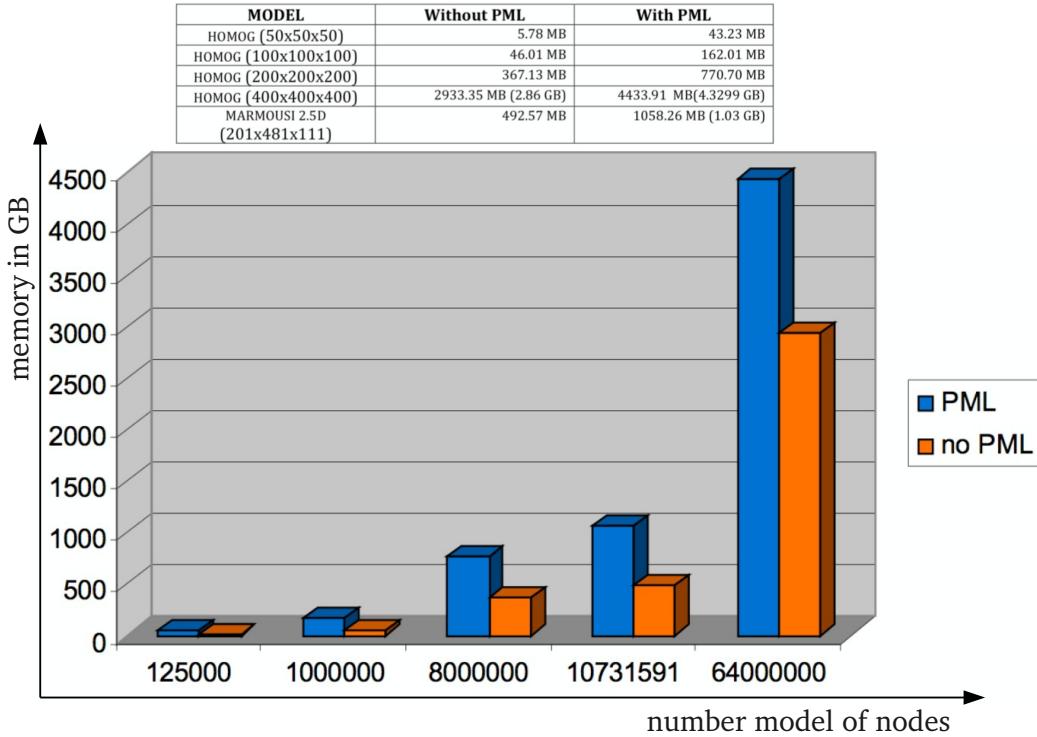


Figure 2.15: Memory requirement table and graph for different models. The thickness of the PML is 20 cells in all the models.

relation between the number of nodes within the model and the number of nodes in the PMLs, and this results are only valid when one or two dimensions are significantly bigger than the rest.

I designed a series of tests aimed to assess the memory demands of the code and the effect of optimisation. The models used to that goal were: Four cubes of increasing size of  $50 \times 50 \times 50$ ,  $100 \times 100 \times 100$ ,  $200 \times 200 \times 200$  and  $400 \times 400 \times 400$  plus a realistic geological model (Marmousi 2.5D). The computer used for the tests has x86-64 architecture and two quad-core microprocessors operating at a clock frequency of 2800 MHz. The total random access memory of the computer is higher than the maximum memory used by the program in all the tests to ensure a good monitoring of the memory usage and run-times.

The impact of the optimisation strategies adopted was tested also for the cubic models, where all models have 20 absorbing cells on each face and the simulation comprises 969

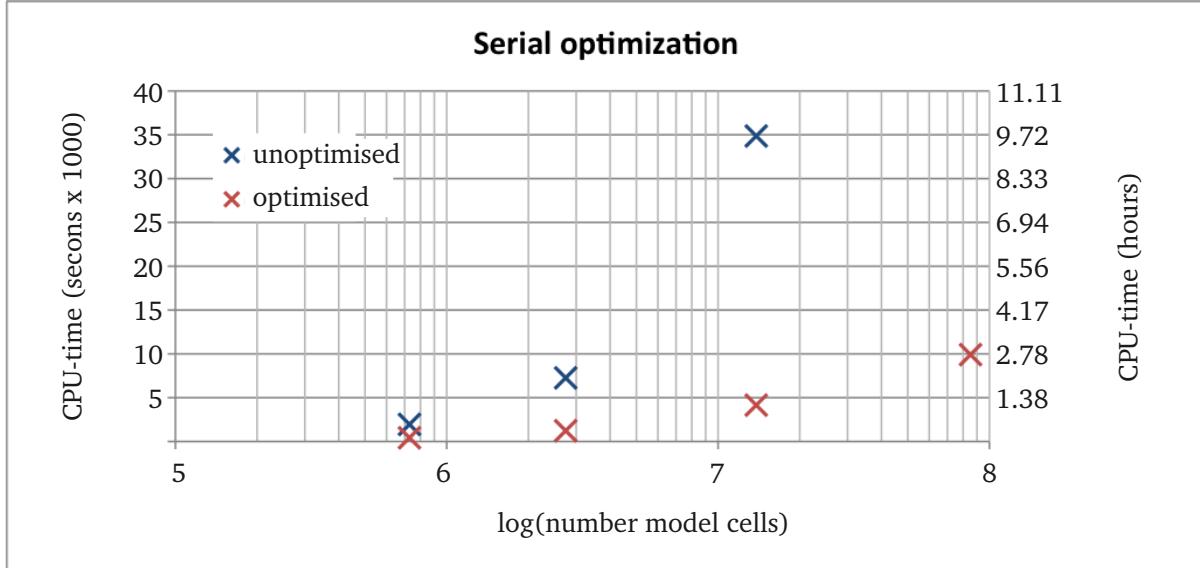


Figure 2.16: Effect of the loop order and the optimisation flags on the run-times. Note that the bigger the model the most accentuated the difference between them. The horizontal axis is logarithmic to avoid excessive concentration of points on the left of the plot.

time-steps. Although accessing the memory in the right order (in the loops) and setting the appropriate flags during compilation is the very first thing to consider when writing efficient codes, I would like to point out the massive impact that it has on the algorithm run-times. In figure 2.16, a comparison between the same code compiled with and without optimisation flags and with the right and wrong ordering of the loops respectively, shows that for big models a speed-up of more than an order of magnitude can be expected.

The improvement in performance of the multi-threading has proved to be very efficient and constant across different model sizes (figure 2.17), with a value of  $\approx 5.5$  using 8 threads. The value is below 8 because the memory bandwidth and the cache size are limited, creating a competition between the threads when they are accessing memory, which limits the ideal value of a 800% decrease in CPU-time.

The response of the speed-up to the number of threads is approximately linear between 1 and 8 threads; using more threads than cores does not seem to accelerate the program any further as shown in figure 2.18. It is interesting to see that multithreading using only one thread does not decrease the performance when compared to a serial run which

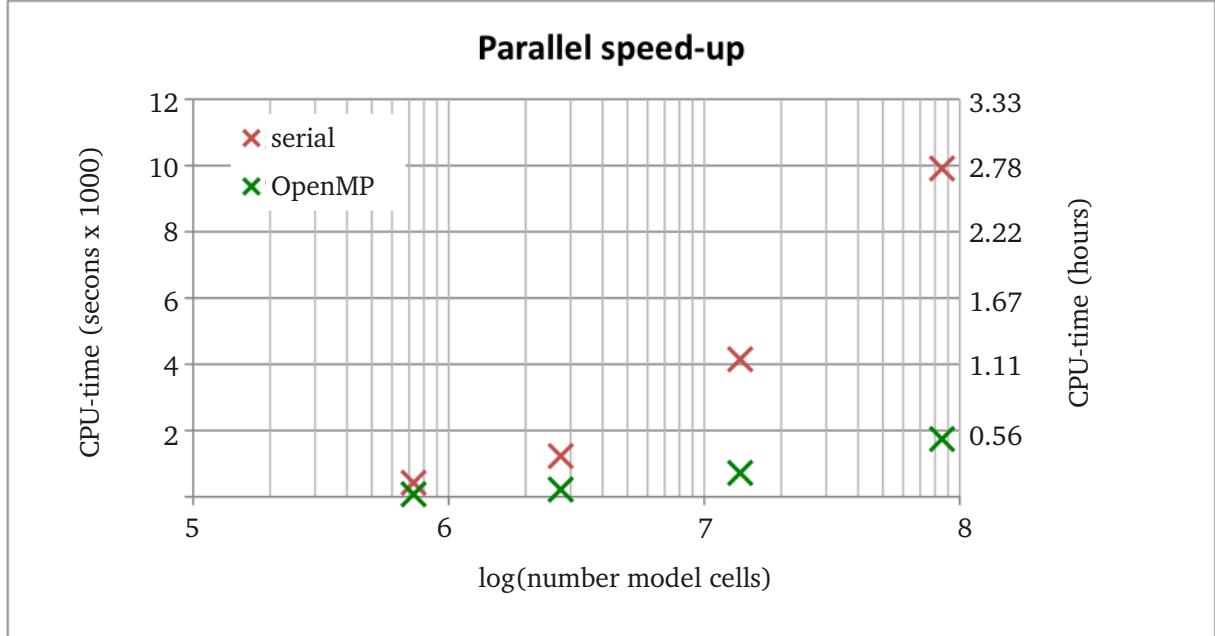


Figure 2.17: Speed-ups for the tested cubic models using 8 threads.

indicates that the time spent setting the threads up (overheads) is not critical. This conclusion is further confirmed by the same speed-up achieved using 8 or 16 threads. Finally, figure 2.18 summarises and corroborates the independence of the size of the model for all the model sizes and for different number of threads.

## 2.5 Conclusions

The method of choice to solve the 3D elastic wave equation is staggered finite differences of 4<sup>th</sup> order in space and 2<sup>nd</sup> order in time. The main reason is its high degree of adaptability, the simplicity in the implementation and its good scalability in shared memory systems. The design of a numerical solution of the wave equation is not a goal per se, and it has been designed as a tool for full-waveform inversion.

The stability conditions for such a relatively low order in space impose more gridpoints per wavelength than higher order schemes used in other depth imaging techniques like reverse time migration. There are two reasons why higher order —and therefore more

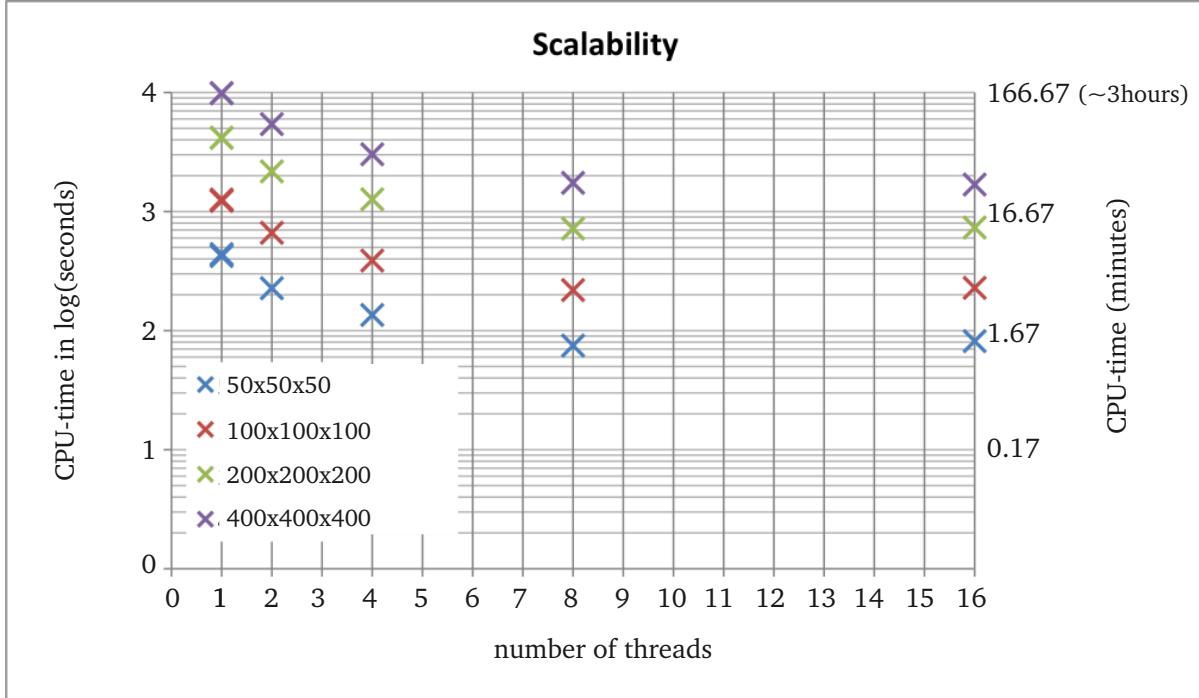


Figure 2.18: The independence of the model size is proved by the invariance of the shape of the trend for each model (for the sake of clarity the vertical axis is logarithmic).

efficient—FD in space have not been used. First and most important, the stencil used provides a good compromise between the theoretical resolution of FWI  $\lambda/2$  and the model mesh size. The second reason is related to the frequency bandwidth of the source. In migration, the source bandwidth typically spans beyond 50 Hz, and therefore, the sampling in space of the models is much finer. Thus, a high order operator does cover a significant number of surrounding points. In full-waveform inversion, on the other hand, the maximum frequencies for 3D real datasets are below 10 Hz, and the spatial sampling is coarser. A small FD operator ensures the correct modelling of waves travelling in the water layer when the depth of the seabed is only of few nodes. It can also capture small features in the models (for example, thin layers) because the size of the operator can be contained in them. If higher order operators were used, they would cover a bigger region of the model and the model properties of that define any existing small feature would take an effective averaged value over the region covered by the operator.

# 3. Inverse theory

In this chapter I present the principles of inversion theory and its application to the seismic problem. Although the elastic wave equation has a non-linear relation between data and model parameters, a review of inverse linear problems is introduced because the inversion of non-linear problems is approached by assuming that the wave equation operator is a linear map in the vicinity of a point in the model space, or in other words, the Born approximation is used to solve inverse problem.

Although, all of the theoretical principles and developments presented in this chapter are not novel, I expose them to support the new formulation of the elastic full-waveform inversion developed in the next chapter.

## 3.1 Introduction

The concept of inverse problem was introduced by Viktor Ambartsumian in the late 1920s. After he graduated from university he started to study the possibility of deriving the laws of quantum mechanics using only atomic physics empirical data. Soon he realised that the magnitude of the task was beyond his capabilities and decided to attack a simpler analogue problem. In 1929 he published a paper in *Zeitschrift für Physik* where he presented a theorem stating that a homogeneous string is uniquely determined by its set of oscillation frequencies [Ambartsumian and Ambartzumian, 1998]. His paper was ignored at the time of publication, but 15 years later it started attracting the attention of the scientific community until inverse theory was finally established as one of the most

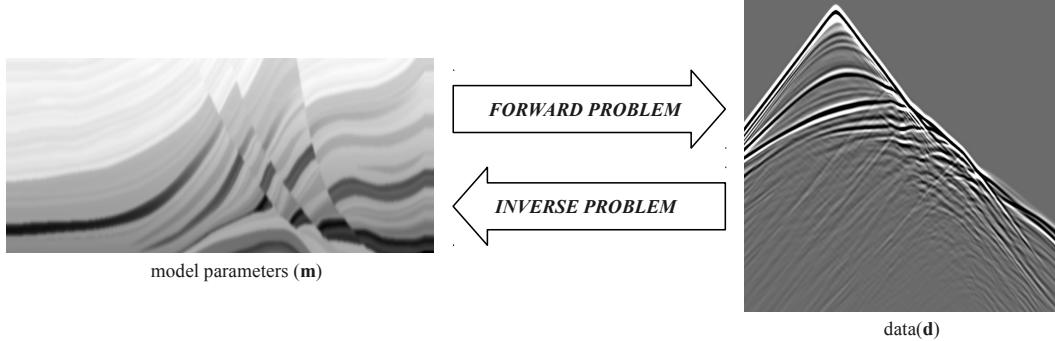


Figure 3.1: Schematic representation of the relation between physical quantities (model parameters) and empirical observations (data). In the depicted image, the model parameters are seismic velocities and density, and the data are pressure seismograms.

important fields in mathematical physics.

The inverse problem is the *dual* concept of the previously presented (in chapter 2) forward or direct problem, where, given a governing law, one tries to predict the result of a particular experiment. Figure 3.1 illustrates the relation between the model and the data spaces.

The direct problem is a straightforward application of the governing laws of physics applied to a known set of variables which describe the properties of the studied system. This set of variables defines the so-called *model*, and the solution of the system (a particular direct problem) uses the laws of physics to obtain a deterministic solution. The model space or parameter space is defined as the set of all possible values taken by the quantities used to parameterise a particular problem. Its dimension is defined by the minimum number of parameters required to describe the problem.

For example,  $\lambda$ ,  $\mu$  and  $\rho$  define a possible model space for the elastic wave equation problem (equations 2.15), where the space is filled with all the possible values that these variables can take within the finite volume that defines the boundary of the problem. One could obviously choose different parameterisations for the same problem. Following with the elastic case,  $V_P$  and  $V_S$  could substitute for  $\lambda$  and  $\mu$  with no effect on the laws of physics describing wave propagation.

The inverse problem, on the other hand, does not generally have a unique solution. Moreover, its solution is non-deterministic and, formally, it should be represented by a probability distribution. The data space is the dual space of the model space, and its dimension is determined by the number of different observable variables measured. Note the non-uniqueness of the dimension of the data space, as opposed to the uniquely defined dimension of the model space (for a particular problem). The dimension of the data space is defined arbitrarily by the number of variables measured during the experiment.

Going back to the elastic wave equation again, imagine a seismic exploration survey where the instrumentation consists of hydrophones streamers (pressure sensors). In this case, the dimension of the data space would be the number of measure points acquired defined as the number of sources times the number of receivers times the number of time samples. Now, if the survey uses ocean-bottom cables (OBCs) equipped with four component (4C) sensors —that is 3 perpendicular geophones plus an hydrophone, then the dimension of the data space would be equal to the dimension of the hydrophone case times four, since we have 4 sensors at each receiver point.

The dimensionality of both model and data spaces may vary from 1 to infinity. When the model parameters or the data are described using functions, then its dimensionality it's infinite. However, it is more common to discretise both model and data spaces, making the dimension of both spaces finite.

In exploration seismology, the model parameter englobes a spatial 3D volume and the data is usually acquired using sensors and sources distributed in 2D horizontal planes at the top of this volume, although sometimes data is acquired by placing sources and/or receivers inside wells.

The forward problem is usually represented by the equation

$$\mathbf{d} = G(\mathbf{m}) \quad (3.1)$$

where  $G$  is an operator representative of the physical laws —for example the wave equation

operator— that relates the data  $\mathbf{d}$  to the model parameters  $\mathbf{m}$ . The inverse problem corresponding to the forward problem represented by equation 3.1 is

$$\hat{\mathbf{m}} = G^{-1}\mathbf{d} \quad (3.2)$$

where the model parameters can now be different from the ones in equation 3.1, in fact they may even have different units. But  $G$  is not always invertible. Indeed, in most problems it is not invertible and waveform inversion is not an exception. Moreover,  $G$  is not even linear, and therefore cannot be written as a linear system of equations (which is very convenient to solve the forward problem using numerical techniques).

Optimisation methods are used to overcome the non-invertibility of the operator  $G$ . The basis of such methods is to define a functional —also referred to as objective or cost function— to be minimised, and to assume that once the global minimum of such objective function is found, the problem is solved. Hereafter, I will refer indistinctively to functional, objective function or solution space. Standard objective functions are of the form

$$F = \|\mathbf{d}_{mod} - \mathbf{d}_{obs}\|_p = \|G(\mathbf{m}) - \mathbf{d}_{obs}\|_p \quad (3.3)$$

where  $\mathbf{d}_{obs}$  is the observed data and  $\mathbf{d}_{mod}$  is the calculated data obtained using a known model, and  $\|\cdot\|_p$  is the  $L^p$  norm (often  $L^2$  or  $L^1$ ). Optimisation methods are based on the assumption that the model  $\mathbf{m}$  that generates the best match for the observed data is the model that best describes the real world.

The functional  $F$  defines the solution space, and its domain of definition is  $\mathbf{m}$ . The solution space may, and normally does, have several minima. Optimisation methods aim to find its global minimum, assuming that local minima are wrong solutions of the inverse problem.

The minimisation of the objective function  $F$  can be achieved using different methods that can be grouped in two main categories:

1. Local methods: a starting model is defined and then the global minimum is searched

in the vicinity of the starting  $F$  value. The most common local optimisation methods are: steepest-descent, conjugate gradients, Newton and quasi-Newton methods.

2. Global methods: the solution space is broadly explored and a minimum is chosen from the set of generated  $F$  values. The most common global optimisation methods are simulated annealing and genetic algorithms [Sambridge and Drikkonen, 1992].

My method of choice for the 3D elastic waveform inverse problem is a local method, specifically a steepest-descent method. As it was shown in the previous chapter 2, the cost of solving the elastic wave equation is of the order of hours, and in its discretised form, the number of variables is of the order of the tens of millions at least. Therefore, applying global search methods to map the solution space would take at the very least, several years to run in standard-sized clusters (several hundred nodes). Given the technical characteristics of present computers 3D, seismic inversion problems are bound to local methods and to design strategies to avoid local minima solutions.

## 3.2 Linear matrix algebra background

Inverse problems are formulated and solved using linear matrix algebra, thus I present in this section a short review of the basic concepts that will be used in this chapter. An important concept is that of vector space, which is a mathematical structure with vectors as elements and two operations: addition of vectors (sum) and multiplication by a scalar. Vectors are used to describe the problem components: data and model parameters. Other important concepts are those of linear maps and matrices. Linear maps are functions that operate over two vector spaces and can be encoded using matrices. Some of the concepts introduced in this section will be accompanied by applications/examples relevant to waveform inversion enclosed in boxes.

### 3.2.1 Vector spaces

Both model and data spaces are vector spaces and by definition their elements can be added together and multiplied by a scalar number. These two operations have to comply with 8 axioms to form a vector space:

1. Associativity of addition:  $(\mathbf{a} + \mathbf{b}) + \mathbf{c} = \mathbf{a} + (\mathbf{b} + \mathbf{c})$
2. Commutativity of addition:  $\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}$
3. Existence of the identity element (addition):  $\mathbf{0} + \mathbf{a} = \mathbf{a}$
4. Existence of inverse elements (addition):  $\mathbf{a} + (-\mathbf{a}) = \mathbf{0}$
5. Existence of the identity element (multiplication):  $1\mathbf{a} = \mathbf{a}$
6. Distributivity (addition):  $\alpha(\mathbf{a} + \mathbf{b}) = \alpha\mathbf{a} + \alpha\mathbf{b}$
7. Distributivity (multiplication):  $(\alpha + \beta)\mathbf{a} = \alpha\mathbf{a} + \beta\mathbf{a}$
8. Scalar over field multiplication:  $\alpha(\beta\mathbf{a}) = (\alpha\beta)\mathbf{a}$

#### Box1. Elastic model and data spaces

Model parameters and data can be expressed as vectors if space and time is discrete—which is usually the case when solving the wave equation using numerical methods.

A model  $\mathbf{m}$  in its vector form is written

$$\mathbf{m} = (m_1^1, m_2^1, m_3^1, \dots, m_n^1, m_1^2, m_2^2, \dots, m_n^2, \dots, m_n^{np})$$

where  $n$  is the number of discrete points of the model and  $np$  is the total number of properties defined per point. Similarly, data can be expressed using vectors

$$\mathbf{d} = (d_1^1, d_2^1, d_3^1, \dots, d_t^1, d_1^2, d_2^2, \dots, d_t^2, \dots, d_t^{nr})$$

where  $t$  is the total number of time-steps and  $nr$  is the total number of receivers. We can define a point in space recording more than one quantity by just giving each quantity a different receiver number. Figure 3.2 show the construction of such vectors from a 3D model and a dataset.

For the elastic wave equation presented in equations 2.15, after discretisation the model vectors look like

$$\mathbf{m} = (\lambda_1, \lambda_2, \dots, \lambda_n, \mu_1, \dots, \mu_n, \rho_1, \dots, \rho_n)$$

which has dimension  $3n$ . The dimension of the data space depends on the number of measured quantities and points of measure. For a four channel OBC (3 components of particle-velocity plus pressure) the data vector would be

$$\mathbf{d} = (v_1^x, v_2^x, \dots, v_m^x, v_1^y, v_2^y, \dots, v_m^y, v_1^z, v_2^z, \dots, v_m^z, p_1, p_2, \dots, p_m)$$

where  $m$  is the number of time-samples times the number of traces and the dimension of this space is  $4m$ .

Two more operations can be defined in vector spaces: inner product and norm. The inner product is defined as an operation over two vector elements and its result is an scalar, represented by  $\langle \cdot, \cdot \rangle$  and it satisfies 3 axioms: conjugate symmetry, linearity and being positive defined. In Euclidean spaces, the inner product is the dot product and is defined as

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i \quad (3.4)$$

The norm is normally represented by  $\|\cdot\|$  and is an unary operation over elements of the vector space which results in a positive scalar. There are different types of norms but we are only interested in the family of  $p$ -norms, defined as

$$\|\mathbf{a}\|_p = \left( \sum_{i=1}^n |a_i|^p \right)^{\frac{1}{p}} \quad (3.5)$$

where  $p \geq 1$ . Norms have the following properties:

1.  $\|\mathbf{a} + \mathbf{b}\| \leq \|\mathbf{a}\| + \|\mathbf{b}\|$

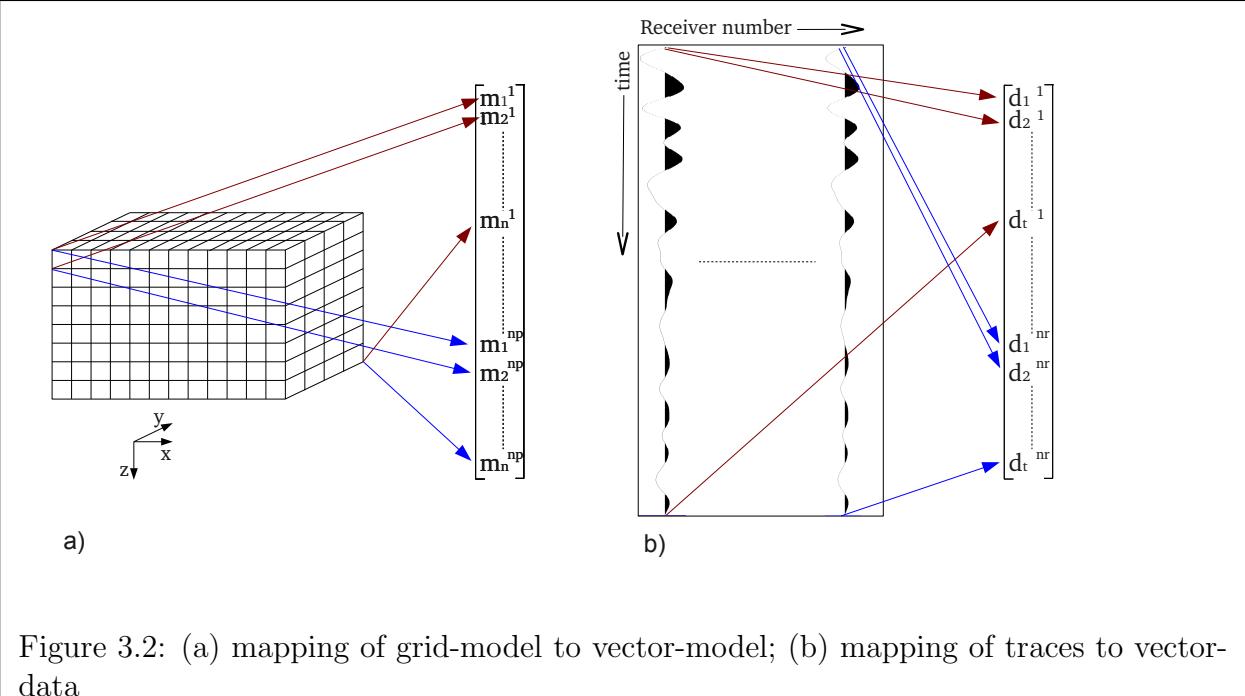


Figure 3.2: (a) mapping of grid-model to vector-model; (b) mapping of traces to vector-data

$$2. \alpha \|\mathbf{a}\| = \|\alpha \mathbf{a}\|$$

$$3. \|\mathbf{a}\| \neq 0 \text{ if } \mathbf{a} \neq 0$$

As mentioned above,  $L^1$  and  $L^2$  norms are the most common in the framework of seismic inversion, although this is not clear yet which one may be better to perform waveform inversion. It probably depends on the domain in which the inversion is done (Laplace, frequency or time) and the problem's specific configuration (such as geometry, bandwidth, quality of the starting model, etc). In the next section it will be shown why an  $L^2$  norm has been chosen and its relation with the choice of domain —time in our case. The  $L^1$  norm takes the form

$$\|\mathbf{a}\|_1 = \sum_{i=1}^n |a_i| \quad (3.6)$$

and is sometimes referred to as Manhattan or Taxicab norm because it represents the length of the vector calculated as the sum of the distances in every dimension, just like the distance a car has to drive in Manhattan (restricted to follow perpendicular streets). For  $p = 2$  we have the  $L^2$  norm which can be interpreted as the euclidean length of a

vector

$$\|\mathbf{a}\| = \sqrt{\sum_{i=1}^n |a_i|^2} \quad (3.7)$$

where the subindex has been dropped for simplicity (from now on, if a norm is different than  $L^2$  it will be noted by a subindex  $\|\cdot\|_p$ ).

### Box 2. Seismic data functional

Equation 3.3 makes use of the norm to evaluate the degree of similarity between model data and observed data. More important, it does it in a quantitative way, by assigning a real number to it, which is important when one wants to address the inverse problem from an automated numerical point of view, without any human interaction. As stated above, the most tested and used norms in waveform inversion are  $L^1$  and  $L^2$ . Following with the previous boxed example, the  $L^1$  and  $L^2$  norms to determine the mismatch between model and observed are

$$L^1 : \|G\mathbf{m} - \mathbf{d}_{obs}\|_1 = \sum_{i=1}^n |d_{mod}^i - d_{obs}^i|$$

$$L^2 : \|G\mathbf{m} - \mathbf{d}_{obs}\|_2 = \sqrt{\sum_{i=1}^n (d_{mod}^i - d_{obs}^i)^2}$$

where the data points are reals, and  $n$  is the total number of data points. As we will see in the next section, the choice of norm has an important effect on the domain in which the problem is formulated. Specifically, using the  $L^1$  norm leads to the backpropagation of squared signals that contain very high frequencies which the FD operators cannot handle.

### 3.2.2 Linear maps and matrices

Linear maps (or linear operators or linear transformations) are mathematical constructs that relate two different vector spaces. Formally, a linear map is a function  $f$  that as-

sociates elements from the vector space  $\Re^n$  onto  $\Re^m$   $f : \Re^n \rightarrow \Re^m$ . If the elements of  $\Re^n$  and  $\Re^m$  are vectors, linear maps  $f$  can be represented by matrices  $\mathbf{A} \in \Re^{m \times n}$ . Take for example a vector of dimension  $n$ , then a matrix  $\mathbf{A}$  associates it to another vector of dimension  $n$  via straightforward multiplication  $\mathbf{Ax} = \mathbf{y}$ . This transformation is linear if

$$\mathbf{A}(\mathbf{x} + \mathbf{y}) = \mathbf{A}(\mathbf{x}) + \mathbf{A}(\mathbf{y}) \quad (3.8)$$

$$\alpha(\mathbf{Ay}) = \mathbf{A}(\alpha\mathbf{y}) \quad (3.9)$$

Matrices can be transposed by exchanging the rows and the columns. Formally, the transpose of a matrix  $\mathbf{A}$  is noted  $\mathbf{A}^T$ , and its elements are related

$$A_{ij} = A_{ji}^T$$

An important characteristic of matrices is its invertibility. The inverse of a matrix  $\mathbf{A}$  is  $\mathbf{A}^{-1}$  and it exists if and only if its determinant  $|\mathbf{A}|$  is non-zero. Some properties of matrix transposes and inverses will be used in the next sections and are worth noting here:

1.  $(\mathbf{A} + \mathbf{B}^T) = \mathbf{A}^T + \mathbf{B}^T$
2.  $(\mathbf{A}^T)^T = \mathbf{A}$
3.  $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$
4.  $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$
5. if  $\mathbf{A}^T = \mathbf{A}$ , then  $\mathbf{A}$  is symmetric

The adjoint definition is as follows: a linear operator (matrix)  $\mathbf{A}$  is adjoint to another linear operator  $\mathbf{B}$  if

$$\langle \mathbf{Ax}, \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{By} \rangle$$

where  $\langle , \rangle$  is the inner product and  $\mathbf{x}, \mathbf{y}$  are vectors. If a matrix has complex entries, then the adjoint matrix or conjugate transpose  $\mathbf{A}^\dagger$  is defined as

$$\mathbf{A}^\dagger = (\mathbf{A}^*)^T \quad (3.10)$$

where  $*$  denotes complex conjugate of the elements of  $\mathbf{A}$ . Conjugate transposes have properties similar to transposes:

1.  $(\mathbf{A} + \mathbf{B})^\dagger = \mathbf{A}^\dagger + \mathbf{B}^\dagger$
2.  $(\mathbf{A}^\dagger)^\dagger = \mathbf{A}$
3.  $(\mathbf{AB})^\dagger = \mathbf{B}^\dagger \mathbf{A}^\dagger$
4.  $(\mathbf{A}^\dagger)^{-1} = (\mathbf{A}^{-1})^\dagger$
5.  $(\alpha\mathbf{A})^\dagger = \alpha^*\mathbf{A}^\dagger$
6. if  $\mathbf{A}^\dagger = \mathbf{A}$ , then  $\mathbf{A}$  is called self-adjoint or Hermitian

### Box 3. Wave equations as linear operators

The forward problem expressed by equation 3.1 is not a linear map, which is equivalent to say that the wave equation does not comply with the conditions abovementioned. Neither the first condition  $G(\mathbf{m}_1 + \mathbf{m}_2) = G(\mathbf{m}_1) + G(\mathbf{m}_2)$  nor the second  $\alpha(G\mathbf{m}) = G(\alpha\mathbf{m})$  are satisfied. Because of this non-linearity,  $G$  cannot be expressed as a matrix operating over vectors of the model space to obtain vectors of the data space.

It is more convenient to pose the wave equation as a linear operator in order to solve it numerically but this linear relation cannot be between model parameters and data. There is, though, a quantity that does comply with this condition, one can notice that there is a linear relation between the source and the wavefield and write the wave equation according to that

$$\mathbf{A}\phi = \mathbf{s} \quad (3.11)$$

where  $\mathbf{A}$  is a wave equation operator,  $\phi$  is the wavefield corresponding to the equation (for example: only pressure for the acoustic case, or stresses and particle-displacements for elastic); and  $\mathbf{s}$  is the source which has non-zero entries at the points of the model where it is inserted. Note that the operator  $\mathbf{A}$  contains the model properties as well as the spatial and temporal differential operators, and the two vector spaces are the source space and the wavefield space.

Take again the acoustic wave equation 2.1 introducing a source  $\mathbf{s}$

$$\nabla^2 \mathbf{p} = \frac{1}{V_P^2} \frac{\partial^2 \mathbf{p}}{\partial t^2} + \mathbf{s}$$

where  $\mathbf{p}$  is the wavefield and  $V_P$  is the P-wave velocity. Discretisation of its temporal part leads to

$$\frac{\mathbf{p}_{l+1} - 2\mathbf{p}_l + \mathbf{p}_{l-1}}{\Delta t^2} = \frac{1}{V_P^2} \nabla \mathbf{p}_l - \mathbf{s}_l$$

where the index  $l$  denotes the current time-step. After space discretisation, the value of the wavefield can be calculated using the values of its two previous time-steps as

$$\mathbf{p}_{l+1} = \Delta t^2 (\mathbf{A} \mathbf{p}_l + \mathbf{s}_l) + 2\mathbf{p}_l - \mathbf{p}_{l-1}$$

It is possible to write this time-stepping scheme in matrix form

$$\mathbf{P} \mathbf{p}_{l+1} + \mathbf{Q} \mathbf{p}_l + \mathbf{R} \mathbf{p}_{l-1} = \mathbf{s}_l$$

where the matrices  $\mathbf{P}$ ,  $\mathbf{Q}$  and  $\mathbf{R}$  have dimensions  $n \times n$ , with  $n$  the total number of model points. All the time-steps can be written in matrix form as well

$$\begin{bmatrix} \mathbf{P} & \mathbf{Q} & \mathbf{R} \\ \mathbf{P} & \mathbf{Q} & \mathbf{R} & 0 \\ \ddots & & & & 0 \\ & \mathbf{P} & \mathbf{Q} & \mathbf{R} \\ 0 & & \mathbf{P} & \mathbf{Q} \\ & & & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{p}_m \\ \mathbf{p}_{m-1} \\ \vdots \\ \mathbf{p}_3 \\ \mathbf{p}_2 \\ \mathbf{p}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{m-1} \\ \mathbf{s}_{m-2} \\ \vdots \\ \mathbf{s}_2 \\ \mathbf{s}_1 \\ \mathbf{s}_0 \end{bmatrix}$$

which has  $m \times n \times n$  columns and the same amount of rows (where  $m$  is the total number of time-steps), hence squared and invertible (as long as its determinant  $\neq 0$ ).

This linear system of equations is solved from bottom to top, so the first step effectively begins with the two previous wavefields that are zero. Therefore, the problem

of solving the linear system of equations is avoided by using an explicit time-marching scheme, which is basically to solve a linear system of equations which associated matrix has a diagonal form and can be solved by straightforward substitution of known values. This is one of the main advantages of the time-domain formulation. In the frequency domain, on the other hand, LU decomposition of the matrix  $\mathbf{A}$  is required to solve the wave equation.

In the next chapter I will derive a similar matrix for the elastic case and use properties of its transpose and inverse to derive navigation directions in the solution space for the elastic inverse problem.

### 3.2.3 Differentiation of vectors and matrices

The next sections include an elevated number of vector differentiations with respect to other vectors, therefore a formal definition of vector and matrix derivatives is provided. Formally, the definition of a vector derivative is

$$\frac{\partial}{\partial \mathbf{a}} = \left( \frac{\partial}{\partial a_1}, \frac{\partial}{\partial a_2}, \dots, \frac{\partial}{\partial a_M} \right)^T \quad (3.12)$$

which is transposed with respect to a normal vector. The derivative of a vector  $\mathbf{b}$  with respect to  $\mathbf{a}$  is defined as

$$\frac{\partial \mathbf{b}}{\partial \mathbf{a}} = \begin{bmatrix} \frac{\partial b_1}{\partial a_1} & \frac{\partial b_1}{\partial a_2} & \dots & \frac{\partial b_1}{\partial a_M} \\ \frac{\partial b_2}{\partial a_1} & \frac{\partial b_2}{\partial a_2} & \dots & \frac{\partial b_2}{\partial a_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial b_N}{\partial a_1} & \frac{\partial b_N}{\partial a_2} & \dots & \frac{\partial b_N}{\partial a_M} \end{bmatrix}$$

where  $M$  and  $N$  are the dimensions of  $\mathbf{a}$  and  $\mathbf{b}$  respectively. Note that the outcome of the operation is a matrix instead of the scalar one usually obtains after multiplication of two vectors. The differentiation of a vector  $\mathbf{b}$  with respect to a scalar  $\alpha$  is defined

$$\frac{\partial \mathbf{b}}{\partial \alpha} = \left( \frac{\partial b_1}{\partial \alpha}, \frac{\partial b_2}{\partial \alpha}, \dots, \frac{\partial b_N}{\partial \alpha} \right)^T$$

and the differentiation of a scalar  $\beta$  with respect to a vector  $\mathbf{a}$  is defined as

$$\frac{\partial \beta}{\partial \mathbf{a}} = \left( \frac{\partial \beta}{\partial a_1}, \frac{\partial \beta}{\partial a_2}, \dots, \frac{\partial \beta}{\partial a_M} \right)$$

The second order derivative of a vector with respect to another vector twice leads to a tensor of rank 3. Following the previous example, the second derivative of  $\mathbf{b}$  with respect to  $\mathbf{a}$  twice results in a tensor with elements

$$\frac{\partial b_i}{\partial a_j \partial a_k} \tag{3.13}$$

with  $i = 1, N, j = 1, M, k = 1, M$ .

### 3.3 Local methods in inverse problems

Large sized inverse problems—especially non-linear ones—rely on optimisation methods to find solutions. A quantity known as the functional or objective function is defined and then minimised to fall below a previously defined degree of accuracy. As previously discussed, this quantity can take the form of the norm between observed and modelled data and has the form of an hypersurface  $F(m_1, m_2, \dots, m_n)$  for a model  $\mathbf{m}$  defined by  $n$  parameters. In some cases (linear problems) this hypersurface is convex and local methods converge to the only minimum and true solution, whereas in other cases it may contain local minima. Since the wave equation does not have a linear relation between model parameters and data, one has to be aware that local methods may lead to incorrect answers. In this section, the developments are made in the context of seismic problems.

#### 3.3.1 Least squares

Least squares is a method used in large inverse problems to find the best fit between two datasets. The first dataset is the observed data and it remains constant throughout the process; the second is generated using physical laws to obtain deterministic data

(forward problem). The objective function was introduced in equation 3.3 as a quantity that accounts for the difference between these two datasets using vector norms. Let us rewrite it here adding a scaling factor of  $1/2$  to simplify future developments

$$F = \frac{1}{2} \|\mathbf{d} - \mathbf{d}_{obs}\| = \frac{1}{2} \|G\mathbf{m} - \mathbf{d}_{obs}\| = \frac{1}{2} (\mathbf{d} - \mathbf{d}_{obs})^\dagger (\mathbf{d} - \mathbf{d}_{obs}) = \frac{1}{2} \Delta \mathbf{d}^\dagger \Delta \mathbf{d} \quad (3.14)$$

where now  $\mathbf{d} = \mathbf{d}_{mod}$  for simplicity, the quantities  $\|\mathbf{d} - \mathbf{d}_{obs}\| = \Delta \mathbf{d}$  are known as residuals and  $^\dagger$  represents the transpose conjugate —data can take complex values, for example under frequency domain formulations. Let  $\mathbf{m}_{min}$  be the element of the model space that minimises  $F$ , then the least-squares method consists in finding this value.

The problem can be approached using local or global methods, and both have benefits and pitfalls. Global methods can better avoid local minima by exploring a more wider range of values of  $\mathbf{m}$ , but they are more expensive computationally speaking. Local methods, on the other hand, are much less expensive to compute, but they cannot ensure that a global minimum is reached because of the small region explored in the solution space. Figure 3.3 shows an schematic representation for a very simple case with only two variables. The left picture shows how a steepest-descent method drives the model towards its closer minimum. If the starting point is  $X_1$ , then the closer minimum is the global minimum and convergence towards the global minimum is achieved, but if the starting model is at  $X_2$ , then the local method converges to a wrong solution (local minimum). In the right picture, a global search maps the entire solution space and then selects the point  $X_m$ , which corresponds to a minimum value of the objective function, and because the whole space is explored, it is assumed to be the global minimum. It is beyond the

scope of this thesis to provide an exhaustive study on optimisation methods and global methods are only presented for completeness.

In global methods the global minimum position is found by taking a representative subgroup of points of  $F$  and choosing the one with minimum value. In this case, the linearity of the problem does not affect the solution. But, as previously discussed, global

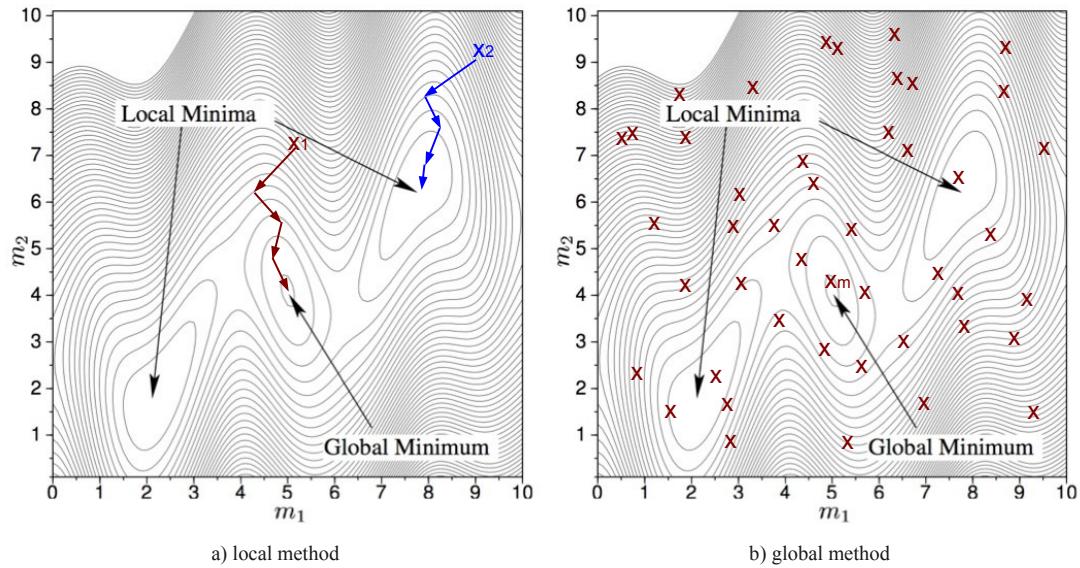


Figure 3.3: Solution space for a simple problem with two model parameters  $m_1$  and  $m_2$ ; (a) example of a local method (steepest-descent): starting at two different points  $X_1$  and  $X_2$  lead to different solutions; (b) example of a global method (Monte Carlo): the global minimum is found calculating many different values of  $F$  and then choosing the model parameters that generate the minimum value of  $F$  at  $X_m$ . The background figure is from [Sirgue, 2003].

methods are too expensive for large-sized model and data spaces, and local methods must be used. The solution space  $F$  has different shapes depending on the nature of the problem, and can be separated in two categories according to that: linear problems have convex solution spaces but non-linear problems may have multiple minima. In the linear case, any local method would converge to the right solution because the gradient of the solution space at any point is directing us to the model that minimises the residuals in an univocal manner. This implies that the starting point of the local optimisation method is insensitive to the final solution because all starting points converge to the same unique minimum. For non-linear problems, on the other hand, the choice of starting model is a crucial step since the solution is dependent on the starting point, as illustrated in figure 3.3 (the solution space is not convex in this case and hence the problem is not linear).

Due to the presence of local minima, the starting point must be relatively close to the global minimum, and that allows the expansion of the objective function as a Taylor

series around the starting model

$$F(m_{0i} + \Delta m_i) = F(m_{0i}) + \Delta m_i \left. \frac{\partial F(m_i)}{\partial m_i} \right|_{m_{0i}} + \Delta m_i \Delta m_j \frac{1}{2} \left. \frac{\partial^2 F(m_i)}{\partial m_i \partial m_j} \right|_{m_{0i}} + O(m)^3 \quad (3.15)$$

where  $m_i$  is the  $i$ th element of the model vector of dimension  $n$ , and  $|_{m_{0i}}$  denotes evaluation of the function at the point  $m_{0i}$ . The expansion is truncated leaving a residual value depending on third order refinements  $O(m_i)^3$ .

Since the gradient of a function in a minimum position always vanishes, we can differentiate the objective function with respect to the model parameters and set it equal to zero in order to find the closest minimum of  $F$  around  $\mathbf{m}_0$ . Note that minimums are not the only stationary points where the derivative is zero, they can be maximums or saddle points as well. Consider the derivative of the functional with respect to the model parameters, the first component of the right-hand side of equation vanishes, and the second and the third are differentiated taking into account that  $\Delta\mathbf{m} = \mathbf{m} - \mathbf{m}_0$  leading to

$$\frac{\partial F}{\partial m_i} = \frac{\partial F(m_{0i})}{\partial m_i} + \Delta m_i^\dagger \frac{\partial^2 F(m_{0i})}{\partial m_i^2} = 0 \quad (3.16)$$

where for the sake of simplicity  $|_{m_{0i}}$  has been replaced by  $F(m_{0i})$ . Now the minimum is found by isolating  $\Delta\mathbf{m}$  in equation 3.16

$$\Delta\mathbf{m} = - \left( \frac{\partial^2 F(\mathbf{m}_0)}{\partial \mathbf{m}^2} \right)^{-1} \frac{\partial F(\mathbf{m}_0)}{\partial \mathbf{m}} \quad (3.17)$$

For linear problems, equation 3.17 provides a solution in one step. If  $F$  is substituted back into its original definition  $F = 1/2(\mathbf{G}\mathbf{m} - d_{obs})^\dagger(\mathbf{G}\mathbf{m} - d_{obs})$  —where now  $\mathbf{G}$  has become a linear operator, for example using the Born approximation of the wave equation—and reinjected in equation 3.15, the error term  $\mathbf{O}(\mathbf{m})^3$  vanishes and the solution given by 3.17 is exact. The objective function  $F$  is quadratic with respect to  $\mathbf{m}$  and thus third order derivatives are zero [Virieux and Operto, 2009]. Although the extremum found may not be a minimum, the Hessian will define whether the point is a maximum, minimum or a

saddle point. If the Hessian has a non-zero determinant, the critical point is a minimum when it is positive definite, a maximum when it is negative definite and a saddle point when it has both positive and negative eigenvalues<sup>1</sup>

However, this is not true when  $G$  is not a linear map and this approximation is a linearisation of  $\mathbf{d}$  with respect to  $\mathbf{m}$  around the studied point. Therefore,  $\Delta\mathbf{m}$  only updates the starting model to a new position moving in the opposite direction of the gradient (first order derivative of  $F$ ) and scaled by the Hessian(second order derivative of  $F$ ). In order to obtain the solution point, one must repeat the process taking the updated model as a starting for the next iteration.

### 3.3.2 Iterative methods

Note that the solutions provided by equation 3.17 will only be valid if the iterative process starts in a point  $\mathbf{m}_0$  that is close enough to the global minimum and if  $F$  around  $\mathbf{m}_0$  is sufficiently monotonic. If these two conditions are fulfilled, then a solution can be found by repeatedly applying equation 3.17 using the model obtained in the previous step as the starting model for the current one. Such iterative scheme will only lead to the true solution in the limit of  $n_{it} \rightarrow \infty$  where  $n_{it}$  is the total number of iterations.

In order to solve the problem in a finite amount of time a stopping criterion is needed. For example one can define a certain threshold of the value  $F$  and stop the iterations when  $F$  falls below this value, or by fixing beforehand the total number of iterations to be performed. In any case, the value of  $n_{it}$  that provides satisfactory results —from a seismic imaging point of view— depends on the approximations and assumptions made when solving the equation 3.17. I will present three different methods: Newton, Gauss-Newton and steepest-descent.

Let  $\Delta\mathbf{d}$  be expanded as  $(\mathbf{d} - \mathbf{d}_{\text{obs}})$  and used to derive expressions for the first and

---

<sup>1</sup>a matrix  $\mathbf{A}$  is positive (negative) definite if all its eigenvalues are positive (negative).

second order derivative terms of equation 3.15

$$\begin{aligned}\nabla F = \frac{\partial F(\mathbf{m})}{\partial \mathbf{m}} &= \frac{1}{2} \Re \left\{ \frac{\partial}{\partial \mathbf{m}} \left[ (\mathbf{d} - \mathbf{d}_{obs})^\dagger (\mathbf{d} - \mathbf{d}_{obs}) \right] \right\} \\ &= \frac{1}{2} \Re \left[ \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right)^\dagger (\mathbf{d} - \mathbf{d}_{obs}) + (\mathbf{d} - \mathbf{d}_{obs})^\dagger \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right) \right] \\ &= \Re \left[ \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right)^\dagger (\mathbf{d} - \mathbf{d}_{obs}) \right]\end{aligned}\quad (3.18)$$

where the real part denoted by  $\Re$  is only retained when data have complex values, for example when the wave equation is formulated in the frequency domain. For the second order derivative calculation it is sufficient to take the derivative of the final expression provided in 3.18

$$\begin{aligned}\mathbf{H} = \frac{\partial^2 F(\mathbf{m})}{\partial \mathbf{m}^2} &= \frac{\partial}{\partial \mathbf{m}} \left\{ \Re \left[ \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right)^\dagger (\mathbf{d} - \mathbf{d}_{obs}) \right] \right\} \\ &= \Re \left[ \left( \frac{\partial^2 \mathbf{d}}{\partial \mathbf{m}^2} \right)^\dagger (\mathbf{d} - \mathbf{d}_{obs}) + \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right)^\dagger \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right) \right]\end{aligned}\quad (3.19)$$

The only unknowns in equations 3.18 and 3.19 are the derivatives of the data with respect to the model parameters since the quantity  $(\mathbf{d} - \mathbf{d}_{obs})$  is already known.  $\nabla F$  and  $\mathbf{H}$  are referred to as the gradient —or Jacobian or sensitivity matrix or Fréchet derivative matrix— and the Hessian respectively.

The calculation of the gradient and the Hessian would not be a problem if an analytical form of the solution of the wave equation for any given problem was available, but unfortunately this is not the case and we rely on numerical techniques to solve it. In section 3.3.3 it will be shown that straight calculation of expressions 3.18 and 3.19 can be avoided using the adjoint-state method.

### Newton method

The Newton method does not make any further assumptions besides the one already stated when the objective function was expanded using a Taylor series. Geometrically, this approximation states that in the vicinity of the point  $\mathbf{m}_0$ , the functional is a polynomial

of second order (quadratic form). The Newton method navigates the solution space by updating the model to the closest stationary point. It uses all the terms in equations 3.18 and 3.19 to find  $\Delta\mathbf{m}$ . The Newton method can be summarized by the model updates

$$\Delta\mathbf{m} = -\Re \left[ \left( \frac{\partial\mathbf{d}}{\partial\mathbf{m}} \right)^\dagger (\mathbf{d} - \mathbf{d}_{obs}) \right] \left\{ \Re \left[ \left( \frac{\partial^2\mathbf{d}}{\partial\mathbf{m}^2} \right)^\dagger (\mathbf{d} - \mathbf{d}_{obs}) + \left( \frac{\partial\mathbf{d}}{\partial\mathbf{m}} \right)^\dagger \left( \frac{\partial\mathbf{d}}{\partial\mathbf{m}} \right) \right] \right\}^{-1} \quad (3.20)$$

The Newton method tends to converge much faster than Gauss-Newton or steepest-descent methods, but is not always applicable. Such better convergence rate is achieved if two conditions are satisfied: the Hessian is invertible and  $F$  has some Lipschitz continuity. The invertibility of the Hessian is unclear since we do not have the means to calculate the Hessian itself in large problems that need to be discretised. The Lipschitz condition imposes the continuity of  $F$  and restricts its variation in the vicinity of  $\mathbf{m}_0$  by imposing a maximum slope value (the gradient  $\partial\mathbf{d}/\partial\mathbf{m}$  cannot be bigger than a certain value in the vicinity of  $\mathbf{m}_0$ ). The second condition is rather useless because the shape of the solution space is not known and therefore its veracity cannot be proved. The first condition can only be confirmed if the Hessian can be explicitly calculated and this is normally not the case. If the two conditions are satisfied in a region  $\mathcal{M}$ , taking any point  $\mathbf{m} \in \mathcal{M}$  as a starting point would lead to convergence to the closest minimum (global or local) in only one step. Nevertheless, the explicit calculation of the Hessian can be avoided when using Quasi-Newton methods, where a Taylor expansion of the gradient lead to an approximation of the Hessian at every iteration. DFP (named after Davidon, Fletcher and Powell) and BFGS (named after Broyden, Fletcher, Goldfarb and Shannon) are examples of such methods, but this is a vast subject on itself and an extensive review will not be given here (see for example [Tarantola, 2005] for a more complete description).

### Gauss-Newton method

The Hessian can be approximated by

$$\mathbf{H} = \left( \frac{\partial^2\mathbf{d}}{\partial\mathbf{m}^2} \right)^\dagger (\mathbf{d} - \mathbf{d}_{obs}) + \left( \frac{\partial\mathbf{d}}{\partial\mathbf{m}} \right)^\dagger \left( \frac{\partial\mathbf{d}}{\partial\mathbf{m}} \right) \approx \left( \frac{\partial\mathbf{d}}{\partial\mathbf{m}} \right)^\dagger \left( \frac{\partial\mathbf{d}}{\partial\mathbf{m}} \right) \quad (3.21)$$

an approximation that assumes a linear relation between data and model parameters and hence removes the second order derivative. Let  $\mathbf{H}_a$  be the approximate Hessian, then the update of the model in this case is given by

$$\Delta \mathbf{m} = \nabla F \mathbf{H}_a^{-1} = \Re \left[ \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right)^\dagger (\mathbf{d} - \mathbf{d}_{obs}) \right] \left\{ \Re \left[ \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right)^\dagger \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right) \right] \right\}^{-1} \quad (3.22)$$

Convergence is to be expected if one of these two conditions is satisfied:

1. The values of  $F$  have to be sufficiently small around the minimum.
2. Non-linearities in  $F$  are small.

Note that any of these conditions ensure

$$\left| \frac{\partial^2 \mathbf{d}}{\partial \mathbf{m}^2} \right| \ll \left| \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right)^\dagger \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right) \right| \quad (3.23)$$

and therefore the second order derivative term can be dropped from the Hessian.

### Steepest-descent method

As it names indicates, steepest-descent—or gradient—methods search the solution space in the opposite direction to the gradient (steepest direction). Because the magnitude of the gradient is not physically sensible, a scaling factor  $\alpha$  is added to equation 3.17 after dropping the Hessian, and we end up with

$$\Delta \mathbf{m} = \alpha \nabla F = -\alpha \Re \left[ \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right)^\dagger (\mathbf{d} - \mathbf{d}_{obs}) \right] \quad (3.24)$$

The choice of alpha is quite critical because if all the assumptions made are correct, we are moving along the gradient —multidimensional straight line—in a solution space which has a convex shape and therefore if we move too far we have passed the minimum, and we end up in a place where the functional may even have a higher value. Conversely, if we move too little, then the convergence rate would be too slow.

Figure 3.4 illustrates the procedure of this method for a simple case with only two model parameters. Note that the shape of the solution space is not known, the only points

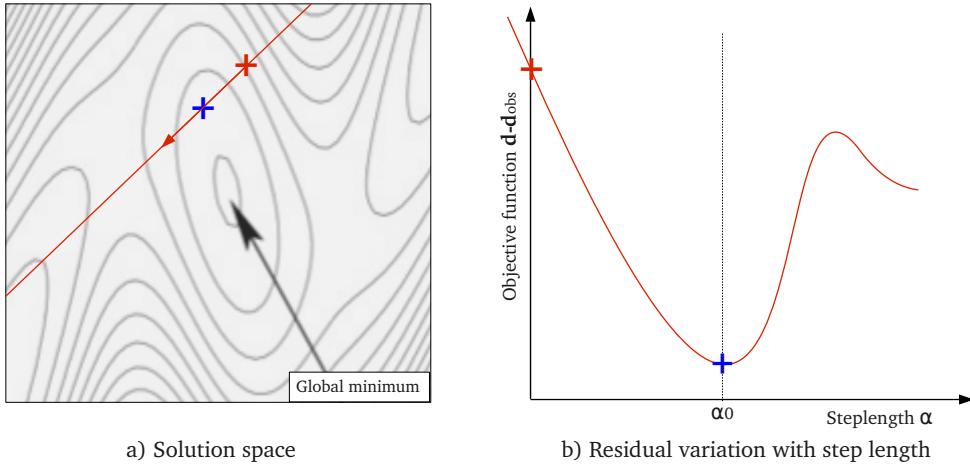


Figure 3.4: + is the point where the gradient is computed, the red line is the gradient direction in (a) (with the arrow pointing downhill) which have residual values  $\mathbf{d} - \mathbf{d}_{obs}$  shown in (b); + is the position of the minimum of  $F$  along the direction of the gradient, and it is achieved when the gradient is scaled by a factor  $\alpha_0$

known are the ones that are explicitly calculated, so in the figure we do not know the real solution space (contour map), and we make the assumption that its shape is quadratic (when using the  $L^2$  norm) and that its real minimum coincides roughly with that of the parabolic approximation.

Finding the best step length leads to another optimisation problem where, again, some approximations need to be made. The obvious solution is to use brute force to populate the values of  $F$  along the gradient direction in order to explicitly know its minimum, but since every point of the curve requires solving the forward problem, it becomes computationally unaffordable (at least if the solution is to be found on a reasonable time scale). A similar problem occurs if a global search (Monte Carlo based methods for example) is conducted. The only option left is to locally optimise the value of the step length.

The simplest solution would be to fit a line between the starting point and a new point found by updating the model in the direction of the gradient by a small amount, but this method has a strong dependence on the curvature of the functional between these two

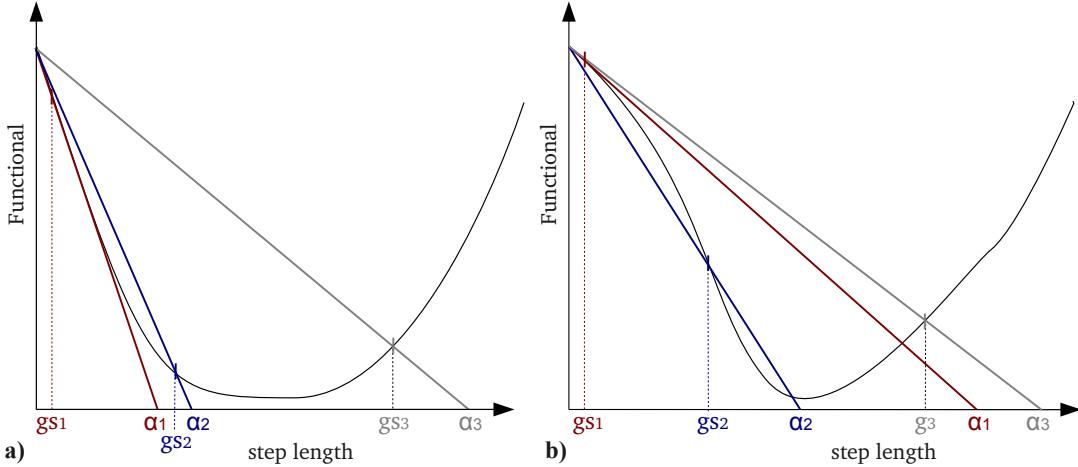


Figure 3.5: For two different functional curves along gradient directions (a) and (b) three different step length guesses are computed, leading to dramatically different results.  $gs_1$ ,  $gs_2$  and  $gs_3$  are the three different guessed step lengths leading to optimal step lengths of  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$ .

points as shown in figure 3.5.

Similarly, if a parabolical fit is used (requiring an extra functional point calculation), due to the unknown real shape of the curve the estimated position of the global minimum may be erroneous. For example, as figure 3.6 shows, the same points that lead to significant reductions of the objective function (figure 3.6-(a)) fail completely for a different functional shape (figure 3.6-(b)). Figures 3.5 show the poor control over the quality of the optimal step length. For example, a small drop in the functional for a particular guess means that we have to take a big step in order to minimise  $F$ ? or it means that the value of  $\alpha$  for the guessed step length is poorly chosen and it should be changed by another more significant value?

To prevent such random out of control updates it is better to minimise the  $L^2$  norm of the residuals under the assumption that the residual change caused by the guessed step length  $gs$  is linear with the change in model parameters. Taking the residuals  $\Delta \mathbf{d}^0$  (corresponding to the difference  $(\mathbf{d}^0 - \mathbf{d}_{obs})$ ) where  $\mathbf{d}^0$  is the data calculated using  $\mathbf{m}_0$ ) and the residuals  $\Delta \mathbf{d}^1$  (similarly defined at  $\mathbf{m}_1$ , we want to find  $\alpha$  that minimises the

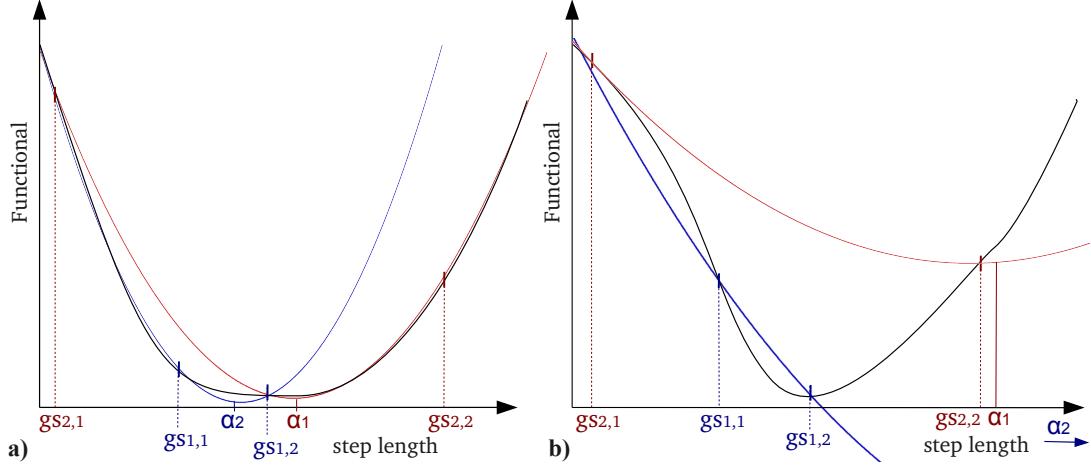


Figure 3.6: Two different sets of points chosen to fit a parabola are shown in red and blue for two different objective function curves (a) and (b).  $gS_{i,1}$  and  $gS_{i,2}$  are the two values of the functional used to fit parabolas with minima  $\alpha_i$  for  $i \in 1, 2$ . Every point calculation costs an extra forward modelling, so this method is, computationally speaking, more expensive than the linear fit, although it may speed up convergence.

expression

$$\Delta\mathbf{d} = \Delta\mathbf{d}^0 - \alpha(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1) \quad (3.25)$$

If we rewrite the objective function in terms of the new linearisation represented by 3.25

$$\begin{aligned} F &= \frac{1}{2}\Delta\mathbf{d}^\dagger\Delta\mathbf{d} = \frac{1}{2}[\Delta\mathbf{d}^0 - \alpha(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1)]^\dagger[\Delta\mathbf{d}^0 - \alpha(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1)] \\ &= \frac{1}{2}(\Delta\mathbf{d}^0)^\dagger\Delta\mathbf{d}^0 - \alpha(\Delta\mathbf{d}^0)^\dagger(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1) + \frac{1}{2}\alpha^2(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1)^\dagger(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1) \end{aligned} \quad (3.26)$$

and take its derivative with respect to  $\alpha$  equal to zero (condition of minimum provided we are closer to it than to another extremal)

$$\frac{\partial F}{\partial \alpha} = -(\Delta\mathbf{d}^0)^\dagger(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1) + \alpha(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1)^\dagger(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1) = 0 \quad (3.27)$$

and finally

$$\alpha = \frac{(\Delta\mathbf{d}^0)^\dagger(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1)}{(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1)^\dagger(\Delta\mathbf{d}^0 - \Delta\mathbf{d}^1)} \quad (3.28)$$

Note that  $\alpha$  only depends on the already computed residuals at the two different points

and therefore its optimal value is directly computed following equation 3.28, using only one extra forward modelling to determine  $\Delta\mathbf{d}^1$ .

It is possible to mitigate the effect of the curvature of the objective function along the gradient direction by using some form of the Hessian. The Hessian takes into account the curvature of the solution space in the vicinity of the point at which is calculated. But the steepest-descent method's efficiency relies on simplifying the optimisation by not having to calculate the Hessian —and especially its inverse, and introducing it again may seem counterproductive. To avoid the calculation of the Hessian, a further approximation is made by only considering the main diagonal of  $\mathbf{H}_a$ . Let  $\mathbf{H}_D$  be a matrix that only contains the diagonal elements of  $\mathbf{H}_a$

$$\mathbf{H}_D = \begin{bmatrix} H_D^{1,1} & & & 0 \\ & H_D^{2,2} & & \\ & & \ddots & \\ 0 & & & H_D^{n,n} \end{bmatrix}$$

then its inverse is simply

$$\mathbf{H}_D^{-1} = \begin{bmatrix} 1/H_D^{1,1} & & & 0 \\ & 1/H_D^{2,2} & & \\ & & \ddots & \\ 0 & & & 1/H_D^{n,n} \end{bmatrix}$$

which is virtually free to compute. The final expression for the parameter update using  $\mathbf{H}_D$  is then

$$\Delta\mathbf{m} = -\alpha \nabla F[\mathbf{H}_D]^{-1} = -\alpha \Re \left[ \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right)^\dagger (\mathbf{d} - \mathbf{d}_{obs}) \right] \left\{ \Re \ diag \left[ \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right)^\dagger \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \right) \right] \right\}^{-1} \quad (3.29)$$

An schematic representation of the effect caused by the approximate diagonal Hessian in a simple two parameter case is depicted in figure 3.7. The direction of the gradient is changed to take into account the curvature of the solution space, so it is expected that the new direction (in blue) is closer to the direction in which the actual minimum is found.

The approximate diagonal Hessian contains the autocorrelation of the wavefield deriva-

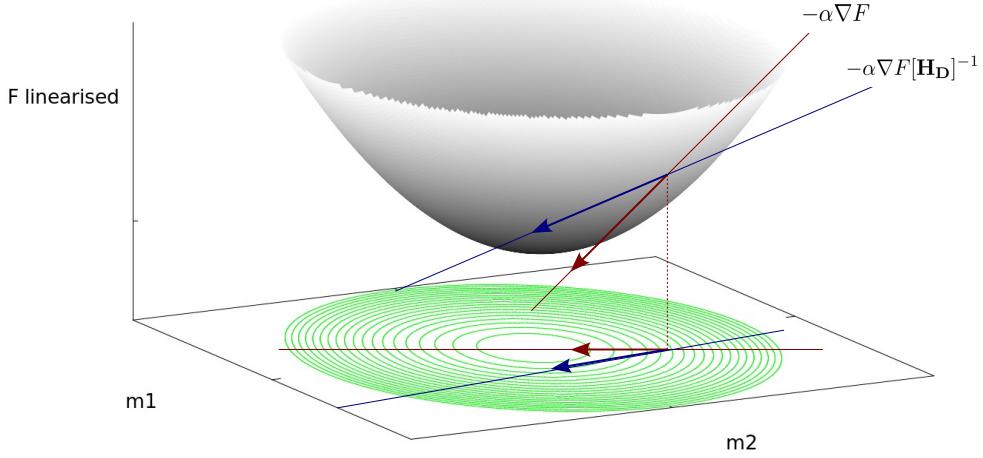


Figure 3.7: The grey paraboloid is the approximation of the solution space that results from assuming that the objective function is quadratic with respect to the model parameters. The red line indicates the direction of the gradient and the blue line the direction of the gradient scaled with the approximate diagonal Hessian. Note that the real shape of the objective function is absent in the plot.

tives which has the effect of removing the amplitude decay of the wavefield and therefore it balances the contributions of the gradient at every point in the model with no extra computational cost associated.

### 3.3.3 Adjoint-state method

In the previous section, a set of formulas to calculate the update of model parameters which minimises the objective function was presented. In all of them  $\Delta\mathbf{m}$  depends on the first (or second in the Newton method) derivatives of the data with respect to the model parameters, although it is not possible to obtain an analytical expression for it due to the nature of the wave equation. Numerical methods cannot be used to directly solve these partial derivatives because of the elevated number of operations that such procedure would imply. Let  $M$  be the total number of model cells and  $N$  the total number of data

points<sup>2</sup>. The Fréchet derivative matrix takes the form

$$\frac{\partial \mathbf{d}}{\partial \mathbf{m}} = \begin{bmatrix} \frac{\partial d_1}{\partial m_1} & \frac{\partial d_1}{\partial m_2} & \dots & \frac{\partial d_1}{\partial m_M} \\ \frac{\partial d_2}{\partial m_1} & \frac{\partial d_2}{\partial m_2} & \dots & \frac{\partial d_2}{\partial m_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial d_N}{\partial m_1} & \frac{\partial d_N}{\partial m_2} & \dots & \frac{\partial d_N}{\partial m_M} \end{bmatrix}$$

The Fréchet derivative matrix has  $N \times M$  elements and considering that in 3D experiments models typically have  $10^6 - 10^8$  cells and  $10^5 - 10^7$  data points the number of elements of the matrix is inconveniently high. Most computers store real values in memory blocks of 4 bytes, then for a model of  $10^7$  cells and for  $10^6$  data points, a computer would need to store  $10^{13}$  reals which translates into  $2.3TB$  (terabytes). Considering that current computers typically have random-access memory (RAM) storage of less than 100 GB (gigabyte), it is not possible to fit the sensitivity matrix in RAM and one would need to store the matrix on disk which would slow down the calculation considerably. Not only memory is an issue, the calculation of the elements of the matrix is massively time consuming as well. If the derivatives are computed using for example first-order finite differences, that would require at the very least one extra forward modelling calculation for each model parameter. Obviously it is impossible to solve the wave equation several millions of times within a reasonable amount of time.

Nonetheless, the gradient of the misfit function can be calculated without explicit calculation of all the partial derivative terms  $\frac{\partial d_i}{\partial m_j}$ . Previously, a linear relation between the wavefields —of which the data is a subset— and the source was introduced as  $\mathbf{A}\phi = \mathbf{s}$  in equation 3.11. Differentiation with respect to  $\mathbf{m}$  leads to

$$\frac{\partial}{\partial \mathbf{m}}(\mathbf{A}\phi) = \frac{\partial \mathbf{A}}{\partial \mathbf{m}}\phi + \mathbf{A}\frac{\partial \phi}{\partial \mathbf{m}} = \frac{\partial \mathbf{s}}{\partial \mathbf{m}} = 0 \quad (3.30)$$

because the source is independent of the model parameters and hence its derivative with

---

<sup>2</sup>The notation for the size of the data and model vectors has been changed from the previous definition given in the **Elastic model and data spaces** box for convenience of future developments.

respect to  $\mathbf{m}$  is zero. Rearranging terms in equation 3.30

$$\frac{\partial \phi}{\partial \mathbf{m}} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \mathbf{m}} \phi \quad (3.31)$$

The second term of equation 3.31 provides an alternative expression for the Fréchet derivative matrix with the new unknowns  $\mathbf{A}^{-1}$  and  $\frac{\partial \mathbf{A}}{\partial \mathbf{m}}$ . Although the inverse of the forward modelling matrix  $\mathbf{A}$  may not be easily obtained, the other term is straightforward to calculate because the matrix  $\mathbf{A}$  depends linearly on the model parameters (after discretising the wave equation using FD). Reinjecting 3.31 into the expression for the gradient 3.24 and considering  $\alpha = 1$  (without loss of generality) for simplicity

$$\nabla F = -\Re \left\{ \left[ \mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \mathbf{m}} \phi \right]^\dagger \mathbf{D}(\phi - \phi_{\text{obs}}) \right\} \quad (3.32)$$

where the data vectors have been substituted by wavefield vectors and a new matrix  $\mathbf{D}$  has been introduced.  $\mathbf{D}$  is a diagonal matrix which has non-zero entries in the cell positions containing data. Besides, the non-zero values can be equal to one, or alternatively they can take different values at different data points if some weighting is desired. For example, it is possible to boost the contribution of the data cells far from the source by assigning higher values to their corresponding positions in  $\mathbf{D}$  to compensate for the amplitude decay with distance if punctual sources are used; or to eliminate noisy traces zeroing their values. Furthermore,  $\mathbf{D}$  can have other uses: introduce uncertainties, make the data-space misfit function dimensionless, select a portion of the data, etc.

It is still inconvenient to calculate the inverse  $\mathbf{A}^{-1}$ . Indeed it can be avoided by introducing a new forward problem

$$\mathbf{A}^\dagger \phi_b = \mathbf{s}_b \quad (3.33)$$

where  $\mathbf{A}^\dagger$  is the adjoint matrix equation and  $\phi_b$  is a wavefield propagated backwards in time due to the structure of the matrix  $\mathbf{A}^\dagger$ . The backpropagation nature of the wavefield

$\phi_b$  is explained as follows: the wave equation operator matrix in the time domain is of the form

$$\begin{bmatrix} \mathbf{P} & \mathbf{Q} & \mathbf{R} \\ \mathbf{P} & \mathbf{Q} & \mathbf{R} & 0 \\ \ddots & & & \\ & \mathbf{P} & \mathbf{Q} & \mathbf{R} \\ 0 & & \mathbf{P} & \mathbf{Q} \\ & & \mathbf{P} & \end{bmatrix} \begin{bmatrix} \mathbf{p}_m \\ \mathbf{p}_{m-1} \\ \vdots \\ \mathbf{p}_3 \\ \mathbf{p}_2 \\ \mathbf{p}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{m-1} \\ \mathbf{s}_{m-2} \\ \vdots \\ \mathbf{s}_2 \\ \mathbf{s}_1 \\ \mathbf{s}_0 \end{bmatrix}$$

where the matrix  $\mathbf{R}$  is 0 for first-order time-derivative formulations. Note that in time domain the conjugate transpose operation reduces to simple transposition because the entries of  $\mathbf{A}$  are real. The structure of the matrix forces the solution of the system to start at the first time-step  $\mathbf{p}_1$  and then proceeds to solve later time-steps using the previously computed values. However, in equation 3.33, the operator  $\mathbf{A}$  is transposed and the upper diagonals containing  $\mathbf{Q}$  and  $\mathbf{R}$  appear now as lower diagonals

$$\begin{bmatrix} \mathbf{P}^T & & & \\ \mathbf{Q}^T & \mathbf{P}^T & & 0 \\ \mathbf{R}^T & \mathbf{Q}^T & \mathbf{P}^T & \\ & \ddots & & \\ 0 & \mathbf{R}^T & \mathbf{Q}^T & \mathbf{P}^T \\ & & \mathbf{R}^T & \mathbf{Q}^T & \mathbf{P}^T \end{bmatrix} \begin{bmatrix} \mathbf{p}_m \\ \mathbf{p}_{m-1} \\ \vdots \\ \mathbf{p}_3 \\ \mathbf{p}_2 \\ \mathbf{p}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{m-1} \\ \mathbf{s}_{m-2} \\ \vdots \\ \mathbf{s}_2 \\ \mathbf{s}_1 \\ \mathbf{s}_0 \end{bmatrix}$$

where now the system is solved starting at the last time-step and proceeds backwards in time until the initial time-step is reached. If the new source is defined

$$\mathbf{s}_b = \mathbf{D}^T(\phi - \phi_{obs}) \quad (3.34)$$

then, the residuals act now as a new source which generate a wavefield  $\phi_b$ . Note  $\mathbf{D}^\dagger = \mathbf{D}^T$  because the weighting matrix only contain real numbers. The new solution of the modified wave equation represented by  $\mathbf{A}^\dagger$  is then

$$\phi_b = (\mathbf{A}^\dagger)^{-1}\mathbf{D}^T(\phi - \phi_{obs}) \quad (3.35)$$

Exploiting the third and the fourth properties of conjugate transpose matrices, equation 3.32 can be rewritten as

$$\nabla F = -\Re \left\{ \phi^\dagger \left( \frac{\partial \mathbf{A}}{\partial \mathbf{m}} \right)^\dagger (\mathbf{A}^\dagger)^{-1} \mathbf{D} (\phi - \phi_{obs}) \right\} \quad (3.36)$$

and by identification of terms with equation 3.35

$$\nabla F = \Re \left\{ \phi^\dagger \left( \frac{\partial \mathbf{A}}{\partial \mathbf{m}} \right)^\dagger \phi_b \right\} = \Re \left\{ \phi_b^\dagger \left( \frac{\partial \mathbf{A}}{\partial \mathbf{m}} \right) \phi \right\} \quad (3.37)$$

where the last expression is only true when the middle term  $\frac{\partial \mathbf{A}}{\partial \mathbf{m}}$  is self-adjoint —symmetric in the time-domain because the entries are real numbers. Then, to compute the gradient  $\nabla F$  —and the approximate and approximate diagonal Hessians, since they only depend on the gradient as  $\mathbf{H}_a = \nabla F^\dagger \nabla F$  and  $\mathbf{H}_D = \text{diag}(\nabla F^\dagger \nabla F)$ — we only need to compute the backpropagated residuals  $\phi_b$ . Note that the backpropagated wavefield is not generated using the same wave equation as that used for the forward model problem. While  $\mathbf{A}$  is the forward problem operator,  $\mathbf{A}^\dagger$  is the backpropagated wavefield operator and they are not necessarily the same, although under certain formulations of the wave equation  $\mathbf{A} = \mathbf{A}^\dagger$  as it will be shown in the next chapter.

The development shown in equations 3.30 to 3.37 is the adjoint-state method, widely used in the geophysical community in a variety of techniques, from stereotomography to reverse time migration.

## 4. Elastic FWI

I present two different derivations of the inverse problem. The first is based on the elastic wave equation formulated under a matrix algebra formulation. The second treats both model and data spaces as probability distributions and uses Green functions' properties to solve the inverse problem in the framework of functional analysis (developed by Albert Tarantola in the early 1980s). Both formulations are equivalent and lead to the same results as they both correspond to the same physical problem. Although both derivations lead to the same results, the matrix formulation is more flexible in terms of the wave equation. In other words, it is more straightforward to derive the equations when using different parameterisations of the elastic wave equation (particle-velocity and stress, particle-displacement and stress, or second order particle-velocity formulations for example).

I would like to remark that the core of this chapter section is a combined effort between Adrian Umpleby and I. We have worked together to formulate the application of the steepest-descent and the adjoint-state methodologies for the elastic wave equation in three dimensions—formulated as the FD system of equations introduced in chapter 2. We have also collaborated in implementing the stand-alone code into the main software developed by our research group.

## 4.1 Matrix algebra formulation

In this section, I derive expressions for the update of  $V_P$  and  $V_S$  following the methodologies previously discussed in this chapter. The application and iterative implementation of such updates will be described in the next section.

There are several possibilities to acquire data in 3D seismic surveys depending on the nature of the recording devices deployed. Data receivers can record pressure (hydrophones) and/or particle velocities in 3 directions (geophones). For the most general case the data vector is represented as

$$\mathbf{d} = (\mathbf{p}, \mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z) = (v_1^x, v_2^x, \dots, v_N^x, v_1^y, v_2^y, \dots, v_N^y, v_1^z, v_2^z, \dots, v_N^z, p_1, p_2, \dots, p_N) \quad (4.1)$$

where  $4N$  is the dimension of the data vector,  $v^x$ ,  $v^y$  and  $v^z$  are particle velocities in the three Cartesian directions and  $p$  is pressure. The elastic wave equation formulated in equations 2.15 has the following model parameter vector associated

$$\mathbf{m} = (\boldsymbol{\rho}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = (\rho_1, \rho_2, \dots, \rho_M, \lambda_1, \lambda_2, \dots, \lambda_M, \mu_1, \mu_2, \dots, \mu_M) \quad (4.2)$$

with  $M$  the total number of model cells. As before  $\rho$ ,  $\lambda$  and  $\mu$  are the density and the Lamé parameters respectively. There are other possible parameterisations of the elastic wave equation that may be more suitable to perform FWI. In exploration geophysics seismic velocities and density are more commonly used to describe elastic media and they can be related with the Lamé parameters as

$$\begin{aligned} V_P &= \sqrt{\frac{\lambda + 2\mu}{\rho}} \\ V_S &= \sqrt{\frac{\mu}{\rho}} \end{aligned} \quad (4.3)$$

and

$$\begin{aligned}\lambda &= \rho(V_P^2 - 2V_S^2) \\ \mu &= \rho V_S^2\end{aligned}\tag{4.4}$$

where  $V_P$  and  $V_S$  are pressure and shear wave propagation velocities respectively. Indeed the wave equation can be formulated using any set of parameters desired; for example  $V_P/V_S$  ratio, impedances  $I_P$  and  $I_S$  or any other imaginable combination.

Elastic FWI equations will be derived using seismic velocities, Lamé parameters and densities only, but the following development could be used for different parameterisations applying the chain rule to the misfit function as it will be shown later in the chapter.

The vector spaces have been already defined in 4.1 and 4.2. An explicit matrix formulation for the acoustic operator  $\mathbf{A}$  was derived previously in **Box 3**. Consider now an elastic wave equation operator  $\mathbf{A}$  which acts over an elastic wavefield as  $\mathbf{A}\phi = \mathbf{s}$ , and a time-step marching scheme similar to the acoustic one. The main difference between the acoustic and the elastic case is the dimension of the vector spaces. The wavefield has now nine different components as opposed to only one in the acoustic case —which is pressure. The new wavefield is composed of six stresses and three particle velocities

$$\phi = (\tau_{xx}, \tau_{yy}, \tau_{zz}, \tau_{xy}, \tau_{xz}, \tau_{yz}, v_x, v_y, v_z)$$

of which four of them are recorded as data as shown in equation 4.1. Besides, the model is now described using three parameters instead of only one ( $V_P$ ) as shown in equation 4.2. The new discretised variables are defined

$$\phi = (T_{xx}, T_{yy}, T_{zz}, T_{xy}, T_{xz}, T_{yz}, U, V, W)\tag{4.5}$$

note that  $T_{ij}, i, j \in x, y, z$  and  $U, V, W$  are discretised vectors although they are not in boldface. From now on, they will refer to vectors and their elements will be noted by

$T_{k,l,m}^{ij}$  where  $k, l, m$  are indices indicating positions in the discretised model.

Another important difference with higher order time-derivative formulations —like the one presented for the acoustic case in the previous chapter— is that the order of the derivatives affects the structure of the matrix operator  $\mathbf{A}$ . In particular, the temporal derivatives are first order which implies (following the nomenclature of the acoustic case) that the matrix  $\mathbf{R}$  is 0 because the wavefield values two time-steps before are no longer needed. The decomposition of the elastic operator  $\mathbf{A}$  in smaller matrices operating over different time-steps is then

$$\begin{bmatrix} \mathbf{P} & \mathbf{Q} \\ & \mathbf{P} & \mathbf{Q} \\ & & \ddots & & 0 \\ & & & \mathbf{P} & \mathbf{Q} \\ 0 & & & \mathbf{P} & \mathbf{Q} \\ & & & & \mathbf{P} \end{bmatrix} \begin{bmatrix} \phi_N \\ \phi_{N-1} \\ \vdots \\ \phi_3 \\ \phi_2 \\ \phi_1 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{N-1} \\ \mathbf{s}_{N-2} \\ \vdots \\ \mathbf{s}_2 \\ \mathbf{s}_1 \\ \mathbf{s}_0 \end{bmatrix}$$

where now, the wavefield  $\phi$  contains all stresses and particle velocities and  $\mathbf{s}$  contains as many entries (with non-zeroes at according to the choice of source insertion variables: Txx,Tyy and Tzz and/or U,V and W).

The system of equations 2.15 after time discretisation and some reordering of its terms

becomes

$$\begin{aligned}
T_{xx}^{l+1} - T_{xx}^l - (\lambda + 2\mu) \frac{\partial U}{\partial x} - \mu \left( \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} \right) &= s_{T_{xx}} \\
T_{yy}^{l+1} - T_{yy}^l - (\lambda + 2\mu) \frac{\partial V}{\partial y} - \mu \left( \frac{\partial U}{\partial x} + \frac{\partial W}{\partial z} \right) &= s_{T_{yy}} \\
T_{zz}^{l+1} - T_{zz}^l - (\lambda + 2\mu) \frac{\partial W}{\partial z} - \mu \left( \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} \right) &= s_{T_{zz}} \\
T_{xy}^{l+1} - T_{xy}^l - \mu \left( \frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) &= s_{T_{xy}} = 0 \\
T_{xz}^{l+1} - T_{xz}^l - \mu \left( \frac{\partial U}{\partial z} + \frac{\partial W}{\partial x} \right) &= s_{T_{xz}} = 0 \\
T_{yz}^{l+1} - T_{yz}^l - \mu \left( \frac{\partial V}{\partial z} + \frac{\partial W}{\partial y} \right) &= s_{T_{yz}} = 0 \\
\rho U^{l+1} - \rho U^l - \frac{\partial T_{xx}}{\partial x} - \frac{\partial T_{xy}}{\partial y} - \frac{\partial T_{xz}}{\partial z} &= s_U \\
\rho V^{l+1} - \rho V^l - \frac{\partial T_{xy}}{\partial x} - \frac{\partial T_{yy}}{\partial y} - \frac{\partial T_{yz}}{\partial z} &= s_V \\
\rho W^{l+1} - \rho W^l - \frac{\partial T_{xz}}{\partial x} - \frac{\partial T_{yz}}{\partial y} - \frac{\partial T_{zz}}{\partial z} &= s_W
\end{aligned} \tag{4.6}$$

where the super-indices  $^l$  ( $^{l+1}$ ) represent the values at the time-step  $l$  ( $l+1$ ). Note that the shear stress sources are zero (in exploration geophysics this is always true, however in earthquake simulations they may take non-zero values) and the staggered nature of the time discretisation has been concealed by the order of the operations but is still implicitly present.

Equation 4.6 defines the form of the matrices  $\mathbf{P}$  and  $\mathbf{Q}$ .  $\mathbf{P}$  operates over variables defined at the same time-step —which actually correspond to two different time-steps

separated  $\Delta t/2$ — and considering only time discretisation has the form

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -(\lambda + 2\mu)\partial_x & -\lambda\partial_y & -\lambda\partial_z \\ 0 & 1 & 0 & 0 & 0 & 0 & -\lambda\partial_x & -(\lambda + 2\mu)\partial_y & -\lambda\partial_z \\ 0 & 0 & 1 & 0 & 0 & 0 & -\lambda\partial_x & -\lambda\partial_y & -(\lambda + 2\mu)\partial_z \\ 0 & 0 & 0 & 1 & 0 & 0 & -\mu\partial_y & -\mu\partial_x & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -\mu\partial_z & 0 & -\mu\partial_x \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -\mu\partial_z & -\mu\partial_y \\ -\partial_x & 0 & 0 & -\partial_y & -\partial_z & 0 & \rho & 0 & 0 \\ 0 & -\partial_y & 0 & -\partial_x & 0 & -\partial_z & 0 & \rho & 0 \\ 0 & 0 & -\partial_z & 0 & -\partial_x & -\partial_y & 0 & 0 & \rho \end{bmatrix}$$

and the matrix  $\mathbf{Q}$  is simply a diagonal matrix

$$\mathbf{Q} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\rho & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\rho \end{bmatrix}$$

The spatial derivatives are introduced via discretisation of space and approximation of the spatial derivatives using FD operators. Such discretisation leads to an expansion of each of the above matrices' entries into new  $9M \times 9M$  matrices with the following non-zero elements: each row of these new matrices corresponds to the value of one variable (stress or particle-velocity) at one cell in the model and contains all the stress and particle-velocities at all the points in the model —hence the dimension: 9 variables  $\times M$  grid points. There are as many rows as columns since all variables need be calculated at all points in the

model. A partial representation illustrates the shape of these matrices

$$\begin{bmatrix} \dots & 1 & \dots & -c2 & -c1 & c1 & c2 & \dots \\ & \vdots & & & & & & \\ & \vdots & & & & & & \\ & \vdots & & & & & & \\ & \vdots & & & & & & \\ & \vdots & & & & & & \\ & \vdots & & & & & & \\ & \vdots & & & & & & \\ & \vdots & & & & & & \\ & \vdots & & & & & & \\ & \vdots & & & & & & \\ & \vdots & & & & & & \end{bmatrix} \begin{bmatrix} \vdots \\ T_{k,l,m}^{xx} \\ \vdots \\ U_{k-1,l-1,m-1} \\ U_{k,l,m} \\ U_{k+1,l+1,m+1} \\ U_{k+2,l+2,m+2} \\ \vdots \end{bmatrix}$$

where only the partial derivative of  $T_{xx}$  in the position  $k, l, m$  is shown, and the mapping of the model onto a vector is such that elements in the  $x$  direction are contiguously stored. The indices of the particle velocities are staggered in space although it may not seem so, as discussed before.

The exhaustive analysis of the operator's structure has been presented to prove its consistency with the development of the inverse problem development to follow. The objective is to prove that the operators are performing the expected operations once they are transposed. Note that all the matrices involved in the solution of the elastic wave equation have only real entries. Therefore, their conjugate transposes are simply the transposed matrices. From the structure of the matrix operators, one can infer that the transpose of the matrix  $\mathbf{A}$  is independent of the time and space discretisation. Thus, a more compact version of the elastic wave equation can be written as

$$\begin{bmatrix} \partial_t & 0 & 0 & 0 & 0 & 0 & -(\lambda + 2\mu)\partial_x & -\lambda\partial_y & -\lambda\partial_z \\ 0 & \partial_t & 0 & 0 & 0 & 0 & -\lambda\partial_x & -(\lambda + 2\mu)\partial_y & -\lambda\partial_z \\ 0 & 0 & \partial_t & 0 & 0 & 0 & -\lambda\partial_x & -\lambda\partial_y & -(\lambda + 2\mu)\partial_z \\ 0 & 0 & 0 & \partial_t & 0 & 0 & -\mu\partial_y & -\mu\partial_x & 0 \\ 0 & 0 & 0 & 0 & \partial_t & 0 & -\mu\partial_z & 0 & -\mu\partial_x \\ 0 & 0 & 0 & 0 & 0 & \partial_t & 0 & -\mu\partial_z & -\mu\partial_y \\ -\partial_x & 0 & 0 & -\partial_y & -\partial_z & 0 & \rho\partial_t & 0 & 0 \\ 0 & -\partial_y & 0 & -\partial_x & 0 & -\partial_z & 0 & \rho\partial_t & 0 \\ 0 & 0 & -\partial_z & 0 & -\partial_x & -\partial_y & 0 & 0 & \rho\partial_t \end{bmatrix} \begin{bmatrix} T_{xx} \\ T_{yy} \\ T_{zz} \\ T_{xy} \\ T_{xz} \\ T_{yz} \\ U \\ V \\ W \end{bmatrix} = \begin{bmatrix} s_{T_{xx}} \\ s_{T_{yy}} \\ s_{T_{zz}} \\ 0 \\ 0 \\ 0 \\ s_U \\ s_V \\ s_W \end{bmatrix} \quad (4.7)$$

where the source terms corresponding to the shear stresses are zero (as discussed above), and the pressure term is distributed over the normal stresses following the relation

$$p = \frac{1}{3}(T_{xx} + T_{yy} + T_{zz}) \quad (4.8)$$

We start defining our set of data residuals as  $\Delta\mathbf{d} = (\phi - \phi_{obs})^T \mathbf{D}(\phi - \phi_{obs})$  where  $\mathbf{D}$  is the previously defined matrix that selects data positions in the model grid. Now we want to apply the steepest-descent method to calculate the update of model parameters using  $\nabla F$  as defined by equations 3.29 and 3.37. But note that to calculate the gradient using the backpropagated wavefield involves using a different operator for the forward modelling calculation of  $\phi$  and for  $\phi_b$  because the transpose of 4.7 is not the same matrix.

It is more convenient to rewrite the operator  $\mathbf{A}$  as a symmetric matrix to be able to backpropagate the residuals using the same equation used for the forward modelling. Formally there is no need to do it, but from an implementation point of view it simplifies the code structure of the algorithm. By applying the following change of variables:

$$\begin{aligned} T_{xx} &= p + q + r \\ T_{yy} &= p + \frac{1}{2}(\sqrt{3} - 1)q - \frac{1}{2}(\sqrt{3} + 1)r \\ T_{zz} &= p - \frac{1}{2}(\sqrt{3} + 1)q + \frac{1}{2}(\sqrt{3} - 1)r \end{aligned} \quad (4.9)$$

and

$$\begin{aligned} 3p &= T_{xx} + T_{yy} + T_{zz} \\ 3q &= T_{xx} + \frac{1}{2}(\sqrt{3} - 1)T_{yy} - \frac{1}{2}(\sqrt{3} + 1)T_{zz} \\ 3r &= T_{xx} - \frac{1}{2}(\sqrt{3} + 1)T_{yy} + \frac{1}{2}(\sqrt{3} - 1)T_{zz} \end{aligned} \quad (4.10)$$

where now  $p$  is a scaled version of pressure by a factor of 3. This change of variables

is equivalent to the one presented in [Castellanos et al., 2011]. The new wave equation operator  $\mathbf{A}_{cv}$  takes then the form

$$\mathbf{A}_{cv} = \begin{bmatrix} \frac{3}{3\lambda+2\mu}\partial_t & 0 & 0 & 0 & 0 & 0 & -\partial_x & -\partial_y & -\partial_z \\ 0 & \frac{3}{2\mu}\partial_t & 0 & 0 & 0 & 0 & -\partial_x & -\frac{\sqrt{3}-1}{2}\partial_y & \frac{\sqrt{3}+1}{2}\partial_z \\ 0 & 0 & \frac{3}{2\mu}\partial_t & 0 & 0 & 0 & -\partial_x & \frac{\sqrt{3}+1}{2}\partial_y & -\frac{\sqrt{3}-1}{2}\partial_z \\ 0 & 0 & 0 & \frac{1}{\mu}\partial_t & 0 & 0 & -\partial_y & -\partial_x & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\mu}\partial_t & 0 & -\partial_z & 0 & -\partial_x \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\mu}\partial_t & 0 & -\partial_z & -\partial_y \\ -\partial_x & -\partial_x & \partial_x & -\partial_y & -\partial_z & 0 & \rho\partial_t & 0 & 0 \\ -\partial_y & -\frac{\sqrt{3}-1}{2}\partial_y & \frac{\sqrt{3}+1}{2}\partial_y & -\partial_x & 0 & -\partial_z & 0 & \rho\partial_t & 0 \\ -\partial_z & \frac{\sqrt{3}+1}{2}\partial_z & -\frac{\sqrt{3}-1}{2}\partial_z & 0 & -\partial_x & -\partial_y & 0 & 0 & \rho\partial_t \end{bmatrix} \quad (4.11)$$

which is symmetric, i.e.  $\mathbf{A}_{cv}^T = \mathbf{A}_{cv}$ . Under this symmetric form, the gradient calculation shown in equation 3.37 describes the wavefields  $\phi_b$  and  $\phi$  with consistent variables —both are solutions of the same form of the wave equation  $\mathbf{A}_{cv}$ . The only part left is the derivative of  $\mathbf{A}_{cv}$  with respect to the model parameters  $\partial\mathbf{A}_{cv}/\partial\mathbf{m}$ .

Previous studies have shown that the choice of model parameters have an impact on the shape of the solution space. Tarantola [Tarantola, 1986] and Mora [Mora, 1987] studied this problem and concluded that P- and S-wave velocities or impedances were better resolved than the Lamé parameters, and that density was generally badly resolved. Consequently, I will only derive expressions for  $V_P$  and  $V_S$ , but the procedure is completely analogous for any parameterisation. Inversion of density will not be explored in the present study, which will be limited to inversion of seismic velocities.

The update  $\Delta\mathbf{m}$  is then reduced —following equation 3.36— to

$$\Delta V_{Pi} = \Re \left\{ \phi_b \frac{\partial \mathbf{A}_{cv}}{\partial V_{Pi}} \phi \right\} \quad \Delta V_{Si} = \Re \left\{ \phi_b \frac{\partial \mathbf{A}_{cv}}{\partial V_{Si}} \phi \right\} \quad (4.12)$$

where the derivative of  $\mathbf{A}_{cv}$  with respect to  $\mathbf{V}_P$  is

$$\frac{\partial \mathbf{A}_{cv}}{\partial V_{Pi}} = \begin{bmatrix} \frac{-18V_{Pi}}{\rho(3V_{Pi}^2 - 4V_{Si}^2)^2} \partial_t & 0 & \cdots & 0 \\ 0 & 0 & & \\ \vdots & & \ddots & \\ 0 & & & 0 \end{bmatrix} \quad (4.13)$$

which only acts over  $p$ ; and with respect to  $\mathbf{V}_S$

$$\frac{\partial \mathbf{A}_{cv}}{\partial V_{Si}} = \begin{bmatrix} \frac{-24V_{Si}}{\rho(3V_{Pi}^2 - 4V_{Si}^2)^2} \partial_t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-3}{\rho V_{Si}^3} \partial_t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-3}{\rho V_{Si}^3} \partial_t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-2}{\rho V_{Si}^3} \partial_t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-2}{\rho V_{Si}^3} \partial_t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-2}{\rho V_{Si}^3} \partial_t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-2}{\rho V_{Si}^3} \partial_t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-2}{\rho V_{Si}^3} \partial_t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-2}{\rho V_{Si}^3} \partial_t \end{bmatrix} \quad (4.14)$$

which acts over all stresses and none of the particle velocities and where the Lamé parameters have been converted to seismic velocities using the relations 4.3-4.4

For the sake of simplicity, the Hessian contributions are not considered and can be applied afterwards. Substitution of 4.13 and 4.14 into 4.12 provides expressions to update seismic velocities. After some matrix operations the derivatives of the cost function with respect to  $\mathbf{V}_P$  and  $\mathbf{V}_S$  become

$$\nabla_{v_p} F = \frac{-18\mathbf{V}_P}{\rho(3\mathbf{V}_P^2 - 4\mathbf{V}_S^2)^2} \int_0^T \overleftarrow{p} \frac{\partial \vec{p}}{\partial t} dt \quad (4.15)$$

$$\begin{aligned} \nabla_{v_s} F = & \frac{-24\mathbf{V}_S}{\rho(3\mathbf{V}_P^2 - 4\mathbf{V}_S^2)^2} \int_0^T \overleftarrow{p} \frac{\partial \vec{p}}{\partial t} dt - \frac{3}{\rho \mathbf{V}_S^3} \int_0^T \overleftarrow{q} \frac{\partial \vec{q}}{\partial t} dt - \frac{3}{\rho \mathbf{V}_S^3} \int_0^T \overleftarrow{r} \frac{\partial \vec{r}}{\partial t} dt \\ & - \frac{2}{\rho \mathbf{V}_S^3} \int_0^T \overleftarrow{T}_{xy} \frac{\partial \vec{T}_{xy}}{\partial t} dt - \frac{2}{\rho \mathbf{V}_S^3} \int_0^T \overleftarrow{T}_{xz} \frac{\partial \vec{T}_{xz}}{\partial t} dt - \frac{2}{\rho \mathbf{V}_S^3} \int_0^T \overleftarrow{T}_{yz} \frac{\partial \vec{T}_{yz}}{\partial t} dt \end{aligned} \quad (4.16)$$

where T is the total time and the right arrows  $\rightarrow$  denote forward wavefield variables and

the left ones  $\leftarrow$  denote transposed backpropagated wavefields. Note that because of the  $\partial_t$  operator in 4.13 and 4.14 the forward wavefield is differentiated with respect to time.

At this stage is safe to undo the change of variables and to rewrite equations 4.15 and 4.16 in terms of regular stresses, because the evaluation of the gradient using the adjoint-state method has been already calculated with the symmetric operator  $\mathbf{A}_{cv}$ . After a few algebraic operations

$$\Delta \mathbf{V}_P = -\alpha_{v_P} \nabla_{v_P} F = -\alpha_{v_P} \frac{-18\mathbf{V}_P}{\rho(3\mathbf{V}_P^2 - 4\mathbf{V}_S^2)^2} \int_0^T (\overleftarrow{T}_{xx} + \overleftarrow{T}_{yy} + \overleftarrow{T}_{zz}) \frac{\partial}{\partial t} (\overrightarrow{T}_{xx} + \overrightarrow{T}_{yy} + \overrightarrow{T}_{zz}) dt \quad (4.17)$$

$$\begin{aligned} \Delta \mathbf{V}_S = -\alpha_{v_S} \nabla_{v_S} F = & -\alpha_{v_S} \left( \frac{8\mathbf{V}_S}{3\rho(3\mathbf{V}_P^2 - 4\mathbf{V}_S^2)^2} - \frac{1}{3\rho\mathbf{V}_S^3} \right) \int_0^T (\overleftarrow{T}_{xx} + \overleftarrow{T}_{yy} + \overleftarrow{T}_{zz}) \frac{\partial}{\partial t} (\overrightarrow{T}_{xx} + \overrightarrow{T}_{yy} + \overrightarrow{T}_{zz}) dt \\ & - \frac{1}{\rho\mathbf{V}_S^3} \int_0^T \left( \overleftarrow{T}_{xx} \frac{\partial \overrightarrow{T}_{xx}}{\partial t} + \overleftarrow{T}_{yy} \frac{\partial \overrightarrow{T}_{yy}}{\partial t} + \overleftarrow{T}_{zz} \frac{\partial \overrightarrow{T}_{zz}}{\partial t} \right) \\ & - \frac{2}{\rho\mathbf{V}_S^3} \int_0^T \left( \overleftarrow{T}_{xy} \frac{\partial \overrightarrow{T}_{xy}}{\partial t} + \overleftarrow{T}_{xz} \frac{\partial \overrightarrow{T}_{xz}}{\partial t} + \overleftarrow{T}_{yz} \frac{\partial \overrightarrow{T}_{yz}}{\partial t} \right) \end{aligned} \quad (4.18)$$

where the time derivatives of the sum of normal stresses  $\partial(\overrightarrow{T}_{xx} + \overrightarrow{T}_{yy} + \overrightarrow{T}_{zz})/\partial t$  can be substituted by  $\nabla \cdot \vec{\mathbf{v}}$  with  $\vec{\mathbf{v}} = (\vec{U}, \vec{V}, \vec{W})$  to simplify the notation using the relation

$$\frac{\partial}{\partial t} (T_{xx} + T_{yy} + T_{zz}) = \rho(3\mathbf{V}_P^2 - 4\mathbf{V}_S^2) \nabla \cdot \mathbf{v} \quad (4.19)$$

Actually, all the stresses' time derivatives could be written in terms of the particle-velocities' spatial derivatives, but further substitutions do not simplify the notation and therefore are not applied.

Finally, making use of these substitutions, the expressions to update  $\mathbf{V}_P$  and  $\mathbf{V}_S$  are

$$\Delta \mathbf{V}_P = -\alpha_{V_P} \frac{2\mathbf{V}_P}{3(3\mathbf{V}_P^2 - 4\mathbf{V}_S^2)} \int_0^T (\overleftarrow{T}_{xx} + \overleftarrow{T}_{yy} + \overleftarrow{T}_{zz}) \nabla \cdot \vec{\nabla} dt \quad (4.20)$$

$$\begin{aligned} \Delta \mathbf{V}_S = & -\alpha_{V_S} \left[ \left( \frac{8\mathbf{V}_S}{9(3\mathbf{V}_P^2 - 4\mathbf{V}_S^2)} + \frac{3\mathbf{V}_P^2 - 4\mathbf{V}_S^2}{3\mathbf{V}_S^2} \right) \int_0^T (\overleftarrow{T}_{xx} + \overleftarrow{T}_{yy} + \overleftarrow{T}_{zz}) \nabla \cdot \vec{\nabla} dt \right. \\ & - \frac{1}{\rho \mathbf{V}_S^3} \int_0^T \left( \overleftarrow{T}_{xx} \frac{\partial \overrightarrow{T}_{xx}}{\partial t} + \overleftarrow{T}_{yy} \frac{\partial \overrightarrow{T}_{yy}}{\partial t} + \overleftarrow{T}_{zz} \frac{\partial \overrightarrow{T}_{zz}}{\partial t} \right) \\ & \left. - \frac{2}{\rho \mathbf{V}_S^3} \int_0^T \left( \overleftarrow{T}_{xy} \frac{\partial \overrightarrow{T}_{xy}}{\partial t} + \overleftarrow{T}_{xz} \frac{\partial \overrightarrow{T}_{xz}}{\partial t} + \overleftarrow{T}_{yz} \frac{\partial \overrightarrow{T}_{yz}}{\partial t} \right) \right] \end{aligned} \quad (4.21)$$

Gradient expressions to update other parameters can be found by applying the chain rule to the derivative of the data with respect to the model parameter  $\mathbf{m}_1$ . If the original gradient is

$$\nabla F_{m_1} = \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}_1} \right)^T (\mathbf{d} - \mathbf{d}_{obs}) \quad (4.22)$$

then,  $\nabla F_{m_2}$  contains the Jacobian matrix of the transformation of variables  $\mathbf{m}_1 \rightarrow \mathbf{m}_2$

$$\nabla F_{m_2} = \left( \frac{\partial \mathbf{d}}{\partial \mathbf{m}_1} \frac{\partial \mathbf{m}_1}{\partial \mathbf{m}_2} \right)^T (\mathbf{d} - \mathbf{d}_{obs}) \quad (4.23)$$

Thus, the gradients for different sets of model parameters can be found without redoing all the development from the very beginning and the original expressions 4.20 and 4.21 are recycled by just adding the Jacobian  $\left( \frac{\partial \mathbf{m}_1}{\partial \mathbf{m}_2} \right)$  to them. For example, P-wave impedance  $\mathbf{I}_P$  and  $\mathbf{V}_P$  are related  $\mathbf{I}_P = \rho \mathbf{V}_P$ , and therefore the gradient  $\nabla F_{I_P}$  is

$$\nabla \mathbf{I}_P = \frac{\partial \mathbf{V}_P}{\partial \mathbf{I}_P} \nabla \mathbf{V}_P + \frac{\partial \mathbf{V}_S}{\partial \mathbf{I}_P} \nabla \mathbf{V}_S = \frac{1}{\rho} \nabla \mathbf{V}_P \quad (4.24)$$

and for  $\mathbf{I}_S = \rho \mathbf{V}_S$ , S-wave impedance becomes

$$\nabla \mathbf{I}_S = \frac{\partial \mathbf{V}_P}{\partial \mathbf{I}_S} \nabla \mathbf{V}_P + \frac{\partial \mathbf{V}_S}{\partial \mathbf{I}_S} \nabla \mathbf{V}_S = \frac{1}{\rho} \nabla \mathbf{V}_S \quad (4.25)$$

For wide-angle geometries, the best parameterisation (in our group's experience) is to update slownesses  $\mathbf{Sl}_P$  and  $\mathbf{Sl}_S$  instead of velocities. This is likely because, slowness

and travel-times have a linear relation, whereas velocities and travel-times do not. Thus, the linearisation of the inverse problem behaves better when the model parameters (slownesses) have a linear relation with one important characteristic of the data (travel-times). Applying the same chain rule derivation as for the impedance case leads to the following expressions for the slowness gradients:

$$\nabla \mathbf{S}_{\mathbf{l}_P} = \frac{\partial \mathbf{V}_P}{\partial \mathbf{S}_{\mathbf{l}_P}} \nabla \mathbf{V}_P + \frac{\partial \mathbf{V}_S}{\partial \mathbf{S}_{\mathbf{l}_P}} \nabla \mathbf{V}_S = -\mathbf{V}_P^2 \nabla \mathbf{V}_P \quad (4.26)$$

and

$$\nabla \mathbf{S}_{\mathbf{l}_S} = \frac{\partial \mathbf{V}_P}{\partial \mathbf{S}_{\mathbf{l}_S}} \nabla \mathbf{V}_P + \frac{\partial \mathbf{V}_S}{\partial \mathbf{S}_{\mathbf{l}_S}} \nabla \mathbf{V}_S = -\mathbf{V}_S^2 \nabla \mathbf{V}_S \quad (4.27)$$

## 4.2 Functional analysis derivation

As a QC measure, I present the gradient expressions published by Tarantola more than 25 years ago and prove that they are exactly the same as those derived in the previous section. Following the work of Tarantola, Mora published in 1987 a paper summarising such derivation which I will follow it in this section. I try to respect the original notation and only partially adapt it to be as consistent as possible with the rest of the thesis.

The local optimisation problem can be attacked from a different angle if one considers data and model parameters as probability distributions of the form

$$P(\mathbf{d}, \mathbf{m}) = e^{-\frac{1}{2}(\Delta \mathbf{d}^\dagger \mathbf{C}_d^{-1} \Delta \mathbf{d} + \Delta \mathbf{m}^\dagger \mathbf{C}_m^{-1} \Delta \mathbf{m})} \quad (4.28)$$

where  $\mathbf{C}_d$  and  $\mathbf{C}_m$  are the covariances of data and model parameters respectively. Due to lack of knowledge, the cross-terms of these matrices are ignored, and their diagonals can take specific values if some a priori information is known, or unity if not. The solution with maximum probability reduces then to the global minimum of the least-square functional

$$S(\mathbf{d}, \mathbf{m}) = \Delta \mathbf{d}^\dagger \mathbf{C}_d^{-1} \Delta \mathbf{d} + \Delta \mathbf{m}^\dagger \mathbf{C}_m^{-1} \Delta \mathbf{m} \quad (4.29)$$

and, as before, taking the partial derivative of the functional with respect to the model parameters leads to

$$\mathbf{g} = \frac{1}{2} = \frac{\partial \mathbf{d}}{\partial \mathbf{m}}^\dagger \mathbf{C}_d^{-1} \Delta \mathbf{d} + \mathbf{C}_m^{-1} \Delta \mathbf{m} = 0 \quad (4.30)$$

where the extremal condition has already been imposed. Linearisation of model parameters and data

$$\delta \mathbf{d} = \mathbf{d}(\mathbf{m} + \delta \mathbf{m}) - \mathbf{d}(\mathbf{m}) = \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \delta \mathbf{m} \quad (4.31)$$

reinjected into 4.30 results in

$$\Delta \mathbf{m} = -\mathbf{H}^{-1} \mathbf{g} \quad (4.32)$$

which is equivalent to the previous equation 3.17 considering  $\mathbf{g} = \nabla F$  and  $\mathbf{H} \equiv$  Hessian.

The Einstein notation and continuous functions (instead of discretised variables) are used in the following development. Besides, the formulation of the elastic wave equation is based on particle displacements  $\mathbf{u}$  rather than velocities  $\mathbf{v}$  and the stress equations have been removed by reinjecting them in the system as functions of  $\mathbf{u}$ . The new system of equations becomes

$$\begin{aligned} \rho \ddot{u}_i - \frac{\partial}{\partial j} \left( c_{ijkl} \frac{\partial}{\partial l} u_k \right) &= f_i \\ c_{ijkl} \frac{\partial}{\partial l} u_k n_j &= T_i \end{aligned} \quad (4.33)$$

where  $f_i$  and  $T_i$  are body forces and tractions respectively and  $i, j, k, l$  are spatial directions  $x, y$  or  $z$ .

The problem reduces (as in the previous section) to find an equivalent expression for the linearisation of the model and data perturbations

$$\Delta \mathbf{d} = \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \Delta \mathbf{m} \quad (4.34)$$

or in continuous form

$$\Delta \mathbf{d} = \int_V dV \frac{\partial \mathbf{d}}{\partial \mathbf{m}(\mathbf{x})} \Delta \mathbf{m}(\mathbf{x}) \quad (4.35)$$

Taking a different notation (for convenience), equation 4.35 becomes

$$\Delta u_i = \int_V dV \frac{\partial u_i}{\partial \mathbf{m}(\mathbf{x})} \Delta \mathbf{m}(\mathbf{x}) \quad (4.36)$$

where  $u_i$  is now the  $i$ th component of the wavefield, and the integral is over volume. By taking the adjoint operator represented by the Fréchet kernel, one can rewrite equation 4.36 as

$$\Delta \hat{\mathbf{m}}(\mathbf{x}) = \sum_{src} \int_0^T dt \sum_{rec} \frac{\partial u_i}{\partial \mathbf{m}(\mathbf{x})} \Delta u_i \quad (4.37)$$

where, as noted in equation 3.2,  $\hat{\mathbf{m}}$  is different from  $\mathbf{m}$  and the summations are for all sources and receivers. The linearisation of model and data perturbation is the Born approximation, and as in the previous section it will only be used to solve the inverse problem —the forward problem does not need use this approximation.

Applying the following change of variables

$$\begin{aligned} u_i &\rightarrow u_i + \delta u_i \\ \rho &\rightarrow \rho + \delta \rho \\ c_{ijkl} &\rightarrow c_{ijkl} + \delta c_{ijkl} \\ f_i &\rightarrow f_i + \delta f_i \\ T_i &\rightarrow T_i + \delta T_i \end{aligned} \quad (4.38)$$

to the elastic wave equation 4.33 and reordering terms one obtains

$$\begin{aligned} \rho \delta \ddot{u}_i - \frac{\partial}{\partial j} \left( c_{ijkl} \frac{\partial}{\partial l} \delta u_k \right) &= \Delta f_i \\ c_{ijkl} \frac{\partial}{\partial l} \delta u_k n_j &= \Delta T_i \end{aligned} \quad (4.39)$$

with  $\Delta f_i$  and  $\Delta T_i$  being the new sources containing all the rest of terms

$$\begin{aligned}\Delta f_i &= \delta f_i - \delta \rho \ddot{u}_i + \partial_j \delta c_{ijkl} \partial_l u_k + O_i(\delta \rho, \delta c_{ijkl}, \delta f_i, \delta T_i)^2 \\ \Delta T_i &= \delta T_i - \delta c_{ijkl} \partial_l u_k n_j + O_i(\delta \rho, \delta c_{ijkl}, \delta f_i, \delta T_i)^2\end{aligned}\quad (4.40)$$

The solution of 4.38 in terms of Green functions can be expressed as

$$\delta u_i = \int_V dV G_{ij} * (\delta f_i - \delta \rho \ddot{u}_j + \partial_k \delta c_{jklm} \frac{\partial u_l}{\partial m}) + \int_S dS G_{ij} * (\delta T_i - \delta c_{jklm} \frac{\partial u_l}{\partial m} n_k) \quad (4.41)$$

where  $*$  indicates convolution. By using several properties of the convolution and assuming isotropy,  $\delta u_i$  becomes

$$\begin{aligned}\delta u_i &= \int_V dV G_{ij} * \delta f_i + \int_S dS G_{ij} * \delta T_i - \int_V dV G_{ij} * \dot{u}_j \delta \rho \\ &\quad - \int_V dV \frac{\partial G_{ij}}{\partial j} * \frac{\partial u_m}{\partial m} \delta \lambda - \int_V dV \frac{\partial G_{ij}}{\partial k} * \left( \frac{\partial u_j}{\partial k} + \frac{\partial u_k}{\partial j} \right) \delta \mu\end{aligned}\quad (4.42)$$

and integrating over the data space using equation 4.37 results in the following parameter perturbations:

$$\begin{aligned}\Delta \hat{\lambda} &= - \int_0^T dt \sum_{rec} \frac{\partial G_{ij}}{\partial j} * \frac{\partial u_m}{\partial m} \delta u_i \\ \Delta \hat{\mu} &= - \frac{1}{2} \int_0^T dt \sum_{rec} \left( \frac{\partial G_{ij}}{\partial k} + \frac{\partial G_{ik}}{\partial j} \right) * \left( \frac{\partial u_j}{\partial k} + \frac{\partial u_k}{\partial j} \right)\end{aligned}\quad (4.43)$$

where only the Lamé parameter updates are shown.

Finally, by making use of the reciprocity principle, the commutativity and time shift properties of the convolution one can define the new wavefield

$$\Psi_j = \sum_{rec} G_{ij}(\mathbf{x}, -t; \mathbf{x}_R, 0) * \delta u_i(\mathbf{x}_R, t) \quad (4.44)$$

and reinjected it into 4.43 to obtain the desired model parameter gradients. Note that this is the backpropagated residual wavefield, expressed as the convolution of the Green function after applying reciprocity to it and running backwards in time ( $-t$  term), and the perturbation of the wavefield  $\delta u_i$  or residual wavefield in the receiver points, thus the

data residuals. After some algebra, and making use of equation 4.23 for the chain rule change of variables derivation, one obtains the following expressions for the update of P- and S-wave velocities (where the sum over receivers has been dropped to clarify the future comparison with the matrix algebra results)

$$\Delta \mathbf{V}_P = 2\rho \mathbf{V}_P \int_0^T dt \frac{\partial u_m}{\partial m} \frac{\partial \Psi_j}{\partial j} \quad (4.45)$$

$$\Delta \mathbf{V}_S = -4\rho \mathbf{V}_S \int_0^T dt \frac{\partial u_m}{\partial m} \frac{\partial \Psi_j}{\partial j} + 2\rho \mathbf{V}_S \int_0^T dt \frac{1}{2} \left( \frac{\partial u_k}{\partial j} + \frac{\partial u_j}{\partial k} \right) \left( \frac{\partial \Psi_k}{\partial j} + \frac{\partial \Psi_j}{\partial k} \right) \quad (4.46)$$

with the nomenclature equivalence  $u_i \equiv \overrightarrow{u}_i$  and  $\Psi_i \equiv \overleftarrow{u}_i$ .

### 4.3 Comparison between formulations

In order to prove that the parameter updates are the same for both derivations, one must rewrite them to contain the same variables in the integrals. I transform the latter equations 4.45-4.46 to contain stresses instead of particle displacements, although the opposite —that is transform the former into particle displacement— should lead to the very same result. To do so, one must first expand the Einstein notation and compare them with the elastic wave equation in terms of particle-displacements and stresses (equation 2.12). For the pressure velocity gradient, adding the three normal stress equations of 2.12 (three middle ones) provides an adequate relation between spatial derivatives of particle-displacements and stresses

$$\begin{aligned} \frac{\partial u_m}{\partial m} &\rightarrow \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = \frac{1}{3\lambda + 2\mu} (\overrightarrow{T_{xx}} + \overrightarrow{T_{yy}} + \overrightarrow{T_{zz}}) \\ \frac{\partial \Psi_j}{\partial j} &\rightarrow \frac{\partial \Psi_x}{\partial x} + \frac{\partial \Psi_y}{\partial y} + \frac{\partial \Psi_z}{\partial z} = \frac{1}{3\lambda + 2\mu} (\overleftarrow{T_{xx}} + \overleftarrow{T_{yy}} + \overleftarrow{T_{zz}}) \end{aligned} \quad (4.47)$$

and using the relation 4.4 to convert the Lamé parameters into seismic velocities, leads to

$$\Delta \mathbf{V}_P = -\alpha_{V_P} 2 \mathbf{V}_P \frac{1}{\rho(3\mathbf{V}_P^2 - 4\mathbf{V}_S^2)^2} \int_0^T (\overleftarrow{T_{xx}}^{disp} + \overleftarrow{T_{yy}}^{disp} + \overleftarrow{T_{zz}}^{disp})(\overrightarrow{T_{xx}} + \overrightarrow{T_{yy}} + \overrightarrow{T_{zz}}) dt \quad (4.48)$$

where  $T_{ii}^{disp} : i \in \{x, y, z\}$  are the backpropagated residuals when the wave equation is expressed in particle displacements and stresses.

An important difference arises here: the forward stresses are not differentiated with respect to time. The explanation for this disagreement between parameter updates (equations 4.17 and 4.48) lies in the nature of the wave equations used. While the relation of particle-displacements to stresses holds true, the two updates for  $\mathbf{V}_P$  were derived from two different versions of the elastic wave equation: one using particle velocities and the other displacements. Due to this difference, the backpropagated residuals are of different nature and this has an impact on the evaluation of the updates. Going back to the matrix notation, one can notice that the main diagonal of the matrix operator in 4.7 contains a time derivative  $\partial_t$  in all the terms of its diagonal, which is consistent with the particle-velocity equation, but under the particle-displacement formulation the first six diagonal term do not contain any  $\partial_t$  while the last three have now second order time derivatives  $\partial_{tt}$ . As a consequence of that, the partial derivative of  $\mathbf{A}_{cv}$  with respect to  $\mathbf{V}_P$  does not have a partial derivative operator and, consequently the middle term  $\frac{\partial \mathbf{A}_{cv}}{\partial \mathbf{V}_P}$  in the adjoint form of the gradient which operates over the forward wavefield  $\phi$  does not contain it either. The forward propagated stresses are the same because the source term has different forms, in particular, the displacement source is the velocity source integrated over time. For the normal stress  $T_{xx}$ , the source insertion has the form

$$\begin{aligned} \partial_t T_{xx} - (\lambda + 2\mu) \partial_x v_x - \lambda(\partial_y v_y + \partial_z v_z) &= s_{T_{xx}} \\ T_{xx} - (\lambda + 2\mu) \partial_x u_x - \lambda(\partial_y u_y + \partial_z u_z) &= \int s_{T_{xx}} dt \end{aligned} \quad (4.49)$$

where  $v_i, u_i, i \in \{x, y, z\}$  are particle velocities and displacements respectively. Now, if the residuals used to generate the backpropagated wavefield were calculated simply as the difference between modelled and observed data, that would lead to different wavefields because the source terms —the residuals— would not comply with the source relation established in equation 4.49. Thus, in order to generate comparable wavefields, the back-propagated residuals should be modified accordingly. Such modification consists on the

differentiation of the residuals for the particle velocity formulation, which is equivalent to differentiate the backpropagated residual wavefield, which in turn is equivalent to differentiate the forward wavefield (equation 3.37).

The update of  $\mathbf{V}_S$  is subject to this difference in operators as well, and therefore, after the transformation of particle-velocities into stresses the partial derivatives of forward wavefields drops as well. The first step is to expand equation 4.46 and add all six equations (2.12) containing stresses on the left hand side to obtain the relation

$$\begin{aligned} \frac{\partial u_k}{\partial j} + \frac{\partial u_j}{\partial k} &\rightarrow 2 \left( \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} \right) + 2 \left( \frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial x} + \frac{\partial u_z}{\partial y} + \frac{\partial u_x}{\partial z} \right) \\ &= \frac{2}{3\lambda + 2\mu} (\overrightarrow{T_{xx}} + \overrightarrow{T_{yy}} + \overrightarrow{T_{zz}}) + \frac{2}{\mu} (\overrightarrow{T_{xy}} + \overrightarrow{T_{xz}} + \overrightarrow{T_{yz}}) \end{aligned} \quad (4.50)$$

and converting  $\lambda, \mu$  to  $\mathbf{V}_P, \mathbf{V}_S$  again and reinjecting into equation 4.46 leads to

$$\begin{aligned} \Delta \mathbf{V}_S = -\alpha_{V_S} \left[ \left( \frac{8\mathbf{V}_S}{9\rho(3\mathbf{V}_P^2 - 4\mathbf{V}_S^2)^2} + \frac{1}{3\rho\mathbf{V}_S^3} \right) \int_0^T (\overleftarrow{T_{xx}^{disp}} + \overleftarrow{T_{yy}^{disp}} + \overleftarrow{T_{zz}^{disp}})(\overrightarrow{T_{xx}} + \overrightarrow{T_{yy}} + \overrightarrow{T_{zz}}) dt \right. \\ \left. - \frac{1}{\rho\mathbf{V}_S^3} \int_0^T dt (\overleftarrow{T_{xx}^{disp}} \overrightarrow{T_{xx}} + \overleftarrow{T_{yy}^{disp}} \overrightarrow{T_{yy}} + \overleftarrow{T_{zz}^{disp}} \overrightarrow{T_{zz}}) \right. \\ \left. - \frac{2}{\rho\mathbf{V}_S^3} \int_0^T dt (\overleftarrow{T_{xy}^{disp}} \overrightarrow{T_{xy}} + \overleftarrow{T_{xz}^{disp}} \overrightarrow{T_{xz}} + \overleftarrow{T_{yz}^{disp}} \overrightarrow{T_{yz}}) \right] \end{aligned} \quad (4.51)$$

where, as previously noted, the backpropagated wavefields are not differentiated with respect to time.

In summary, the equivalence of the two formulations is satisfied by virtue of the following equivalences

$$\begin{aligned} (\overleftarrow{T_{xx}^{disp}} + \overleftarrow{T_{yy}^{disp}} + \overleftarrow{T_{zz}^{disp}})(\overrightarrow{T_{xx}} + \overrightarrow{T_{yy}} + \overrightarrow{T_{zz}}) &= (\overleftarrow{T_{xx}^{vel}} + \overleftarrow{T_{yy}^{vel}} + \overleftarrow{T_{zz}^{vel}}) \frac{\partial}{\partial t} (\overrightarrow{T_{xx}} + \overrightarrow{T_{yy}} + \overrightarrow{T_{zz}}) \\ (\overleftarrow{T_{xx}^{disp}} \overrightarrow{T_{xx}} + \overleftarrow{T_{yy}^{disp}} \overrightarrow{T_{yy}} + \overleftarrow{T_{zz}^{disp}} \overrightarrow{T_{zz}}) &= (\overleftarrow{T_{xx}^{vel}} \frac{\partial}{\partial t} \overrightarrow{T_{xx}} + \overleftarrow{T_{yy}^{vel}} \frac{\partial}{\partial t} \overrightarrow{T_{yy}} + \overleftarrow{T_{zz}^{vel}} \frac{\partial}{\partial t} \overrightarrow{T_{zz}}) \\ (\overleftarrow{T_{xy}^{disp}} \overrightarrow{T_{xy}} + \overleftarrow{T_{xz}^{disp}} \overrightarrow{T_{xz}} + \overleftarrow{T_{yz}^{disp}} \overrightarrow{T_{yz}}) &= (\overleftarrow{T_{xy}^{vel}} \frac{\partial}{\partial t} \overrightarrow{T_{xy}} + \overleftarrow{T_{xz}^{vel}} \frac{\partial}{\partial t} \overrightarrow{T_{xz}} + \overleftarrow{T_{yz}^{vel}} \frac{\partial}{\partial t} \overrightarrow{T_{yz}}) \end{aligned} \quad (4.52)$$

where  $T_{ij}^{vel} : i, j \in \{x, y, z\}$  are the backpropagated residuals for the particle-velocity formulation. Note that the forward propagated stresses are the same independently of the wave equation used.

## 4.4 Implementation

This section details the computer implementation of the inverse problem. Thanks to the adjoint-state methodology used to obtain the parameter updates, most of the time is spent computing forward problems. Due to limitations in computer memory, a number of strategies are adopted to adapt the algorithm to fit into the capabilities of current computers. The most restrictive limitation is that storing the wavefield at all points of the model for all the time-steps is unaffordable, and a methodology to overcome this problem will be presented.

### 4.4.1 Code structure

The inverse problem has been solved from a mathematical point of view and appropriate expressions to update model parameters have been presented. The iterative nature of the local optimisation method results in computer intensive algorithms, mainly due to the elevated number of iterations computed (typically go from ten up to thousands in most extreme cases).

The procedure from a computational point of view consists of a few basic steps. To start, all the input files are read: starting model, field data, geometry and parameter files, and sources. Using these input files, the forward and backpropagated residual wavefields are computed and correlated to obtain gradient expressions for the model parameters. The gradients are used to calculate the step-length, and then the model is updated. These instructions are repeated iteratively until the number of desired iterations is reached and the final model is written. Figure 4.1 illustrates the structure of the algorithm. Most of the instructions are straightforward to implement, but due to computational limitations some of them need to be approached carefully.

The number of sources in the problem determines the number of forward problems to be solved, specifically three for each source at every iteration. The first one consist of computing the forward wavefield corresponding to the starting model, then the backprop-

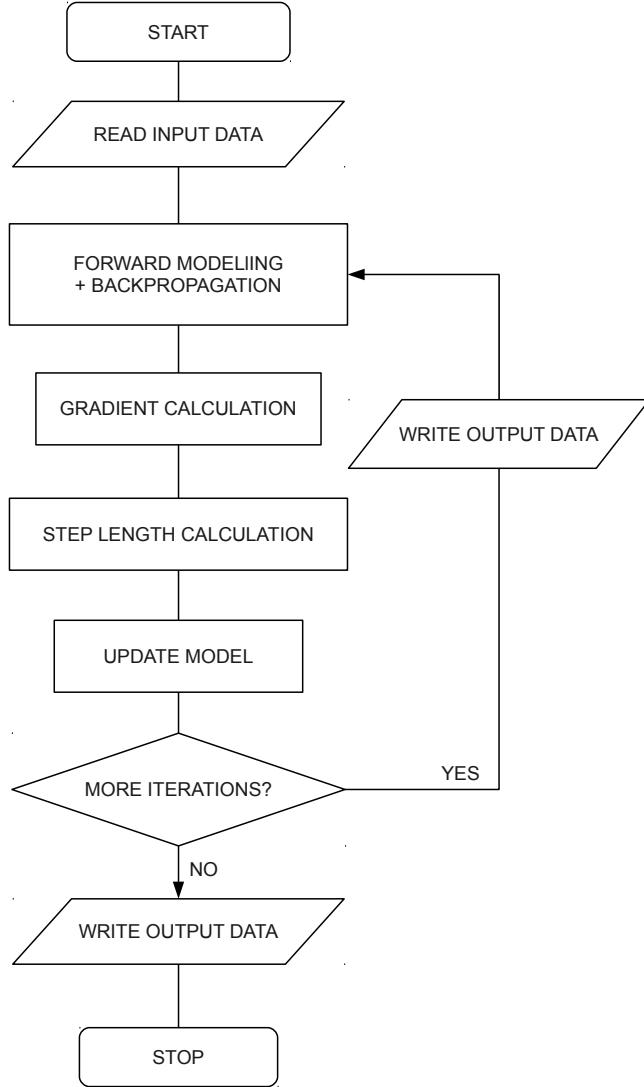


Figure 4.1: Flowchart diagram of the inverse problem for any wave equation formulation.

agated wavefield is computed, and the third provides the optimum scaling of the updates (step-length). For problems with elevated numbers of sources, the common procedure is to parallelise the code for distributed memory systems or clusters in order to reduce the computational times required to run the program. In the inverse case, tasks are shared in a similar way to that shown in the previous chapter for the forward modelling. Figure 4.2 shows the distribution of tasks between the scheduler or master node and the slaves.

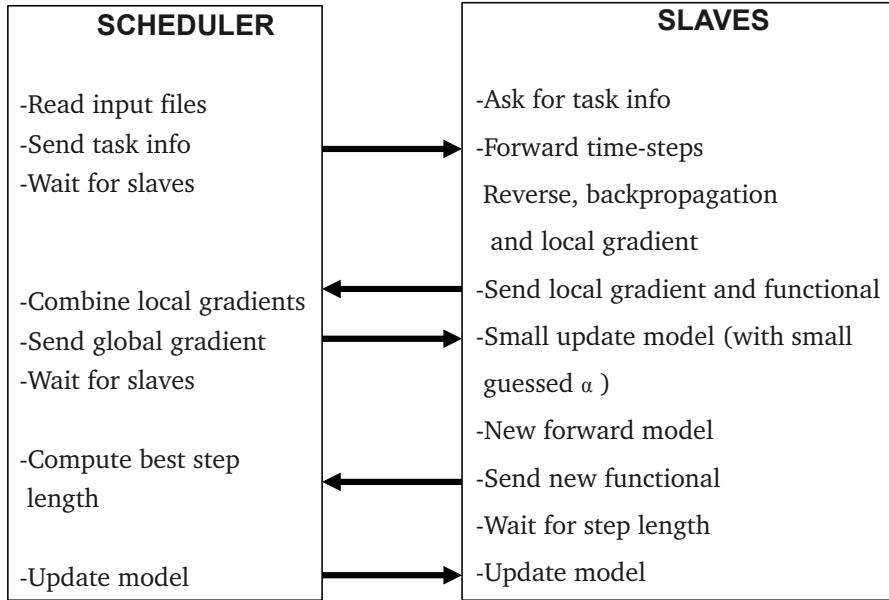


Figure 4.2: Interaction between the scheduler node and the slaves. The former is responsible to distribute tasks across the slave nodes, which bear most of the computational load.

At each iteration, each slave is given a shot along with all the information it needs to calculate it, and then passes the results to the scheduler. After all the shots have been distributed and all the forward, backpropagation and gradient calculations have been computed by the slaves, the master calculates the parameter update by combining all the information. Then the global update is sent to all the slaves and they apply it to calculate the best step-length, which is passed again to the master to compute a global step-length. The model is updated and the loop starts again until the desired number of iterations is reached.

The extra forward calculations to compute the correlation between forward and back-propagated residual wavefields require the wavefields everywhere at every single time-step. A simple, and rather impractical, approach would be to store the forward wavefield and then use it to correlate with the backpropagated wavefield as the latter is computed. An efficient way to do that would be to use RAM memory for the storage, because disk mem-

ory is much slower to access. But the amount of memory required is usually one (if not more) orders of magnitude above the available RAM.

Such memory limitation is not a problem in other domain formulations. The time-stepping nature of the solution is responsible for the gargantuan memory requirement. In the frequency domain, on the other hand, the wavefield is computed per temporal frequency, and its size is only that of the discretised model, which can be well fit in RAM. Fortunately, such constraint can be overcome by storing only a subset of the full forward wavefield as discussed in the next section.

#### 4.4.2 Strategies to reduce storage of wavefields

There are five ways, to the extent of my knowledge, to reduce the amount of information contained in the total forward wavefield  $\phi$  without loss of accuracy.

The first is to exploit the redundancy of information of  $\phi$ . Due to the nature of the Earth's subsurface (which limits seismic velocity values between 500 and 7000  $m/s$  in active source seismic experiments) and the range of frequencies used in FWI, the stability criterion 2.26 combined with the spatial sampling required to avoid dispersion, produce an oversampling of the wavefields because the time-steps are smaller than they need be to capture the range of frequencies propagated. The number of time-steps of the forward wavefield could be considerably reduced to approach the limit imposed by the Nyquist frequency —typically the number of samples in the wavefield could be reduced by a factor of 10. The Nyquist frequency of a time series is defined by

$$\nu_{Ny} = \frac{1}{2\Delta t} \quad (4.53)$$

and it states that the minimum number of points required to represent a particular signal has a lower limit. In the presence of frequencies higher than the Nyquist, an aliased signal is produced, which means that the energy of frequencies higher than Nyquist fold back into the spectrum and get remapped below  $\nu_{Ny}$ .

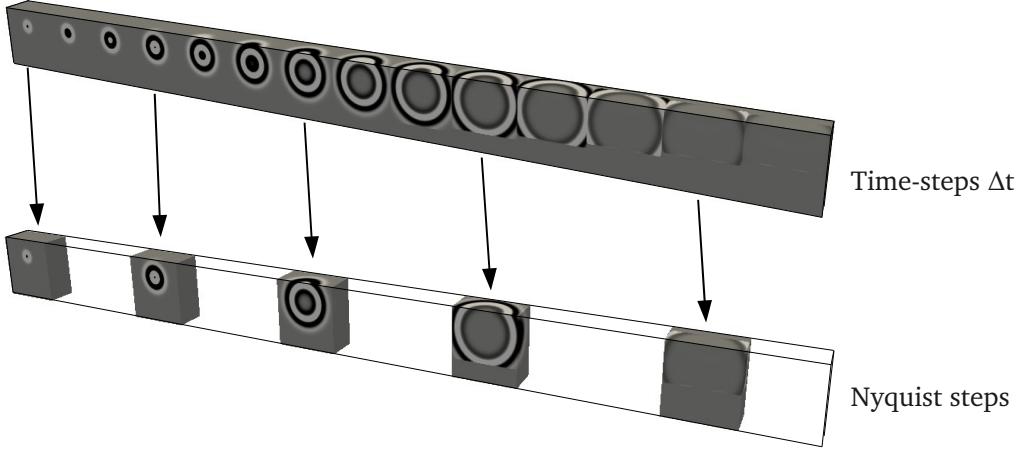


Figure 4.3: All the snapshots of the wavefield (above) are undersampled to reduce memory usage(below).

The forward wavefield can therefore be resampled at a larger time-step interval  $\Delta t_{Ny}$  as long as it complies with 4.53. Figure 4.3 shows schematically this procedure for a homogeneous model.

Another method to minimise the amount of information to be kept in memory is to store the minimum amount of information necessary to recalculate the forward wavefield. This opens the possibility of computing the correlation of forward and backpropagated residual wavefields ‘on the fly’. But the backpropagation —as it name indicates— runs backward in time, and the recalculation should be performed backwards as well. Thus, at each time-step, the correlation (parameter update calculation) reduces to a multiplication of the two simultaneously calculated wavefields added to the existing cumulative parameter update. After the last time-step (corresponding to the first in the forward problem) correlation is computed and added, the update is complete and completely equivalent to that computed storing all the time-steps.

Due to the symmetry of the wave equation with respect to time, with the adequate initial conditions, the solution can be reversed. Given a model and a source, one can calculate the discrete solution of the wave equation. At some point the propagated energy

has expanded and fills the model volume. At this point one can reverse the time direction of the solution and run the problem backwards until the expanding energy has collapsed again. The nature of the numerical discretisation in time will determine the minimum number of time steps required to reverse the wavefield in a consistent way. In our case only the two last time-steps are needed because the time derivative is of first order as well as the FD time operator. Thus, at any point of the time marching scheme it is possible to reverse the wavefield by knowing its values in the last two time-steps. But, in order to limit the size of the computational domain, absorbing boundaries are defined in the faces of the model (with the exception of the free surfaces) to simulate an infinite medium, and therefore part of the energy is lost by damping in the absorbing boundary regions. However, this approach fails to recover the original wavefields when attenuation is considered.

It is not possible to recover the energy lost because of the limited precision of the stored values. Although, in principle it is possible if the floating points used in the computer implementation are big (in bytes) enough to capture the vanishing wavefield with reasonable accuracy, but this would be impractical because memory will grow linearly with the size of the floating points, and the total memory would increase rather than reduce. We define the internal faces as those enclosing the region of the model free of absorption and the externals as the ones defining the actual edges of the model. One can then store  $\phi$  at all time-steps only in the internal faces of the model, and reinject it during the backpropagation steps to recreate the original  $\phi$ , avoiding the absorbing boundary regions (between the internal and the external faces of the model). The internal faces are only planes of nodes and our FD implementation is of 4<sup>th</sup> order in space, and needs values at adjacent nodes to compute the solution at a point. As a consequence,  $\phi$  must be stored not only at the face but in its surrounding points as well. For our spatial FD operator, the two adjacent inner planes for each internal face must be kept as well, as shown in figure 4.4. Under this approach, only the last wavefield need to be kept in memory over all the model, and the stored values at the boundaries are reinjected as new sources to

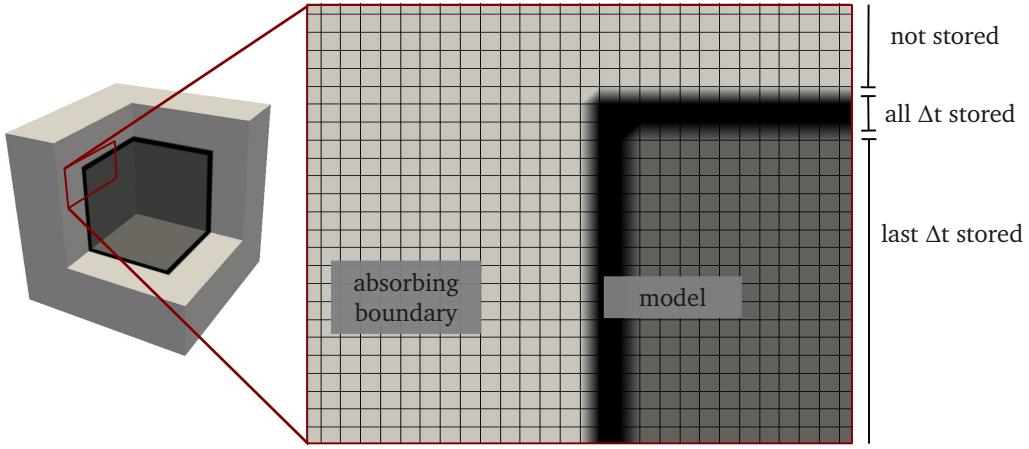


Figure 4.4: The absorbing boundary regions where the wavefield is not stored is depicted in light grey; the actual model, in dark grey, only is stored at the final time-step; the black region is where the wavefield is stored for all time-steps. The thickness of the black region is of two nodes, as the overlaid grid highlights.

regenerate the wavefield backwards in time.

For models containing a few millions of cells or more (the exact number depends on the shape of the volume), it is more efficient to recalculate the wavefield than to adopt the first approach where the full wavefield is undersampled. Most commonly, real data are acquired over volumes that after discretisation are bigger than a few million cells, and therefore this is the strategy implemented in the inversion algorithm.

The third method would be to undersample the wavefield violating the Nyquist criterion and interpolate it to fill the gaps. I have not explored this option in depth due to the low accuracy of a straightforward application and lack of time to examine it rigorously.

The fourth method would be to compress the wavefields and store them either in RAM or in disk memory. Effective compression strategies significantly reduce the size of the required memory to store all the wavefield information. For example, Fourier transforms can be used as a compression tool to reduce the size of the time-domain wavefields by storing only their values for a limited number of frequencies [Sirgue et al., 2007].

Finally, it is also possible to design an absorbing boundary region with scatterers

that act at the required lengths to cover the full bandwidth of the data. The scattering boundaries reflect the energy back to the model in a non-coherent form, hence not affecting the calculation of the correlation of wavefields. Under such approach, only the last few snapshots of the wavefield (the number depends on the order of the FD time operator) are stored and the wavefield is reconstructed by calculating the wave equation backwards in time [Clapp, 2009, Shen and Clapp, 2011].

## 4.5 Conclusions

In this chapter I have described the problem of inverting elastic seismic data and the different existent options to attack the problem. The problem is posed as the minimisation of an objective function which describes the misfit between field and modelled data. The underlying reason not to attempt a solution of the problem using a more direct approach is the impossibility to invert the wave equation operator  $G$  in  $G\mathbf{m} = \mathbf{d}$ . To minimise the objective function, the only attainable option is to rely on local optimisation methods due to the unaffordable computational cost of global methods. In particular the method of choice is a steepest-descent method combined with an optional preconditioning given by the diagonal of the approximate Hessian.

I have presented the mathematical development of the elastic full-waveform inversion equations using two different approaches. Despite apparent differences in the resulting expressions I have proven that they are completely equivalent and that the differences in formal treatment of the physical concepts do not affect the outcome of the problem. The matrix algebra development is more flexible in terms of wave equation parameterisations: once the formal development is complete, a change of equations is easily incorporated and new expressions are found with the only difficulty of finding an appropriate change of variables that makes the matrix operator  $\mathbf{A}$  symmetric. In the functional analysis development, a change of parameterisation has an impact on the early stages of the development and the derivation of new expressions is more complex. For example, if

particle velocities are used instead of displacements some of the terms involved will have to be integrated over time.

The translation of the mathematical concepts into computer language needs a careful analysis of the algorithm design. Leaving aside the computational times required to run realistic models—which are not too critical, the main limitation is the unrealistic amount of memory that a straight implementation would require. The adjoint-state method requires the correlation of two different wavefields that cannot be computed simultaneously because the sources of one of them depend on the other. The storage of the first wavefield in memory is impossible unless disk memory is used in the process, but I/O to and from disk would slow down the process to the point that the run-times would become impractically large. I propose two solutions to this problem both involving the storage of a subset of the initial wavefield. The first one is to undersample it in time at all points of the model and this is a valid and efficient approach when the size of the model is limited. Our choice, however, is to store the values of the wavefield in the inner boundaries of the model and reinject it during the calculation of the second wavefield, which is more efficient for bigger models. Since small models do not pose a memory problem, our choice is a more practical approach.

# 5. Pure transmission datasets

To examine the capability of the FWI algorithm, I have run a series of tests with synthetic models and different acquisition geometries. In experiments with synthetic models, a known model is used to generate synthetic field data to be inverted. The first part of the chapter explores the quality of the recovered models in transmission experiments where sources are on one face (or two) of the model and data is recorded at the opposite face. This provides a good illumination of the target and decreases the non-linearity of the inversion algorithm by using only transmitted data. In all the experiments presented both  $V_P$  and  $V_S$  were recovered simultaneously.

## 5.1 Introduction

In the first set of tests I have tried to generate an optimal framework to run elastic FWI: the first test model is simple, the computational workload is kept low, and the source-receiver distribution is designed to highlight the contribution of transmitted data. I have chosen two models previously tested by my research group on 3D acoustic FWI: a cube with eight cubic anomalies inside [Warner et al., 2007b] and a perturbed 1D gradient model with two channels [Stekl et al., 2007]. The inverted parameters are seismic velocities  $V_P$  and  $V_S$ , hence the parameter updates are computed using expressions 4.15 and 4.16. No spatial preconditioning—approximate diagonal Hessian—is applied because the sources are plane waves, and therefore the amplitude does not decay with the distance as happens with point sources.

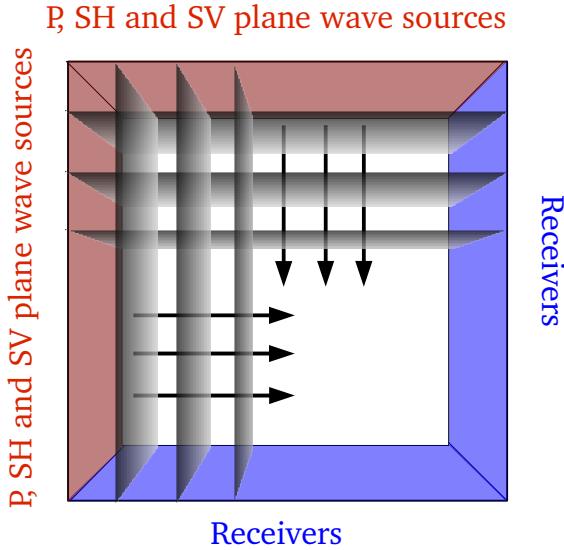


Figure 5.1: Spatial distribution of sources and receivers. There are six plane wave sources (red) recorded only in their opposite faces (blue). The grey planes in the model represent the shape of the waves' path, and the arrows its direction.

These realisations are a proof of concept to demonstrate the feasibility and validity of the method. They were conducted at the early stages of my research project and the software used to run the inversions was a prototype not parallelised for distributed memory systems. Therefore, the number of iterations is reduced to a minimum to limit CPU-times when running on a regular desktop computer. Besides, I did not realise of the importance of the parameterisation of the forward modelling, and I just converted the stresses to particle velocities in the parameter updates 4.45 and 4.46 given by Tarantola and Mora in [Tarantola, 1986] and [Mora, 1987]. I realised my misconception of the problem later on, but I want to show the results obtained with the wrong expressions to assess its impact and help me explain why other people have managed to achieve good results with such incorrect expressions.

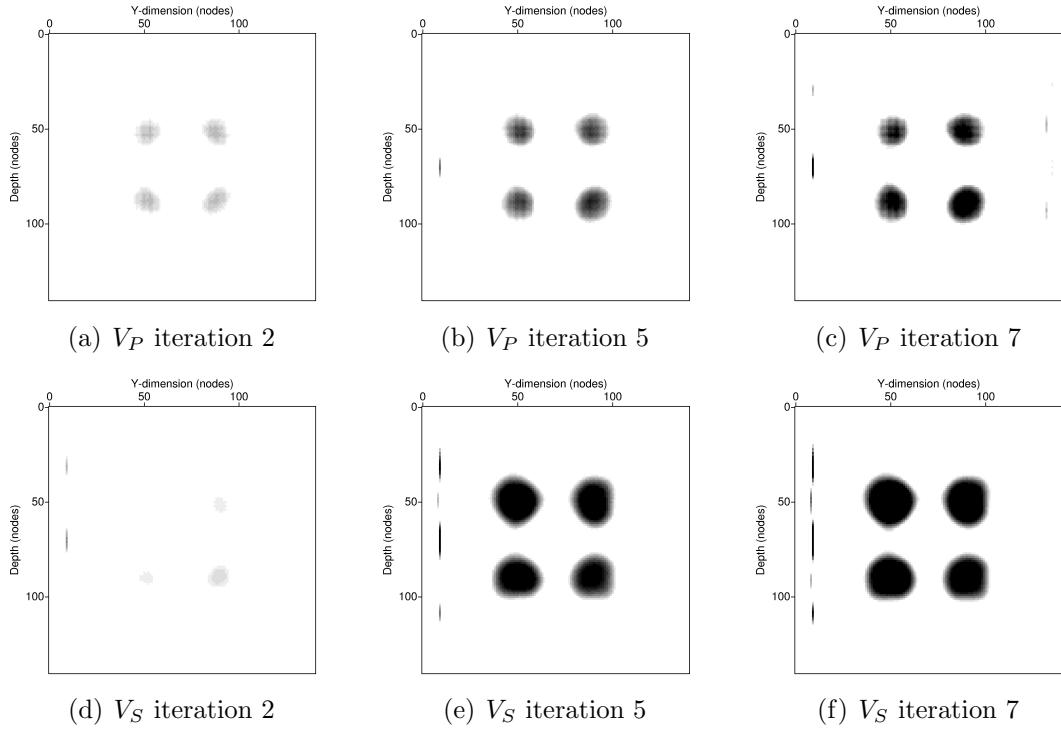


Figure 5.2: 2D slice of the model showing the evolution as iterations proceed. The The colourscale is black and white with the minimum and maximum values matching those of the true model.

## 5.2 Eight cubes model

The eight cubes model is a homogeneous cube of dimensions  $140 \times 140 \times 140$  nodes with eight cubic anomalies of size  $20 \times 20 \times 20$ , where the grid spacing is 20 m. The background velocities are 2500 m/s and 1440 m/s for  $V_P$  and  $V_S$  respectively, the anomalies have 5% higher velocity values and 30 PML cells surround the model. The  $V_P/V_S$  ratio is kept constant in all the model and equal to  $\sqrt{3}$ . The source signature is a ricker wavelet with a peak frequency at 10 Hz. The medium is illuminated using P- and S-wave plane sources (as pictured in figure 5.1), with two shear sources per face oscillating perpendicular to each other. Each source is fired independently and recorded only in the opposite face of the cube so there are six sources in total.

The sources are implemented as particle velocities, hence to generate plane waves all the nodes in the face should be excited simultaneously. For the pressure sources all the nodes contained in the corresponding plane are excited. The generation of shear plane

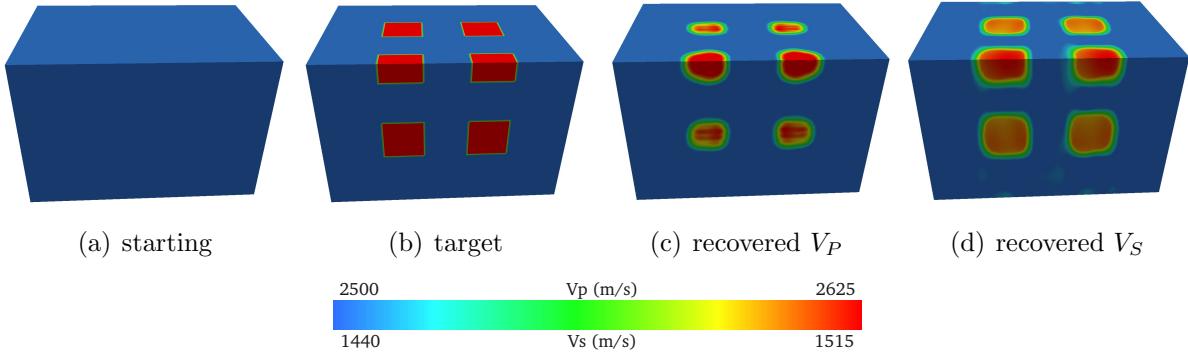


Figure 5.3: Cutaways of the the starting (a), target (b) and recovered models for  $V_P$  (c) and  $V_S$  (d).

waves, though, is not achieved by exciting all the nodes in the plane because, as shown by figure 2.6, the values would cancel each other and only the edges will have non zero values. For the first S-wave source all the nodes in one of directions of the plane are excited but in the other direction only every second node is excited. A  $90^\circ$  rotation of this pattern provides the other shear plane wave source. Source and receiver planes are not at the very edges of the model but buried 5 nodes deep to avoid artifacts from the absorbing boundaries. For example, the P-wave source located at the top of the model is generated by inserting the same source signature to the vertical component of the particle velocity (only one of the grey circles in figure 2.6).

The inversion is set to perform 10 iterations and starts from a homogeneous velocity model. At each iteration the  $V_P$  gradient is a combination of the two plane P-wave sources and the  $V_S$  is a combination of the four S-wave sources. As iterations proceed the anomalies grow in amplitude showing the solution space is navigating towards the global minimum (figure 5.2).

The asymmetry in the models is due to the different amount of energy involved in the gradient calculation at different model regions. The superposition of the plane waves travelling in the model generates a pattern that leaks into the gradient, this is why the top right and bottom left anomalies have the same amplitude and a  $-45^\circ$  symmetry axis. The results for this experiment prove that the algorithm is recovering the target model successfully. Figure 5.3 shows the starting, the target and the recovered models.

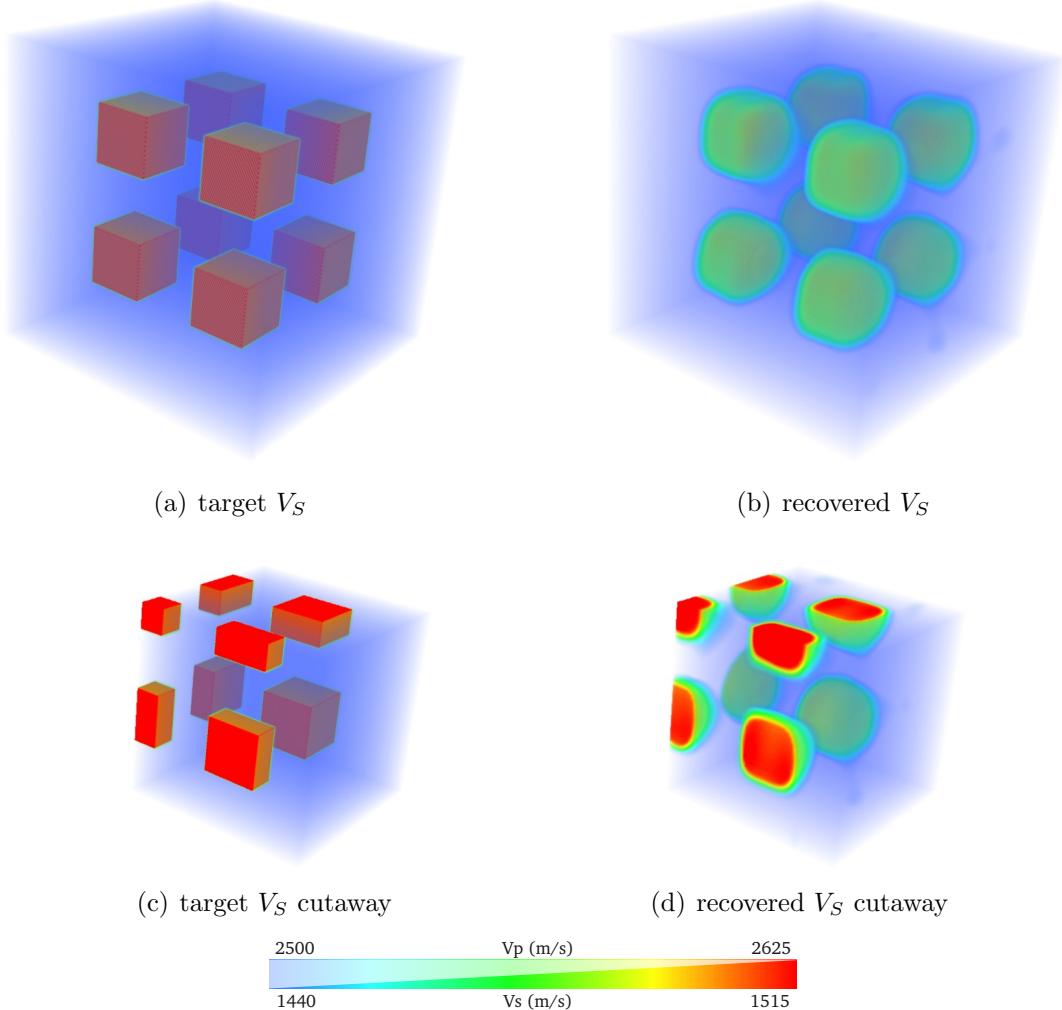


Figure 5.4: Transparency views of target (a) and recovered  $V_S$  (b). The cutaways show a good match of amplitude in the recovered anomalies in the middle and a smooth transition to the background velocity in the edges.

Note that recovered anomalies are not as sharp as the target model due to the lack of higher frequencies in the source and to the lack of short aperture angles. As figure 5.4(b) shows, the anomalies have blurry edges and the correct amplitude is only achieved in the central regions. Besides, the recovered shear velocity model is better resolved due to two factors: the wavelength of the shear waves is smaller at the same frequency —because the propagation velocity is smaller; and the medium is illuminated with four S-wave and only two P-wave sources.

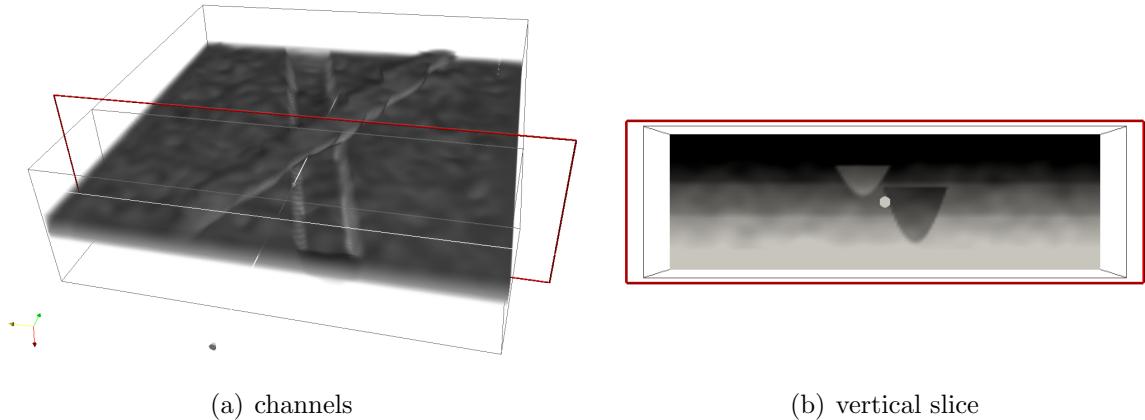


Figure 5.5: Geometrical view of the model features: (a) to highlight the channel structure, transparency is applied to the lower and higher values; (b) 2D vertical slice of the model exhibiting the opposite nature of the channels velocities: the shallower has higher and the deeper has lower velocities than the surrounding background velocity field.

### 5.3 Two channel model

The next experiment with transmission data involves a more realistic model (from a geological standpoint) with a non-homogeneous background velocity and two channels. The target model geometrical distribution of velocities is portrayed in figure 5.5. The minimum and maximum values for  $V_P$  ( $V_S$ ) are 2000 m/s (1154 m/s) and 2800 m/s (1732 m/s) respectively. The size of the model is  $160 \times 160 \times 51$  and the grid spacing is 13 m.

This test represents a further challenge to the FWI algorithm for two reasons:

1. The background (model without the channels) velocity in the target model corresponds to a perturbed 1D gradient and the starting model is a plain 1D model with no perturbation. The starting model is generated by taking the first trace of the model and expanding it to fill all the volume. The starting model, then, is not the same as the target without the anomalies as in the eight cubes case, but has different values at every point (with the exception of the first trace).
  2. The medium is illuminated from only one plane and recorded only in the opposite face of the source (figure 5.6).

The channels are not present in the starting model and the goal of the inversion is to recover them. The source signature is the same used for the previous case as well as the

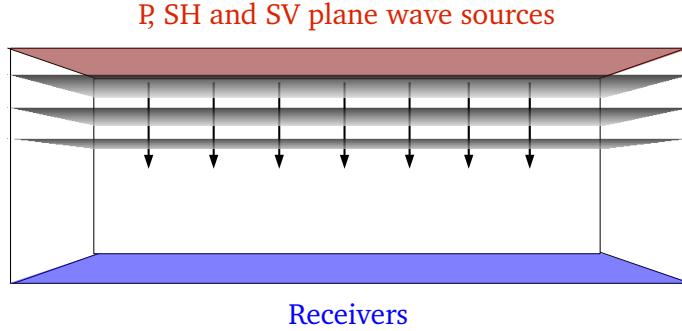


Figure 5.6: Spatial distribution of sources and receivers (see figure 5.1 for the description of colours, planes, etc)

peak frequency and the thickness of the PMLs, and the ratio of the seismic velocities is also  $V_P/V_S = \sqrt{3}$ .

The inversion performed a total of five iterations. The results are shown in figure 5.7, where as before, the model has been illuminated with two shear and only one pressure plane waves, hence the higher saturation in the channel colours in 5.7(d).

## 5.4 Conclusions

As mentioned previously, these tests are a proof of concept, and the goal is to be able to recover the target models to a certain degree of accuracy to demonstrate the algorithm is responding well. As it will be shown in the next chapter, the accuracy of the recovered models is significantly improved by using the correct model update expressions, even when the geometry of the experiment is not optimal —sources and receivers placed in the surface of the model only.

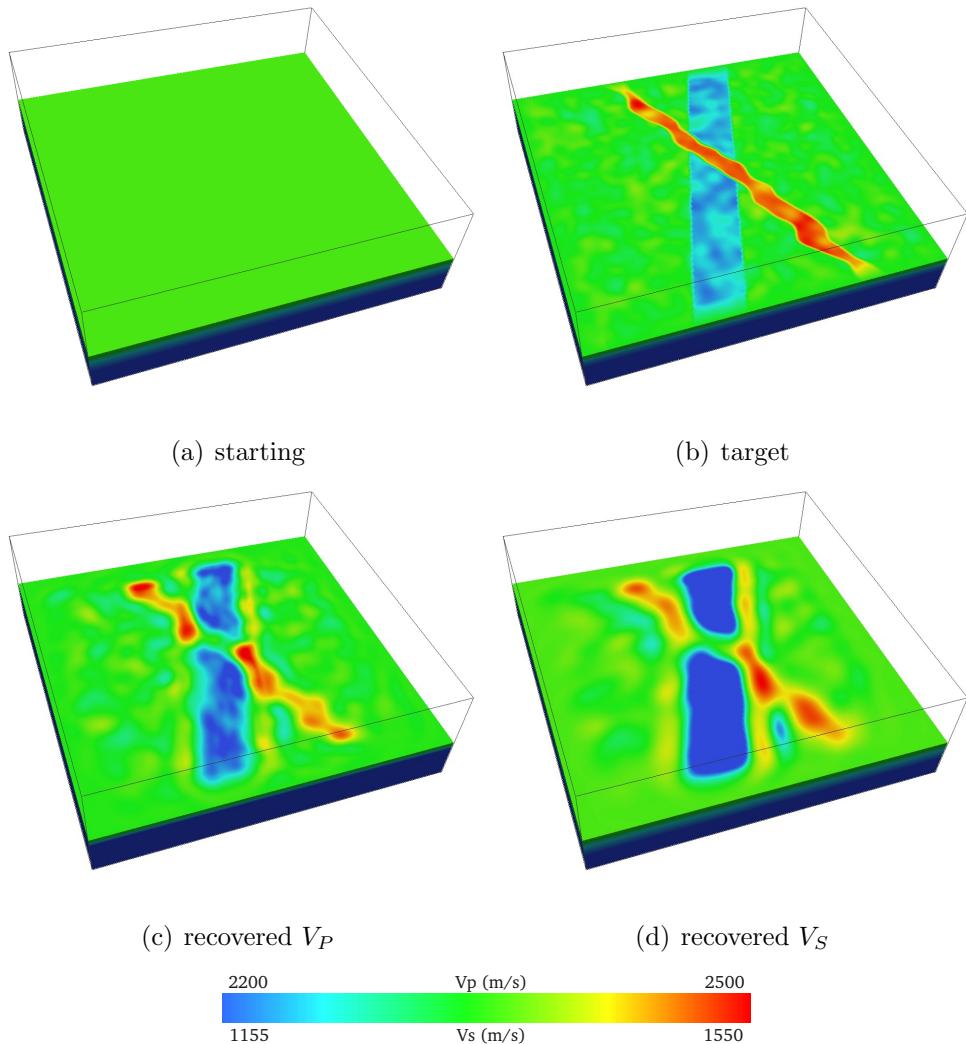


Figure 5.7: Horizontal view at a depth of 260 m of: (a) starting model, a 1D velocity gradient; (b) the target model; (c) recovered  $V_P$  after five iterations; and (d) recovered  $V_S$  after five iterations. Note that the colour scale has been clipped to brighten the channels.

# 6. Surface acquisition

In this chapter, I present the results of tests similar to those in the previous chapter, but with acquisition geometries that restrict the positions of sources and receivers on the shallow part of the models, to analyse the efficiency of the algorithm in real surveys. In all the datasets, both P-wave and S-wave velocities were inverted simultaneously.

## 6.1 Introduction

Real acquisition surveys, in general only acquire data at the surface of the target volume, with the exception of walkaway VSP (vertical seismic profile) or cross-hole experiments, which will not be investigated. In order to demonstrate the potential of 3D elastic FWI, synthetic experiments with all sources and receivers placed in the shallow part of the model must successfully recover the subsurface properties. This represents a significant reduction in the linear response of the problem compared to the previous transmission experiments. In addition, I examined the effect of the Poisson ratio on the inversions by comparing the results for models with constant and non-constant  $V_P/V_S$  ratios.

In the following sections, all sources and receivers are placed in the shallow parts of the models. The software employed to perform the inversions has all the correct expressions for the gradient and parallelisation is fully implemented at all levels. The parameters inverted are slownesses instead of seismic velocities, however little differences are observed in the final images with both parameterisations. The gradients are preconditioned using the approximate diagonal Hessian to normalise the wavefield's amplitude. Another important

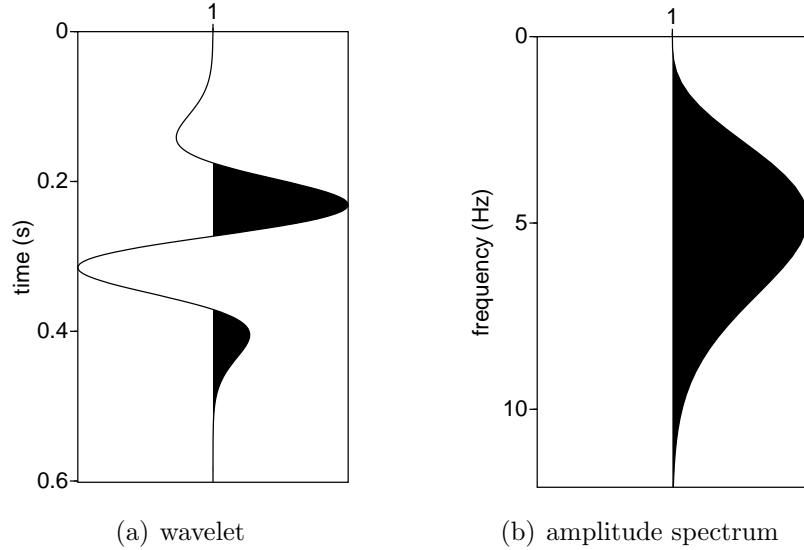


Figure 6.1: Source signature and amplitude spectrum.

difference with respect to the transmission experiments is that not all frequencies are included in the inversion process. Instead, the total number of iterations is divided into blocks of increasing frequency content. At each iteration block, the wavefields are filtered with a Gaussian function which has a peak at the frequency of interest. As iterations proceed and the inversion moves to the next block, the Gaussian translates its peak to the next frequency and broadens to keep part of the low frequency energy. For the sake of simplicity, hereafter I will refer to blocks of frequency  $x$  Hz as just frequencies, where  $x$  is the central frequency of the Gaussian filter.

Finally, only pressure sources —explosions— are used and only pressure is recorded in the receivers. This represents an important restriction in terms of the amount of available information that is inverted, but most seismic acquisitions operate under these acquisition parameters. Therefore, to demonstrate the potential application to field data, FWI it must be able to handle this limitation.

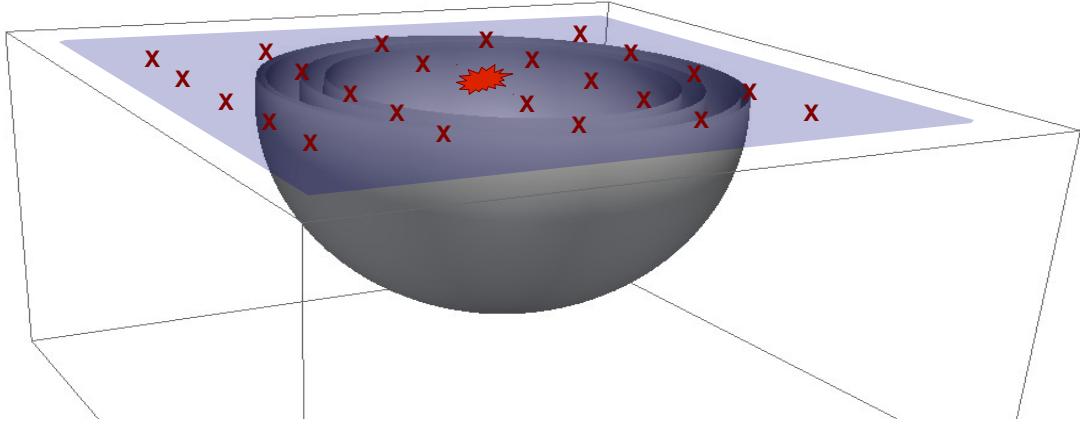


Figure 6.2: Experiment geometry depicting all 25 sources in red and 8100 receivers in blue. The shape of the wavefield in this case is spherical as shown by the grey isosurfaces which correspond to a constant amplitude values of the wavefield for the different shaped source.

## 6.2 One cube validation tests

Elastic imaging of homogeneous models with homogeneous anomalies using only pressure sources and receivers poses an important problem: shear-wave energy is not generated in the sources and is not recorded in the receivers. P-S conversions are the main source of information to image  $V_S$ , but they are not recorded in the pressure receivers making it difficult to image the elastic properties of the target models. As a consequence, the main source of information to recover the elastic properties of the medium comes from the P-S-P —and higher order— conversions that occur within the anomalies as internal reflections. However, the AVO response of the P waves depends on  $V_S$ , which helps to mitigate the lack of suitable converted waves.

I use a very simple model —a cuboid with a cuboid anomaly— to evaluate the response of the inversion with respect to the coupling between  $V_P$  and  $V_S$ . Due to the interleaved nature of the data and the model parameters, the solution spaces corresponding to each seismic velocity are not independent and some cross-talk is expected.

The model dimensions are  $101 \times 101 \times 51$ , the anomalies are  $10 \times 10 \times 5$  and the grid spacing is 20 m (6.3). The anomalies are in the centre of the XY plane and at a depth of

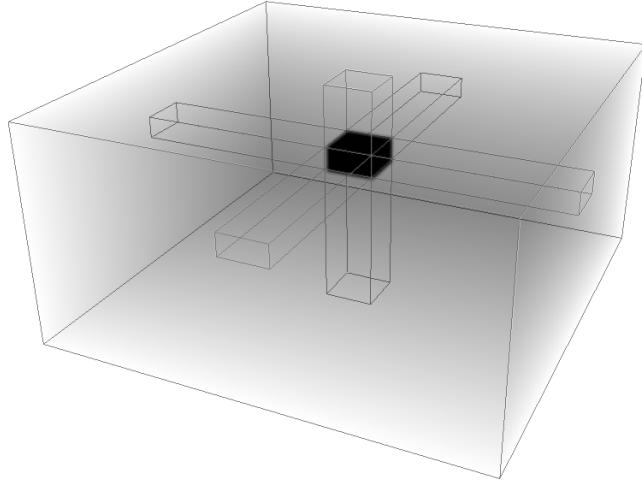


Figure 6.3: The anomaly coordinates (given in nodes) are:  $x \in \{45 \dots 55\}$ ,  $y \in \{45 \dots 55\}$  and  $z \in \{14 \dots 19\}$ .

300 m. The model is surrounded by 15 absorbing cells, but five of them are within the model except for the top face —where sources and receivers are placed, which has the 15 extra cells added on top of the model so both forward and backpropagated wavefields are not damped in their originating points. There are 25 sources and 8100 receivers, both homogeneously distributed at a depth of 60 m as shown in figure 6.2. The inversion is designed to increase the frequency in six iteration blocks of 5 frequencies each, thus a total of 30 iterations are performed overall.

The source signature for all the shots is a ricker wavelet differentiated with respect to time with peak frequency after differentiation of 5 Hz (Figure 6.1). The starting models are homogeneous cuboids with the correct background velocities. A total of ten realisations with different  $V_P/V_S$  ratios are conducted for the models listed in table 6.1. The first column of the table indicates the name of the model, the second and third indicate the amplitude contrast with respect to the background for both seismic velocities. The last two columns report the success of the inversion from a qualitative point of view. The first two are acoustic inversions of data generated with the acoustic wave equation and they serve as a benchmark to compare the quality of the elastic cases. All possible combinations corresponding to the fixed perturbation criteria  $+/- 12.5\%$  are executed. The results are

Table 6.1: One cube models

model name	$V_P$ anomaly	$V_S$ anomaly	inverted $V_P$	inverted $V_S$
acoustic-plusvp	+12.5%	NA	correct	NA
acoustic-minusvp	-12.5%	NA	correct	NA
plusvp-novs	+12.5%	0	correct	correct
minusvp-novs	-12.5%	0	partially correct	correct
novp-plusvs	0	+12.5%	correct	incorrect
novp-minusvs	0	-12.5%	correct	incorrect
plusvp-plusvs	+12.5%	+12.5%	correct	correct
plusvp-minusvs	+12.5%	-12.5%	correct minor artifacts	incorrect
minusvp-plusvs	-12.5%	+12.5%	correct major artifacts	incorrect
minusvp-minusvs	-12.5%	-12.5%	correct	correct

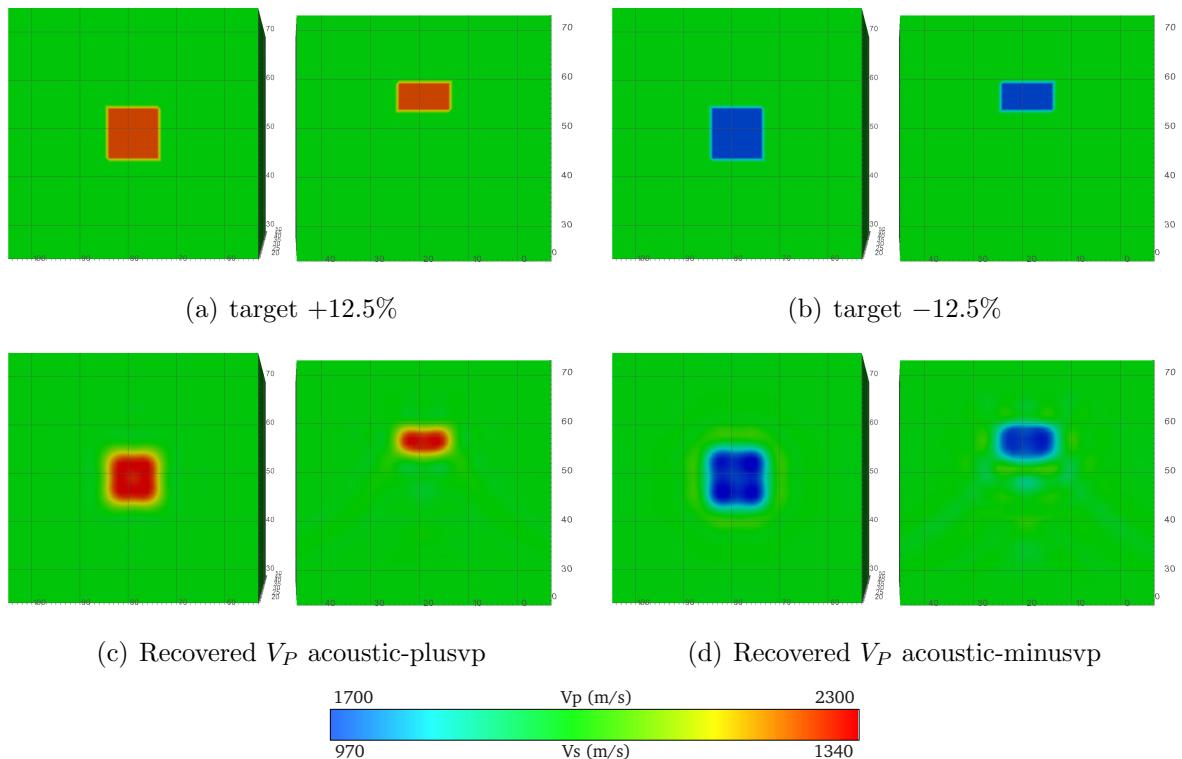


Figure 6.4: Vertical and horizontal slices of the target model for both positive (a) and negative (b) anomalies. Recovered models for the purely acoustic cases after 30 iterations (c) and (d). The  $x$  and  $y$  directions in the plot are equivalent due to the symmetry of the experiment.

shown in figures 6.4, 6.5 and 6.6; the slices are at  $z = 17$  ( $XY$  plane) and  $x = 50$  ( $YZ$  plane) nodes.

In the purely acoustic case, the inversion successfully recovers both positive and negative anomalies (figure 6.4). Assuming that in synthetic experiments a purely acoustic inversion outperforms I will use these results as a benchmark to compare the elastic results.

The inversion results for the models with only one of the two seismic velocities perturbed are shown in figure 6.5. It is interesting to observe how the shear wave velocity cannot be imaged properly in absence of  $V_P$  heterogeneities (see (e-h) in figure 6.5). The model updates for  $V_P$  obey the relation 4.17, which only depends on normal stresses, but the model updates for  $V_S$  (relation 4.18) depend both on normal and shear stresses. Thus, when the target  $V_P$  is homogeneous, the update for  $V_S$  have only contributions from the shear stresses which seems to drive the direction of the updates away from the global minimum. In this case, the local minimum in which the recovered shear velocity models falls seem to be complementary to the recovered  $V_P$  models (this effect is especially clear in the vertical slices) which suggests that the solution space for both realisations has some symmetry. The incorrect recovery of the shear-wave velocity models can be explained by the lack of P-S-P conversions and the nature of the receivers (which not record shear-wave energy), as stated in the beginning of the section.

The results for the cases where both anomalies are present are pictured in figure 6.6. The target models that have the same anomaly polarity (plusvp-plusvs and minusvp-minusvs) are inverted successfully and yield images with the same resolution than the purely acoustic inversion depicted in figure 6.4 and a slightly different shape. When the anomalies have opposite polarity the inversion fails to provide a satisfactory  $V_S$  model and artifacts are introduced in the final  $V_P$  model. Figures 6.6(c) and 6.6(d) show how the P-wave velocity model is not free of artifacts but still updates the perturbation in the correct direction by increasing its value in the model region. However, the S-wave velocity model is updated by modifying the regions of the model where there are no  $V_P$  anomalies, resulting in models that are somehow reciprocal to each other, thus presenting the same behaviour observed in figures 6.6(e) and 6.6(f).

The opposite trend in the updates can be explained by analysing the parameter updates expressions (equations 4.20 and 4.21). Both updates have a term in common, the correlation of the pressure wavefield expressed as the sum of the normal stresses. Moreover, the  $V_P$  updates only depend on this wavefield variable, while the  $V_S$  gradient contains the correlation of other wavefields as well —normal and shear stress wavefields. The conflict arises when this common term is pushing the  $V_P$  and  $V_S$  models in opposite directions, but the shear velocity update can compensate for that in the non-common terms, and this is the reason why it is updating areas outside the anomaly region: because in these areas the compressional velocity is already correct and therefore the pressure wavefield term should be zero, leaving the shear stress terms ‘free’ to introduce artifacts into the  $V_S$  model. A side-effect is that some trade-off occurs due to the artifacts introduced in  $V_S$ , which leak into the  $V_P$  model and produce the complementary patterns in the final models. This cross-talk creates a feedback loop that increase the amplitude of the artifacts as iteration proceeds.

Although the realisations corresponding to non-realistic models may appear unnecessary, it is my firm belief that exploring the limits, even beyond geological consistency, should help us to understand the potentials and pitfalls of elastic full-waveform inversion. In models with more heterogeneous velocity distributions, the scattered energy coming from different points of the volume creates more wave conversions that help improve the imaging of the elastic shear-wave properties, as demonstrated by the better imaged models presented in the next sections.

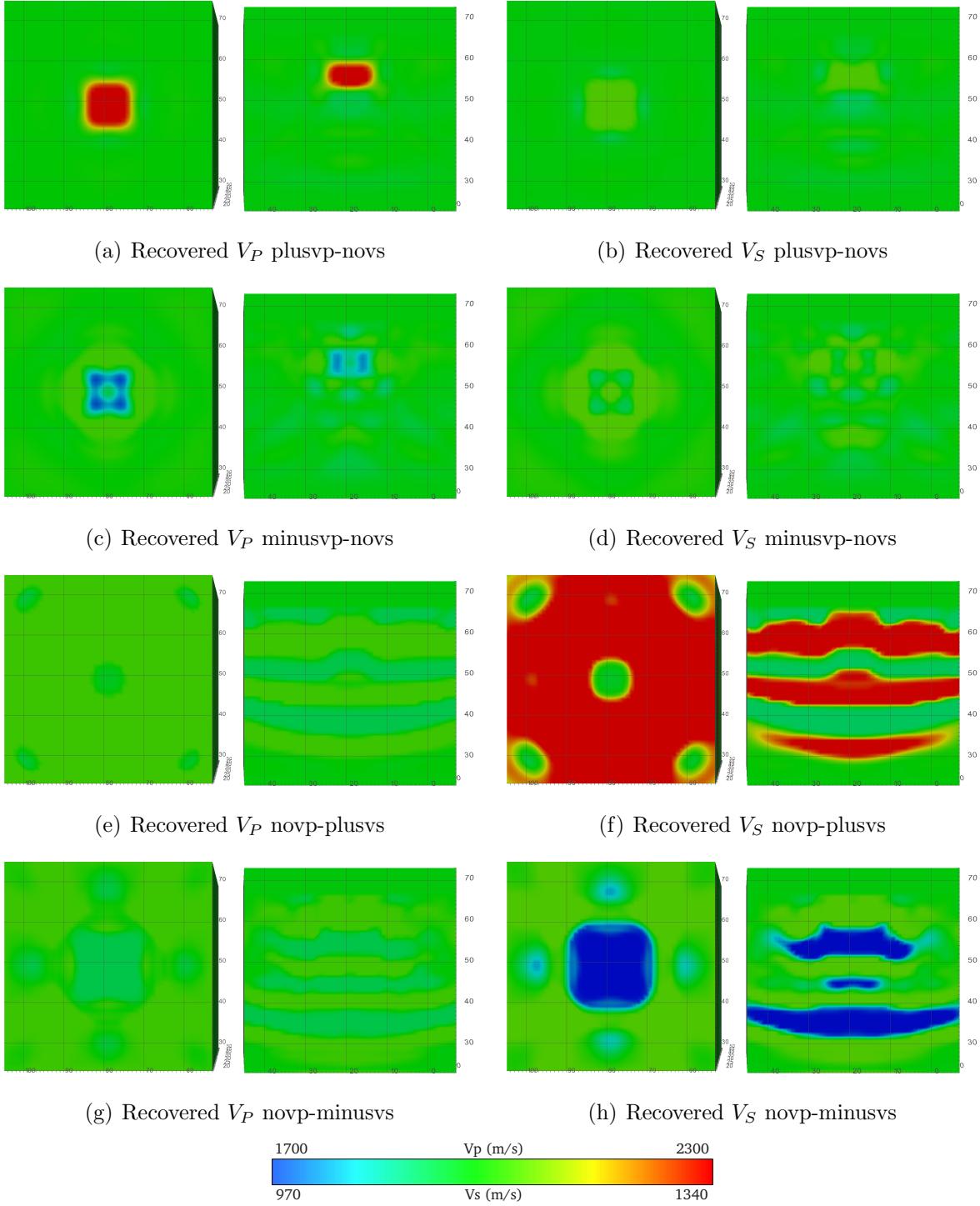


Figure 6.5: A positive anomaly in  $V_P$  is recovered well (a) and there is no trade-off with  $V_S$  (b); a negative anomaly is only recovered partially (c) and there are no leaks to  $V_S$  (d); a positive anomaly in  $V_S$ , while leaving  $V_P$  unchanged (e) as expected, gets stuck in a local minimum and therefore is not recovered (f); the recovered  $V_S$  model for a negative anomaly does not converge to the global minimum either (h), although  $V_P$  remains correctly unchanged (g).

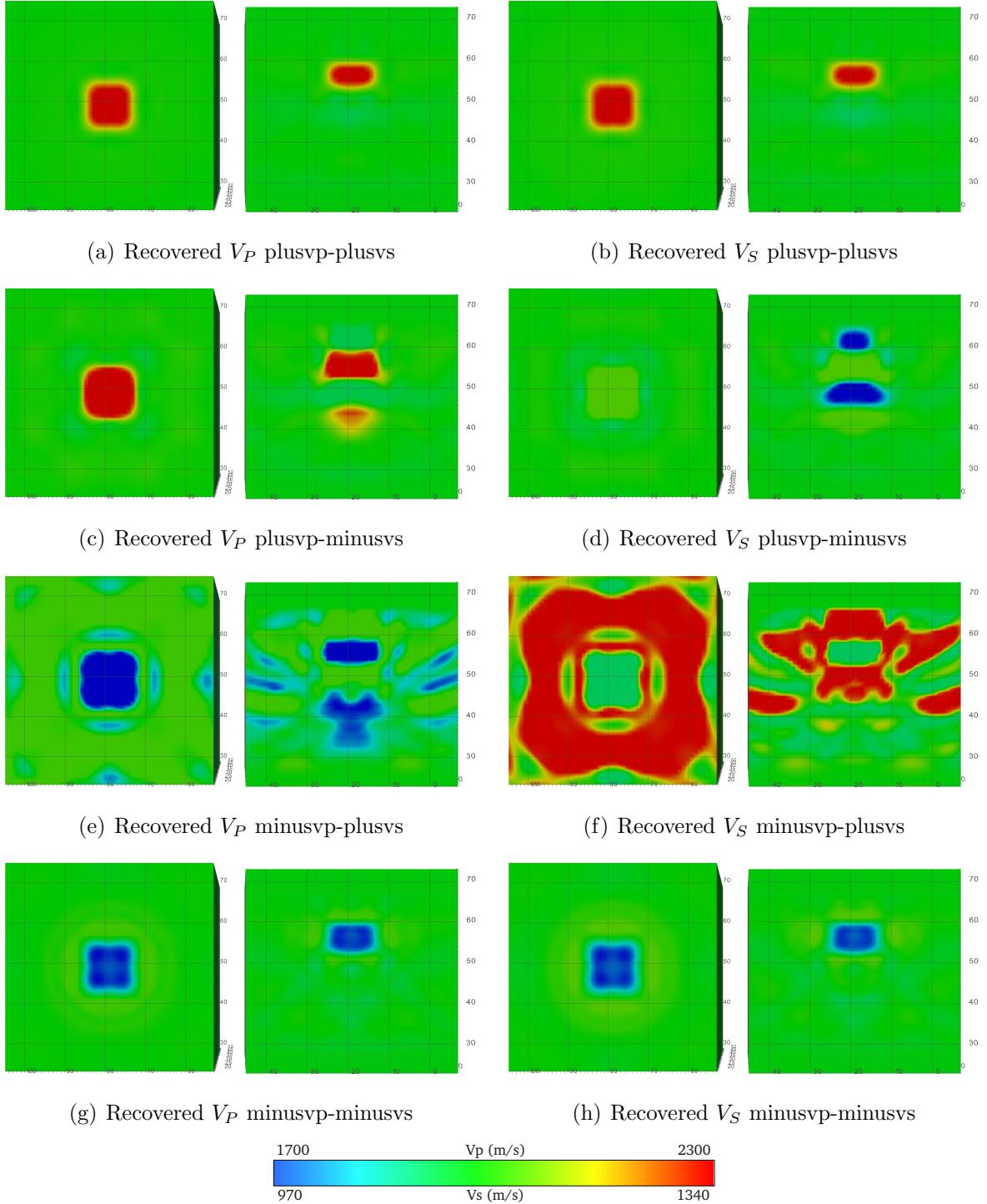


Figure 6.6: Positive anomalies for both seismic velocities are imaged correctly (a,b); when the target has positive  $V_P$  and negative  $V_S$  anomalies, the former is well recovered with some minor artifacts (c) while the latter is not recovered at all (d); the opposite anomaly configuration leads to similar results although in this case the recovered  $V_P$  presents higher amplitude artifacts (e) and the recovered  $V_S$  is still completely wrong (f); a target model with both anomalies negative leads to a correct recovered models for both seismic velocities (g,h)

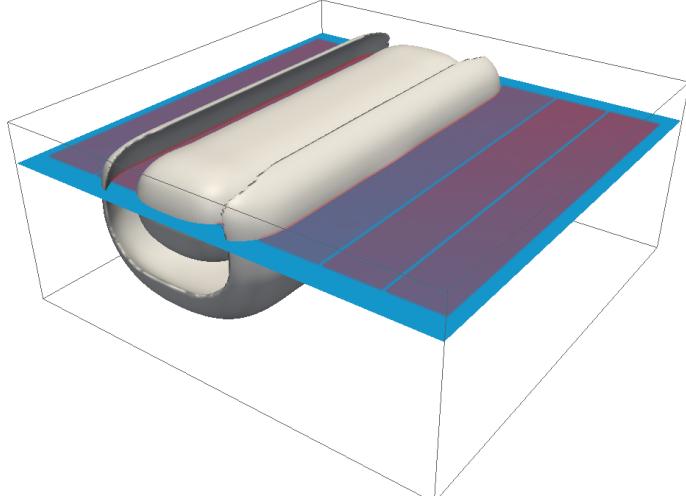


Figure 6.7: Composite sources (red patches) and receivers (blue plane) are located at a depth of 130 m. The shape of the wavefield generated by one of the composite sources is described as a constant amplitude isosurface.

### 6.3 Two channel surface acquisition

I revisit the two channel model introducing a new geometry layout: both sources and receivers are now in a plane buried at a depth of 130 m. The total number of composite pressure sources is seven, each of them containing 2840 simultaneous point sources. The composite sources cover contiguous rectangular areas that cover all the extent of the model in the  $XY$  plane as shown in figure 6.7. The composite shots are approximately perpendicular to the direction of the channels.

The thickness of the PMLs is now reduced from 30 (in the transmission two channel experiment) to 15 nodes per face. The model is padded with 10 extra cells per face except for the top face which is padded with 15, thus the absorbing boundaries start damping the wavefields in the five external cells of the model (with the exception of the top absorbing boundary which does not overlap the model). The source wavelet used is the same as in the transmission experiment —peak frequency at 10 Hz.

I have generated three different datasets using three different versions of the two channel model. Table 6.2 summarises their different features. The inversion increases the frequency content every five iterations from 4 to 16 Hz. The wave equation refers to

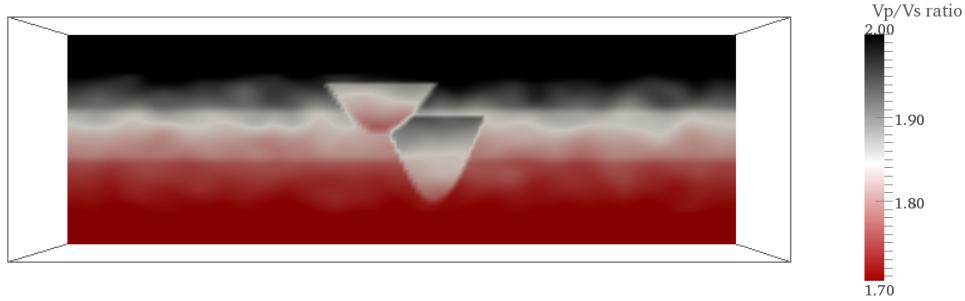


Figure 6.8: 2D vertical slice of the  $V_P/V_S$  ratio for the twoch-elastic model described in table 6.2. The ratio spans from 2.0 in the shallow areas to 1.7 in the deepest parts of the model.

the equation used to generate the synthetic field data and to invert it. As before, the acoustic data serves as a benchmark to compare against the elastic FWI results. The only difference between the two elastic inversions is the  $V_P/V_S$  ratio in the target models: one has a constant ratio and the other has a ratio that varies (figure 6.8) according to

$$V_S = \frac{1}{2}V_P + 0.26(V_P - 2000) \quad (6.1)$$

Table 6.2: Two channel inversions

model name	wave equation	$V_P/V_S$ ratio	iterations	frequencies
twoch-acoustic	acoustic	NA	35	4,6,8,10,12,14,16 Hz
twoch-elconst	elastic	1.65	35	4,6,8,10,12,14,16 Hz
twoch-elastic	elastic	variable	35	4,6,8,10,12,14,16 Hz

The  $V_P$  starting and target models are the same for all three realisations, the starting  $V_S$  models however are not the same in the twoch-elconst and twoch-elastic runs. Both are 1D models generated from the first trace of each target model, and hence are different since the ratios between the seismic velocities are not the same.

Figures 6.9, 6.10, 6.11 and 6.12 show depth slices of the starting, target and recovered models. The three depths are selected to emphasize the successful recovery of the model

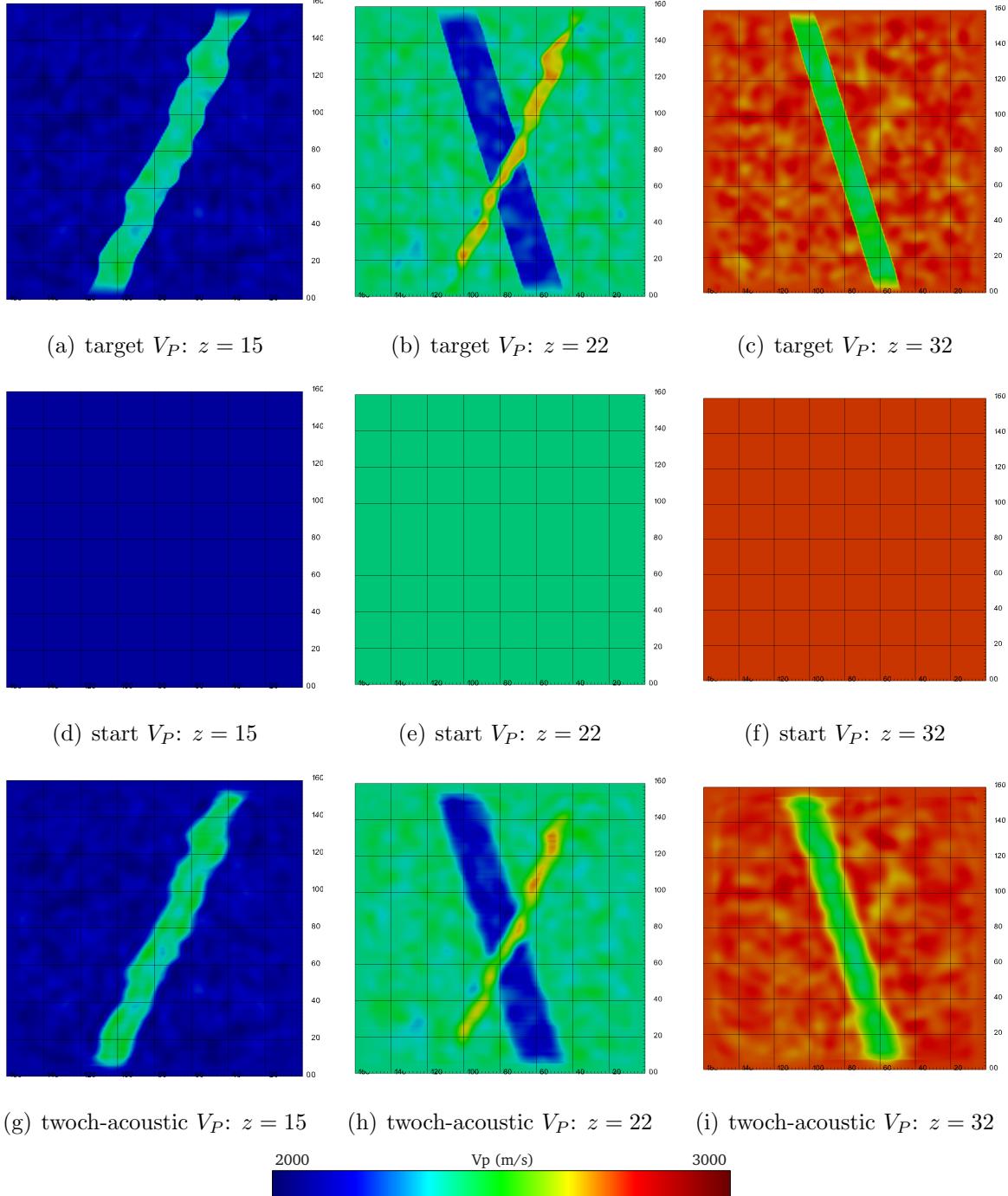


Figure 6.9:  $XY$  planes of the target and start  $V_P$  models (a-f) at different depths given in nodes; same depth slices for the purely acoustic inversion results (g-i).

in the top and bottom channels as well as their intersection. The P-wave velocity fields are duly imaged in all three cases (figures 6.9, 6.10), however the variable  $V_P/V_S$  ratio provides a slightly less accurate image compared to the other two. Note that not only

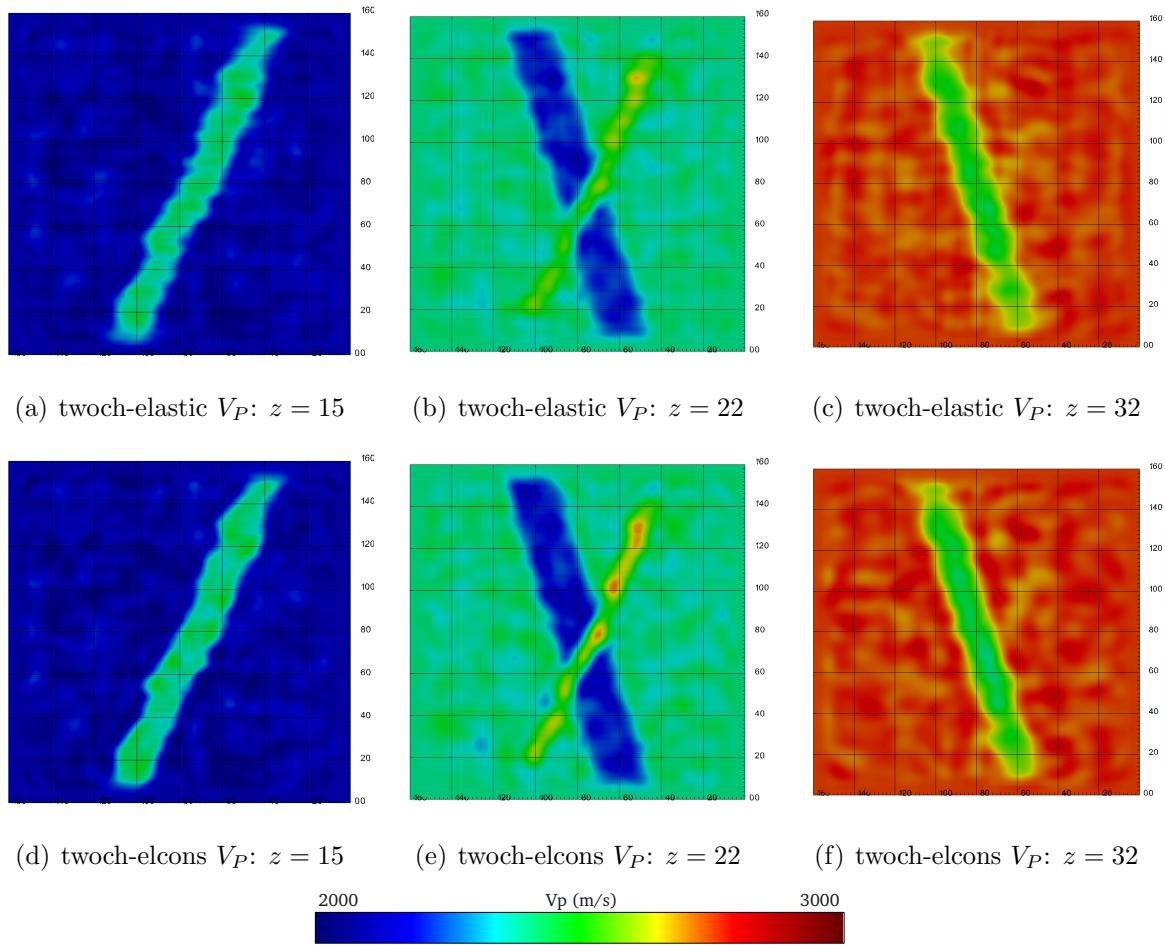


Figure 6.10: Depth slices for the recovered  $V_P$  models corresponding to the inversion of twoch-elastic (a-c) and twoch-elconst (d-f).

the channels are resolved, the background velocity perturbations are imaged with a high degree of accuracy.

The shear wave velocity fields are also well imaged although, as shown in figures 6.11 and 6.12, the images are not as highly resolved as the acoustic ones, especially for the two-elastic model; a closer look at its shallowest slice reveals a faint acquisition footprint in the form of horizontal stripes across the channel —due to their low amplitude they only become apparent when the contrast of velocities is strong enough, that is on top of the channel, leaving the background unaffected by this stripping. Besides, the perturbations in the background are better imaged in the constant  $V_P/V_S$  ratio inversion, which confirms the hypothesis of a better behaved solution space for constant Poisson’s ratio models.

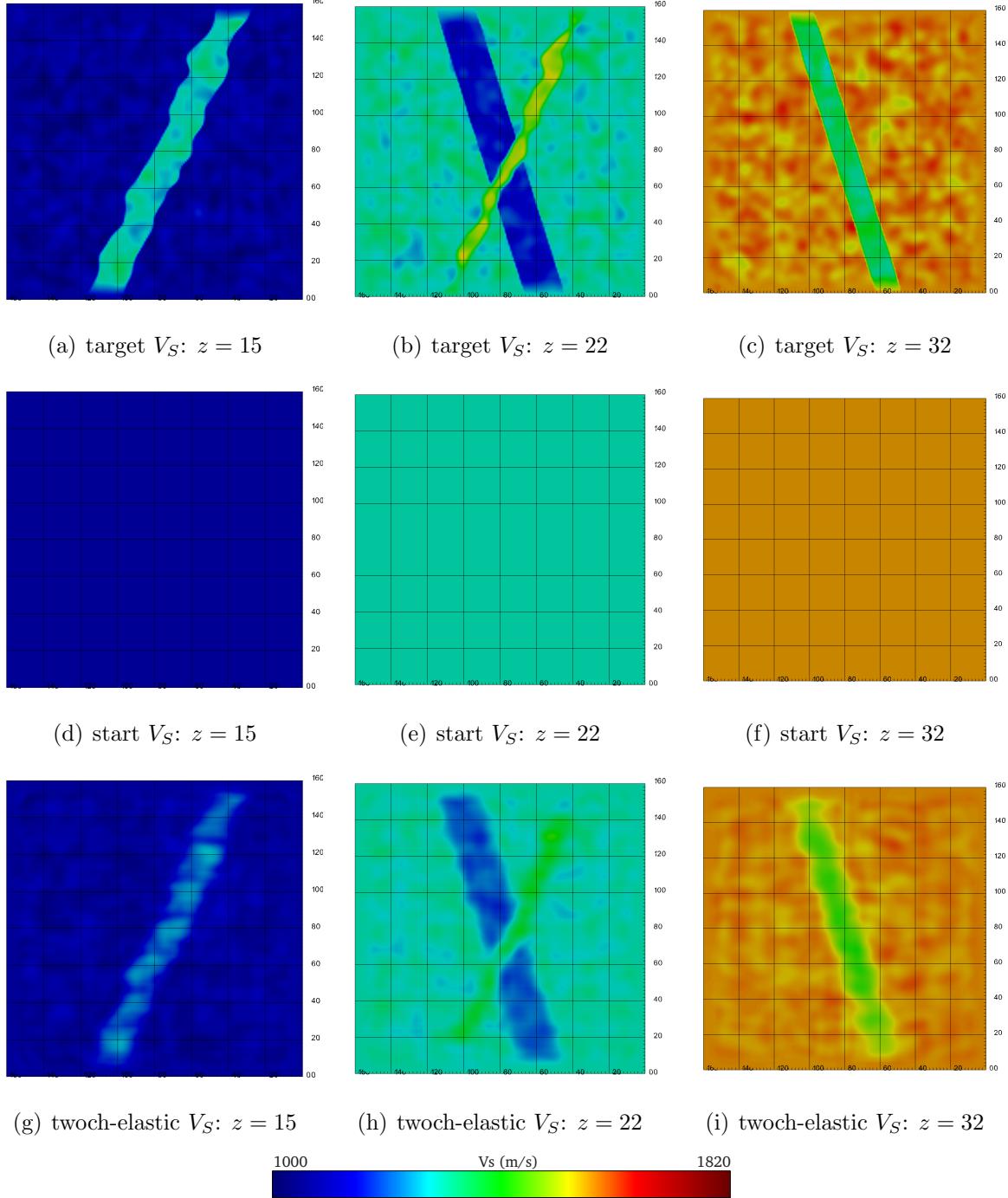


Figure 6.11:  $XY$  planes of the target and start  $V_S$  models (a-f); inversion results for the twoch-elastic case (g-i).

In addition, the absence of shear-wave sources and 3 component receivers restricts the accuracy of the  $V_S$  images, which will be improved if the medium were illuminated with shear waves and they could be recorded. Note how the  $V_S$  images improve with depth,

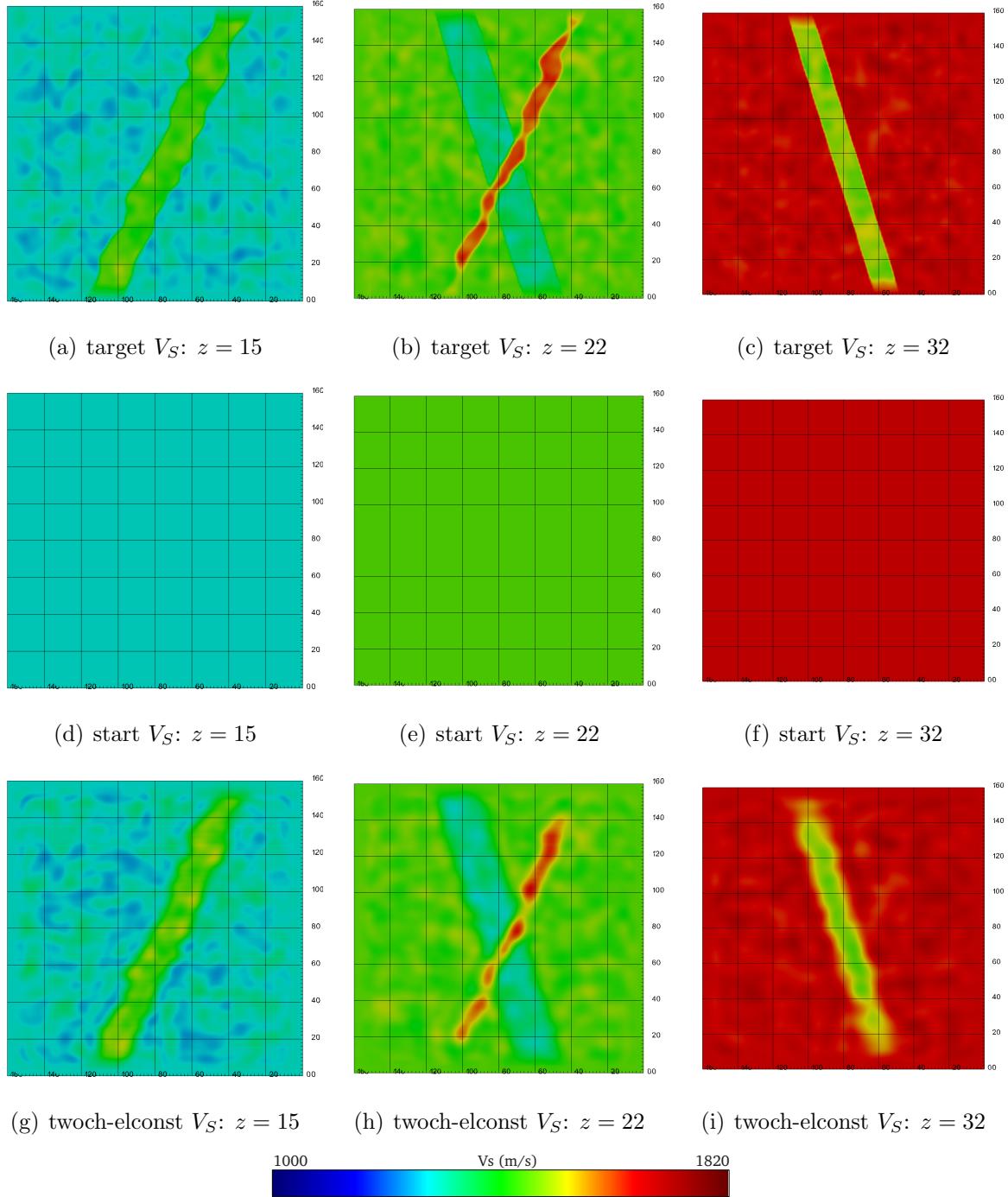


Figure 6.12:  $XY$  planes of the target and start  $V_S$  models (a-f); inversion results for the twoch-elconst case (g-i).

because the waves involved in the imaging of the deeper parts have been subject to more P-S-P conversion than the shallow waves involved in the imaging of the shallow parts (figures 6.11 and 6.12).

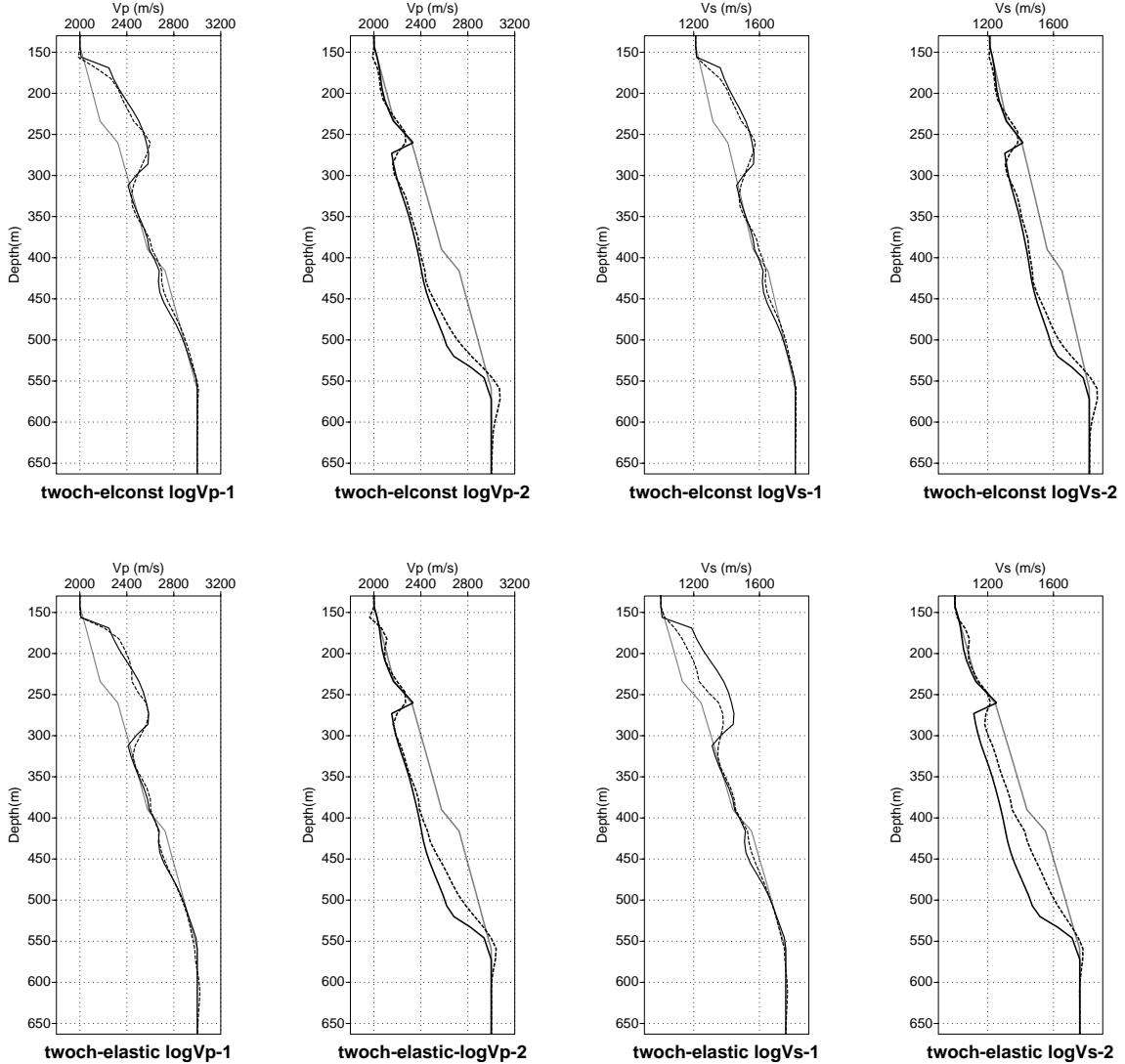


Figure 6.13: Vertical logs for the twoch-elconst and twoch-elastic models at locations: log 1  $x = 1170$  m,  $y = 845$  m; log 2  $x = 715$  m,  $y = 1105$  m. The grey line is the starting model, the black is the target and the dashed is the recovered model.

A more detailed evaluation of the result's accuracy is shown in the velocity logs provided by figure 6.13. The position of the logs in relation to the model is shown in figure 6.14. For comparison purposes, the targets are plotted next to the recovered models. The lower accuracy of the recovered model for the variable  $V_P/V_S$  ratio case becomes apparent in this picture, where the shape of the surface is smoother than the target.

The footprint acquisition can be seen more clearly as well in the form of stripes in the channel. Nonetheless the shape of the channel is well imaged in both cases, confirming

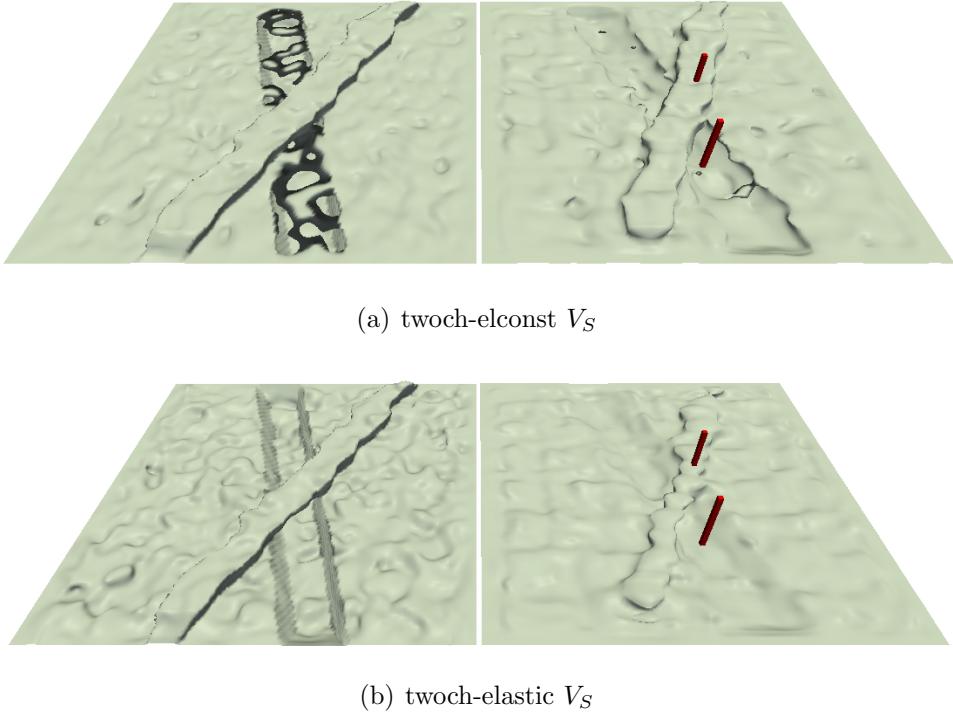


Figure 6.14: Isosurfaces corresponding to a velocity of 1300 m/s for the target (left) and recovered (right)  $V_S$  models. The positions of the logs are indicated with red bars, the front and rear correspond to logs 1 and 2 respectively.

the success of the local optimisation in evolving to the global minimum. The set of experiments described in this section provide the first realistic inversion in terms of acquisition geometry and geological veracity. Besides, the inversion of data corresponding to models with realistic  $V_P/V_S$  ratios results in accurate images of both seismic velocities.

## 6.4 Marmousi 3D

The final suite of data were generated with a 3D version of the Marmousi model. The first dataset is purely acoustic and it will be inverted under this wave equation approximation, and the other is generated with a variable  $V_P/V_S$  ratio. The model dimensions are  $201 \times 201 \times 60$  nodes and the grid spacing is 24 m. The first two nodes of the model correspond to the water layer and therefore do not get updated across iterations, their value is fixed to 1500 m/s and 0 m/s for  $V_P$  and  $V_S$  respectively. The model and the  $V_P/V_S$  ratio are

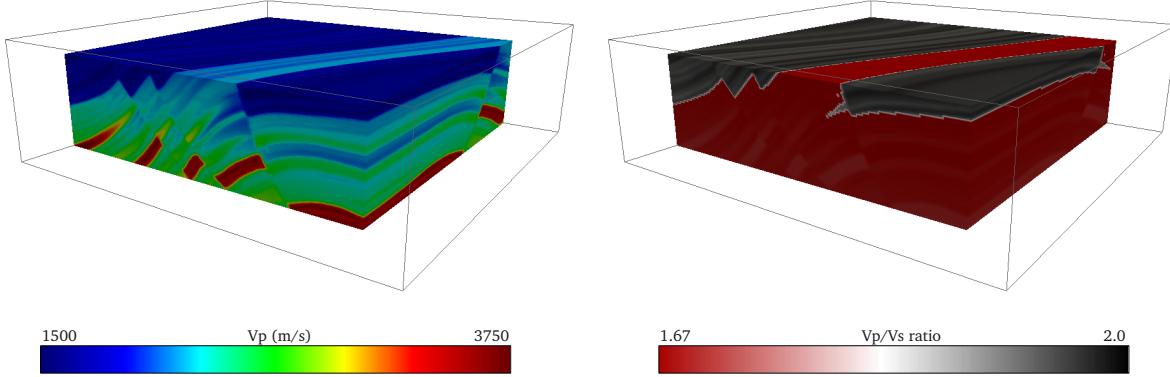


Figure 6.15: Cutaways of the 3D Marmousi (left) and the  $V_p/V_s$  ratio (right). Note that the models vary in all three directions (not just an extension of a 2D model like the one shown in figures 2.8 and 2.9).

depicted in figure 6.15.

A 3D shot gather corresponding to the both modellisations is shown in figure 6.16. The total amplitude of the gathers has been normalised and the values have been clipped to highlight the relevant events in the seismograms. The shear wave phenomena introduced in the elastic cases—for example, the presence of a strong elastic impedance contrast in the seabed where the velocity of shear waves goes from zero to some value around 800 m/s—creates a completely different distribution of events in terms of amplitude.

Table 6.3: Marmousi 3D inversions

model name	marm-acoustic	marm-elastic
wave equation	acoustic	elastic
iterations	35 (5 iterations/frequency)	35 (5 iterations/frequency)
frequencies inverted	3,4,5,6,7,8,9 Hz	3,4,5,6,7,8,9 Hz
PML thickness	20 cell/face (25 top face)	20 cells/face (25 top face)
normalised traces	no	yes
gradient preconditioning	approximate diagonal Hessian	approximate diagonal Hessian

A total of 240 point sources—not grouped in composite shots in this case—and 10000 receivers are spread to cover the model surface at a depth of 24 m. The maximum offset is given by the diagonal of the surface of the model. The source wavelet is a derivative of

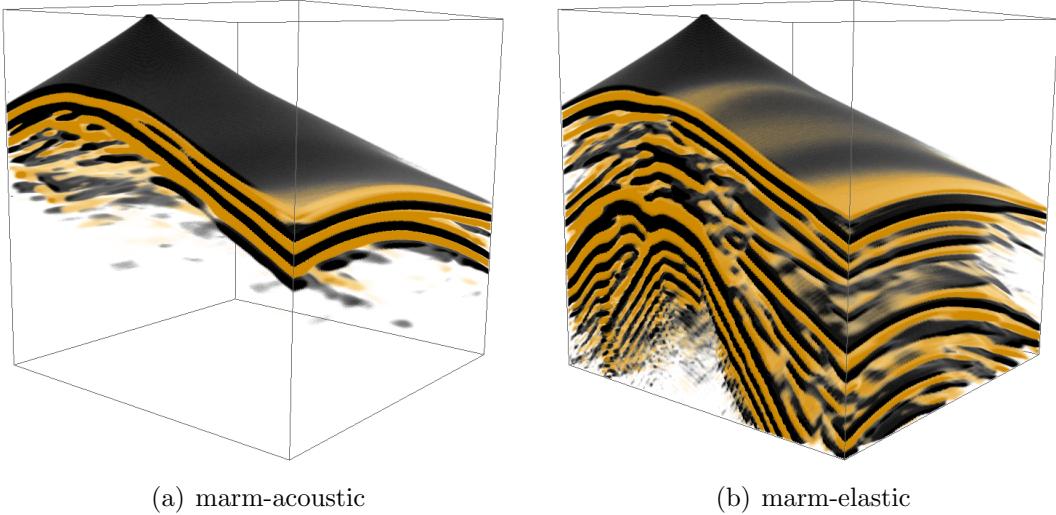


Figure 6.16: Normalised 3D shot gathers for a source positioned at  $x = 720$  m,  $y = 600$  m,  $z = 24$  m. The recorded energy corresponding to elastic events becomes apparent in (b) and (c), while in the acoustic case the energy is concentrated around the first arrivals. The amplitude values span from -0.001 to +0.001.

a ricker as well, with a peak frequency at 5 Hz because higher frequency contents would cause instabilities in the wavefield propagation given the grid-spacing and the time-step. The model has PMLs in all the faces, and therefore no free-surface is considered at the top of the model. Table 6.3 summarises the main characteristics of the inversions.

The gradient is preconditioned using the approximate diagonal Hessian to compensate for amplitude decay in the simulations. In the acoustic case, trace normalisation does not have an important effect on the recovered images, but in the elastic case trace amplitude normalisation is required to lessen the contribution of the Scholte waves, produced at acoustic-elastic interfaces. The comparatively higher amplitude of such interface waves dominates the inversion if the original amplitude is preserved. The starting models are smoothed versions of the targets.

Following the two-channel strategy to evaluate the performance of the inversions, the results are presented in three different formats: horizontal slices at three different depths, velocity logs at two different locations and, isovelocity surfaces. Besides, two vertical slices in the  $XZ$  and  $YZ$  planes are presented. The horizontal slices show good agreement between the final images and the target models (figures 6.17 and 6.18), however for the

elastic inversion, the quality of the recovered  $V_P$  degrades. The horizontal slides show how the shape of the undulating layer intersections is clearly imaged in the inverted models.

Vertical slices of the models at  $x = 75$  and  $y = 125$  are shown in figures 6.19-6.22. The shallow region (first few nodes) in the middle of the recovered  $V_P$  model for the elastic case, presents some instability due to the presence of a strong vertical and horizontal contrast, forming a step-shape. This kind of structures normally create strong high amplitude Scholte waves in fluid-solid interfaces, that in turn contaminate the parameter updates because they are less linear than the rest of the propagated waves.

The lower resolution is due to the limited bandwidth of the source: the models were discretised with a smaller grid-spacing then it would be possible to model (and therefore invert) higher frequencies. But the extra computational cost would be quite significant; if, for example, the grid spacing was reduced to half its value, then the computational time would increase 16 times: twice for each spatial direction and twice due to the consequent reduction of the temporal discretisation (increasing the number of times-steps).

Velocity logs provide a good measure of the fitting between recovered and target models from an accurate quantitative point of view. In addition, well log information is usually available in real acquisition surveys, and therefore is one of the few QC tools available to validate the inversion results. Figure 6.23 presents a comparison of starting, target and recovered 1D velocity profiles. The important feature to notice in the evaluation of the velocity logs is the trend of the inverted models which moves towards the target curve in a consistent manner. The fact that the shape is not exactly the same is due to the limited bandwidth of the source as mentioned above. The trade-off between  $V_P$  and  $V_S$  becomes apparent in the deeper regions of the model: in the first log, the high-velocity layer is imaged with approximately the right amplitude for  $V_P$  but  $V_S$  is overestimated, whereas in the second log  $V_P$  is underestimated while the  $V_S$  amplitude presents a good fit with the target model.

Figures 6.24 and 6.25 aim to highlight the continuity and coherency of isovelocity surfaces in the recovered models. Both recovered shallow horizons ( $V_P = 2000$  m/s,

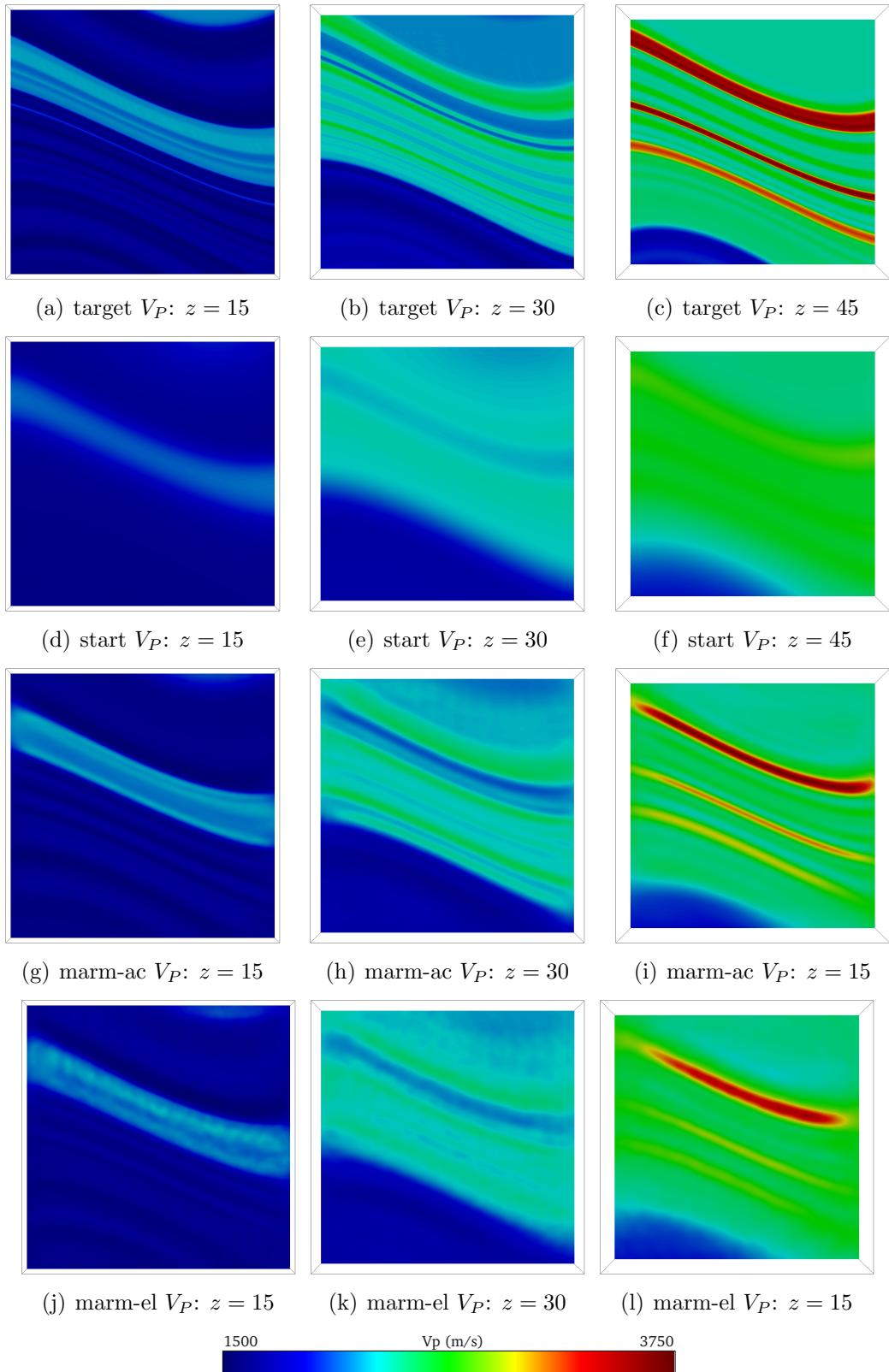


Figure 6.17: Depth slices for the target (a-c), start (d-f) and recovered  $V_P$  models for the acoustic (g-i) and elastic (j-l) inversions.

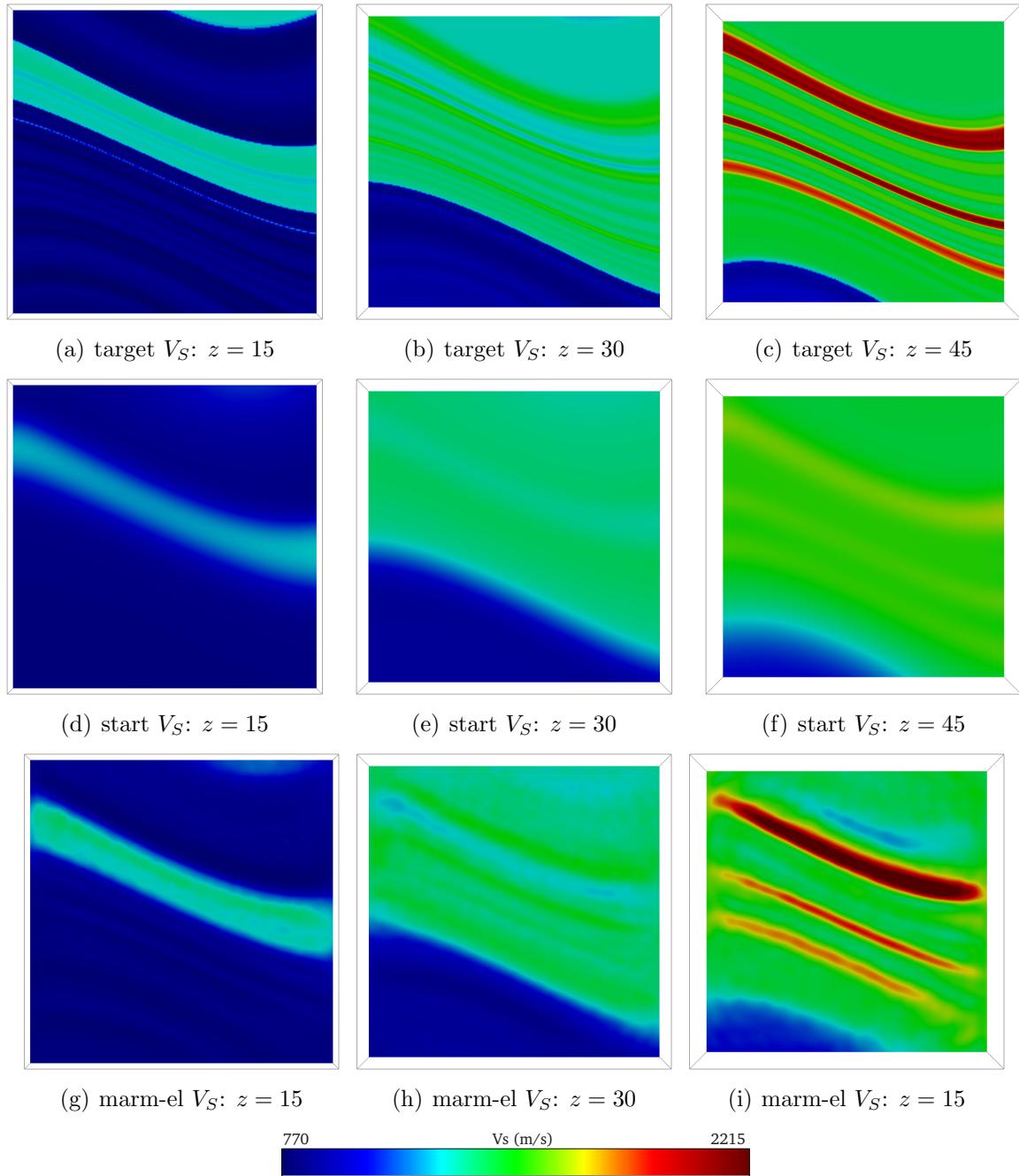


Figure 6.18: Depth slices for the target (a-c), start (d-f) and recovered (g-i)  $V_S$  models.

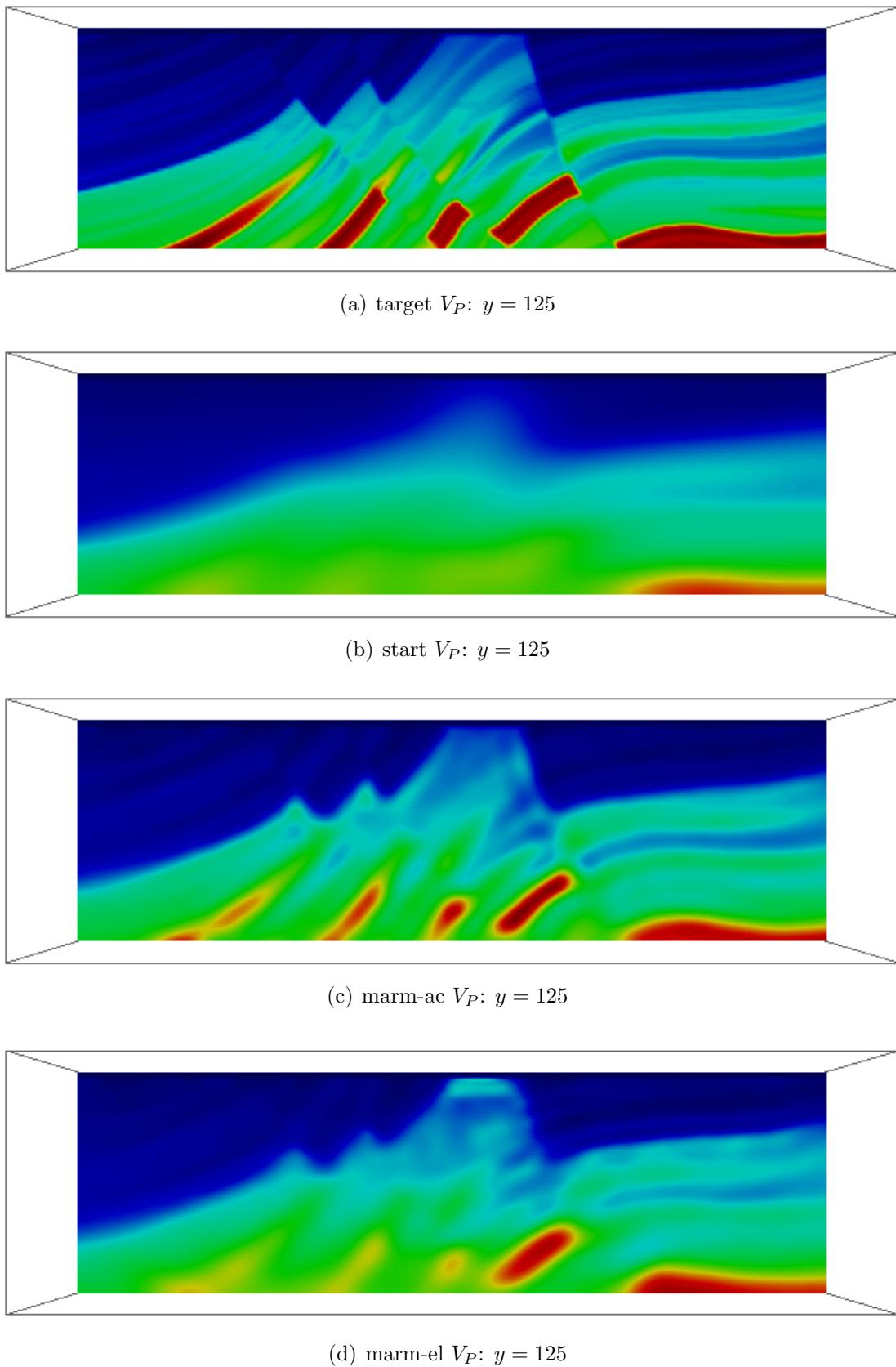


Figure 6.19: Vertical slices of the  $V_P$  models at  $y = 125$  nodes for the target (a), starting (b) and recovered models from acoustic (c) and elastic (d) inversions. Due to the presence of Scholte waves, the middle shallow region is better imaged in the acoustic (c) than in the elastic (d) inverted models.

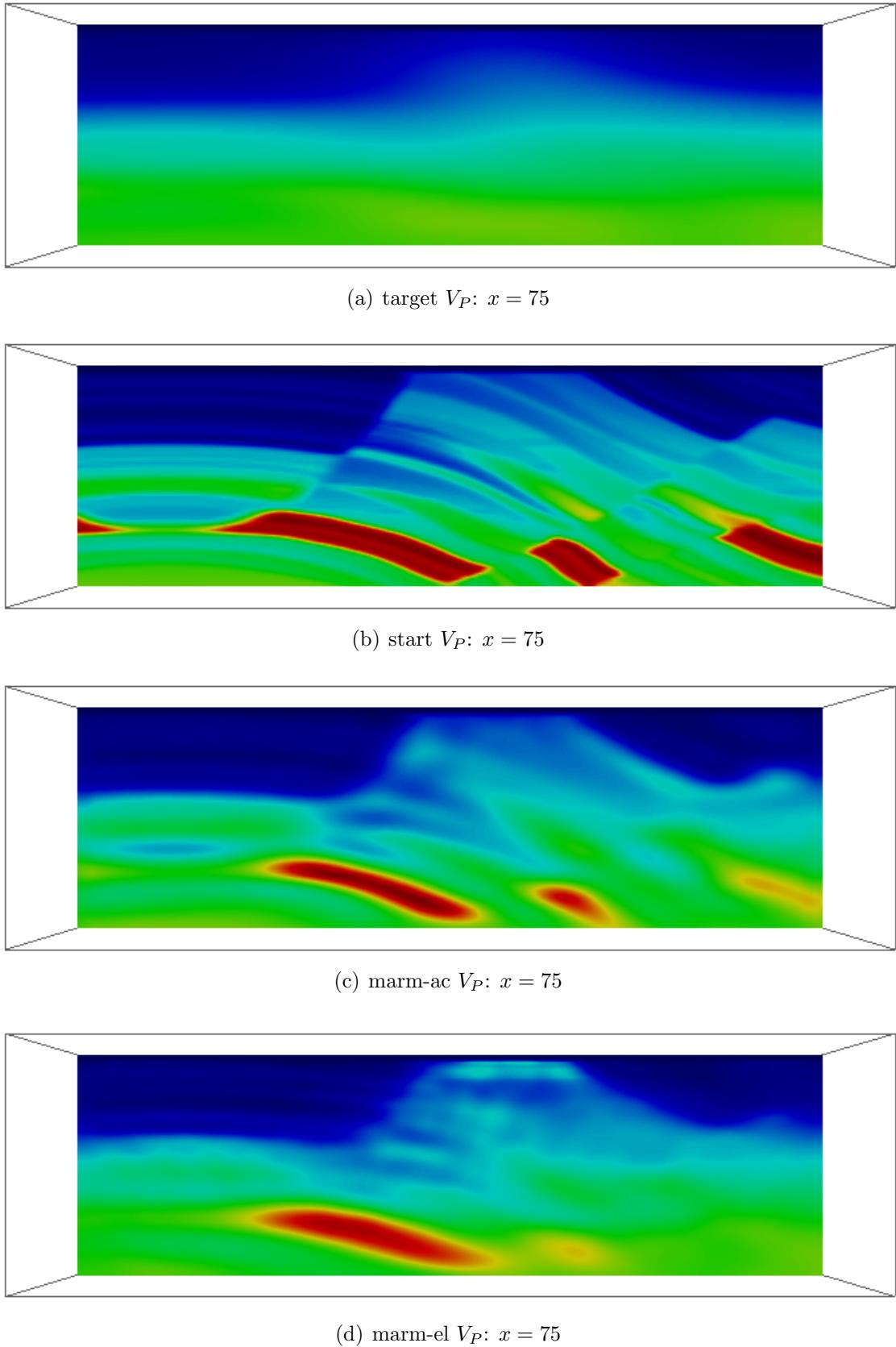


Figure 6.20: Vertical slices of the  $V_P$  models at  $x = 75$  nodes for the target (a), starting (b) and recovered models from acoustic (c) and elastic (d) inversions. The same differences than in figure 6.19 between (c) and (d) are observed.

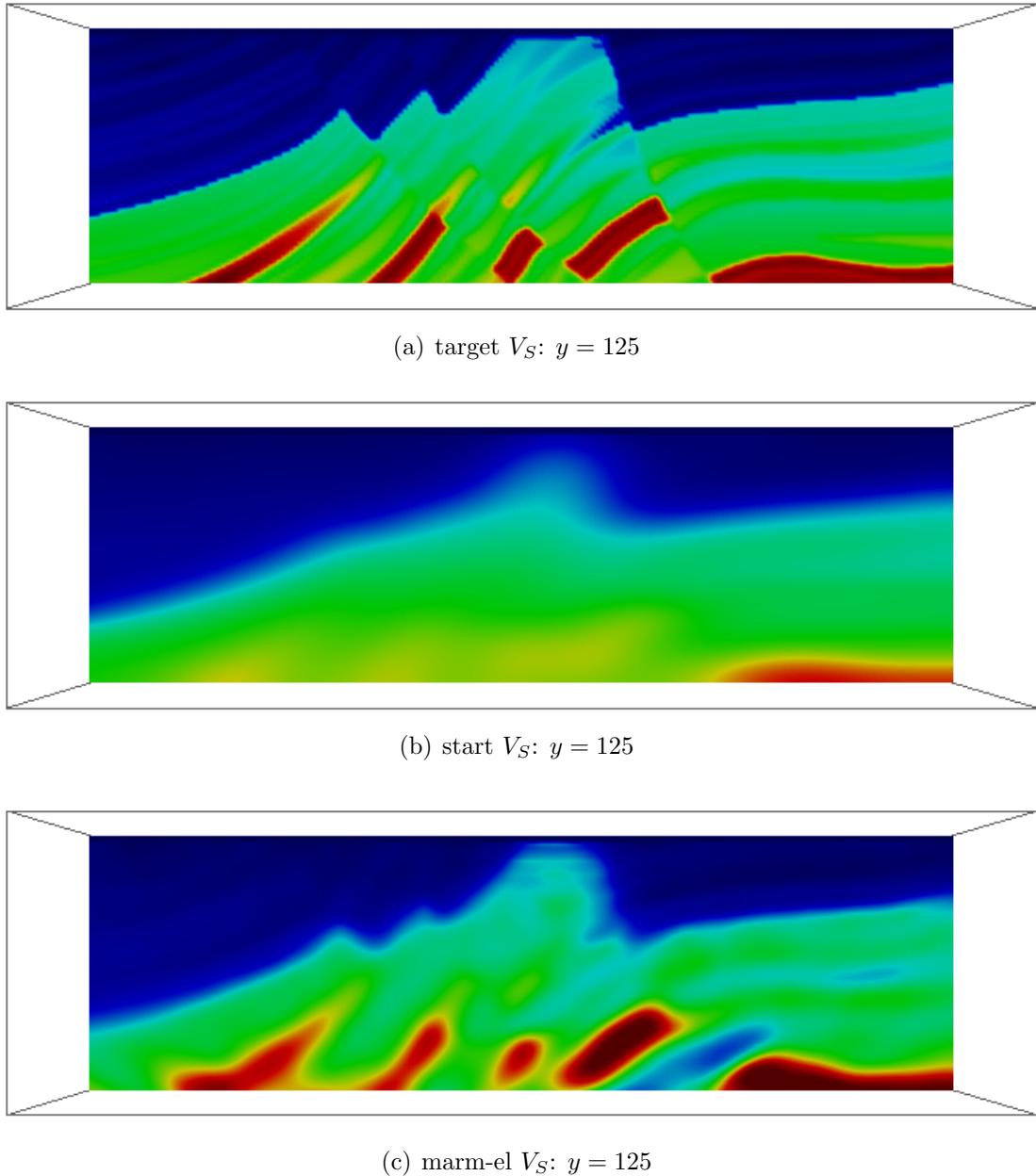


Figure 6.21: Vertical slices of the  $V_S$  models at  $y = 125$  nodes for the target (a), starting (b) and recovered model (c). The shallow middle region does not suffer from the artifacts introduced by the Scholte waves as much as the recovered  $V_P$  models. Note that in the deeper regions the amplitude of the high-velocity intrusion layers is overestimated.

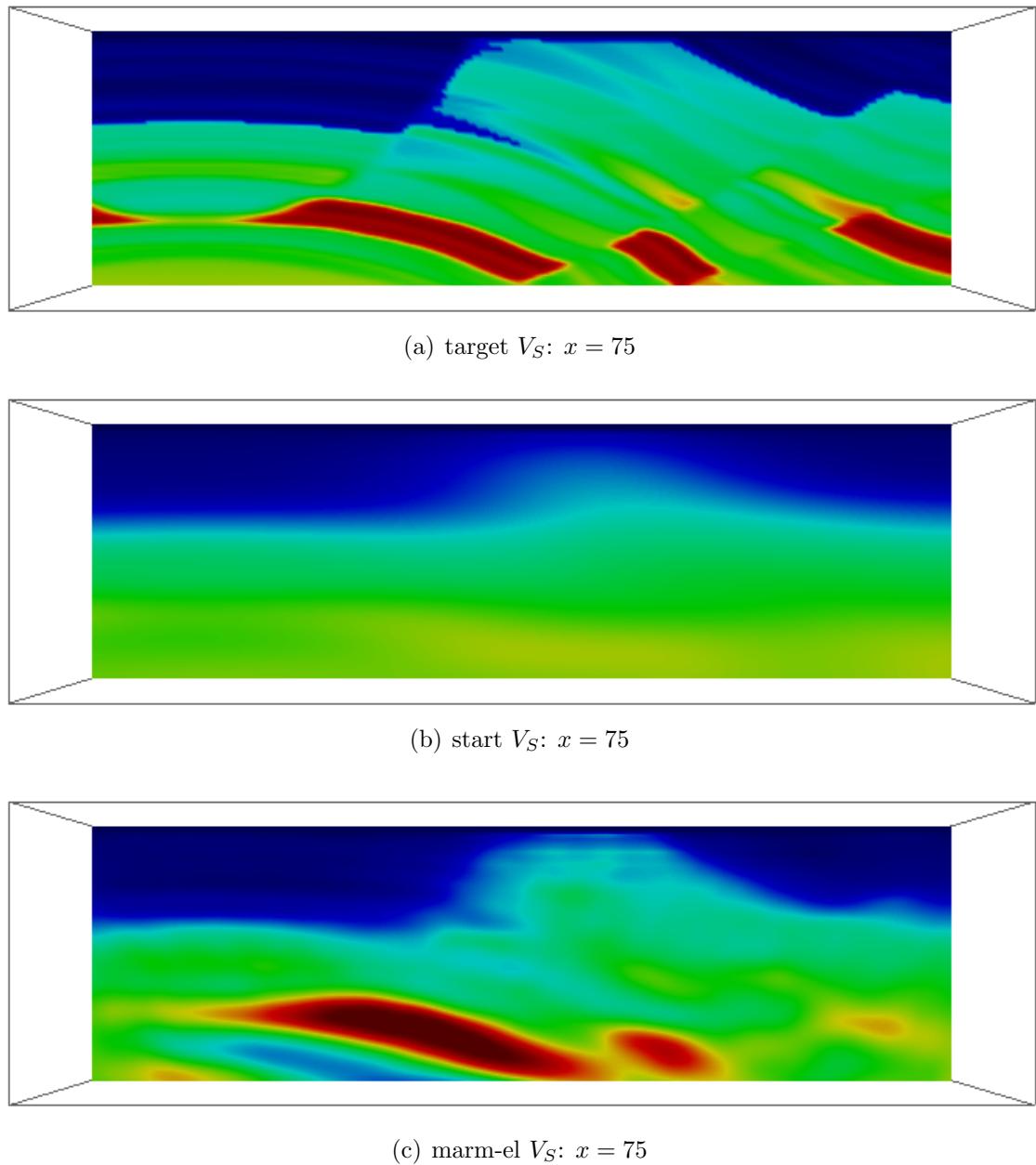


Figure 6.22: Vertical slices of the  $V_S$  models at  $y = 125$  nodes for the target (a), starting (b) and recovered model (c).

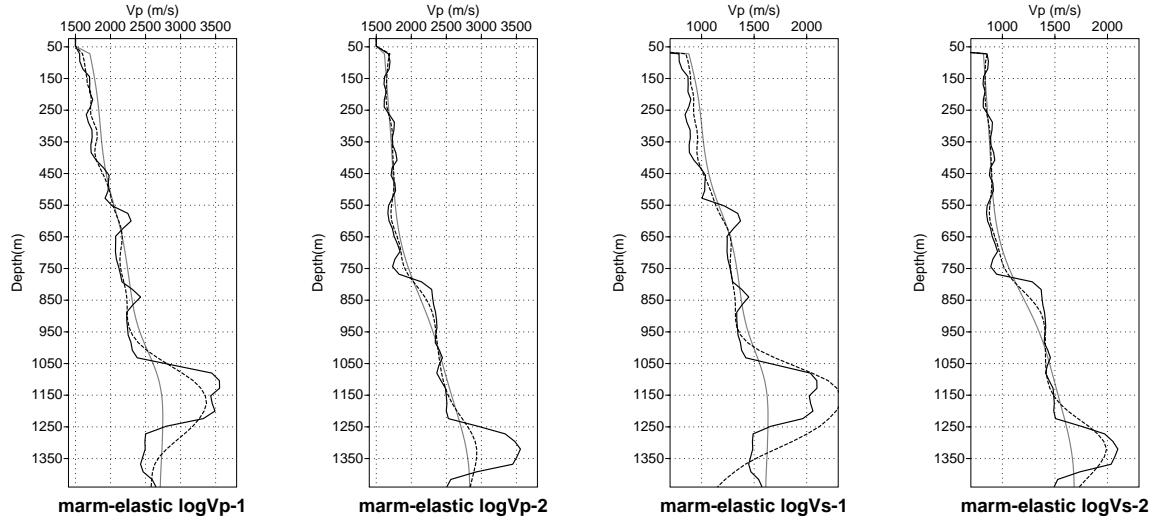


Figure 6.23: Vertical logs for the recovered  $V_P$  and  $V_S$  velocity models at locations: log 1  $x = 3120$  m,  $y = 2160$  m; log 2  $x = 1680$  m,  $y = 3480$  m. The grey, black and dashed lines correspond to the starting model, target and recovered models respectively.

$V_S = 1100$  m/s) recover the ripples between the two logs from a starting model that did not contain them. Similarly, the deeper horizons ( $V_P = 2800$  m/s,  $V_S = 1800$  m/s) show how the high-velocity layers are successfully imaged from a smooth starting model. As a consequence, models obtained with 3D FWI provide high resolution images which have more potential than the velocity models obtained with other ray-based methods. A direct application is that interpretation of the geological features can be done directly on the depth imaged velocity images. Thus, geological horizons can be extracted in a completely automated way from the recovered models instead of migrated images. In the 3D Marmousi case, I have selected two different horizons for each seismic velocity to demonstrate the capabilities of FWI to accurately select horizons. The shallower horizon defines the interface between the soft and consolidate rocks clearly appreciated in figures 6.19-6.22 as blue-green transitions. The deeper horizon separates the high velocity layer intrusions (dark red areas in figures 6.19-6.22).

Data generated with the target, starting and recovered models (figure 6.26) reveal how the inversion process operates on the data space. The model updates increase the resolution of the starting model, increasing the number of reflectors and modifying the

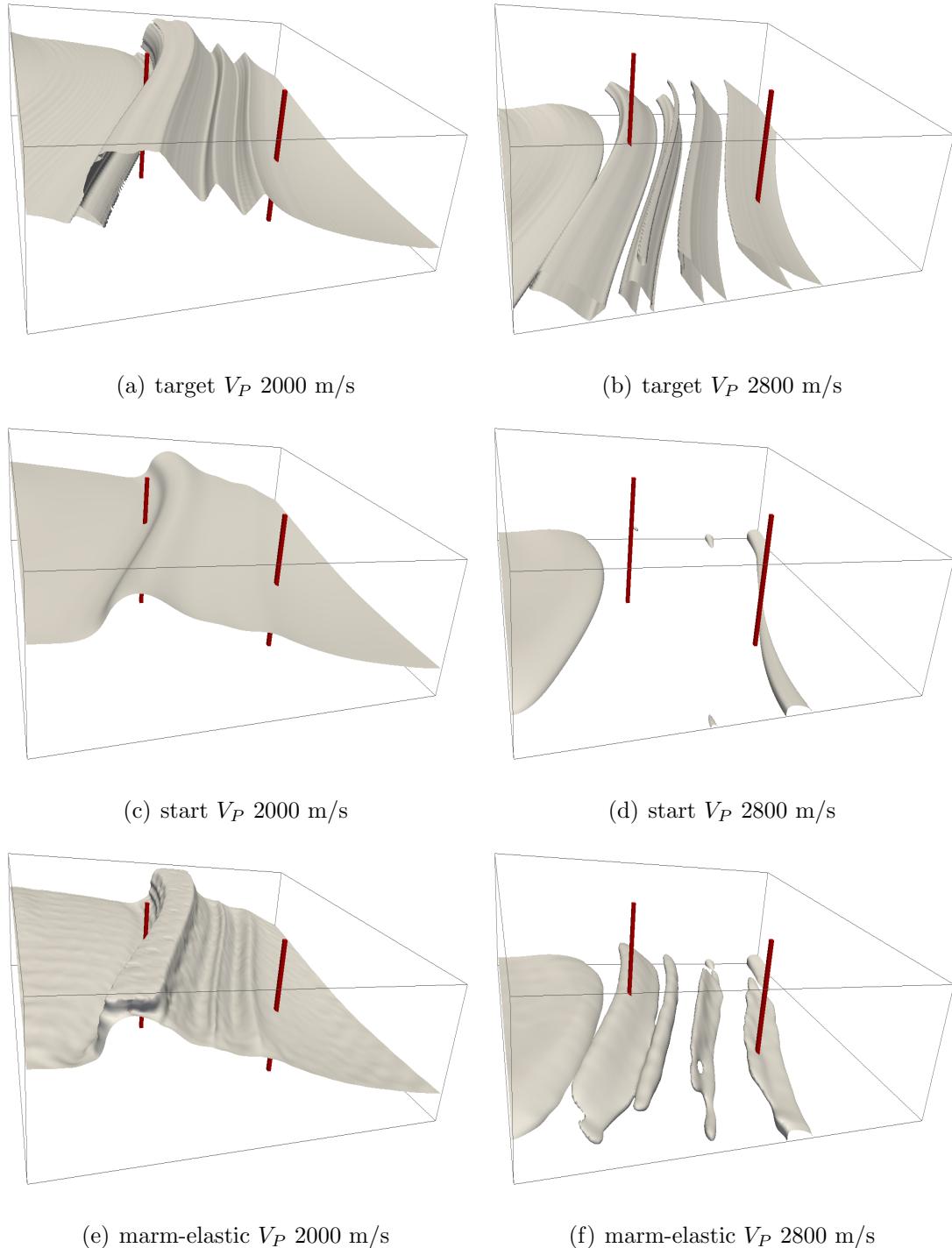


Figure 6.24: Isosurfaces corresponding to 2000 m/s and 2800 m/s for the target (top), start (middle) and recovered (bottom)  $V_P$  models. The positions of the logs are indicated with red bars: log 1: left; log 2: right.

travel-times —within a cycle— to mimic the true data.

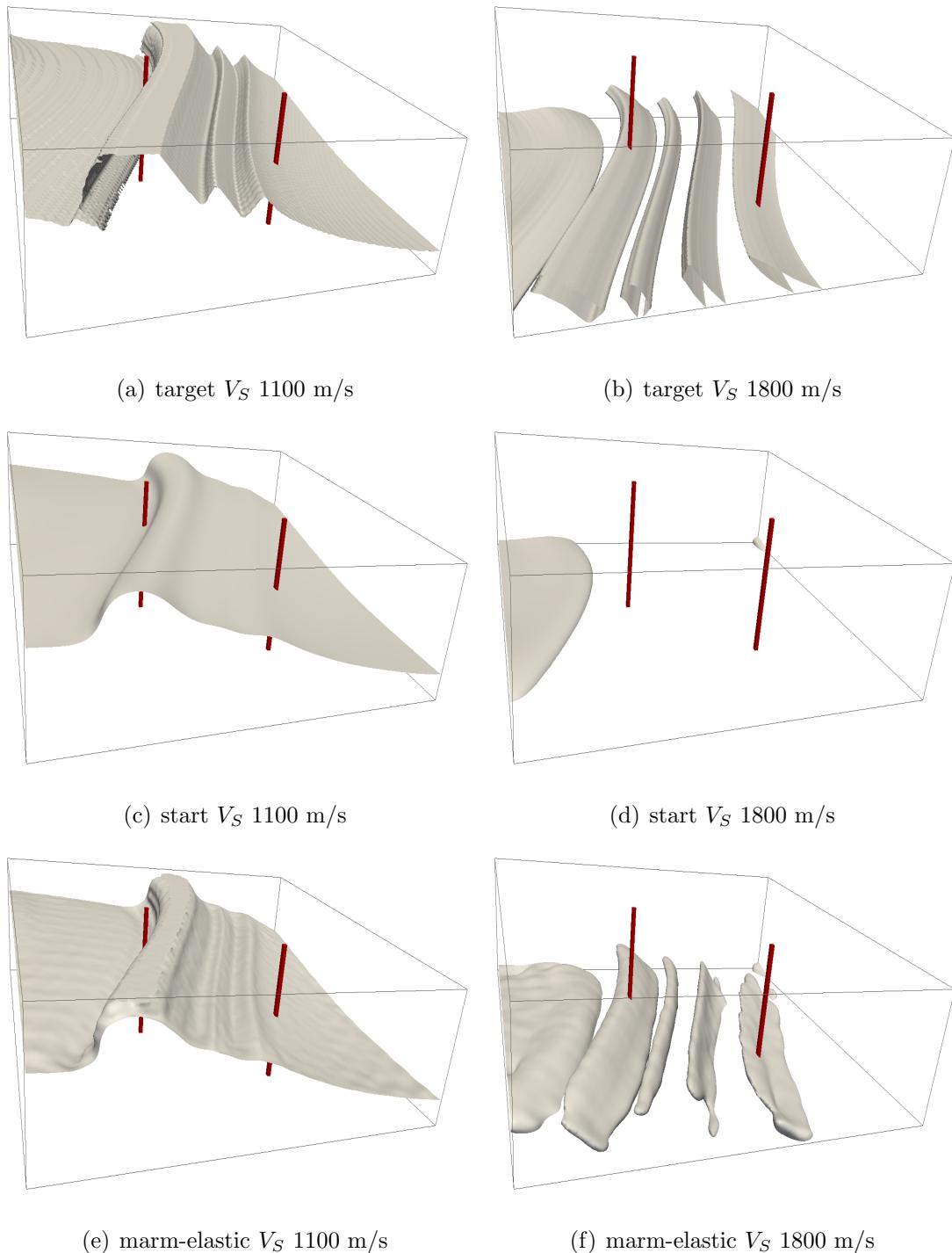


Figure 6.25: Isosurfaces corresponding to 1100 m/s and 1800 m/s for the target (top), start (middle) and recovered (bottom)  $V_S$  models. As in the previous figure, the positions of the logs are indicated with red bars: log 1: left; log 2: right.

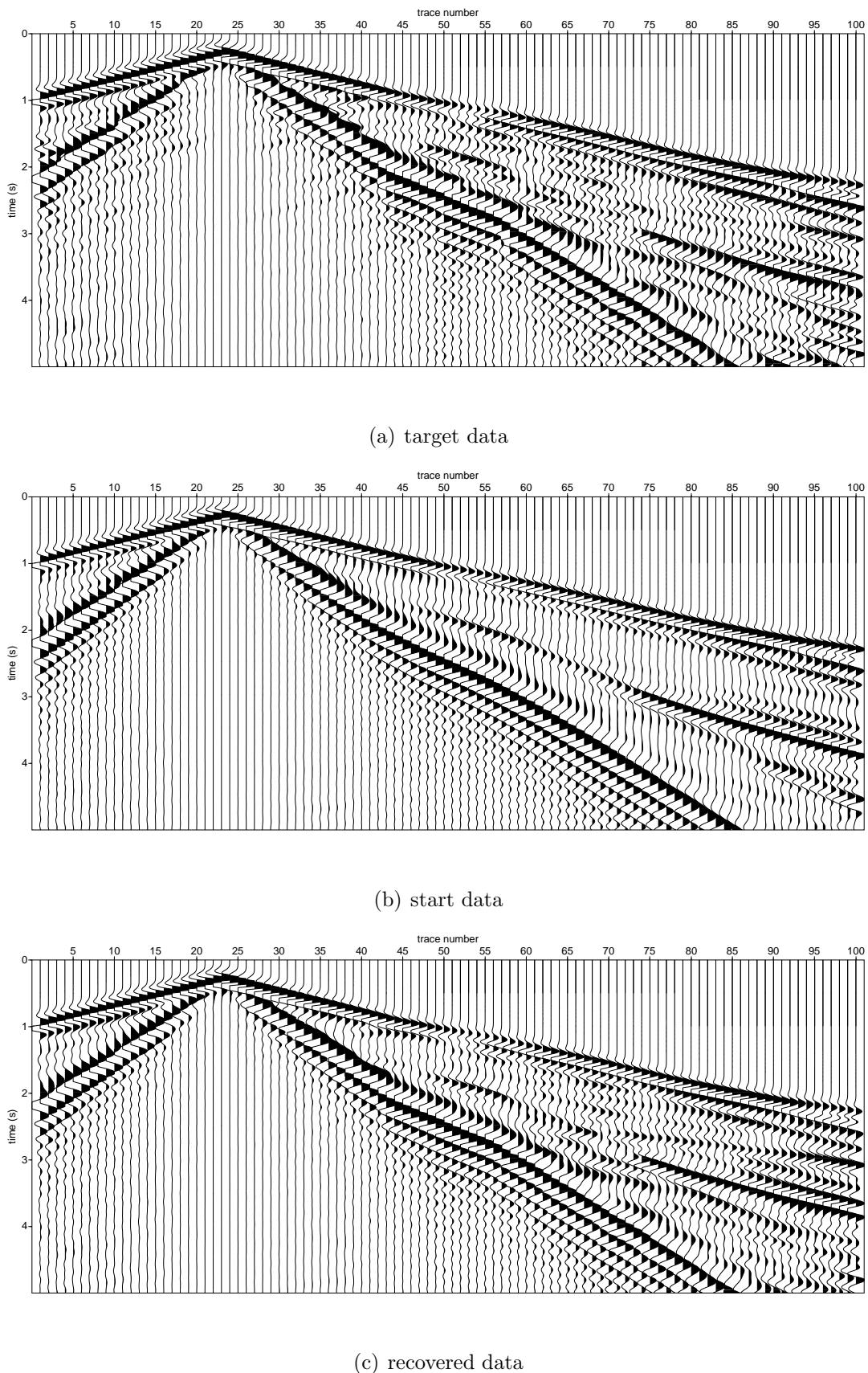


Figure 6.26: Normalised seismograms corresponding to a line of receivers at  $x = 75$  for a shot in  $x = 75, y = 75$  (all units are given in nodes).

## 6.5 Objective function evolution

One of the most challenging problems in FWI is to quality control the results. Normal procedures involve evaluation of the objective function across iterations and assume that an overall reduction of the residuals is symptomatic of a good behaviour. Generally I do not agree that this is a valid QC to monitor the inversion of seismic data because the traces normally need normalisation (matching amplitudes of synthetic and field data) before calculating the residuals —giving more weight to the phase information during the inversion process— and therefore the absolute value of the functional is quite meaningless. Moreover, as the inversion progress, the data spectrum is usually changed from low to high frequencies and such high frequencies augment the density of information of the traces and, thus, the functional.

The objective functions for the one cuboid anomaly FWI experiments are shown in figure 6.27. Note that even though both acoustic inversions recovered the anomalies, the difference in the residuals is significant. Every five iterations a peak appears, corresponding to the next frequency block. For the elastic cases, even for the ones that accurately recover the target models, the functional increases considerably. The increment is caused by a combination of factors: the bandwidth increases with the frequency blocks, and including more energy increases the value of the functional; in some cases the sources involved are different at each iteration and therefore the functional does not represent the misfit of the same data; finally, the updates in the shear- and compressional-velocity properties result in more heterogeneous models that create more energy conversions recorded (principally P-S-P) in the receivers, increasing the amount of energy recorded in the receivers, and hence, the objective function. As a consequence, the functional values are not comparable from one iteration to another. In addition, the functional is only used to calculate the step-length —there is no ‘memory’ between iterations as in BFGS or conjugate-gradient methods. To compute the step-length, a small perturbation is applied to the model in the direction of the gradient. In this case, the functional corresponding

to the perturbed model  $\mathbf{m} + \delta\mathbf{m}$  is always smaller than the functional corresponding to  $\mathbf{m}$ . Indeed, this is the only situation where the functional values are comparable because the model is almost the same — $\delta\mathbf{m}$  is sufficiently small— and the frequency content in the data is unchanged.

For the two channel model, a similar behaviour is observed: the acoustic inversion results in an overall reduction of the functional while the elastic ones increase its value in the last iterations (figure 6.28(a)). Note that in the elastic case, the spikes corresponding to the frequency blocks cannot be appreciated, probably due to the dependency of the data on both seismic velocities. The functional for the Marmousi inversion evolves similarly, presenting a rather chaotic trend for the elastic inversion (figure 6.28(b)).

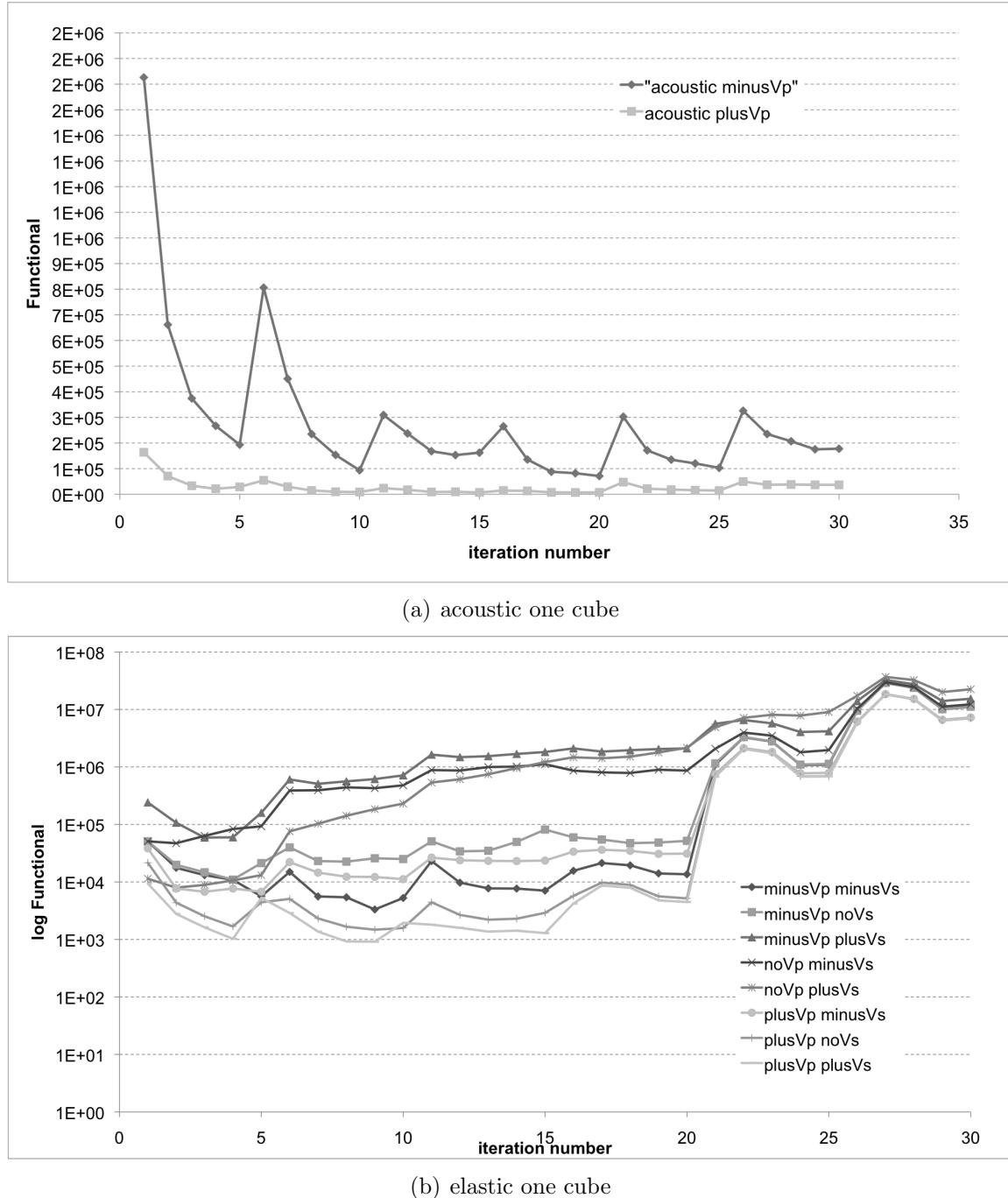


Figure 6.27: (a) the functional is reduced at each frequency block for the two acoustic inversions although the absolute values are very different for the positive and negative anomalies; (b) all the elastic functionals have higher values at the end of the inversion, although the successful iterations show some tendency to decrease at early stages (in this case the functional axis is logarithmic to appreciate the changes at each iteration). The increase in the functional is due to the fact that the functional is computed under different circumstances at each iteration.

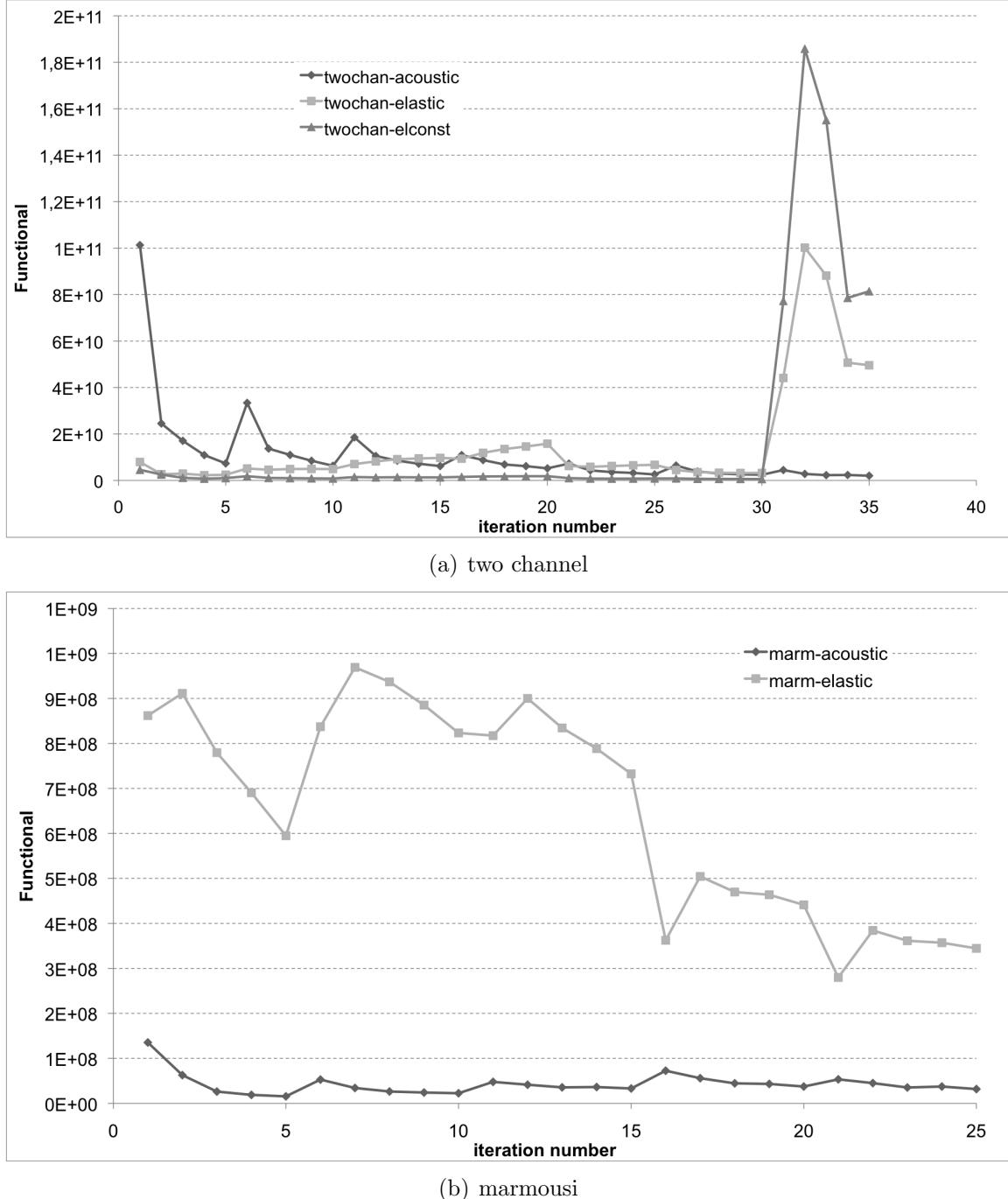


Figure 6.28: Objective function evolution for the two channel (a) and Marmousi (b) experiments. Again, the elastic and the acoustic inversions have very different trends.

## 6.6 Conclusions

In this section I applied the principles of the previous chapter to perform 3D elastic full-waveform inversion with realistic synthetic datasets. The acoustic inversions produce slightly better resolved images, especially for the elastic models that do not have a constant Poisson’s ratio probably due to the uncoupling of the solution spaces —expressed as a function of position— for both seismic velocities. Due to the high non-linearity of the problem, it is hard to predict whether this trade-off is critical, although instinctively one may be inclined to think that a constant Poisson’s ratio would generate similar shapes in the solution space, maybe elongated in the functional direction. However, this is hard to prove. On the other hand, the solution space can be thought of as a unique hypersurface with both seismic velocities treated independently even if they share the same position in space.

Indeed, the trade-off between seismic velocities has a major impact on the recovered models: the geometry of the imaged anomalies for  $V_P$  and  $V_S$  follow the same shape-pattern in each particular case. This effect becomes very distinctive when the recovered models have artifacts (see, for example, figures 6.6(c)-6.6(f)). In this case, the local minima in which the recovered shear velocity models fall seem to be complementary to the  $V_P$  models, which suggests that the solution space for both realisations has some kind of symmetry. For the realistic models (in terms of geological features) the differences in the Poisson’s ratio are not as extreme as for the one cube anomaly models and therefore is more difficult to identify if the effect of the trade-off presents some reciprocity in the imaged model spaces. In any case, in realistic cases this does not have a critical effect due to the limited range of values of  $V_P/V_S$ , except in some situations where strong shear waves in fluid-solid interfaces occur.

Results generated by FWI provide high-resolution images but not all the detailed features in a recovered model are necessarily correct. To QC the output is one of the most challenging problems in the FWI community; especially when it is applied to seismic

exploration of hydrocarbons, where the risk assessment drives the strategies adopted by companies. The usual procedure is to compare the inversion images against well-log data and measure their degree of fitness. Another standard strategy is to depth migrate the data with the velocity model and study the continuity of the reflectors in the stacks. 3D elastic migration is a complex topic and this is beyond the scope of my project to study the potential improvements of data migration with FWI velocity models, and I did not apply this QC method. Mostly in industry applications, flattening the reflections in the gathers is also a standard practice to assess the quality of the velocity models.

### 6.6.1 CPU times

The run-times for the inverted datasets are summarised in table 6.4. A direct comparison of the times is not very pragmatic because I have used different resources for each model. The number of nodes has a major impact on the total time required to complete the computations because the parallelisation is designed to distribute shots across nodes. If the number of nodes is lower than the number of shots one wants to invert, then they need to be recycled to run more than one shot increasing the time proportionally to the number of recycled shots per node. A strategy I adopted to mitigate this effect is to run multiple shots at each node to reduce the computational time. The number of shots is still the same but if the memory of the nodes is enough to host more than one shot, this strategy decreases by a few percent the run-times. Obviously, the number of time-steps, iterations, shots/iteration, cluster nodes, shots/node, and the size of the model and PMLs have a direct effect on the computational time required to solve both the forward and the inverse problem.

Note that the ratio between acoustic and elastic is close to two even the number of variables—and calculations—that describe the problem increases by a factor of nine in the elastic case. There is a combination of factors that explain why the elastic run-times are not bigger:

1. A significant amount of time is spent accessing memory. In the acoustic wave equation contains a second-order time differential. As a result, pressure needs to be stored at two time-steps, while in the elastic case all the variables only need to be kept for one time-step.
2. The threading parallelism becomes less efficient as the workload decreases. The domain in the models inverted is relatively small (especially considering that the model is decomposed in different parts handled to each thread), so it means the acoustic is doing considerably less work per step than the elastic, therefore the parallel thread efficiency is significantly lower.
3. The acoustic stencil is more complex: it has 37 points for each cell (per calculated wavefield), whereas elastic only really has 8 or 12 (for stress and particle velocity respectively).

Table 6.4: CPU-times

<b>model</b>	<b>cube-ac</b>	<b>cube-el</b>	<b>twoch-ac</b>	<b>twoch-el</b>	<b>marm-ac</b>	<b>marm-el</b>
forward modelling	00:02:30	00:04:25	00:02:25	00:04:00	00:28:00	00:42:00
inversion	03:35:00	05:55:00	03:12:00	06:32:00	13:30:00	27:30:00
cluster nodes	4	4	4	4	31	31
shots/node	4	4	2	2	1	1
time-steps	940	940	720	720	2000	2000
threads	6	6	6	6	12	12
model size	101 × 101 × 51	101 × 101 × 51	161 × 161 × 51	161 × 161 × 51	201 × 201 × 60	201 × 201 × 60
PML thickness/face	10 cells (top:15)	10 cells (top:15)	10 cells (top:15)	10 cells (top:15)	20 cells (top:25)	20 cells (top:25)
shots/iteration	25	25	7	7	120	120
number of iterations	30	30	35	35	25	25

## 7. Conclusions and further research

The objective of this thesis was to design an efficient implementation of 3D elastic full-waveform inversion suitable to operate on real-sized models. To achieve this goal, I have made use of well established wave modelling and inversion techniques. In a collaborative effort, Adrian Umpleby and I have reformulated and programmed the inverse problem using matrix algebra and discovered the sensitivity to the wave equation formulation in the application of the adjoint-state method. As a consequence, direct substitution in the parameter updates derived by Tarantola and Mora — who used a displacement-stress formulation— would lead to incorrect gradient expressions in the velocity-stress parameterisation used to solve the elastic wave equation.

I have applied 3D elastic FWI to several synthetic models to validate the effectiveness and efficiency of our implementation. Inversion of synthetic datasets corresponding to the studied benchmark models (one and eight cuboid anomalies) defines the theoretical limitations of the method, while inversion of realistic datasets generated with more geologically consistent models (two channel and 3D Marmousi) has confirmed its potential.

The extension from acoustic to elastic increases the computational cost significantly. However, a more realistic description of wave propagation phenomena is worth the extra cost because:

1. The amplitude versus offset (AVO) behaviour is more accurate.
2. Elastic parameters are more sensitive to fluid presence, of great interest in hydrocarbon exploration.
3. Elastic parameters have an empirical relation with electromagnetic properties and

other rock physics characteristics, opening new possibilities for further joint-inversions of multi-physic datasets.

4. Geological interpretation of multiple highly resolved images (as opposed to only  $V_P$  images) provides with an extra QC tool to evaluate FWI results.

## 7.1 Cross-talk between elastic parameters

By nature, the interleaved relation between elastic parameters and seismic data introduces a new level of complexity in the local optimisation methods used to solve the inverse problem. Ignoring density, the acoustic wave equation poses a unique relation between data and model parameters which is lost when elastic phenomena are taken into account. As a result, the local optimisation method suffers from parameter update competition and therefore increases the number of local minima in which the objective function may fall. Furthermore, the presence of higher-order wave propagation phenomena —for example, Scholte waves— degrades the quality of the inverse problem linearisation.

To improve the performance of local optimisation methods in such a complex scenario, I believe that further study of the following aspects involved in the inversion process should be explored:

1. Order in which the updates are applied. Which strategy is best to update the model parameter: simultaneous inversion? Fix one of the parameters while the other is updated? Or is it better to alternate updates in a flip-flop fashion?
2. Impact of the model update parameterisation on the shape of the solution space. Elastic parameters can be represented using many different properties: seismic velocities and impedances, Lamé parameters, Young modulus, Bulk modulus and Poisson’s ratio, as well as their different combinations. The search for an optimal choice will involve the derivation of new gradient expressions and the evaluation of their response to local optimisation methodologies.
3. Potential of other optimisation methods. Although a pure global search is still

intractable due to computer limitations, exploration of hybrid methods or extension to multiple-local optimisations will improve the probability of approaching the global minimum of the solution space.

4. Preconditioning operators based on the correlation between elastic properties. Maybe the origin of the problem can be turned into a positive discrimination criteria to constrain the independent parameter updates. For example, the Poisson's ratio only take values in the range of  $\{-1, 0.5\}$ ; or, a priori knowledge of rock properties limit the expected values of elastic parameters.

The best and probably unrealistic solution would be to find a parameterisation of the wave equation that naturally splits the compressional and shear waves into different equations, each depending on different sets of parameters. At a first glance, this separation seems unattainable: straightforward substitution of the stress variables into particle-velocities or vice-versa do not separate shear and compressional wavefields, nor their dependency with the elastic parameters  $\rho$ ,  $\mu$  and  $\lambda$ . However, I am reluctant to rule this possibility out before a more exhaustive investigation of the rigorous mathematical demonstration.

## 7.2 Anisotropic elastic inversion

The full stiffness tensor  $c_{ijkl}$  fully describes wave propagation phenomena under the approximation of linear elasticity by introducing a complete description of the media properties in the wave equation. Thus, the natural evolution of FWI is to take all the elements of the elastic tensor into account to generate seismic data as accurately as possible. The extra cost associated with such extension can be afforded using present computer hardware. The anisotropic behaviour (which has a direct impact on real data) introduced by the stiffness tensor will improve the fitness between real and modelled data.

A completely different problem arises when the elements of the stiffness tensor are not only imposed to model data, but also considered as variables subject to be inverted. In

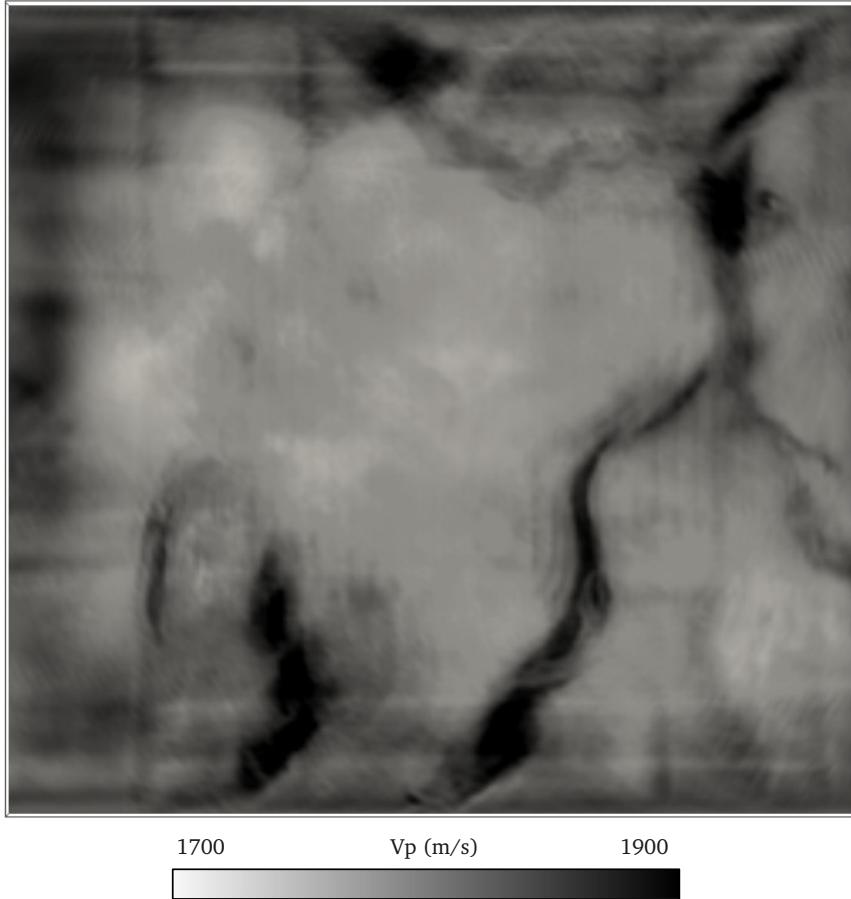


Figure 7.1: Acoustic image of the channels at a depth of 200 m.

this case, the abovementioned trade-off between inverted parameters poses an even more complex problem that I would like to investigate in my future research endeavours.

### 7.3 Field data application

The final purpose of my work is to apply 3D elastic FWI to real data. Currently, I am working with another member of my research group, Tenice Nangoo, in an OBC survey placed on top of an oil and gas field. An important difference is that real datasets are more difficult to invert than their synthetic counterparts due to anisotropy, attenuation and the fact that the same forward modelling tool is used for both generation and inversion of

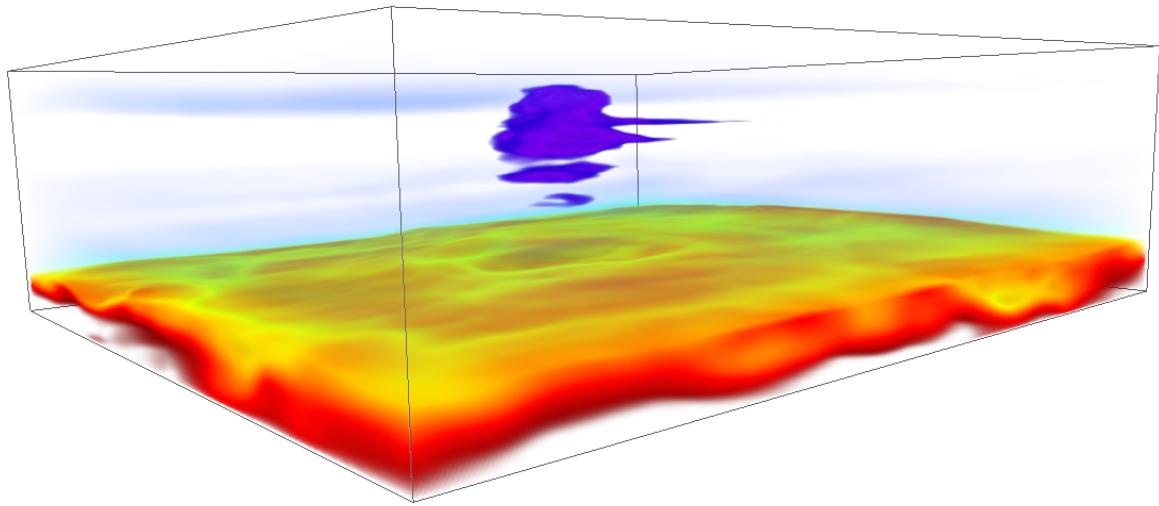


Figure 7.2: 3D view of the target gas cloud.

synthetic datasets.

The target volume has already been subjected to anisotropic acoustic 3D FWI. The resulting high-resolution images highlight several features in the recovered model. We have selected two of these features as our elastic targets: a system of shallow channels (depicted in figure 7.1) and a gas cloud located at a depth of approximately 1 km (figure 7.2). The rationale behind our decision is that shallow regions are easier to image and that both features present sufficiently strong velocity contrasts.

Even though the starting model used in this acoustic inversion did not contain any hint of channel presence, FWI succeeds in detecting them. Therefore, we expect elastic FWI to be sensitive to their presence as well, and provide one of the first 3D elastic FWI applications of field data.

The gas cloud, on the other hand, needs to be included as a blocky low velocity anomaly in the starting model to avoid cycle skipping in the data: the extremely low values of the compressional wave velocities contained in it can go as low as 1450 m/s, and not including this velocity anomaly in the starting model would cause a phase shift in the modelled data (compared to the real) of more than 360°, thus cycle-skipped, and the

inversion would fail. As a consequence, the starting  $V_S$  model must also include the gas cloud. Indeed, the shorter shear wavelengths increase the risk of cycle-skipping and the inclusion of the gas cloud in the starting  $V_S$  model is more critical.

# Bibliography

- [Aki and Richards, 2002] Aki, K., and P. Richards, 2002, Quantitative seismology: Univ Science Books.
- [Alterman and Abramovici, 1965] Alterman, Z., and F. Abramovici, 1965, Propagation of a P-pulse in a solid sphere: Bulletin of the Seismological Society of America, **55**, 821–861.
- [Alterman and Karal, 1968] Alterman, Z., and F. Karal, 1968, Propagation of elastic waves in layered media by finite difference methods: Bulletin of the seismological Society of America, **58**, 367–398.
- [Ambartsumian and Ambartzumian, 1998] Ambartsumian, V., and R. Ambartzumian, 1998, A life in astrophysics: selected papers of Viktor A. Ambartsumian: Allerton Press.
- [Asvadurov et al., 2004] Asvadurov, S., V. Druskin, M. Guddati, and L. Knizhnerman, 2004, On optimal finite-difference approximation of pml: SIAM Journal on Numerical Analysis, 287–305.
- [Ben-Hadj-Ali et al., 2008] Ben-Hadj-Ali, H., S. Operto, and J. Virieux, 2008, Velocity model building by 3D frequency-domain, full-waveform inversion of wide-aperture seismic data: Geophysics, **73**, VE101.
- [Berenger, 1996] Berenger, J., 1996, Three-dimensional perfectly matched layer for the absorption of electromagnetic waves: Journal of computational physics, **127**, 363–379.

- [Berenger, 1994] Berenger, J.-P., 1994, A perfectly matched layer for the absorption of electromagnetic waves: *Journal of Computational Physics*, **114**, 185 – 200.
- [Bohlen, 2002] Bohlen, T., 2002, Parallel 3-D viscoelastic finite difference seismic modelling\* 1: *Computers & geosciences*, **28**, 887–899.
- [Brenders and Pratt, 2007] Brenders, A., and R. Pratt, 2007, Efficient waveform tomography for lithospheric imaging: implications for realistic, two-dimensional acquisition geometries and low-frequency data: *Geophysical Journal International*, **168**, 152–170.
- [Brossier et al., 2009] Brossier, R., S. Operto, and J. Virieux, 2009, Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion: *Geophysics*, **74**, WCC105.
- [Brossier et al., 2010] ———, 2010, Which data residual norm for robust elastic frequency-domain full waveform inversion?: *Geophysics*, **75**, R37.
- [Bunks et al., 1995] Bunks, C., F. Saleck, S. Zaleski, and G. Chavent, 1995, Multiscale seismic waveform inversion: *Geophysics*, **60**, 1457.
- [Castellanos et al., 2011] Castellanos, C., V. Etienne, G. Hu, S. Operto, R. Brossier, and J. Virieux, 2011, Algorithmic and methodological developments towards full waveform inversion in 3D elastic media.
- [Cerjan et al., 1985] Cerjan, C., D. Kosloff, R. Kosloff, and M. Reshef, 1985, A nonreflecting boundary condition for discrete acoustic and elastic wave equations: *Geophysics*, **50**, 705–708.
- [Červený et al., 1977] Červený, V., I. Molotkov, I. Molotkov, and I. Pšenčík, 1977, Ray method in seismology: Univerzita Karlova.
- [Chew and Liu, 1996] Chew, W., and Q. Liu, 1996, Perfectly matched layers for elastodynamics: a new absorbing boundary condition: Urbana, **51**, 61801.
- [Clapp, 2009] Clapp, R. G., 2009, Reverse time migration with random boundaries: SEG Technical Program Expanded Abstracts, **28**, 2809–2813.

- [Clayton and Engquist, 1977] Clayton, R., and B. Engquist, 1977, Absorbing boundary conditions for acoustic and elastic wave equations: *Bulletin of the Seismological Society of America*, **67**, 1529.
- [Clayton and Stolt, 1981] Clayton, R., and R. Stolt, 1981, A Born-WKBJ inversion method for acoustic reflection data: *Geophysics*, **46**, 1559–1567.
- [Collino and Tsogka, 1998] Collino, F., and C. Tsogka, 1998, Application of the PML absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media: *Rapport de recherche-institut national de recherche en informatique et en automatique*.
- [Crase et al., 1990] Crase, E., A. Pica, M. Noble, J. McDonald, and A. Tarantola, 1990, Robust elastic nonlinear waveform inversion; application to real data: *Geophysics*, **55**, 527.
- [Drossaert and Giannopoulos, 2007] Drossaert, F., and A. Giannopoulos, 2007, A non-split complex frequency-shifted PML based on recursive integration for FDTD modeling of elastic waves: *Geophysics*, **72**, T9.
- [Epanomeritakis et al., 2008] Epanomeritakis, I., V. Akçelik, O. Ghattas, and J. Bielak, 2008, A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion: *Inverse Problems*, **24**, 034015.
- [Etienne et al., 2010] Etienne, V., S. Operto, J. Virieux, and Y. Jia, 2010, Computational issues and strategies related to full waveform inversion in 3D elastic media: methodological developments.
- [Gauthier et al., 1986] Gauthier, O., J. Virieux, and A. Tarantola, 1986, Two-dimensional nonlinear inversion of seismic waveforms: Numerical results: *Geophysics*, **51**, 1387.
- [Gélis et al., 2007] Gélis, C., J. Virieux, and G. Grandjean, 2007, Two-dimensional elastic full waveform inversion using born and rylov formulations in the frequency domain: *Geophysical Journal International*, **168**, 605–633.

- [Graves, 1996] Graves, R., 1996, Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differences: *Bulletin of the Seismological Society of America*, **86**, 1091–1106.
- [Guasch and Warner, 2010] Guasch, L., and M. Warner, 2010, 3D Elastic Wavefield Inversion in the Time Domain: EAGE, **27**, 14–17.
- [Guasch et al., 2010] Guasch, L., M. Warner, I. Stekl, A. Umpleby, and N. Shah, 2010, 3D Elastic Wavefield Tomography: AGU Fall Meeting Abstracts, 03.
- [Heincke et al., 2006] Heincke, B., M. Jegen, and R. Hobbs, 2006, Joint inversion of MT, gravity and seismic data applied to sub-basalt imaging: SEG Technical Program Expanded Abstracts, **25**, 784–789.
- [Jegen et al., 2009] Jegen, M., R. Hobbs, P. Tarits, and A. Chave, 2009, Joint inversion of marine magnetotelluric and gravity data incorporating seismic constraints: Preliminary results of sub-basalt imaging off the Faroe Shelf: *Earth and Planetary Science Letters*, **282**, 47–55.
- [Johnson, 1984] Johnson, O., 1984, Three-dimensional wave equation computations on vector computers: *Proceedings of the IEEE*, **72**, 90–95.
- [Komatitsch and Tromp, 1999] Komatitsch, D., and J. Tromp, 1999, Introduction to the spectral element method for three-dimensional seismic wave propagation: *Geophys. J. Int.*, **139**, 806–822.
- [Komatitsch and Tromp, 2003] ———, 2003, A perfectly matched layer absorbing boundary condition for the second-order seismic wave equation: *Geophysical Journal International*, **154**, 146–153.
- [Komatitsch and Vilotte, 1998] Komatitsch, D., and J. Vilotte, 1998, The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures: *Bulletin of the Seismological Society of America*, **88**, 368.

- [Korenaga et al., 2001] Korenaga, J., W. Holbrook, R. Detrick, and P. Kelemen, 2001, Gravity anomalies and crustal structure at the southeast Greenland margin: *Journal of geophysical research*, **106**, 8853–8870.
- [Korenaga et al., 2000] Korenaga, J., W. Holbrook, G. Kent, P. Kelemen, R. Detrick, H. Larsen, J. Hopper, and T. Dahl-Jensen, 2000, Crustal structure of the southeast Greenland margin from joint refraction and reflection seismic tomography: *Journal of Geophysical Research*, **105**, 21591–21.
- [Kristek et al., 2002] Kristek, J., P. Moczo, and R. Archuleta, 2002, Efficient methods to simulate planar free surface in the 3D 4th-order staggered-grid finite-difference schemes: *Studia Geophysica et Geodaetica*, **46**, 355–381.
- [Lailly, 1983] Lailly, P., 1983, The seismic inverse problem as a sequence of before stack migrations: Conference on inverse scattering: theory and application, Society for Industrial and Applied Mathematics, Philadelphia, PA, 206–220.
- [Levander, 1988] Levander, A., 1988, Fourth-order finite-difference P-SV seismograms: *Geophysics*, **53**, 1425.
- [Lindman, 1975] Lindman, E. L., 1975, Free-space” boundary conditions for the time dependent wave equation: *Journal of Computational Physics*, **18**, 66 – 78.
- [Lo and Inderwiesen, 1994] Lo, T., and P. Inderwiesen, 1994, Fundamentals of seismic tomography: Soc of Exploration Geophysicists.
- [Luo and Schuster, 1990] Luo, Y., and G. Schuster, 1990, Parsimonious staggered grid finite-differencing of the wave equation: *Geophys. Res. Lett.*, **17**, 155–158.
- [Luo and Schuster, 1991] Luo, Y., and G. T. Schuster, 1991, Wave-equation travelttime inversion: *Geophysics*, **56**, 645–653.
- [Madariaga, 1976] Madariaga, R., 1976, Dynamics of an expanding circular fault: *Bulletin of the Seismological Society of America*, **66**, 639–666.

- [Marcinkovich and Olsen, 2003] Marcinkovich, C., and K. Olsen, 2003, On the implementation of perfectly matched layers in a three-dimensional fourth-order velocity-stress finite difference scheme: *J. Geophys. Res.*, **108**, 2276.
- [Minkoff, 2003] Minkoff, S., 2003, Spatial parallelism of a 3D finite difference velocity-stress elastic wave propagation code: *SIAM Journal on Scientific Computing*, **24**, 1–19.
- [Moczo et al., 2000] Moczo, P., J. Kristek, and L. Halada, 2000, 3D fourth-order staggered-grid finite-difference schemes: stability and grid dispersion: *Bulletin of the Seismological Society of America*, **90**, 587–603.
- [Moczo et al., 2002] Moczo, P., J. Kristek, V. Vavrycuk, R. Archuleta, and L. Halada, 2002, 3D heterogeneous staggered-grid finite-difference modeling of seismic motion with volume harmonic and arithmetic averaging of elastic moduli and densities: *Bulletin of the Seismological Society of America*, **92**, 3042–3066.
- [Mora, 1987] Mora, P., 1987, Nonlinear two-dimensional elastic inversion of multioffset seismic data: *Geophysics*, **52**, 1211–1228.
- [Mora, 1988] ———, 1988, Elastic wave-field inversion of reflection and transmission data: *Geophysics*, **53**, 750–759.
- [Nolet, 1987] Nolet, G., 1987, Seismic wave propagation and seismic tomography: *Seismic Tomography*, **1**.
- [Operto et al., 2004] Operto, S., C. Ravaut, L. Imrota, J. Virieux, A. Herrero, and P. Dell’Aversana, 2004, Quantitative imaging of complex structures from dense wide-aperture seismic data by multiscale travelttime and waveform inversions: a case study: *Geophysical Prospecting*, **52**, 625–651.
- [Operto et al., 2007] Operto, S., J. Virieux, P. Amestoy, J. L’Excellent, L. Giraud, and H. Ali, 2007, 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study: *Geophysics*, **72**, SM195.

- [Operto et al., 2006] Operto, S., J. Virieux, J. Dessa, and G. Pascal, 2006, Crustal seismic imaging from multifold ocean bottom seismometer data by frequency domain full waveform tomography: Application to the eastern Nankai trough: *Journal of Geophysical Research-Solid Earth*, **111**, B09306.
- [Papazachos, 2006] Papazachos, C., 2006, Implementation of a non-splitting formulation of Perfect Matching Layer in a 3D-4th order staggered-grid velocity-stress finite-difference scheme: 13 ECEES and 30 General Assembly of the ESC.
- [Pitarka, 1999] Pitarka, A., 1999, 3D elastic finite-difference modeling of seismic motion using staggered grids with nonuniform spacing: *Bulletin of the Seismological Society of America*, **89**, 54.
- [Plessix, 2006] Plessix, R., 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: *Geophysical Journal International*, **167**, 495–503.
- [Plessix, 2009] ———, 2009, Three-dimensional frequency-domain full-waveform inversion with an iterative solver: *Geophysics*, **74**, WCC149.
- [Plessix and Cao, 2011] Plessix, R., and Q. Cao, 2011, A parametrization study for surface seismic full waveform inversion in an acoustic vertical transversely isotropic medium: *Geophysical Journal International*.
- [Podvin and Lecomte, 1991] Podvin, P., and I. Lecomte, 1991, Finite difference computation of traveltimes in very contrasted velocity models: a massively parallel approach and its associated tools: *Geophysical Journal International*, **105**, 271–284.
- [Pratt and Worthington, 1990] Pratt, R., and M. Worthington, 1990, Inverse theory applied to multi-source cross-hole tomography.: *Geophysical Prospecting*, **38**, 287–310.
- [Pratt, 1999] Pratt, R. G., 1999, Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model: *Geophysics*, **64**, 888–901.

- [Pratt and Shipp, 1999] Pratt, R. G., and R. M. Shipp, 1999, Seismic waveform inversion in the frequency domain, part 2: Fault delineation in sediments using crosshole data: *Geophysics*, **64**, 902–914.
- [Randall, 1988] Randall, C., 1988, Absorbing boundary condition for the elastic wave equation: *Geophysics*, **53**, 611.
- [Randall, 1989] Randall, C. J., 1989, Absorbing boundary condition for the elastic wave equation: Velocity-stress formulation: *Geophysics*, **54**, 1141–1152.
- [Saenger et al., 2000] Saenger, E., N. Gold, and S. Shapiro, 2000, Modeling the propagation of elastic waves using a modified finite-difference grid: *Wave motion*, **31**, 77–92.
- [Sambridge and Drijkoningen, 1992] Sambridge, M., and G. Drijkoningen, 1992, Genetic algorithms in seismic waveform inversion: *Geophysical Journal International*, **109**, 323–342.
- [Santosa, 1984] Santosa, F., 1984, Inverse problems of acoustic and elastic waves: Siam.
- [Sears et al., 2008] Sears, T., S. Singh, and P. Barton, 2008, Elastic full waveform inversion of multi-component OBC seismic data: *Geophysical Prospecting*, **56**, 843–862.
- [Shen and Clapp, 2011] Shen, X., and R. Clapp, 2011, Random boundary condition for low-frequency wave propagation: SEG Technical Program Expanded Abstracts.
- [Sheriff et al., 1995] Sheriff, R., L. Geldart, and J. Lees, 1995, *Exploration seismology*: Cambridge University Press Cambridge.
- [Shin and Cha, 2008] Shin, C., and Y. Cha, 2008, Waveform inversion in the laplace domain: *Geophysical Journal International*, **173**, 922–931.
- [Shipp and Singh, 2002] Shipp, R., and S. Singh, 2002, Two-dimensional full wavefield inversion of wide-aperture marine seismic streamer data: *Geophysical Journal International*, **151**, 325–344.
- [Sirgue, 2003] Sirgue, L., 2003, Inversion de la forme d'onde dans le domain fréquentiel de données sismiques grands offsets: PhD thesis, Queen's University, Canada.

- [Sirgue et al., 2009] Sirgue, L., O. Barkved, J. Gestel, O. Askim, and J. Kommedal, 2009, 2D waveform inversion on Valhall wide-azimuth OBS: 71st Conference & Technical Exhibition, EAGE: Extended Abstracts, U, **38**.
- [Sirgue et al., 2007] Sirgue, L., J. T. Etgen, U. Albertin, and S. Brandsbergh-Dahl, 2007, System and method for 3D frequency-domain waveform inversion based on 3D time-domain modeling: Patent.
- [Sirgue and Pratt, 2004] Sirgue, L., and R. Pratt, 2004, Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies: *Geophysics*, **69**, 231.
- [Smith, 1974] Smith, W., 1974, A nonreflecting plane boundary for wave propagation problems: *Journal of Computational Physics*, **15**, 492–503.
- [Stekl and Pratt, 1998] Stekl, I., and R. Pratt, 1998, Accurate viscoelastic modeling by frequency-domain finite differences using rotated operators: *Geophysics*, **63**, 1779.
- [Stekl et al., 2007] Stekl, I., M. Warner, and A. Umpleby, 2007, 3D frequency domain waveform inversion-synthetic shallow channel example: Presented at the Proceedings of the European Association of Geoscientists and Engineers, 69th Conference and Exhibition, London, England.
- [Stekl et al., 2008] ———, 2008, Practical 3D wavefield tomography on field datasets: EAGE 70th Conference Proceeding.
- [Tarantola, 1984a] Tarantola, A., 1984a, Inversion of seismic reflection data in the acoustic approximation: *Geophysics*, **49**, 1259–1266.
- [Tarantola, 1984b] ———, 1984b, Linearized inversion of seismic reflection data: *Geophysical prospecting*, **32**, 998–1015.
- [Tarantola, 1986] ———, 1986, A strategy for nonlinear elastic inversion of seismic reflection data: *Geophysics*, **51**, 1893–1903.
- [Tarantola, 2005] ———, 2005, Inverse problem theory and methods for model parameter estimation: Society for Industrial Mathematics.

- [Vidale, 1988] Vidale, J., 1988, Finite-difference calculation of travel times: Bulletin of the Seismological Society of America, **78**, 2062.
- [Vidale, 1990] ———, 1990, Finite-difference calculation of traveltimes in three dimensions: Geophysics, **55**, 521–526.
- [Virieux, 1984] Virieux, J., 1984, Sh-wave propagation in heterogeneous media: Velocity-stress finite-difference method: Geophysics, **49**, 1933–1942.
- [Virieux, 1986] ———, 1986, P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method: Geophysics, **51**, 889.
- [Virieux and Operto, 2009] Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, WCC1.
- [Wang and Tang, 2003] Wang, T., and X. Tang, 2003, Finite-difference modeling of elastic wave propagation: A nonsplitting perfectly matched layer approach: Geophysics, **68**, 1749.
- [Wang and Rao, 2009] Wang, Y., and Y. Rao, 2009, Reflection seismic waveform tomography: Journal of Geophysical research, **114**, B03304.
- [Warner et al., 2007a] Warner, M., I. Stekl, and A. Umpleby, 2007a, Full Wavefield Seismic Tomography - Iterative Forward Modelling in 3D: EAGE 69th Conference Proceeding.
- [Warner et al., 2007b] ———, 2007b, Full waveform seismic tomography-iterative forward modelling in 3D: Presented at the 69th EAGE meeting, London, UK, Expanded Abstracts.
- [Warner et al., 2008] ———, 2008, Efficient and Effective 3D Wavefield Tomography: EAGE 70th Conference Proceeding.
- [Yokota and Matsushima, 2004] Yokota, T., and J. Matsushima, 2004, Seismic waveform tomography in the frequency-space domain: selection of the optimal temporal frequency for inversion: Exploration Geophysics, **35**, 19–24.

[Yomogida and Etgen, 1993] Yomogida, K., and J. Etgen, 1993, 3-D wave propagation in the Los Angeles basin for the Whittier-Narrows earthquake: Bulletin of the Seismological Society of America, **83**, 1325–1344.



## APPENDIX A

FD equations inside the model (in the PML the spatial FD operator is of 2<sup>nd</sup> order instead of 4<sup>th</sup> as shown in Appendix B). The medium parameters are not defined in the same positions as some of the variables involved in the calculation and their values are calculated according to equations 2.22 and 2.23.

## PARTICLE-VELOCITIES:

$$\begin{aligned}
U_{i+1/2,j,k}^{t+1/2} = & U_{i+1/2,j,k}^{t-1/2} + \frac{\Delta t}{h\rho} \left[ c_1(T_{i+1,j,k}^{xx,t} - T_{i,j,k}^{xx,t}) - c_2(T_{i+2,j,k}^{xx,t} - T_{i-1,j,k}^{xx,t}) \right. \\
& + c_1(T_{i+1/2,j+1/2,k}^{xy,t} - T_{i+1/2,j-1/2,k}^{xy,t}) - c_2(T_{i+1/2,j+3/2,k}^{xy,t} - T_{i+1/2,j-3/2,k}^{xy,t}) \\
& \left. + c_1(T_{i+1/2,j,k+1/2}^{xz,t} - T_{i+1/2,j,k-1/2}^{xz,t}) - c_2(T_{i+1/2,j,k+3/2}^{xz,t} - T_{i+1/2,j,k-3/2}^{xz,t}) \right] \quad (\text{A-1})
\end{aligned}$$

$$\begin{aligned}
V_{i,j+1/2,k}^{t+1/2} &= V_{i,j+1/2,k}^{t-1/2} + \frac{\Delta t}{h\rho} \left[ c_1(T_{i+1/2,j+1/2,k}^{xy,t} - T_{i-1/2,j+1/2,k}^{xy,t}) - c_2(T_{i+3/2,j+1/2,k}^{xy,t} \right. \\
&\quad - T_{i-3/2,j+1/2,k}^{xy,t}) + c_1(T_{i,j+1,k}^{yy,t} - T_{i,j,k}^{yy,t}) - c_2(T_{i,j+2,k}^{yy,t} - T_{i,j-1,k}^{yy,t}) \\
&\quad \left. + c_1(T_{i,j+1/2,k+1/2}^{yz,t} - T_{i,j+1/2,k-1/2}^{yz,t}) - c_2(T_{i,j+1/2,k+3/2}^{yz,t} - T_{i,j+1/2,k-3/2}^{yz,t}) \right] 
\end{aligned} \tag{A-2}$$

$$\begin{aligned} W_{i,j,k+1/2}^{t+1/2} = & \quad W_{i,j,k+1/2}^{t-1/2} + \frac{\Delta t}{h\rho} \left[ c_1(T_{i+1/2,j,k+1/2}^{xz,t} - T_{i-1/2,j,k+1/2}^{xz,t}) - c_2(T_{i+3/2,j,k+1/2}^{xz,t} \right. \\ & - T_{i-3/2,j,k+1/2}^{xz,t}) + c_1(T_{i,j+1/2,k+1/2}^{yz,t} - T_{i,j-1/2,k+1/2}^{yz,t}) - c_2(T_{i,j+3/2,k+1/2}^{yz,t} \\ & - T_{i,j-3/2,k+1/2}^{yz,t}) + c_1(T_{i,j,k+1}^{zz,t} - T_{i,j,k}^{zz,t}) - c_2(T_{i,j,k+2}^{zz,t} - T_{i,j,k-1}^{zz,t}) \left. \right] \quad (\text{A-3}) \end{aligned}$$

STRESSES:

$$\begin{aligned} T^{xx,t+1} &= T^{xx,t} + \frac{\Delta t}{h} \left\{ (\lambda + 2\mu) \left[ c_1(U_{i+1/2,j,k}^{t+1/2} - U_{i-1/2,j,k}^{t+1/2}) - c_2(U_{i+3/2,j,k}^{t+1/2} - U_{i-3/2,j,k}^{t+1/2}) \right] \right. \\ &\quad + \lambda \left[ c_1(V_{i,j+1/2,k}^{t+1/2} - V_{i,j-1/2,k}^{t+1/2}) - c_2(V_{i,j+3/2,k}^{t+1/2} - V_{i,j-3/2,k}^{t+1/2}) \right. \\ &\quad \left. \left. + c_1(W_{i,j,k+1/2}^{t+1/2} - W_{i,j,k-1/2}^{t+1/2}) - c_2(W_{i,j,k+3/2}^{t+1/2} - W_{i,j,k-3/2}^{t+1/2}) \right] \right\} \end{aligned} \quad (\text{A-4})$$

$$\begin{aligned} T^{yy,t+1} &= T^{yy,t} + \frac{\Delta t}{h} \left\{ (\lambda + 2\mu) \left[ c_1(V_{i,j+1/2,k}^{t+1/2} - V_{i,j-1/2,k}^{t+1/2}) - c_2(V_{i,j+3/2,k}^{t+1/2} - V_{i,j-3/2,k}^{t+1/2}) \right] \right. \\ &\quad + \lambda \left[ c_1(U_{i+1/2,j,k}^{t+1/2} - U_{i-1/2,j,k}^{t+1/2}) - c_2(U_{i+3/2,j,k}^{t+1/2} - U_{i-3/2,j,k}^{t+1/2}) \right. \\ &\quad \left. \left. + c_1(W_{i,j,k+1/2}^{t+1/2} - W_{i,j,k-1/2}^{t+1/2}) - c_2(W_{i,j,k+3/2}^{t+1/2} - W_{i,j,k-3/2}^{t+1/2}) \right] \right\} \end{aligned} \quad (\text{A-5})$$

$$\begin{aligned} T^{zz,t+1} &= T^{zz,t} + \frac{\Delta t}{h} \left\{ (\lambda + 2\mu) \left[ c_1(W_{i,j,k+1/2}^{t+1/2} - W_{i,j,k-1/2}^{t+1/2}) - c_2(W_{i,j,k+3/2}^{t+1/2} - W_{i,j,k-3/2}^{t+1/2}) \right] \right. \\ &\quad + \lambda \left[ c_1(U_{i+1/2,j,k}^{t+1/2} - U_{i-1/2,j,k}^{t+1/2}) - c_2(U_{i+3/2,j,k}^{t+1/2} - U_{i-3/2,j,k}^{t+1/2}) \right. \\ &\quad \left. \left. + c_1(V_{i,j+1/2,k}^{t+1/2} - V_{i,j-1/2,k}^{t+1/2}) - c_2(V_{i,j+3/2,k}^{t+1/2} - V_{i,j-3/2,k}^{t+1/2}) \right] \right\} \end{aligned} \quad (\text{A-6})$$

$$\begin{aligned} T^{xy,t+1} &= T^{xy,t} + \frac{\Delta t}{h} \mu \left[ c_1(U_{i+1/2,j+1,k}^{t+1/2} - U_{i+1/2,j,k}^{t+1/2}) - c_2(U_{i+1/2,j+2,k}^{t+1/2} - U_{i+1/2,j-1,k}^{t+1/2}) \right. \\ &\quad \left. + c_1(V_{i+1,j+1/2,k}^{t+1/2} - V_{i,j+1/2,k}^{t+1/2}) - c_2(V_{i+2,j+1/2,k}^{t+1/2} - V_{i-1,j+1/2,k}^{t+1/2}) \right] \end{aligned} \quad (\text{A-7})$$

$$\begin{aligned} T^{xz,t+1} &= T^{xz,t} + \frac{\Delta t}{h} \mu \left[ c_1(U_{i+1/2,j,k+1}^{t+1/2} - U_{i+1/2,j,k}^{t+1/2}) - c_2(U_{i+1/2,j,k+2}^{t+1/2} - U_{i+1/2,j,k-1}^{t+1/2}) \right. \\ &\quad \left. + c_1(W_{i+1,j,k+1/2}^{t+1/2} - W_{i,j,k+1/2}^{t+1/2}) - c_2(W_{i+2,j,k+1/2}^{t+1/2} - W_{i-1,j,k+1/2}^{t+1/2}) \right] \end{aligned} \quad (\text{A-8})$$

$$\begin{aligned} T^{yz,t+1} &= T^{yz,t} + \frac{\Delta t}{h} \mu \left[ c_1(V_{i,j+1/2,k+1}^{t+1/2} - V_{i,j+1/2,k}^{t+1/2}) - c_2(V_{i,j+1/2,k+2}^{t+1/2} - V_{i,j+1/2,k-1}^{t+1/2}) \right. \\ &\quad \left. + c_1(W_{i,j+1,k+1/2}^{t+1/2} - W_{i,j,k+1/2}^{t+1/2}) - c_2(W_{i,j+2,k+1/2}^{t+1/2} - W_{i,j-1,k+1/2}^{t+1/2}) \right] \end{aligned} \quad (\text{A-9})$$

where  $i, j, k$  correspond to  $x, y, z$  coordinates and  $t$  is the time-step number. The elastic tensor is symmetric, which means that  $T^{lm}$  is equal to  $T^{ml}$ , with  $m, l \in \{x, y, z\}$ .

## APPENDIX B

The following development is from Papazachos *et al* in [Papazachos, 2006], where I changed the nomenclature for the sake of consistency within the present report.

As mentioned in the Chapter 2, stretching the coordinates in the absorbing layer leads to a damping in the amplitudes of the wave while they are travelling inside the PML. Such a stretch can be written in the  $\hat{n}$  direction as:

$$\frac{\partial}{\partial \tilde{n}} = \frac{i\omega}{i\omega + d(n)} \quad (\text{A-1})$$

where  $\tilde{n}$  is the new complex coordinate,  $\omega$  is the frequency and  $d(n)$  is the damping discrete function shown in figure 2.10(b). The corresponding equation of motion is then:

$$\rho \frac{\partial \mathbf{v}}{\partial t} = \tilde{\nabla} \cdot \boldsymbol{\tau} \quad (\text{A-2})$$

where  $\mathbf{v}$  is the particle velocity vector and  $\boldsymbol{\tau}$  is the stress tensor. If we take the inverse Fourier Transform of the equation A-1

$$\mathcal{F}^{-1} \left( 1 - \frac{d(n)}{d(n) + i\omega} \right) = \delta(t) - d(n)e^{-d(n)t} \quad (\text{A-3})$$

and

$$\mathcal{F}^{-1} \left( 1 - \frac{\partial}{\partial \tilde{n}} \right) = (1 + \phi_n) \frac{\partial}{\partial n}, \quad \phi_n = -d(n)e^{-d(n)t}_* \quad (\text{A-4})$$

which leads to the following expression for the time domain equation of motion in the

PML (considering damping in all directions):

$$\rho \frac{\partial \mathbf{v}}{\partial t} = \begin{pmatrix} (1 + \phi_x) \frac{\partial \tau_{xx}}{\partial x} & (1 + \phi_y) \frac{\partial \tau_{yx}}{\partial y} & (1 + \phi_z) \frac{\partial \tau_{zx}}{\partial z} \\ (1 + \phi_x) \frac{\partial \tau_{xy}}{\partial x} & (1 + \phi_y) \frac{\partial \tau_{yy}}{\partial y} & (1 + \phi_z) \frac{\partial \tau_{zy}}{\partial z} \\ (1 + \phi_x) \frac{\partial \tau_{xz}}{\partial x} & (1 + \phi_y) \frac{\partial \tau_{yz}}{\partial y} & (1 + \phi_z) \frac{\partial \tau_{zz}}{\partial z} \end{pmatrix} \quad (\text{A-5})$$

Applying the same coordinate stretching to the constitutive laws and applying the inverse Fourier Transform —in order to obtain the expression in the time domain— leads to:

$$\frac{\partial \boldsymbol{\tau}}{\partial t} = \mathbf{c}_{ijkl} \frac{\partial}{\partial \mathbf{t}} \begin{pmatrix} (1 + \phi_x) \frac{\partial u}{\partial x} & \frac{1}{2} \left( (1 + \phi_x) \frac{\partial v}{\partial x} + (1 + \phi_y) \frac{\partial u}{\partial y} \right) & \frac{1}{2} \left( (1 + \phi_x) \frac{\partial w}{\partial x} + (1 + \phi_z) \frac{\partial u}{\partial z} \right) \\ \frac{1}{2} \left( (1 + \phi_x) \frac{\partial v}{\partial x} + (1 + \phi_y) \frac{\partial u}{\partial y} \right) & (1 + \phi_y) \frac{\partial v}{\partial y} & \frac{1}{2} \left( (1 + \phi_y) \frac{\partial w}{\partial y} + (1 + \phi_z) \frac{\partial v}{\partial z} \right) \\ \frac{1}{2} \left( (1 + \phi_x) \frac{\partial w}{\partial x} + (1 + \phi_z) \frac{\partial u}{\partial z} \right) & \frac{1}{2} \left( (1 + \phi_y) \frac{\partial w}{\partial y} + (1 + \phi_z) \frac{\partial v}{\partial z} \right) & (1 + \phi_z) \frac{\partial w}{\partial z} \end{pmatrix} \quad (\text{A-6})$$

where  $\mathbf{c}_{ijkl}$  is the stiffness tensor. Discretisation of equations (A-6) and (A-5) using 2<sup>nd</sup> order in time and space finite differences leads to:

### PARTICLE-VELOCITIES:

$$\begin{aligned} U_{i+1/2,j,k}^{t+1/2} = & U_{i+1/2,j,k}^{t-1/2} + \frac{\Delta t}{h\rho} \left[ (T_{i+1,j,k}^{xx,t} - T_{i,j,k}^{xx,t}) + (T_{i+1/2,j+1/2,k}^{xy,t} - T_{i+1/2,j-1/2,k}^{xy,t}) \right. \\ & \left. + (T_{i+1/2,j,k+1/2}^{xz,t} - T_{i+1/2,j,k-1/2}^{xz,t}) + R_{i,j+1/2,k+1/2}^{xx,t} + R_{i,j+1/2,k+1/2}^{xy,t} + R_{i,j+1/2,k+1/2}^{xz,t} \right] \end{aligned} \quad (\text{A-7})$$

$$\begin{aligned} V_{i,j+1/2,k}^{t+1/2} = & V_{i,j+1/2,k}^{t-1/2} + \frac{\Delta t}{h\rho} \left[ (T_{i+1/2,j+1/2,k}^{xy,t} - T_{i-1/2,j+1/2,k}^{xy,t}) + (T_{i,j+1,k}^{yy,t} - T_{i,j,k}^{yy,t}) \right. \\ & \left. + (T_{i,j+1/2,k+1/2}^{yz,t} - T_{i,j+1/2,k-1/2}^{yz,t}) + R_{i+1/2,j,k+1/2}^{yx,t} + R_{i+1/2,j,k+1/2}^{yy,t} + R_{i+1/2,j,k+1/2}^{yz,t} \right] \end{aligned} \quad (\text{A-8})$$

$$\begin{aligned} W_{i,j,k+1/2}^{t+1/2} = & W_{i,j,k+1/2}^{t-1/2} + \frac{\Delta t}{h\rho} \left[ (T_{i+1/2,j,k+1/2}^{xz,t} - T_{i-1/2,j,k+1/2}^{xz,t}) + (T_{i,j+1/2,k+1/2}^{yz,t} - T_{i,j-1/2,k+1/2}^{yz,t}) \right. \\ & \left. + (T_{i,j,k+1}^{zz,t} - T_{i,j,k}^{zz,t}) + R_{i+1/2,j+1/2,k}^{zx,t} + R_{i+1/2,j+1/2,k}^{zy,t} + R_{i+1/2,j+1/2,k}^{zz,t} \right] \end{aligned} \quad (\text{A-9})$$

$$R_{l,m,n}^{ij,t} = e^{-d_{l,m,n}^j \Delta t} R_{l,m,n}^{ij,t-1} - \frac{h}{2} d_{l,m,n}^j \Delta t \left[ e^{-d_{l,m,n}^j \Delta t} \frac{\partial T_{ij}^{t-1}}{\partial j} \Big|_{l,m,n} + \frac{\partial T_{ij}^t}{\partial j} \Big|_{l,m,n} \right], \quad i,j \in x,y,z \quad (\text{A-10})$$

STRESSES:

$$\begin{aligned} T^{xx,t+1} &= T^{xx,t} + \frac{\Delta t}{h} \left\{ (\lambda + 2\mu) \left[ (U_{i+1/2,j,k}^{t+1/2} - U_{i-1/2,j,k}^{t+1/2}) \right] \right. \\ &\quad \left. + \lambda \left[ (V_{i,j+1/2,k}^{t+1/2} - V_{i,j-1/2,k}^{t+1/2}) + (W_{i,j,k+1/2}^{t+1/2} - W_{i,j,k-1/2}^{t+1/2}) \right] \right. \\ &\quad \left. P_{i+1/2,j+1/2,k+1/2}^{xx,t+1/2} + P_{i+1/2,j+1/2,k+1/2}^{yy,t+1/2} + P_{i+1/2,j+1/2,k+1/2}^{zz,t+1/2} \right\} \quad (\text{A-11}) \end{aligned}$$

$$\begin{aligned} T^{yy,t+1} &= T^{yy,t} + \frac{\Delta t}{h} \left\{ (\lambda + 2\mu) \left[ (V_{i,j+1/2,k}^{t+1/2} - V_{i,j-1/2,k}^{t+1/2}) \right] \right. \\ &\quad \left. + \lambda \left[ (U_{i+1/2,j,k}^{t+1/2} - U_{i-1/2,j,k}^{t+1/2}) + (W_{i,j,k+1/2}^{t+1/2} - W_{i,j,k-1/2}^{t+1/2}) \right] \right. \\ &\quad \left. P_{i+1/2,j+1/2,k+1/2}^{xx,t+1/2} + P_{i+1/2,j+1/2,k+1/2}^{yy,t+1/2} + P_{i+1/2,j+1/2,k+1/2}^{zz,t+1/2} \right\} \quad (\text{A-12}) \end{aligned}$$

$$\begin{aligned} T^{zz,t+1} &= T^{zz,t} + \frac{\Delta t}{h} \left\{ (\lambda + 2\mu) \left[ (W_{i,j,k+1/2}^{t+1/2} - W_{i,j,k-1/2}^{t+1/2}) \right] \right. \\ &\quad \left. + \lambda \left[ (U_{i+1/2,j,k}^{t+1/2} - U_{i-1/2,j,k}^{t+1/2}) + (V_{i,j+1/2,k}^{t+1/2} - V_{i,j-1/2,k}^{t+1/2}) \right] \right. \\ &\quad \left. P_{i+1/2,j+1/2,k+1/2}^{xx,t+1/2} + P_{i+1/2,j+1/2,k+1/2}^{yy,t+1/2} + P_{i+1/2,j+1/2,k+1/2}^{zz,t+1/2} \right\} \quad (\text{A-13}) \end{aligned}$$

$$\begin{aligned} T^{xy,t+1} &= T^{xy,t} + \frac{\Delta t}{h} \mu \left[ (U_{i+1/2,j+1,k}^{t+1/2} - U_{i+1/2,j,k}^{t+1/2}) + (V_{i+1,j+1/2,k}^{t+1/2} - V_{i,j+1/2,k}^{t+1/2}) \right. \\ &\quad \left. P_{i,j,k+1/2}^{yx,t+1/2} + P_{i,j,k+1/2}^{xy,t+1/2} \right] \quad (\text{A-14}) \end{aligned}$$

$$\begin{aligned} T^{xz,t+1} &= T^{xz,t} + \frac{\Delta t}{h} \mu \left[ (U_{i+1/2,j,k+1}^{t+1/2} - U_{i+1/2,j,k}^{t+1/2}) + (W_{i+1,j,k+1/2}^{t+1/2} - W_{i,j,k+1/2}^{t+1/2}) \right. \\ &\quad \left. P_{i,j+1/2,k}^{zx,t+1/2} + P_{i,j+1/2,k}^{xz,t+1/2} \right] \quad (\text{A-15}) \end{aligned}$$

$$\begin{aligned}
T^{yz,t+1} &= T^{yz,t} + \frac{\Delta t}{h} \mu \left[ (V_{i,j+1/2,k+1}^{t+1/2} - V_{i,j+1/2,k}^{t+1/2}) + (W_{i,j+1,k+1/2}^{t+1/2} - W_{i,j,k+1/2}^{t+1/2}) \right. \\
&\quad \left. P_{i+1/2,j,k}^{yz,t+1/2} + P_{i+1/2,j,k}^{zy,t+1/2} \right]
\end{aligned} \tag{A-16}$$

$$\begin{aligned}
P_{l,m,n}^{ij,t+1/2} &= e^{-d_{l,m,n}^j \Delta t} P_{l,m,n}^{ij,t-1/2} - \frac{1}{2} d_{l,m,n}^j \Delta t \left[ e^{-d_{l,m,n}^j \Delta t} \frac{\partial V_i^{t-1/2}}{\partial j} \Big|_{l,m,n} \right. \\
&\quad \left. + \frac{\partial V_i^{t+1/2}}{\partial j} \Big|_{l,m,n} \right], \quad i, j \in x, y, z
\end{aligned} \tag{A-17}$$

where  $d^j$  is the damping function along the  $j$  direction.

