Tuning Up Global Optimization Techniques to Solve the Reliability Problem in Nonlinear Geophysical Inversions

Sergio Alberto Abreo Carrillo

This dissertation is submitted for the degree of

Doctor in Engineering

Advisor: Dr-Ing. Oscar M. Reyes T.

Advisor: Ph.D. Ana B. Ramirez S.

Universidad Industrial de Santander

Faculty of Physicomechanical Engineering

School of Electrical, Electronic and Telecommunications Engineering

Bucaramanga

2018

To the most important people in my life. My parents Alberto and Elizabeth, my brother Daniel, my uncle Elberto and my beautiful princess Cindy.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisors Professors Oscar M. Reyes T. and Ana B. Ramirez S. for the continuous support in my doctoral research, for their patience, motivation, and guidance. I could not have imagined having better advisors for my doctoral study.

I would like to thanks to all the CPS professors and students that helped me during this process; in special to the Electronic Engineers Juan Lopez, Jheyston Serrano, David Abreo, Fabian Noriega and Johan Suarez. Thanks guys for all the effort made in your master's thesis and in the development of codes that takes advantage of GPU potentiality.

I would also like to thank my family. To my parents Alberto and Elizabeth, my brother Daniel and my uncle Elberto for supporting me spiritually and emotionally at each challenge that I take in my life.

Finally, I would like to thank Ecopetrol and Colciencias for the financial support to develop this doctoral research. In special to the Ecopetrol official Herling Gonzalez for all his support in this project. This work has been supported by Colombian Oil Company ECOPETROL and COLCIENCIAS as a part of the research project grant No. 0266-2013. The author gratefully acknowledges to the Industrial University of Santander.

Glossary

d Cost function dimension.

D Number of cost function dimensions.

e Current experiment.

E Number of experiments.

H Hessian Matrix.

k Current iteration.

K Number of iterations.

n Current sample.

P Number of particles.

r Current receiver.

R Number of receivers.

s Current source.

S Number of sources.

t Current time.

T Total time.

ω Current frequency.

w Weight of inertia.

W Number of frequencies.

c_1 Local search weight.

c_2 Global search weight.

$C_{s,r}$ Cycle-skipping coefficient.

d_{mod} Modeled data.

d_{obs} Observed data.

dt Particles step size.

Δt Time step.

Δx Spatial step in the x direction.

Δz Spatial step in the z direction.

f_0 Central frequency.

g_s Source wavelet.

i^{th} Swarm's particle.

N_t Number of time steps.

N_x Number of points in the x direction.

N_z Number of points in the z direction.

$p(x,z)$ Seismic wavefield.

R_1 Random value uniformly distributed.

R_2 Random value uniformly distributed.

t_0 Time delay.

$\boldsymbol{v}_{i,d}^{k}$ Current particle velocity.

$\boldsymbol{v}_{i,d}^{k+1}$ New particle velocity.

$\boldsymbol{x}_{i,d}^{k}$ Current particle position.

$\boldsymbol{x}_{i,d}^{k+1}$ New particle position.

$\boldsymbol{x}_{i,d}^{l}$ Best particle location.

$\boldsymbol{x}_{g,d}$ Best swarm location.

$\mathfrak{D}$ Data space.

$F\{\mathbf{v}_{i,d}^{k}\}$ Cost function operator.

$H\{\cdot\}$ Hilbert operator.

$\mathbf{J}(\cdot)$ Jacobian matrix.

$L(\mathbf{v})[\cdot]$ d'Alembertian operator.

$L^{*}(\mathbf{v})[\cdot]$ The adjoint of the d'Alembertian operator.

$M(\cdot,\cdot)$ Functional that measures the difference between d_{obs} and d_{mod}.

$\mathfrak{M}$ Model space.

$\mathfrak{M}^{*}$ Dual space of $\mathfrak{M}$.

$\mathscr{R}$ Resolution matrix.

$R\{\cdot\}$ Time-domain finite-difference operator.

$R^{T}\{\cdot\}$ Transpose of $R\{\cdot\}$.

$T\{\cdot\}$ Operator that maps $\mathfrak{D} \mapsto \mathfrak{T}$.

$\mathfrak{T}$ Transformed space.

$\mathfrak{T}^{*}$ Dual space of $\mathfrak{T}$.

$\mathfrak{W}$ Wavefield space.

$\mathfrak{W}^*$ Dual space of $\mathfrak{W}$.

$\langle \cdot, \cdot \rangle_d$ inner product in $\mathfrak{D}$.

$\langle \cdot, \cdot \rangle_w$ inner product in $\mathfrak{W}$.

$\langle \cdot, \cdot \rangle_m$ inner product in $\mathfrak{M}$.

θ_{mod} Modeled data phase.

θ_{obs} Observed data phase.

λ First order adjoint field.

μ Second order adjoint field.

$\mathscr{L}$ The Lagrangian.

Acronyms

CDF Cumulative Density Function.

CPU Central Processing Unit.

CPML Convolutional Perfectly Matched Layer.

CUDA Compute Unified Device Architecture.

ECDF Empirical Cumulative Density Function.

FDTD Finite Differences in Time Domain.

FWI Full Waveform Inversion.

GPU Graphics Processing Unit.

i.i.d. independent and identically distributed.

KS Kolmogorov, Smirnov.

MAP Maximum A Posteriori.

ML Maximum Likelihood.

MMSE Minimum Mean Square Error.

PSO Particle Swarm Optimization.

PDF Probability Density Function.

QCs Quality Controls.

RAM Random Access Memory.

RMSE Root Mean Square Error.

SMs Streaming Multiprocessors.

SOASM Second Order Adjoint State Method.

VR Variance reduction.

RESUMEN

TÍTULO: PUESTA A PUNTO DE TÉCNICAS DE OPTIMIZACIÓN GLOBAL PARA RESOLVER EL PROBLEMA DE FIABILIDAD EN INVERSIONES GEOFÍSICAS NO LINEALES [1]

AUTOR: SERGIO ALBERTO ABREO CARRILLO [2]

PALABRAS CLAVE: INVERSIÓN DE ONDA COMPLETA, INVERSIÓN GEOFÍSICA NO LINEAL, OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS, CUANTIFICACIÓN DE LA INCERTIDUMBRE.

DESCRIPCIÓN:

La inversión de datos geofísicos es un proceso que implica la solución de un problema no-lineal, que usualmente esta mal puesto y demanda un elevado costo computacional. Es por ello, que en el estado del arte se han desarrollado diferentes técnicas para tratar de resolver este problema. En este trabajo doctoral, se propone una metodología novedosa en la que el proceso de inversión de datos geofísicos se aborda desde el punto de vista de la optimización; usando una técnica de optimización global (optimización por enjambre de partículas, PSO por su nombre en ingles) en conjunto con una técnica de optimización local (inversión de onda completa, FWI por su nombre en ingles) para tratar de resolver el problema. Además, se propone el uso de la transformada de Hilbert en conjunto con la matriz Hessiana para identificar la calidad de los resultados.

La metodología propuesta utiliza arquitecturas de cómputo en paralelo (unidades de procesamiento gráfico, GPU por su nombre en ingles) para reducir el costo computacional de las tres etapas formuladas. La validación de la metodología se hace a través del uso de cuatro modelos geológicos sintéticos de prueba, los cuales se pueden considerar como complejos debido a que contienen variaciones laterales y verticales de velocidad, zonas de falla e intrusiones de alta velocidad (IAV). Los modelos elegidos son: el Marmousi, Canadian foothills, Hess y BP.

La metodología propuesta en esta tesis doctoral prueba que es posible usar PSO en conjunto con la FWI para obtener resultados muy buenos en modelos geológicos con IAV menores o iguales al 6% del area de interés presentando dificultades para IAV mayores al 20%. Además, la información ofrecida por la transformada de Hilbert permitió identificar cuando el proceso de inversión estaba avanzando correcta o incorrectamente.

[1] Tesis Doctoral

[2] Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Directores: Dr-Ing. Oscar M. Reyes T. y Ph.D. Ana B. Ramirez S.

ABSTRACT

TITLE: TUNING UP GLOBAL OPTIMIZATION TECHNIQUES TO SOLVE THE RELIABILITY PROBLEM IN NONLINEAR GEOPHYSICAL INVERSIONS [3]

AUTHOR: SERGIO ALBERTO ABREO CARRILLO [4]

KEYWORDS: FULL WAVEFORM INVERSION, NONLINEAR GEOPHYSICAL INVERSION, PARTICLE SWARM OPTIMIZATION, UNCERTAINTY QUANTIFICATION.

DESCRIPTION:

The inversion of geophysical data is a process that involves the solution of a non-linear problem, which is usually ill-posed and demands a high computational cost. Therefore, in the state of the art, different techniques have been developed to try to solve this problem. In this doctoral dissertation, a novel methodology is proposed in which the geophysical data processing is addressed from an optimization point of view where a global optimization technique (Particle Swarm Optimization, PSO) is used together with a local optimization technique (Full Waveform Inversion, FWI) to solve the problem. Besides, the use of the Hilbert transform together with the Hessian matrix is proposed to value the quality of the results.

The proposed methodology uses parallel computing architectures (i.e. Graphical Processing Units, GPUs) to reduce the computational cost of the three formulated stages. The validation of the methodology is done through the use of four synthetic geological models, which can be considered complex since they contain lateral and vertical velocity fluctuations, fault zones and strong velocity variations (SVV). The chosen models are: the Marmousi, Canadian Foothills, Hess and BP.

The methodology proposed in this doctoral dissertation proves that it is possible to use PSO together with FWI to obtain high-quality results in geological models with an SVV $\leq 6\%$ of the area of interest presenting difficulties for SVV $> 20\%$. Besides, the obtained results suggest that information offered by the Hilbert transform can be used as a quality control (QC) during the inversion process.

[3] Doctoral Dissertation

[4] Faculty of Physico-mechanical Engineering. School of Electrical, Electronic and Telecommunications Engineering. Advisors: Dr-Ing. Oscar M. Reyes T. and Ph.D. Ana B. Ramirez S.

Table of Contents

Introduction **29**

1.1 Motivation 29

1.1.1 Geophysical Framework 31

1.2 Thesis Main Contributions and Organization 31

2 First Challenge: Starting Point **35**

2.1 Particle Swarm Optimization 36

2.1.1 Parameters Selection 39

2.1.2 Cost function 40

2.2 Transformed spaces and Metrics 41

2.2.1 Time domain 42

2.2.2 Frequency domain 43

2.2.3 Hilbert domain 44

2.3 Workflow 45

2.4 Statistical Framework 48

2.5 Experimental Description 51

2.6 Discussion 60

3 Second Challenge: Local optimization technique **65**

3.1 Step Forward 66

3.1.1 Linear interpolation 67

3.1.2 L-BFGS 67

3.2 Stability and Numerical Dispersion 69

3.2.1 FDTD 69

3.3 FWI Computational Strategies 72
3.3.1 Gradient Computation 72
3.3.2 GPU Implementation 73
3.3.3 FWI Strategies 74
3.4 Multi-Scale FWI 77
3.4.1 Experiment Description 77
3.5 Discussion 81

4 Third Challenge: Uncertainty Quantification 85

4.1 Hessian Matrix Formulation 86
4.1.1 First Order Adjoint State Method 86
4.1.2 Second Order Adjoint State Method 88
4.2 Hessian-vector-products 90
4.2.1 Computing the derivative of the p field 90
4.2.2 Computing the adjoint field λ 91
4.2.3 Computing the derivative of α 92
4.2.4 Computing the adjoint field μ 92
4.2.5 Computing the second derivate of the p field 93
4.2.6 Hessian Kernels 93
4.2.7 Computing the kernels 97
4.3 Hessian matrix 98
4.3.1 Computational Cost 101
4.4 Covariance and Resolution Matrices 102
4.5 Uncertainty Quantification 103
4.6 Discussion 113

5 Testing the Proposed Methodology 115

5.1 Experiment Setup 116
5.2 Marmousi Model 118
5.2.1 Final Comments 128

5.3 Canadian Foothill Model 131
5.3.1 Final Comments 138
5.4 Hess Model 139
5.4.1 Final Comments 146
5.5 BP 2004 Model 150
5.5.1 Final Comments 159
5.6 Discussion 160

6 Conclusions and Future work **163**
6.1 PSO Exploration 163
6.2 FWI 164
6.3 Uncertainty Quantification 166

Appendix A Stability and Numerical Dispersion **173**

Appendix B FWI implementation using Strategy I **177**

List of Figures

1.1 Scheme of the methodology proposed to address the Geophysical inversion problem from an optimization point of view. 30

1.2 Seismic acquisition elements: source (1), interfaces (2), receivers (3) and data (4). (a) On-shore and (b) off-shore acquisitions. 32

1.3 Real seismic trace. 32

1.4 Example of one shot gather. 33

2.1 Misfit function for truly nonlinear inverse problems. 35

2.2 Convex function in two variables. 39

2.3 PSO search space exploration. 39

2.4 Schematic of cycle-skipping artifacts in Full Waveform Inversion (FWI). 42

2.5 Calculation of cycle-skipping coefficients in frequency domain 44

2.6 Calculation of cycle-skipping coefficients in Hilbert domain 46

2.7 Workflow of the full implementation 46

2.8 Estimated velocity model from the i^{th} swarm's particle at iteration k. 47

2.9 Setting the Particle Swarm Optimization (PSO) parameters P and K for the geophysical inversion with $w = 0.7$, $c_1 = 0.8$, $c_2 = 1.1$, $D = 211 \times 68$ and $E = 100$. 49

2.10 Statistical behavior of a random dimension for different number of experiments in three different domains. 50

2.11 Cube with the best results after the 100 PSO experiments. 51

2.12 Diffracting square model. 52

2.13 PSO and FWI results using correlation metric. 53

2.14 PSO and FWI results using frequency metric. 54

2.15 PSO and FWI results using Hilbert metric. 54
2.16 Area illuminated by the array of sources and receivers during the acquisition. . 54
2.17 Spectra of source wavelet (g_s) used for the PSO and the multi-scale FWI tests. . 55
2.18 Seismic trace spectra after using PSO and the multi-scale FWI. 55
2.19 Execution time of the PSO implementation for different values of K and P with $w = 0.7$, $c_1 = 0.8$, $c_2 = 1.1$, $D = 211 \times 68$ and $E = 100$. 56
2.20 Multi-metric behavior when the ℓ_2-norm is applied over the difference between d_{obs} and d_{mod}. 57
2.21 Multi-metric behavior when the ℓ_2-norm is applied over the difference between the original velocity model and $fwi\{\hat{\boldsymbol{x}}_{g,d}\}$. 57
2.22 Comparison between the empirical data and the estimated Probability Density Function (PDF). 59
2.23 Variance of the 100 best results in time using a) lineal scale and b) logarithmic scale $(20 \times \log(dB))$. 59
2.24 Variance of the 100 best results in frequency using a) lineal scale and b) logarithmic scale $(20 \times \log(dB))$. 60
2.25 Variance of the 100 best results in Hilbert domain using a) lineal scale and b) logarithmic scale $(20 \times \log(dB))$. 60
2.26 Correlation matrices of the 100 best results in (a) time, (b) frequency and (c) Hilbert domains. 61

3.1 Linear interpolation principle. 68
3.2 Numerical dispersion curves for different α values when $\theta = 0$, $\frac{\pi}{2}$, π, $\frac{3\pi}{2}$, or 2π. 71
3.3 Numerical dispersion curves for different α values when $\theta = \frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$, or $\frac{7\pi}{4}$. 71
3.4 Snapshots of the propagation field p_s. 72
3.5 Snapshots of the backpropagation field q_s. 72
3.6 Initial velocity model for multi-scale at 3 Hz, 6 Hz and 9 Hz. 78
3.7 Final velocity models after multi-scale at 3 Hz, 6 Hz and 9 Hz using one source and different strategies. 79

3.8 Final velocity models after a Multi-scale at 3 Hz, 6 Hz and 9 Hz using 51 sources and: (a) strategy I, (b) strategy II and (c) strategy III.v2. 80

4.1 Interaction area of the source, perturbation and receiver. 93
4.2 Kernel $k^{2\rightarrow 1}$ for one source, one perturbation and one receiver. 95
4.3 Kernel $k^{1\rightarrow 2}$ for one source, one perturbation and one receiver. 95
4.4 Kernel $k^{1\longleftrightarrow 1}$ for one source, one perturbation and one receiver. 96
4.5 $H(\mathbf{v})\upsilon_j$ for one source, one perturbation and one receiver. 97
4.6 Hessian kernels. (a) Kernel $k^{2\rightarrow 1}$, (b) Kernel $k^{1\rightarrow 2}$, and (c) Kernel $k^{1\longleftrightarrow 1}$. . . . 98
4.7 One column of the Hessian matrix. 98
4.8 Hessian matrices using (a) one source and, (b) fifty-one sources with a central frequency of 3 Hz. 99
4.9 Eigenvalues of the Hessian Matrices at 3 Hz for one source and fifty-one sources. 100
4.10 Posterior standard deviation of the velocity model estimated in Figure 3.7a using one source for (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz. 105
4.11 Posterior standard deviation of the velocity model estimated in Figure 3.8a using fifty-one sources for (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz. 106
4.12 Variance reduction (VR) of the posterior standard deviation for the velocity model estimated of Figure 3.7a using one source for (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz. 107
4.13 Variance reduction (VR) of the posterior standard deviation for the velocity model estimated of Figure 3.8a using fifty-one sources for (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz. 108
4.14 Uncertainty Quantification for the test point P_4 at 6 Hz using one source and fifty-one sources. 110
4.15 Resolution for the velocity model estimated in Figure 3.7a using one source at (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz. 111
4.16 Resolution for the velocity model estimated in Figure 3.8a using fifty-one sources at (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz. 112

5.1 Marmousi velocity model. 119
5.2 Marmousi velocity model section selected to test the proposed methodology. . . 120
5.3 Difference in m/s between the original Marmousi velocity model and the FWI reconstruction. The sources used in the experiment are market in red color and the illuminated area is enclosed by a black ellipse. 121
5.4 Instantaneous phase difference between the observed data and the modeled data over the PSO and full-offset FWI result for the Marmousi velocity model. . . . 122
5.5 Full Hessian matrix evaluated at 3 Hz over a) PSO and b) FWI. 124
5.6 Normalized eigenvalue magnitudes of the Hessian matrix evaluated over the PSO and FWI results for 7011 dimensions. 126
5.7 Illumination areas where the Hessian matrix is evaluated for the PSO and the FWI results. 127
5.8 Normalized eigenvalue magnitudes of the Hessian matrix evaluated over the PSO and FWI results. 129
5.9 Marmousi's Hessian matrix evolution over 120 FWI's iterations for the squared area of Figure 5.7b. 130
5.10 Cross section through the foothills of the Canadian rockies. 131
5.11 Canadian foothills velocity model section selected to test the proposed methodology. 133
5.12 Difference in m/s between the original Canadian foothills velocity model and the FWI reconstruction. The sources used in the experiment are market in red color and the illuminated area is enclosed by a black ellipse. 134
5.13 Instantaneous phase difference between the observed data and the modeled data over the PSO and full-offset FWI result for the Canadian foothills velocity model. 135
5.14 Illumination area where the Hessian matrix is evaluated for the PSO and the FWI results. 136
5.15 Canadian Foothills' matrix evolution over 63 FWI's iterations for the squared area of Figure 5.14. 137
5.16 Hess velocity model. 139

5.17 Section of the Hess velocity model. 140
5.18 Hess experiment results for the full-offset and the short-far-offset strategies. . . 141
5.19 Instantaneous phase difference between the observed data and the modeled data over the PSO and the full-offset FWI results for the Hess velocity model. 144
5.20 Instantaneous phase difference between the observed data and the modeled data over the PSO and the short-far-offset FWI results for the Hess velocity model. . 145
5.21 Illumination area where the Hessian matrix is evaluated for the PSO and the FWI results. 146
5.22 Hessian matrix evolution over 63 full-offset FWI's iterations for the squared area of Figure 5.21. 147
5.23 Hessian matrix evolution over 55 short-far-offset FWI's iterations for the squared area of Figure 5.21. 148
5.24 BP 2004 velocity model. 150
5.25 BP velocity model section selected to test the proposed methodology. 150
5.26 BP experiment results for the full-offset and the short-far-offset strategies. . . . 152
5.27 Instantaneous phase difference between the observed data and the modeled data over the PSO and the full-offset FWI results for the BP velocity model. 154
5.28 Instantaneous phase difference between the observed data and the modeled data over the PSO and the short-far-offset FWI results for the BP velocity model. . . 155
5.29 Illumination area where the Hessian matrix is evaluated for the PSO and the FWI results. 156
5.30 Hessian matrix evolution over 48 far-offset FWI's iterations for the squared area of Figure 5.29. 157
5.31 Hessian matrix evolution over 80 short-far-offset FWI's iterations for the squared area of Figure 5.29. 158

List of Tables

2.1 Setting the PSO parameters w, c_1 and c_2 for the geophysical inversion. 49

2.2 Kolmogorov, Smirnov (KS) test results of the random dimension vector in the three different domains. 58

2.3 Estimation of σ^2 and μ using Maximum Likelihood (ML), Maximum A Posteriori (MAP) and Minimum Mean Square Error (MMSE). 58

2.4 Total time estimation of the PSO exploration over 2D and 3D models using the Graphics Processing Unit (GPU) Tesla k40c of Nvidia. 63

3.1 Nvidia Tesla K40c architecture. 74

3.2 FWI computational strategies. 77

3.3 Root Mean Square Error (RMSE) between true and estimated model for all strategies and different number of sources. 79

3.4 Theoretical and measured RAM requirements for all strategies including the execution time for different number of shots. 81

4.1 Characteristics of the theoretical and computed Hessian matrices. 100

4.2 Execution time of the computation of one-column of the Hessian matrix in CPU and GPU. 101

4.3 Execution time and RAM size used to compute the full matrix $\mathbf{H}$ (7182×7182 elements) for different sources using a GPU Tesla K40c. 101

4.4 Special points to test the variance reduction and resolution 109

5.1 Parameters and values for the Marmousi experiment. 121

5.2 Characteristics of the Hessian matrices computed over the PSO and FWI results using the whole Marmousi velocity model and the yellow areas defined in Figure 5.7. 126
5.3 Parameters and values for the Canadian foothills experiment. 132
5.4 Parameters and values for the Hess experiment. 142
5.5 Parameters and values for the BP experiment. 151

List of Appendices

Stability and Numerical Dispersion **173**

FWI implementation using Strategy I **178**

Introduction

1.1 Motivation

The main goal of the geophysical inversion is to look for a geophysical model that can reproduce observed data through a mathematical operator applied in a modeling stage. The three main elements in the geophysical inversion are: the observed data, produced by a source applied over an area of interest and recorded by superficial receivers; the mathematical operator used for the modeling stage and selected from the wave equations family depending on the physical effects that are taken into account; and the cost function, that measures the difference between the modeled and observed data.

In the geophysical case, the inversion is under-determined because the number of independent observations is always fewer than the number of unknown parameters. It is also ill-posed because different geophysical models can generate the same observed data. And it is non-linear because the wavefield propagation through the underground layers is affected by porosity, elasticity and anisotropy effects among others. These three issues make the geophysical inversion an open challenge that can be addressed from different points of view (Beydoun & Alkhalifah, 2015).

In this thesis, the geophysical inversion is addressed from an optimization point of view where global and local optimization techniques work together to find the geophysical model that better emulates the observed data. Particle Swarm Optimization (PSO) and Full Waveform Inversion (FWI) are the global and local optimization techniques, respectively, chosen in this thesis to perform the geophysical inversion (see Figure 1.1). In particular, PSO has been se-

lected because it allows to explore different points of the cost function at the same time (this is a characteristic of the swarm family meta-heuristics) increasing the probability of reaching the global minimum neighborhood. Then the average of the best PSO results is used as the initial velocity model for the FWI. Besides, the uncertainty quantification (UQ) area, that aims to measure how far is the final velocity model from the true model, is addressed using: a statistical analysis over the PSO results, the second order adjoint state theory (Fichtner & Trampert, 2011) to compute the Hessian kernels on the final velocity models and, the Hilbert transform on both the observed and modeled data.

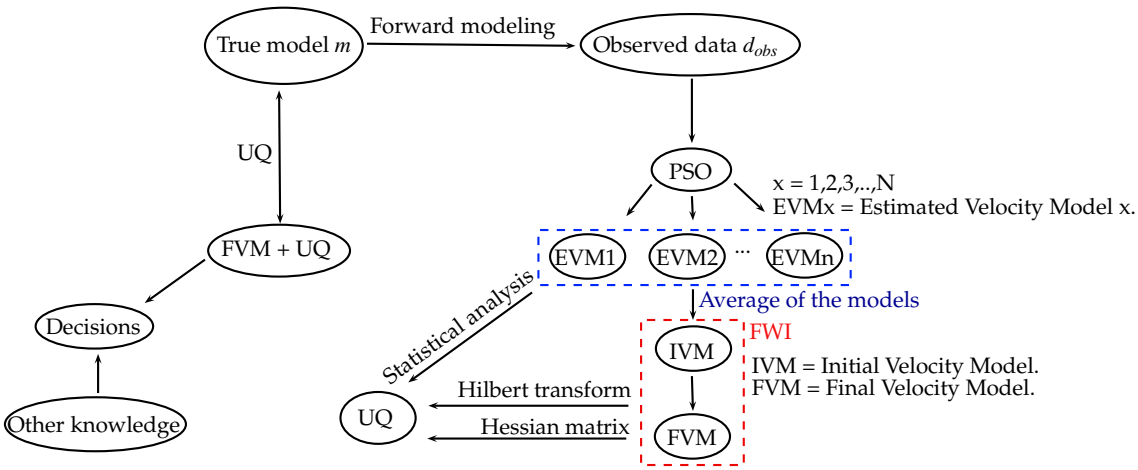


Figure 1.1 Scheme of the methodology proposed to address the Geophysical inversion problem from an optimization point of view.

The computer cluster of CPS research group[5] (CPS-cluster) is used to run all the tests presented in this document and prove the proposed methodology. This cluster has three nodes, each one with two GPUs Tesla k40c/m and two Intel(R) Xeon(R) CPU E5-2620 v3 processors at 2.40 GHz. Each node is administered by the operative system Linux Debian and has 12 GiB of RAM per GPU and 256 GiB of RAM for the two processors.

[5]Connectivity and Signal Processing research group belongs to the school of Electrical, Electronic and Telecommunications at the Industrial University of Santander, Bucaramanga, Colombia.

1.1.1 Geophysical Framework

In geophysics, physical principles are applied in different kinds of geophysical methods to study the Earth's subsurface. This thesis uses one of those geophysical methods called seismic method to build highly accurate subsurface images. In the seismic method, the acoustic waves are generated from a source, i.e., a vibrator or explosive on land or compressed air at sea (see Figure 1.2, step 1). These acoustic waves travel through the subsurface layers until they reach their geological boundaries producing reflections (see Figure 1.2, step 2) and refractions. Part of the reflected energy returns to surface and it is recorded by geophones on land or hydrophones at sea (see Figure 1.2, step 3). The information recorded by each receiver is a signal called a seismic trace (Figure 1.3) and the group of all seismic traces produced by one source is known as shot gather (Figure 1.4). During a seismic acquisition, seismic traces are stored in field (see Figure 1.2, step 4) as shot gathers and then they are processed using complex computational algorithms to extract the geological structures and to build the subsurface image. Finally, the subsurface image is used to decide, for example, whether to drill or not, in the case of an Oil and Gas industry hydrocarbon extraction.

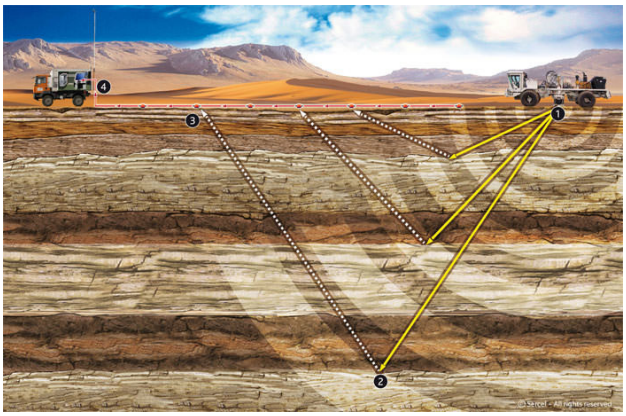
1.2 Thesis Main Contributions and Organization

The most valuable contributions of this thesis are:

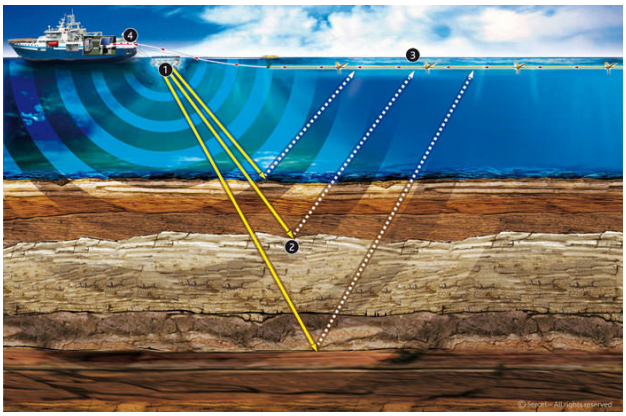
1. The formulation of a novelty methodology that includes: how to use a global optimization technique, as PSO, to address the starting point requirements in FWI and, a practical framework to compute the Hessian matrix from the use of the second order adjoint state theory.

2. A strategy to use the Hilbert transform and the Hessian matrix as two new Quality Controls (QCs) to address the uncertainty quantification area in FWI.

3. The design of computational strategies that take advantage of parallel architectures as provided by GPUs to make feasible the use of the proposed methodology (i.e., the PSO exploration, FWI and uncertainty quantification) in small datasets.



(a)



(b)

Figure 1.2 Seismic acquisition elements: source (1), interfaces (2), receivers (3) and data (4). (a) On-shore and (b) off-shore acquisitions. (Sercel, 2016).

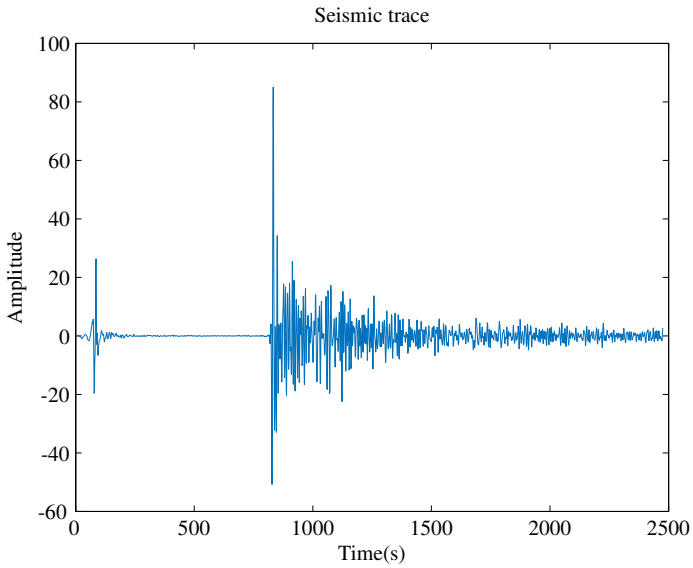


Figure 1.3 Real seismic trace.

The organization of this document is as follows:

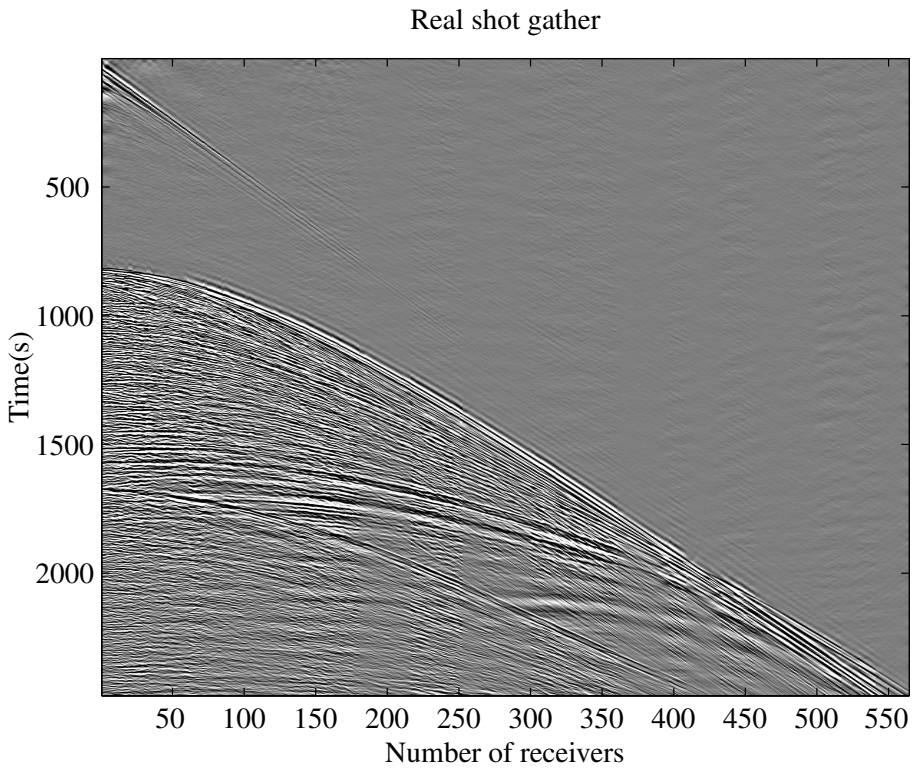


Figure 1.4 Example of one shot gather.

In Chapter 2, the first of the three challenges of using the FWI is addressed. Specifically, the challenge is about the selection of the initial velocity model for the FWI. Particle Swarm Optimization (PSO) is introduced as the global optimization technique selected to perform the cost function exploration of the non-linear geophysical inversion problem. Its theoretical background and parameters are discussed and adapted to the geophysical context. Three different metrics are discussed and used to perform the exploration of the non-linear geophysical cost function. The FWI with a toy model is used to validate the global exploration results. A statistical analysis is done over those results and, at the end, the possibility of using PSO over a real scenario is discussed.

Chapter 3 introduces the FWI presenting the mathematical formulation of the gradient, different computational strategies to compute it and the velocity model update using the steepest descent method together with L-BFGS. Mathematical expressions to calculate the Random Access Memory (RAM) requirements and numerical dispersion are presented. At the end, the discussion section address the possibility of using GPUs for FWI over real data, as a solution

to the second challenge, i.e., the computational cost.

Chapter 4 presents the Second Order Adjoint State Method (SOASM) to compute the Hessian matrix through the Hessian matrix-vector products. The SOASM is illustrated from an optimization point of view working together with linear and dual spaces for greater understanding. The relation between the adjoint fields and the Lagrange multipliers is presented together with a link between the Hessian matrix and the last challenge, i.e., the uncertainty quantification of the FWI results.

In Chapter 5, the proposed methodology is tested over sections of four different synthetic velocity models (Marmousi, Canadian foothills, Hess and BP) to identify its strengths and weaknesses. For each experiment, the PSO and FWI results are illustrated together with the Hessian matrix computed in a squared section of 121 pixels located inside a well illuminated area from the sources and receivers position. The uncertainty quantification topic is addressed from the use of the Hilbert transform and the eigenvalues characteristics of the Hessian matrix as QCs during the inversion process. Moreover, a short-far-offset strategy working together with a muting process to avoid get trapped in a local minimum is discussed in the last section.

The last chapter presents the conclusions about the proposed methodology and provides some guidance about the topics that must be addressed in the future to build a stronger framework for the FWI.

2. First Challenge: Starting Point

According to Snieder (1998), it is possible that for some geological structures the wavefield can not return to surface due to fast velocity areas producing forbidden regions or can not sample their geological characteristics due to slow velocity areas producing plateau regions. Additionally, as the inversion problem is non linear for the seismic case, the cost function has more than one local minimum besides the global minimum (multiple minima) as Figure 2.1 illustrates.

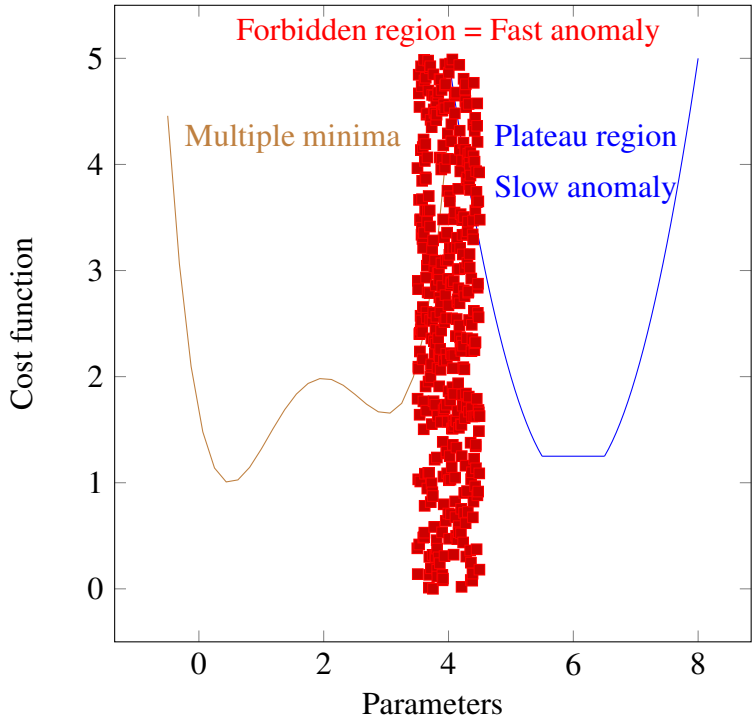


Figure 2.1 Misfit function for truly nonlinear inverse problems. Adapted from Snieder (1998)

This chapter presents the use of PSO, in the geophysical inversion context, to explore the multiple minima cost function and to reach the global minimum neighborhood. This, taking into account that PSO convergence is not affected by the starting point selection due to its exploratory behavior, unlike local optimization techniques as FWI. Shot gathers are analyzed

in different transformed spaces to extract as much information as possible allowing the PSO exploration. In this thesis, the use of PSO is proposed to reaches the global minimum neighborhood of the cost function, in order to it can be used as the starting point for FWI, instead of directly reaching the global minimum itself.

This chapter is organized as follows: first, PSO equations are presented and then three different cost functions, i.e., metrics, are formulated to measure the difference between the modeled and observed data. The PSO parameters selection is presented in the statistical framework section and the metrics behavior is discussed in the experiment description section. A synthetic velocity model is used to test PSO in the geophysical context. A discussion section closes this chapter.

2.1 Particle Swarm Optimization

PSO is an optimization technique inspired by the observation of the social behavior of individuals in nature as bird flocking and fish schooling. Originally attributed to James & Russell (1995) and improved by Shi & Eberhart (1998). Each individual is represented by the i^{th} swarm's particle that moves through the dimension d of the D dimensional search space with the velocity $\boldsymbol{v}_{i,d}^{k}$ at the position $\boldsymbol{x}_{i,d}^{k}$ during the current iteration k. Both $\boldsymbol{v}_{i,d}^{k}$ and $\boldsymbol{x}_{i,d}^{k}$ define the particle behavior where the initial position of each particle is randomly determined and the initial velocity is set to zero. The particles' evolution is defined by two parameters: its new location

$$\boldsymbol{x}_{i,d}^{k+1} = \boldsymbol{x}_{i,d}^{k} + dt \times \boldsymbol{v}_{i,d}^{k+1}, \tag{2.1}$$

where dt is the particles step size which is usually set to one, and its new velocity

$$\boldsymbol{v}_{i,d}^{k+1} = w \times \boldsymbol{v}_{i,d}^{k} + c_1 \times R_1 \times \frac{(\boldsymbol{x}_{i,d}^{l} - \boldsymbol{x}_{i,d}^{k})}{dt} + c_2 \times R_2 \times \frac{(\boldsymbol{x}_{g,d} - \boldsymbol{x}_{i,d}^{k})}{dt}, \tag{2.2}$$

where w is the weight of inertia that controls the particles movement; c_1 and c_2 are the local and global search weights, respectively; R_1 and R_2 are random values uniformly distributed

between 0 and 1; $\boldsymbol{x}_{i,d}^{l}$ is the best particle location and $\boldsymbol{x}_{g,d}$ is the best swarm location.

The cost function is evaluated at new particles positions $f(\boldsymbol{x}_{i,d}^{k+1})$. If a new best local position ($f(\boldsymbol{x}_{i,d}^{k+1}) < f(\boldsymbol{x}_{i,d}^{l})$) or a new best swarm position ($f(\boldsymbol{x}_{i,d}^{k+1}) < f(\boldsymbol{x}_{g,d})$) or both are found then $\boldsymbol{x}_{i,d}^{l}$ or $\boldsymbol{x}_{g,d}$ or both are updated with $\boldsymbol{x}_{i,d}^{k+1}$. This process is repeated until the stopping criterion is reached. Algorithm 1 summarizes the PSO implementation.

PSO can be applied over unknown cost functions or non differentiable cost functions because it does not need the gradient of the cost function to compute the particles movement (see Equation 2.1) which represents a big advantage over Newton's method.

Algorithm 1 PSO pseudo-code

$k = 1$ ▷ K, number of iterations
for each particle $i = 1, \cdots, P$ **do** ▷ P, number of particles
  for each dimension $d = 1, \cdots, D$ **do** ▷ D, number of dimensions
    $\boldsymbol{x}_{i,d}^{k} = \mathscr{U}(0,1) \times (\boldsymbol{x}_{max} - \boldsymbol{x}_{min}) + \boldsymbol{x}_{min}$ ▷ Initialize the particles position
    $\boldsymbol{v}_{i,d}^{k} = 0$ ▷ Initialize the particles velocity
  end for
  $\boldsymbol{x}_{i,d}^{l} \leftarrow \boldsymbol{x}_{i,d}^{k}$ ▷ Initialize the best particle location
  $\phi_i = f(\boldsymbol{x}_{i,d}^{k})$ ▷ Save the cost function value
end for
$_i \leftarrow min(\phi)$ ▷ Locate the best swarm particle position
$\boldsymbol{x}_{g,d} \leftarrow \boldsymbol{x}_{i,d}^{k}$ ▷ Initialize the best swarm location
while Stopping criterion is not met **do** ▷ Search space exploration
  for each particle $i = 1, \cdots, P$ **do** ▷ P, number of particles
    for each dimension $d = 1, \cdots, D$ **do** ▷ D, number of dimensions
      $R_1,\ R_2 \sim \mathscr{U}(0,1)$ ▷ Pick random numbers
      $\boldsymbol{v}_{i,d}^{k+1} = w \times \boldsymbol{v}_{i,d}^{k} + c_1 \times R_1 \times \frac{(\boldsymbol{x}_{i,d}^{l} - \boldsymbol{x}_{i,d}^{k})}{dt} + c_2 \times R_2 \times \frac{(\boldsymbol{x}_{g,d} - \boldsymbol{x}_{i,d}^{k})}{dt}$ ▷ Equation 2.2
      $\boldsymbol{x}_{i,d}^{k+1} = \boldsymbol{x}_{i,d}^{k} + dt \times \boldsymbol{v}_{i,d}^{k+1}$ ▷ Equation 2.1
    end for
    if $f(\boldsymbol{x}_{i,d}^{k+1}) < f(\boldsymbol{x}_{i,d}^{l})$ **then**
      $\boldsymbol{x}_{i,d}^{l} \leftarrow \boldsymbol{x}_{i,d}^{k+1}$ ▷ Update the best particle location
    end if
    if $f(\boldsymbol{x}_{i,d}^{k+1}) < f(\boldsymbol{x}_{g,d})$ **then**
      $\boldsymbol{x}_{g,d} \leftarrow \boldsymbol{x}_{i,d}^{k+1}$ ▷ Update the best swarm location
    end if
  end for
  $k = k + 1$ ▷ Iteration counter
end while

2.1.1 Parameters Selection

If PSO is applied over a convex function (see Figure 2.2), all the particles naturally move in the global minimum direction (blue area in Figure 2.2). However, when PSO is applied

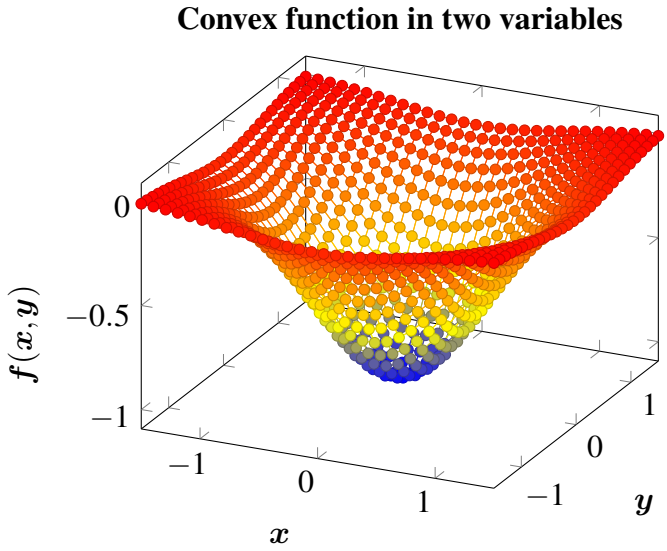


Figure 2.2 Convex function in two variables. Adapted from Rao (2009)

over a non-convex function each particle moves to its nearest local minimum (see blue arrows, Figure 2.3) dragged by its own weight, creating subgroups around the local minima of the cost function (this behavior is controlled by c_1). Additionally, there is a social behavior where the

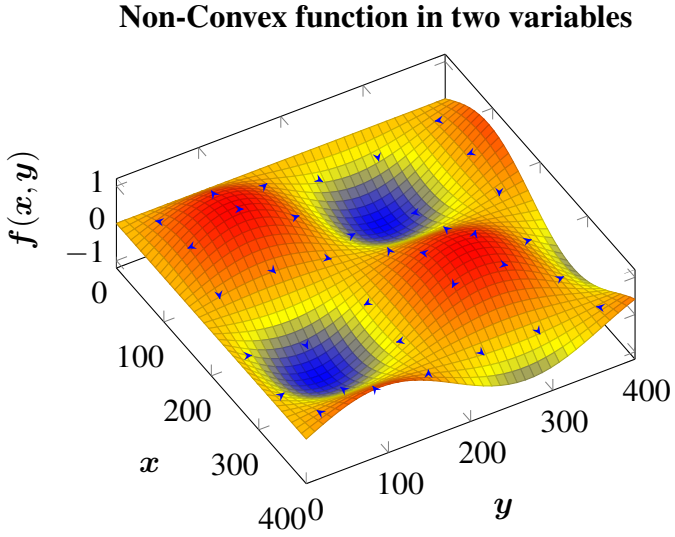


Figure 2.3 PSO search space exploration.

subgroup that contains the particle with the lowest value of the cost function becomes a gravity

point and starts to attract all the particles. During the attraction process, the gravity point changes if a different particle finds a point with a lower value of the cost function (the social behavior is controlled by c_2). The inertia weight w allows to tune the particle swarm behavior between local exploitation and global exploration. According to Taherkhani & Safabakhsh (2016) the particles tend to do global exploration if $0.8 < w < 1.2$ and they do local exploitation if $0.1 < w < 0.4$. A balanced exploration is reached when $w = 0.729$ (Taherkhani & Safabakhsh, 2016) and $c_1 = c_2 = 1.496$ (Clerc, 1999). However, in the-state-of-the-art it is recommended to adjust the local and global weights (c_1 and c_2) depending on the kind of application because a different set of values can reach better results (e.g. $w = 0.729$, $c_1 = 2.04$ and $c_2 = 0.9488$ (Carlisle & Dozier, 2001)). The PSO parameters selected for all the tests are presented in section 2.4.

2.1.2 Cost function

In the geophysical context, the i^{th} particle at iteration k of PSO $\boldsymbol{x}_{i,d}^k$ represents a velocity model $\mathbf{v} \in \mathfrak{M}$, the model space, with a dimensional size $D = N_x \times N_z$ where N_x represents the number of points in the x direction and N_z the number of points in the z direction; and the cost function evaluation $f(\boldsymbol{x}_{i,d}^k)$ represents a mathematical operator applied over the velocity model $F\{\mathbf{v}_{i,d}^k\}$.

In this thesis, the cost function is defined as

$$F\{\mathbf{v}_{i,d}^k\} \triangleq M(T\{R\{L(\mathbf{v}_{i,d}^k)[p(x,z)]\}\}, T\{d_{obs}\}) \tag{2.3}$$

where

- $p(x,z) \in \mathfrak{W}$ is the seismic wave field;
- $L(\mathbf{v})[\cdot] = \nabla^2[\cdot] - \frac{1}{\mathbf{v}^2(x,z)}\frac{\partial^2[\cdot]}{\partial t^2}$ is the d'Alembertian operator that maps $\mathbf{v} \mapsto \mathfrak{W}$;
- $R\{\cdot\}$ is the time-domain finite-difference operator that maps $\mathfrak{W} \mapsto \mathfrak{D}$;
- $d_{obs} \in \mathfrak{D}$ is the observed data produced during a seismic acquisition (see Figure 2.1, step 4);

- $T\{\cdot\}$ is the operator that maps $\mathfrak{D} \mapsto \mathfrak{T}$;
- $M(\cdot,\cdot) \in \mathfrak{T}^*$ is the functional that measures the difference between the modeled and the observed data; such that $\mathfrak{T}^*$ is the dual space[1] of $\mathfrak{T}$.

The wave equation operator defined for the modeling stage is

$$\frac{1}{\mathbf{v}^2(x,z)}\frac{\partial^2 p(x,z)}{\partial t^2} = \frac{\partial^2 p(x,z)}{\partial x^2} + \frac{\partial^2 p(x,z)}{\partial z^2} + src(x,z), \tag{2.4}$$

where t is the time variable and $src(x,z)$ is the source at the spatial coordinates $x,\ z$. After using Equation 2.4, $p(x,z)$ becomes $p(x,z,t)$ which represents a cube of size $N_x \times N_z \times N_t$ where N_t is the number of time steps. The operator $R\{\cdot\}$ takes the $p(x,0,t)$ face, i.e., the shot gather belongs to $\mathfrak{D}$, from the $p(x,z,t)$ cube and the operator $T\{\cdot\}$ brings the shot gather into a transformed domain in which $M(\cdot,\cdot)$ is applied. Equation 2.3 can be rewritten in compact form as

$$F\{\mathbf{v}_{i,d}^k\} = M(T\{d_{mod}\}, T\{d_{obs}\}) \tag{2.5}$$

where d_{mod} is the modeled pressure field at the receiver positions using Equation 2.4.

2.2 Transformed spaces and Metrics

In this thesis the cycle-skipping metric, that is illustrated in Figure 2.4, is used together with PSO to find the FWI starting point. For comparison, the cycle-skipping measurement is performed in time, frequency and Hilbert domains, each one using its own mathematical operator. The cycle-skipping count is computed in the following way:

- First, the cycle-skipping coefficients $C_{s,r}$ are computed according to the domain, as detailed in the following subsections.
- Second, the average of all the cycle-skipping coefficients in the receivers direction $\hat{C}_S$ is computed producing one cycle-skipping coefficient per shot.

[1]The dual of a linear space is just an *ad hoc* space, closely resembling the original space, except that if the quantities of the linear space have physical dimensions, the quantities of the dual space must have the reciprocal physical dimensions in order to the dual product $k = \langle linear\ space, dual\ space \rangle$ returns a scalar (Tarantola, 2005). $\mathfrak{T}^*$ is the space of all the linear forms over $\mathfrak{T}$.

- And third, the average of all the cycle-skipping coefficients in the source direction $\hat{C}$ is computed producing one cycle-skipping coefficient per velocity model.

At the end, a new velocity model is accepted if its cycle-skipping coefficient is lower than the previous ones.

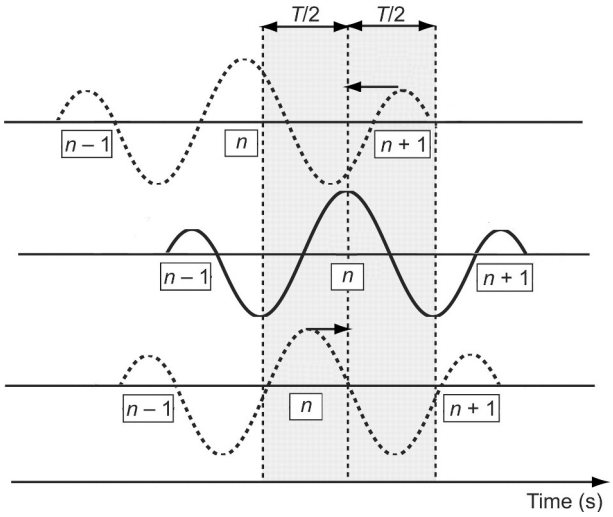


Figure 2.4 Schematic of cycle-skipping artifacts in FWI. The solid black line represents a monochromatic seismogram of period T as a function of time. The upper dashed line represents the modeled monochromatic seismograms with a time delay greater than $T/2$. In this case, FWI will update the model such that the $n+1^{th}$ cycle of the modeled seismograms will match the n^{th} cycle of the observed seismogram, leading to an erroneous model. In the bottom example, FWI will update the model such that the modeled and recorded n^{th} cycle are in-phase because the time delay is less than $T/2$. Adapted from Virieux & Operto (2009).

2.2.1 Time domain

In the time domain, the cycle-skipping coefficients correspond to correlation coefficients per receiver computed as

$$C_{s,r} = \frac{\Sigma_n (d^{s,r}_{obs}(n) - \hat{d}^{s,r}_{obs})(d^{s,r}_{mod}(n) - \hat{d}^{s,r}_{mod})}{\sqrt{(\Sigma_n (d^{s,r}_{obs}(n) - \hat{d}^{s,r}_{obs})^2)(\Sigma_n (d^{s,r}_{mod}(n) - \hat{d}^{s,r}_{mod})^2)}}, \tag{2.6}$$

where s, r and n are the current source, receiver and sample, respectively; and $\hat{d}^{s,r}_{obs}$ or $\hat{d}^{s,r}_{mod}$ represent the average in the samples direction.

Before using this metric, it is necessary to apply an amplitude correction over all the seismic traces to guarantee that correlation takes into account the effects of the secondary reflections, because the first arrival usually has most of the energy of the trace, hiding the secondary reflections. The amplitude correction used in this chapter is

$$d_{obs}^{s,r}(n) = d_{obs}^{s,r}(n) \times q^2(n), \tag{2.7}$$

where $q^2(n)$ is a polynomial function that correct the wavefield exponential decay.

2.2.2 Frequency domain

Frequency domain is formulated to analyze the frequency content of each receiver. The Fourier transform is applied in the time direction over the modeled and the observed data producing ${d_{mod}}^{s,r}(\omega)$ and ${d_{obs}}^{s,r}(\omega)$, respectively. The phase information of both ${\theta_{mod}}^{s,r}(\omega)$ and ${\theta_{obs}}^{s,r}(\omega)$ is saved neglecting the amplitude information.

In this domain, the cycle-skipping coefficients $C_{s,r}$ are computed using Algorithm 2 as Figure 2.5 illustrates.

Algorithm 2 CS in frequency domain

for each source $s = 1, \cdots, S$ **do** ▷ S, number of sources
  for each receiver $r = 1, \cdots, R$ **do** ▷ R, number of receivers
    $Counter = 0$ ▷ Clear the counter
    for each frequency $\omega = 0, \cdots, W$ **do** ▷ W, frequencies fewer than 10 Hz
      $dif = |{\theta_{mod}}^{s,r}(\omega) - {\theta_{obs}}^{s,r}(\omega)|$ ▷ Phase delay
      if $dif > \pi$ **then**
        $Counter = Counter + 1$ ▷ Cycle-skipping counter
      end if
    end for
    $C_{s,r} = Counter$ ▷ Saving the CS count
  end for
end for

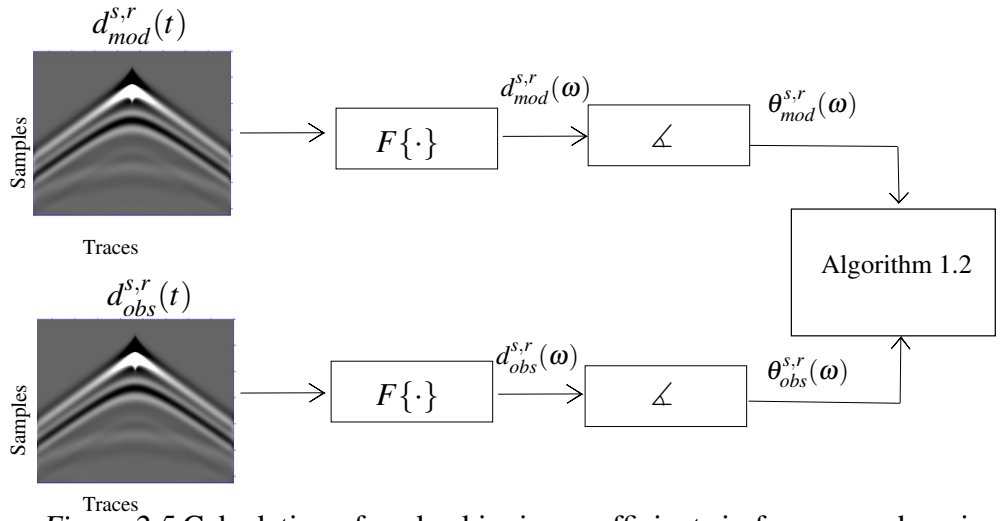


Figure 2.5 Calculation of cycle-skipping coefficients in frequency domain

2.2.3 Hilbert domain

The complex trace is defined as

$$\mathbf{d}^{s,r}_{obs}(t) = {d_{obs}}^{s,r}(t) + jH\{{d_{obs}}^{s,r}(t)\}, \tag{2.8}$$

where $\mathbf{d}_{obs}^{s,r}(t)$ is the analytical signal, j represents the imaginary part and $H\{\cdot\}$ is the Hilbert operator. The complex trace allows the extraction of seismic attributes such as the reflector intensity and the phase wrapping, where the instantaneous phase for the observed data is computed as

$$\theta_{obs}{}^{s,r}(t) = \arctan\left(\frac{H\{d_{obs}{}^{s,r}(t)\}}{d_{obs}{}^{s,r}(t)}\right). \tag{2.9}$$

For the complex trace the cycle-skipping coefficients $C_{s,r}$ are computed using Algorithm 3 as Figure 2.6 illustrates.

Algorithm 3 CS in Hilbert domain

for each source $s = 1, \cdots, S$ **do** ▷ S, number of sources
  for each receiver $r = 1, \cdots, R$ **do** ▷ R, number of receivers
    $Counter = 0$ ▷ Clear the counter
    for each time step $t = 0, \cdots, T$ **do** ▷ T, Total time
      $dif = |\theta_{mod}{}^{s,r}(t) - \theta_{obs}{}^{s,r}(t)|$ ▷ Phase delay
      if $dif > \pi$ **then**
        $Counter = Counter + 1$ ▷ Cycle-skipping counter
      end if
    end for
    $C_{s,r} = Counter$ ▷ Saving the CS count
  end for
end for

2.3 Workflow

Figure 2.7 illustrates the formulated workflow including the propagator module after defining the PSO parameters, the transformed spaces, the metrics and the cost function. In this workflow, the observed data is produced synthetically from the original velocity model and the modeled data is computed from the estimated velocity model using in both cases Equation 2.4. The observed data is computed only once at the beginning while a new modeled data is computed by each particle at each iteration step.

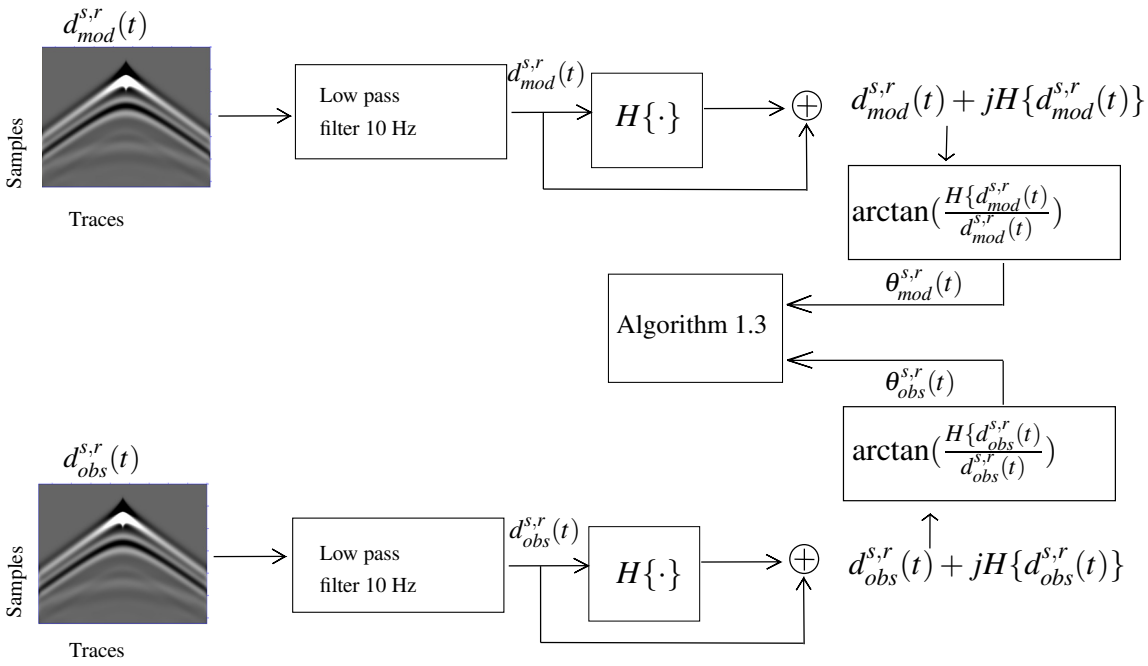


Figure 2.6 Calculation of cycle-skipping coefficients in Hilbert domain

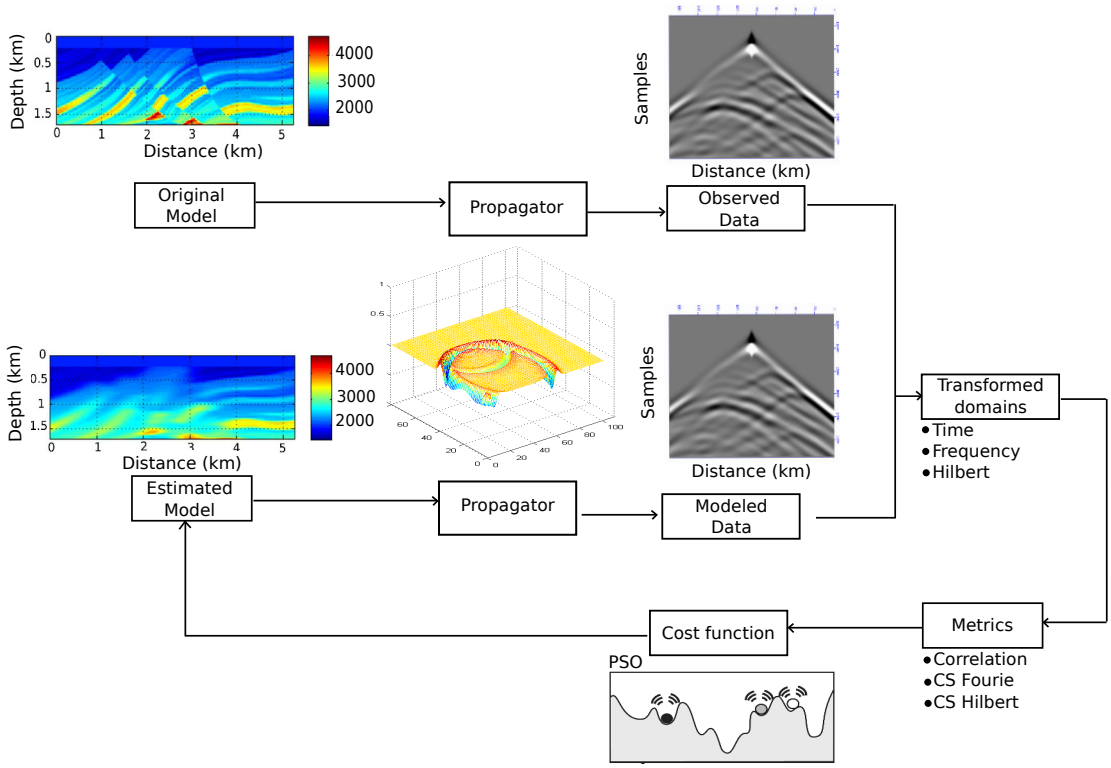


Figure 2.7 Workflow of the full implementation

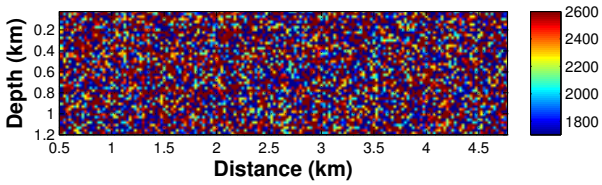
The observed and modeled data are brought to the transformed domains to apply a specific metric. Then PSO algorithm decides from the cost function value whether to save or eliminate the estimated velocity model. In the last step, the PSO algorithm verifies if the stopping crite-

rion is reached to finish the process. Otherwise, a new random velocity model is produced and the loop starts again.

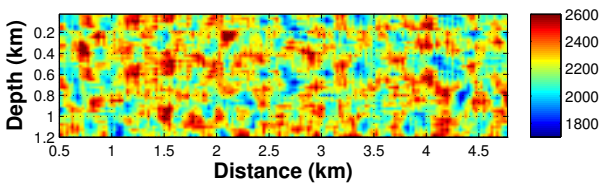
As a consequence of the random velocity values on each dimension, there are areas with high velocity contrast that produce strong reflections. The uniform smoother filter

$$h_{rec}[\cdot,\cdot] = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{2.10}$$

has been applied over all the estimated velocity models to overcome this problem and reduce the energy of the reflections. Figure 2.8a shows one of the estimated velocity models produced by a particle while Figure 2.8b illustrates its smoothing version which has been used to produce the modeled data.



(a)



(b)

Figure 2.8 Estimated velocity model from the i^{th} swarm's particle at iteration k. (a) Random velocity model and (b) smoothing model.

2.4 Statistical Framework

The PSO implementation was validated with five test functions in two dimensions (Ackley, Beale, Rastrigin, Rosenbrock and Step) using 50 particles, 30 iterations per particle and running 100 experiments per function.

For the geophysical context, the test functions are replaced by the new cost function (see Equation 2.5) where the dimensionality D increases to $N_x \times N_z$. Since the new cost function has more dimensions then the number of particles P must be increased to perform a better exploration. Likewise, the number of iterations K per particle must be also increased, to extend the exploration time. According to Shaw & Srivastava (2007), an acceptable PSO result is obtained in a resistivity inversion when $D = 1$, $P = 300$, $K = 100$, $w = 1$, $c_1 = 2.8$ and $c_2 = 3.1$.

The first velocity inversion using the workflow is performed with the PSO parameters recommended by Shaw & Srivastava (2007) but changing the dimensions to $D = 18$. Table 2.1 illustrates that the best PSO result is obtained when $w = 0.7$, $c_1 = 0.8$ and $c_2 = 1.1$. Therefore w, c_1 and c_2 keep these values for all the following PSO tests.

The final adjustments are about P and K for the dimensionality ($D = 211 \times 68$) of the synthetic velocity models used in this thesis. According to Figure 2.9, the highest correlation coefficient $\hat{C}$ is obtained when $P = 500$, $K = 100$ and the number of experiments E is set to 100. Then E was increased to 200 and a random dimension was selected to analyze its behavior in the experiments direction using the three different domains. From Figure 2.10, performing more than 100 experiments does not produce a significant change on the random dimension variance allowing us to keep $E = 100$. This experiment configuration is used for all the following PSO tests.

In this way, one experiment produces 500 velocity models where only one is the best global. After running 100 experiments, a cube with the 100 best global velocity models is build to al-

Table 2.1
Setting the PSO parameters w, c_1 and c_2 for the geophysical inversion.

w	c_1	c_2	Correlation in time
1	2.8	3.1	0.6969
1.2	2.8	3.1	0.6845
0.729	2.8	3.1	0.7354
0.7	2.8	3.1	0.7602
0.2	2.8	3.1	0.6367
0.7	2.8	2.1	0.7800
0.7	2.8	1.1	0.8455
0.7	1.8	1.1	0.8140
0.7	1	1.1	0.9292
0.7	0.8	1.1	0.9914
0.72	0.8	1.1	0.8322
0.729	0.8	1.1	0.8682

Note: In the original velocity model $\boldsymbol{x}_{g,1} = \boldsymbol{x}_{g,2} = \boldsymbol{x}_{g,5} = \boldsymbol{x}_{g,6} = 1500$ m/s, $\boldsymbol{x}_{g,3} = \boldsymbol{x}_{g,4} = 2500$, $\boldsymbol{x}_{g,7} = \boldsymbol{x}_{g,8} = \boldsymbol{x}_{g,9} = \boldsymbol{x}_{g,10} = \boldsymbol{x}_{g,11} = \boldsymbol{x}_{g,12} = 2000$ m/s and $\boldsymbol{x}_{g,13} = \boldsymbol{x}_{g,14} = \boldsymbol{x}_{g,15} = \boldsymbol{x}_{g,16} = \boldsymbol{x}_{g,17} = \boldsymbol{x}_{g,18} = 2800$ m/s.

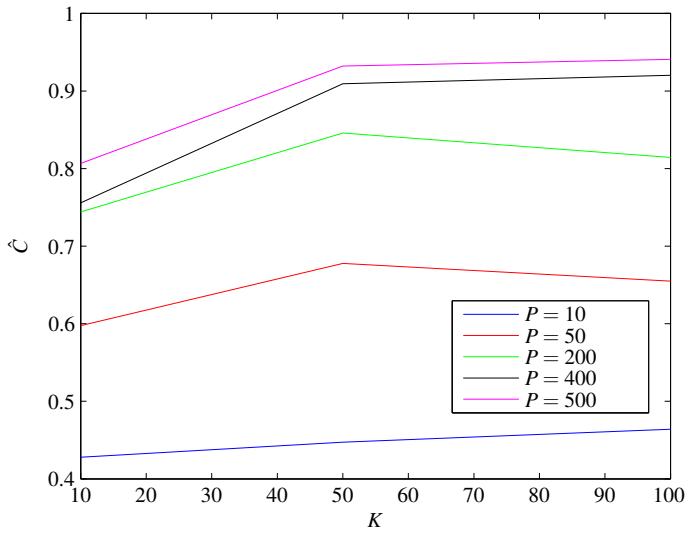
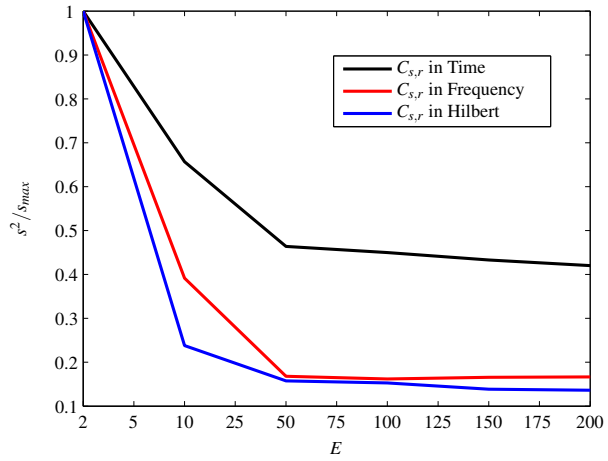


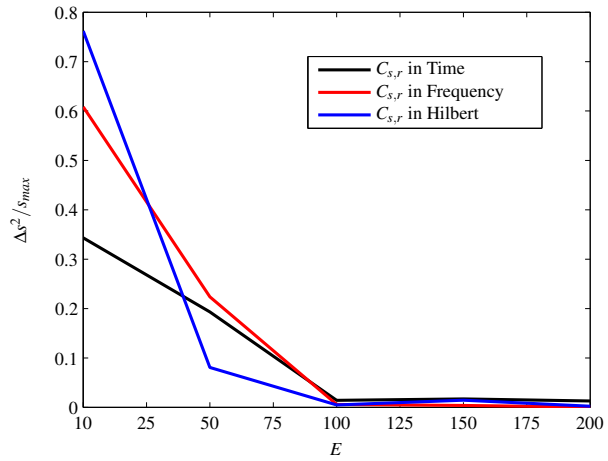
Figure 2.9 Setting the PSO parameters P and K for the geophysical inversion with $w = 0.7$, $c_1 = 0.8$, $c_2 = 1.1$, $D = 211 \times 68$ and $E = 100$.

lows the statistical analysis. Figure 2.11 shows the projection of one of the random dimensions in the experiments direction of the cube $\boldsymbol{x}_{g,d}^{\boldsymbol{e}}$.

From this projection, it is possible to build the Empirical Cumulative Density Function (ECDF) to be compared with a reference Cumulative Density Function (CDF) (e.g. Gaussian, Laplacian or Uniform) through the KS test (Wilcox, 2005). The KS test allows the identifica-



(a)



(b)

Figure 2.10 Statistical behavior of a random dimension for different number of experiments in three different domains. (a) Variance and (b) variance's rate.

tion of the PDF that better describes the behavior of the projected dimension. Once the PDF has been defined, the statistical estimation (e.g. mean, median, variance, etc..) can be done using the empirical data.

In other words, the cube with the 100 best PSO results allows to define the statistical behavior of each dimension. For example, it is possible to determine if the dimensions are independent and identically distributed (i.i.d.). It is also possible to build the covariance and correlation matrices to analyze the dimensions dependence. From the average in the experiment direction it is possible to obtain the average of the best velocity models $\hat{\boldsymbol{x}}_{g,d}$ which is used as the initial velocity model for the forthcoming FWI tests.

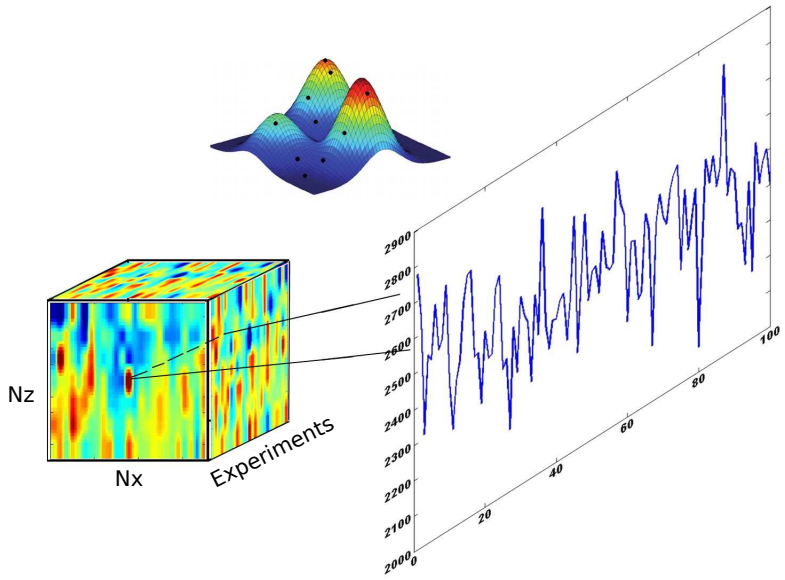


Figure 2.11 Cube with the best results after the 100 PSO experiments.

2.5 Experimental Description

In this section, a synthetic velocity model is used as an example to test the proposed methodology where the PSO results are used to generate the starting point for the FWI. The acquisition geometry and the model characteristics are described. The average in the experiment direction of the best velocity models, $\hat{\boldsymbol{x}}_{g,d}$, are illustrated together with the FWI results. The computational cost tendency of the PSO implementation, the metrics behavior and a statistical analysis of a random dimension vector in the three domains close the section.

The velocity model to test PSO is a diffracting square model, illustrated in Figure 2.12, with a size of $N_x \times N_z = 211 \times 68$ and a spatial grid of 25 m $(\Delta x = \Delta z = 25\ m)$. There is a diffracting element with velocity of 2500 m/s inside the background velocity of 2000 m/s; in this model, the diffracting element is a square of size 9×9 centered at the position $x = 2650$ m and $z = 850$ m. The upper side of the velocity model represents the free surface conditions. At the left, right and lower sides of the velocity model, Convolutional Perfectly Matched Layer (CPML) is applied to reduce the energy that comes from the non-natural boundaries. The details of CPML are discussed in Pasalic & McGarry (2010).

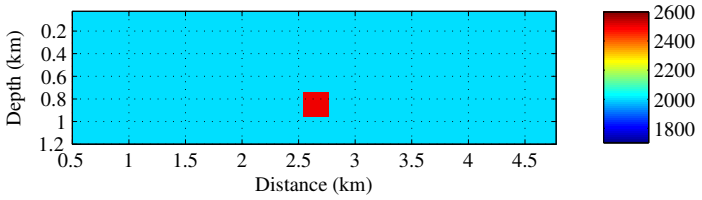


Figure 2.12 Diffracting square model.

Five sources are used to perform the acquisition. They are placed at 525 m, 1600 m, 2625 m, 3725 m and 4775 m with a depth of 125 m. The source wavelet is defined as

$$g_s = -2(\pi f_0)^2(t-t_0)\exp(-(\pi f_0)^2(t-t_0)^2), \tag{2.11}$$

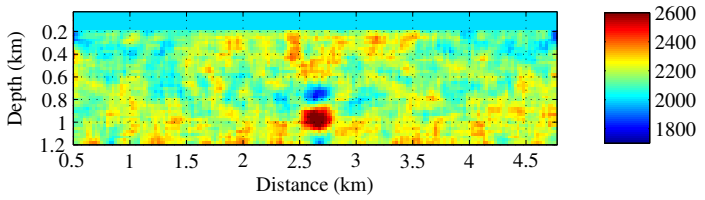
where f_0 is the central frequency in Hertz, and t_0 is the time delay parameter in seconds. The observed data d_{obs} is produced using a set of 171 receivers placed in line 125 m of depth every 25 m, from $x = 525$ m to $x = 4775$ m. Each receiver records 2.5 s at 250 samples per second ($\Delta t = 4 \times 10^{-3}$ s).

The full PSO implementation depicted in Figure 2.7 is applied using the diffracting square model of Figure 2.12 as the original model to obtain the cubes with the 100 best velocity models through the metrics in the three different domains. The average in the experiment direction of the best velocity models $\hat{\boldsymbol{x}}_{g,d}$ is illustrated in Figures 2.13a, 2.14a and 2.15a for time, frequency and Hilbert domains, respectively. From these results, Hilbert and time domains allow to identify a contrast near the true position of the diffracting square element (see Figure 2.12) while the frequency domain tries to resolve the background velocity.

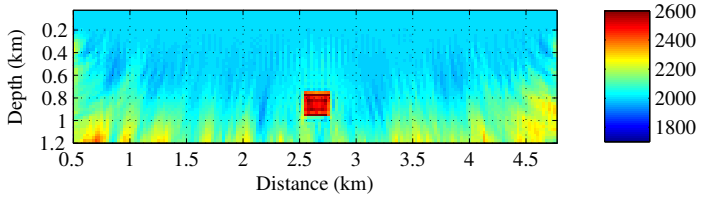
The FWI is computed using the average in the experiment direction of the best velocity models $\hat{\boldsymbol{x}}_{g,d}$ as the initial velocity model and the following geometry: 21 sources distributed using Equation 3.11 at 125 m of depth; the source wavelet defined in Equation 3.12; 171 receivers recording 3.5 s at 1 ms per sample, placed at the same depth of PSO test; and 40 iterations per frequency step. The results after the multi-scale FWI implementation $fwi\{\hat{\boldsymbol{x}}_{g,d}\}$ at 3 Hz, 6 Hz and 9 Hz are illustrated in Figures 2.13b, 2.14b and 2.15b for the time, frequency and Hilbert

metrics, respectively. For all three domains the FWI resolves the diffracting square element and the background velocity inside the area illuminated by the geometry (see Figure 2.16). The details of the FWI implementation are discussed in Chapter 3.

The spectra of the source wavelets g_s used for the PSO and FWI tests, illustrated in Figure 2.17, demonstrate that the whole inversion process (PSO + FWI) resolves from the low frequency information (2 Hz) to the high frequency details (12 Hz). As an example, the spectrum of a seismic trace recorded at the position 1250 m, illustrated in Figure 2.18, shows that the PSO result (blue color) resolves for frequencies lower than 6 Hz and the FWI result (red color) resolves for frequencies between 6 Hz and 12 Hz.



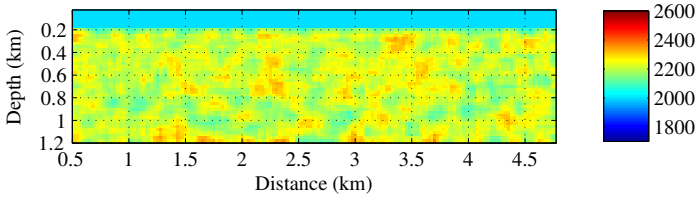
(a)



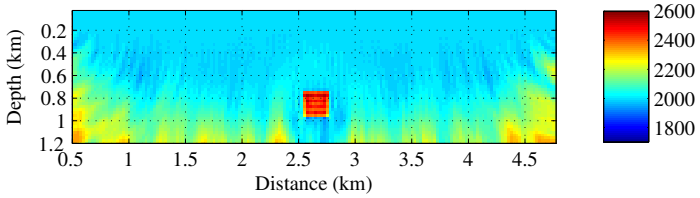
(b)

Figure 2.13 PSO and FWI results using correlation metric. (a) $\hat{\boldsymbol{x}}_{g,d}$ using correlation metric when $w = 0.7$, $c_1 = 0.8$, $c_2 = 1.1$, $P = 500$, $K = 100$ and the number of experiments is set to 100 and, (b) $fwi\{\hat{\boldsymbol{x}}_{g,d}\}$ after a multi-scale at 3 Hz, 6 Hz and 9 Hz.

Five different tests were performed to measure the computational cost as the number of particles P and the number of iterations per particle K increase. According to the results (see Figure 2.19), the computational cost has an exponential tendency, spending less than four hours in this geophysical experiment ($D = 211 \times 68$). It must be highlighted that the particle's exploration, the filtering process and the modeling stage to produce d_{mod} are implemented inside a GPU Tesla K40c using the Compute Unified Device Architecture (CUDA) language; and,

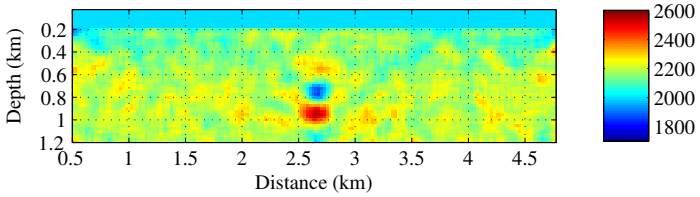


(a)

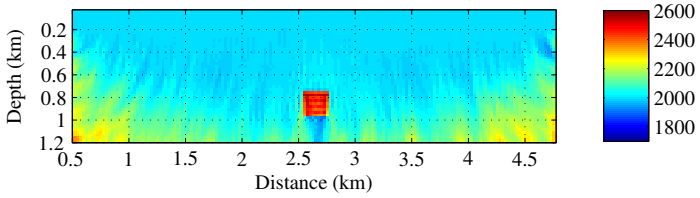


(b)

Figure 2.14 PSO and FWI results using frequency metric. (a) $\hat{\boldsymbol{x}}_{g,d}$ using frequency metric when $w = 0.7$, $c_1 = 0.8$, $c_2 = 1.1$, $P = 500$, $K = 100$ and $E = 100$ and (b) $fwi\{\hat{\boldsymbol{x}}_{g,d}\}$ after a multi-scale at 3 Hz, 6 Hz and 9 Hz.



(a)



(b)

Figure 2.15 PSO and FWI results using Hilbert metric. (a) $\hat{\boldsymbol{x}}_{g,d}$ using Hilbert metric when $w = 0.7$, $c_1 = 0.8$, $c_2 = 1.1$, $P = 500$, $K = 100$ and $E = 100$ and (b) $fwi\{\hat{\boldsymbol{x}}_{g,d}\}$ after a multi-scale at 3 Hz, 6 Hz and 9 Hz.

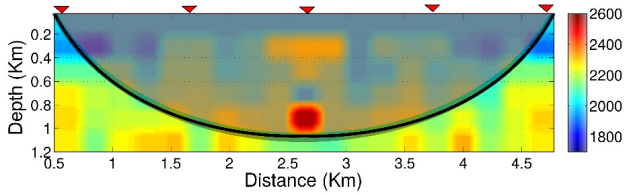


Figure 2.16 Area illuminated by the array of sources and receivers during the acquisition.

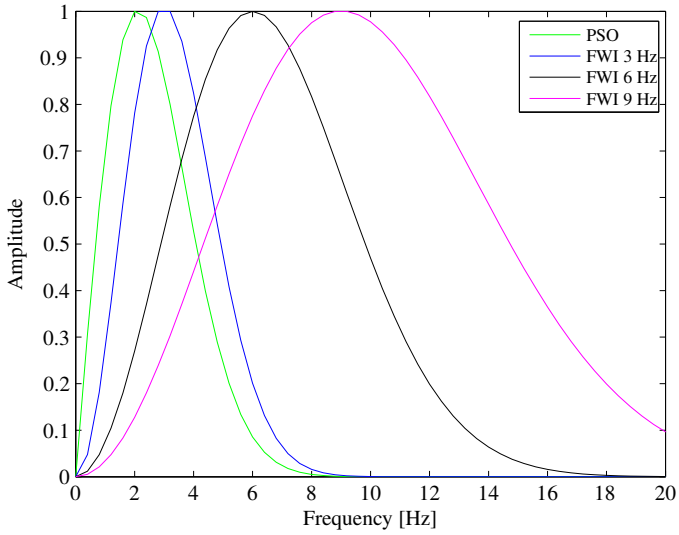


Figure 2.17 Spectra of source wavelet (g_s) used for the PSO and the multi-scale FWI tests. In green color, the spectrum of g_s using Equation 2.11 with $f = 3$ Hz is depicted. In blue, black and magenta colors, the spectra of g_s using Equation 3.12 with $f = 3$, 6 and 9 Hz, respectively are depicted.

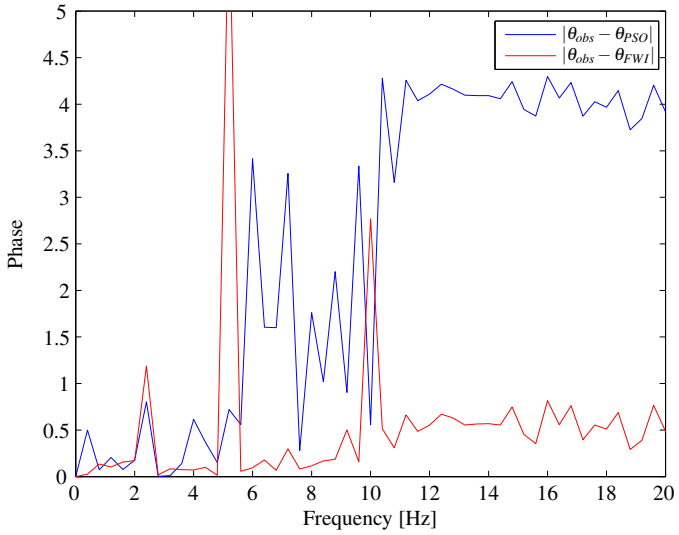


Figure 2.18 Seismic trace spectra after using PSO and the multi-scale FWI. The difference between the phase of d_{obs} and d_{mod} using PSO is depicted in blue color, while the difference between the phase of d_{obs} and d_{mod} using FWI is depicted in red color.

the transformed domains, the metrics and the PSO algorithm are implemented in an Intel(R) Xeon(R) Central Processing Unit (CPU) E5-2620 v3 @ 2.4 GHz with 256 GiB of RAM using the Standard-C language. As a consequence of the use of GPUs for the particles exploration, the performance for $400 \leq P \leq 500$ with $K = 100$ is almost the same.

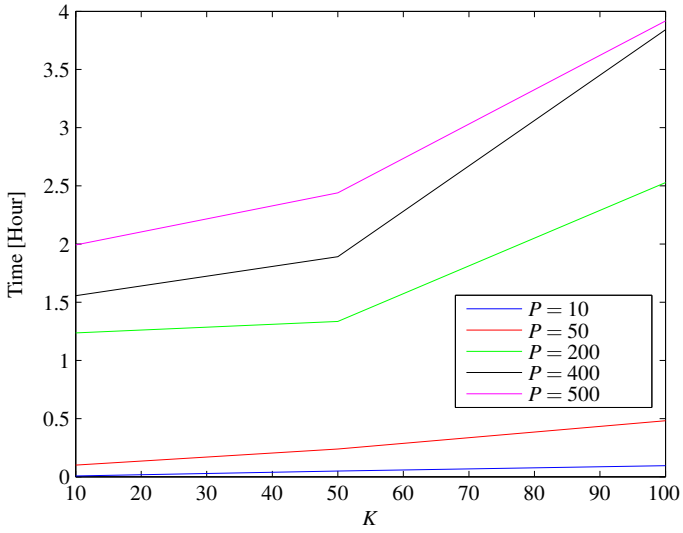


Figure 2.19 Execution time of the PSO implementation for different values of K and P with $w = 0.7$, $c_1 = 0.8$, $c_2 = 1.1$, $D = 211 \times 68$ and $E = 100$. These tests were executed on an Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.4 GHz with 256 GiB of RAM and a GPU Tesla K40c with 12 GiB of RAM.

A multi-metric implementation is performed to define the metric's behavior when they are working either alone or together. To use more than one metric, the $\hat{\boldsymbol{x}}_{g,d}$ of the previous PSO tests is saved as $\boldsymbol{x}_{g,d}$ for the forthcoming PSO implementation. The ℓ_2-norm of the difference between d_{obs} and d_{mod} is used to measure the metric performance where the modeled data d_{mod} is obtained from Equation 2.4 over $fwi\{\hat{\boldsymbol{x}}_{g,d}\}$. The six combinations were tested to identify the best sequences. Figure 2.20 illustrates the results: Hilbert domain (marked with the green circle) offers the best performance when only one metric is applied. However, if two metrics were applied then the best performance is reached using first Hilbert and then the frequency domain (marked with the red circles). Finally, if all the three metrics are applied, the best combination is to use first the Hilbert domain, then the frequency domain and in the last step the correlation in the time domain (marked with the blue circles).

Since these tests are performed over synthetic models, it is also possible to measure the ℓ_2-norm but now using the difference between the original velocity model (Figure 2.12) and the models obtained from $fwi\{\hat{\boldsymbol{x}}_{g,d}\}$ (Figure 2.13b, Figure 2.14b and Figure 2.15b). It must be highlighted that in real life it is impossible to run this type of test because the original velocity model is unknown. From the results (see Figure 2.21), the multi-metric's sequences to obtain

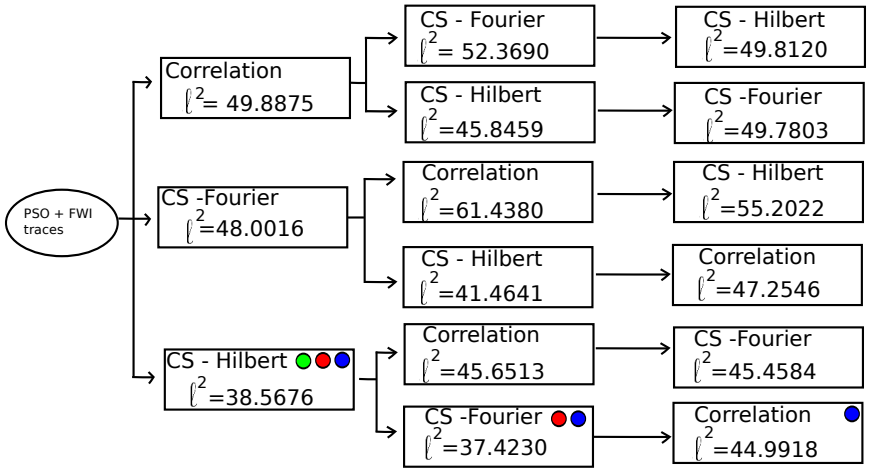


Figure 2.20 Multi-metric behavior when the ℓ_2-norm is applied over the difference between d_{obs} and d_{mod}.

the best performance using two and three domains are different from the sequences based on the seismic traces (Figure 2.20). However, in both cases the use of the Hilbert domain produces the lowest ℓ_2-norm values when only one metric is applied.

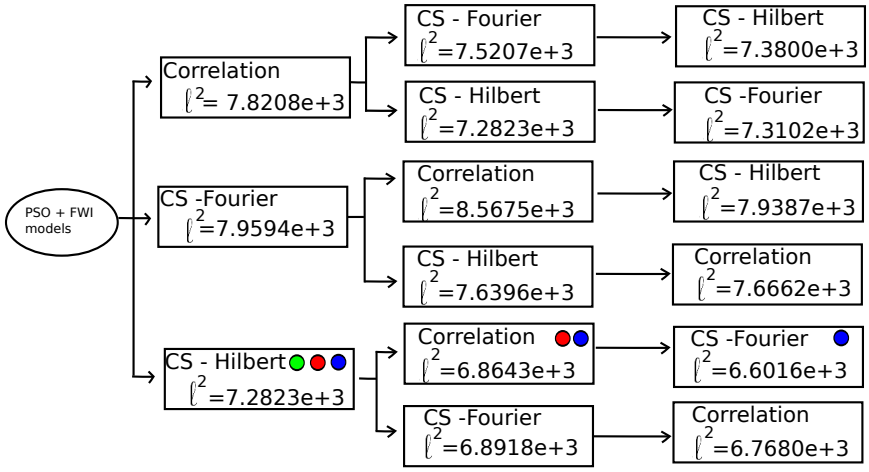


Figure 2.21 Multi-metric behavior when the ℓ_2-norm is applied over the difference between the original velocity model and $fwi\{\hat{\boldsymbol{x}}_{g,d}\}$.

Each PSO test produces a cube with the 100 best results (see Figure 2.11) per metric. A random dimension vector $\boldsymbol{x}^e_{g,d}$ is extracted from these three cubes (time, frequency and Hilbert domains) to perform the statistical analysis. In the KS test, the null hypothesis is not rejected when

$$\sqrt{E}D_m < K_\alpha \tag{2.12}$$

where D_m is the difference between the ECDF and the CDF, and K_α is a value extracted from the tables in Lopes, Reid & Hobson (2007) using the significance level. For this statistical analysis, the significance level is set to 0.05 which produces $K_\alpha = 1.36$. Since $E = 100$ then $D_m < 0.136$.

The ECDF of the random dimension on each domain is compared with the Gaussian, Uniform and Laplacian CDFs. The D_m values of these comparisons are presented in Table 2.2 where the rejected ones are market in red color.

Table 2.2
KS test results of the random dimension vector in the three different domains.

Domains $\diagdown$ D_m	Gaussian	Uniform	Laplacian
Time	0.10748	0.15075	0.32406
Frequency	0.04181	0.10357	0.17157
Hilbert	0.09207	0.14975	0.14974

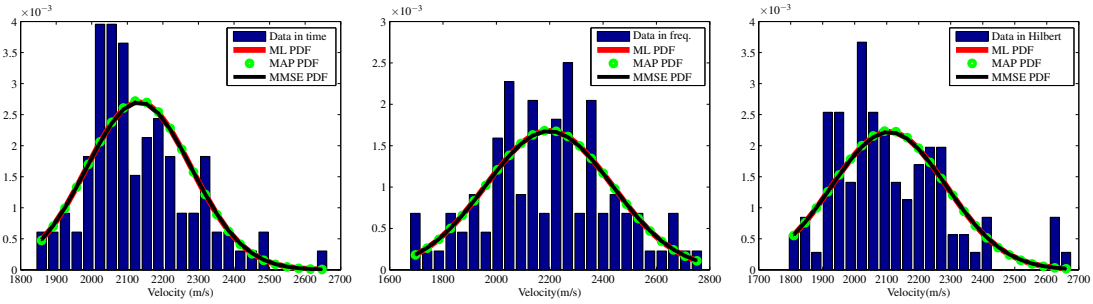
According to the KS test, the Gaussian PDF is not rejected in all the three different domains and the Uniform PDF is only accepted in the frequency domain.

Once the PDF has been defined (Gaussian), it is necessary to estimated σ^2 and μ to emulate the behavior of the extracted data. There are different ways to compute this estimation and in this chapter both parameters are estimated using the ML, MAP and MMSE concepts. According to Table 2.3, ML, MAP and MMSE agree in the estimation of μ and disagree in the estimation of σ^2 for all the domains.

Table 2.3
Estimation of σ^2 and μ using ML, MAP and MMSE.

Estimator $\diagdown$ Domains	Time	Fourier	Hilbert
μ_{ML}	$2.1328e+03$	$2.1973e+03$	$2.1072e+03$
μ_{MAP}	$2.1328e+03$	$2.1973e+03$	$2.1072e+03$
μ_{MMSE}	$2.1328e+03$	$2.1973e+03$	$2.1072e+03$
σ^2_{ML}	147.0682	236.7654	178.8374
σ^2_{MAP}	146.3383	235.5904	177.9499
σ^2_{MMSE}	147.8091	237.9582	179.7383

Figure 2.22 illustrates the empirical data and the estimated PDF using the ML, MAP and MMSE concepts in Time, Frequency and Hilbert domains. For all the domains, the estimated PDFs are almost equal and all of them can be used for an statistical analysis.



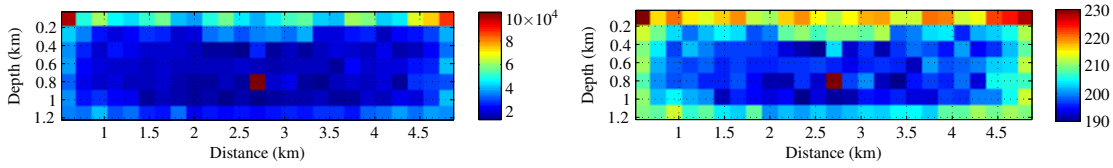
(a) *(b)* *(c)*

Figure 2.22 Comparison between the empirical data (blue histogram) and the estimated PDF using ML, MAP and MMSE concepts in (a) Time, (b) Frequency and (c) Hilbert domains.

Finally, the variances of the 100 best results for each dimension σ_d^2 can be computed from the empirical data as

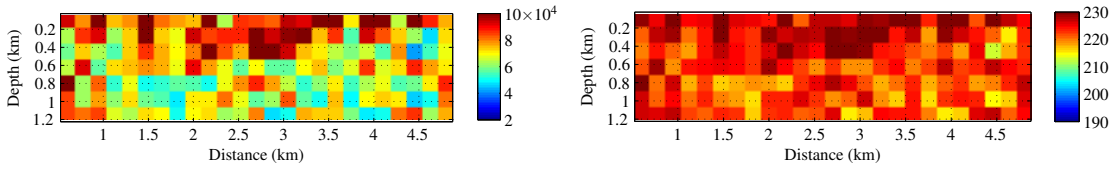
$$\sigma_d^2 = \frac{1}{E}\sum_{e=1}^{E}(\boldsymbol{x}_{g,d}^{e} - \hat{\boldsymbol{x}}_{g,d})^2, \tag{2.13}$$

using a lineal scale in time (Figure 2.23a), frequency (Figure 2.24a) and Hilbert (Figure 2.25a) domains or logarithmic scale (Figure 2.23b, Figure 2.24b and Figure 2.25b). From the three domains, the time produces the lowest variance values (Figure 2.23) while the frequency produces the highest variance values (Figure 2.24).



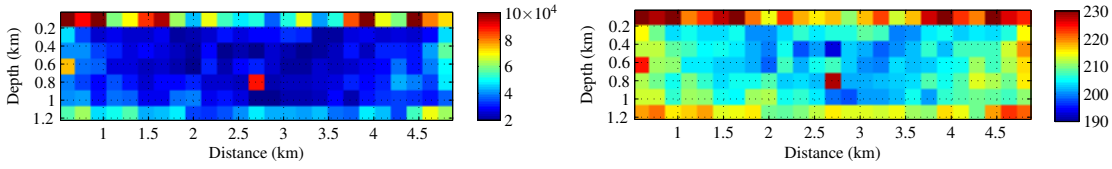
(a) *(b)*

Figure 2.23 Variance of the 100 best results in time using a) lineal scale and b) logarithmic scale ($20 \times \log(dB)$).



(a) *(b)*

Figure 2.24 Variance of the 100 best results in frequency using a) lineal scale and b) logarithmic scale ($20 \times \log(dB)$).



(a) *(b)*

Figure 2.25 Variance of the 100 best results in Hilbert domain using a) lineal scale and b) logarithmic scale ($20 \times \log(dB)$).

From the best 100 results, it is also possible to compute the covariance matrix as

$$\sigma_{i,j} = \frac{\sum_{e=1}^{E}(x_{g,i}^{e} - \hat{x}_{g,i})(x_{g,j}^{e} - \hat{x}_{g,j})}{E-1}, \tag{2.14}$$

and the correlation matrix as

$$r_{i,j} = \frac{\sigma_{i,j}}{\sqrt{\sigma_{i,i}\sigma_{j,j}}}. \tag{2.15}$$

According to Figure 2.26, the dimensions are strongly correlated for all the domains where the highest values are reached in the time (Figure 2.26a) and Hilbert (Figure 2.26c) domains.

2.6 Discussion

In this chapter, a methodology is formulated to find a starting point for the FWI based on the cycle-skipping concept. It is proposed to use PSO for the cost function exploration and three different domains to analyze the cycle-skipping effects. A workflow of the full implementation of this methodology is explained and a statistical framework is proposed to analyze the results. An experiment is formulated to estimate the execution time and to validate the methodology.

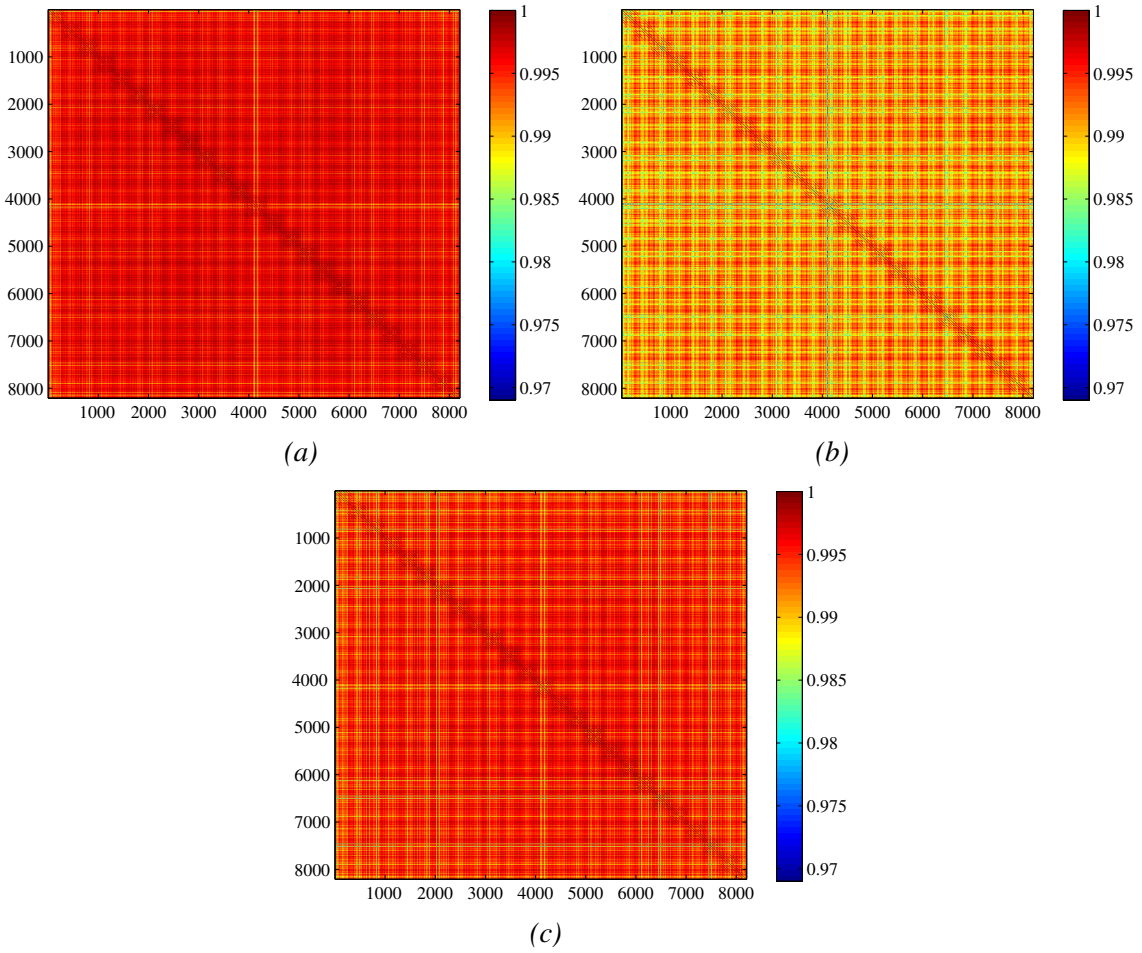


Figure 2.26 Correlation matrices of the 100 best results in (a) time, (b) frequency and (c) Hilbert domains.

The parameters recommended by Shaw & Srivastava (2007) for a resistivity inversion ($w = 1$, $c_1 = 2.8$ and $c_2 = 3.1$) were used in the first PSO implementation. After performing various experiments it was found that for this velocity inversion $w = 0.7$, $c_1 = 0.8$ and $c_2 = 1.1$ reach a better performance than Shaw & Srivastava (2007) in terms of the ℓ_2-norm between the observed data d_{obs} and the modeled data d_{mod}. Probably, this set of parameters do not allow to get the best possible PSO result for each model, but they allow to reach the neighborhood of the global minimum, which is our goal.

The number of particles P and iterations per particle K strongly depend on the cost function dimension D. This is because a greater number of dimensions implies a larger area to be explored and then more particles are needed to perform the exploration of the new cost function. In addition, an increase in the number of iterations per particle means more time to perform the

exploration. For example, when PSO was used over five bi-dimensional ($D = 2$) test functions, the values of $P = 50$ and $K = 30$ allowed to reach the global minimum neighborhood. However, in the geophysical example ($D = 211 \times 68$), it was necessary to increase both the number of particles ($P = 500$) and the number of iterations per particle ($K = 100$) to reach the global minimum neighborhood. Additionally, this thesis exploits the natural parallelism of the PSO algorithm through their parallel implementation on GPUs to reduce the computational cost but the effects of incrementing D must be carefully handled to avoid unfeasible PSO explorations.

The cycle-skipping effect was measured in three different domains. The best performance was obtained in the Hilbert domain (see Figure 2.20) because the complex trace allows to keep the time delay together with the phase information. And the worse performance was obtained in the frequency domain (see Figure 2.20) because the time delay information is missed. For a future study of the cycle-skipping effect in other domains, it is highly recommended to choose domains that preserve both the time delay and the phase information.

The metric formulated in this thesis only uses the cycle-skipping effect. However, more geophysical information (e.g wavelet shape, amplitude information, diving waves, etc.) can be incorporated to the metric to produce constraints that reduce the search space. But the geophysical information must be taken with special care because to many constraints can produce overfitting models since each recorded wave carries information about the subsurface characteristics.

From the statistical test it is seen that the dimensions behavior can be adjusted to a Gaussian PDF for all the domains (see Figure 2.22), but from the correlation matrices (see Figure 2.26) it is seen that the dimensions are strongly correlated. This allows to conclude that the dimensions are not i.i.d. Therefore, statistical techniques for non i.i.d. data must be take into account to perform a proper analysis (Beskos & Stuart (2009), Cichocki, Rutkowski, Barros & Oh (2000), Paganoni, Secchi & others (2015) and Zhang, Song, Gretton & Smola (2009)).

PSO correctly resolves the low frequency information through the cycle-skipping exploration and then FWI resolves the high frequency information through the multi-scale implementation. Figure 2.17 illustrates the source wavelet spectra used in the whole process. It was extracted a random seismic trace from the observed data and its phase was compared with the PSO and the multi-scale FWI result. The example presented in Figure 2.18 illustrates that the PSO result (blue color) reduces the phase difference in the frequency interval $0 \leq f \leq 6$ Hz and the FWI result (red color) reduces the phase difference in the frequency band $6 \leq f \leq 20$ Hz. Since this behavior is replicated on all the traces then the inversion process can reach a velocity model with characteristics of the original one (Figure 2.13b, Figure 2.14b and Figure 2.15b).

In this chapter it has been shown that PSO can reach starting points inside the global minimum neighborhood for FWI implementations when there is a lack of a priori information. Moreover, PSO could be a complementary technique that can work together with the conventional seismic techniques (e.g. tomography) to obtain better starting points for the FWI.

The PSO exploration of the geophysical example ($D = 211 \times 68$) is feasible because the 5×10^6 propagations spend 1.39 hours since, in this case, one propagation requires 1 ms using the GPU Tesla k40c of Nvidia (see Table 2.4). For a 2D real model ($D = 10000 \times 8000$) the number of propagations would increase to 5×10^{10} and the time of one propagation would be 60 seconds using the same GPU. Therefore, the total time would be 9.6×10^4 years (see Table 2.4). Finally, a 3D synthetic model ($D = 211 \times 68 \times 111$) would spend 634.2 years using the same GPU because it would be needed 1×10^9 propagations where one propagation spends 20 seconds (see Table 2.4).

Table 2.4
Total time estimation of the PSO exploration over 2D and 3D models using the GPU Tesla k40c of Nvidia.

Model	D	P	K	E	$P \times K \times E$	Propagation (s)	Time spent
Geophysical example	8000	500	100	100	5×10^6	1×10^{-3}	1.39 hours.
2D real model	8×10^7	5×10^6	100	100	5×10^{10}	60	9.6×10^4 years.
3D synthetic model	16×10^5	1×10^5	100	100	1×10^9	20	634.2 years.

The PSO exploration for a 2D real model or a 3D synthetic model will be feasible when the propagation time over a number of dimensions $D = 1 \times 10^7$ *can be lower than* 1×10^{-6} *seconds.*

3. Second Challenge: Local optimization technique

Full Waveform Inversion (FWI) is a non-linear inversion method that iteratively estimates sub-surface properties, such as the seismic velocity. The main components of FWI are: a cost function to measure the misfit between the observed and modeled data, a wave propagator to compute the modeled data and an initial velocity model that is iteratively updated until the cost function reaches an adequate value.

Usually the misfit function is the least squares error function given by

$$\Phi(\mathbf{v}) = \frac{1}{2}\|G(\mathbf{v}) - \mathbf{d}^{obs}\|_2^2, \tag{3.1}$$

where G is the forward non-linear operator, $\mathbf{v} \in \mathbf{R}^{N_x \times N_z}$ is the velocity model, N_x and N_z are the model dimensions, and $\mathbf{d}^{obs}$ is the data recorded at the surface. To estimate $\mathbf{v}$ from $\mathbf{d}^{obs}$ is an ill-posed problem, and the solution is non-unique. From an optimization point of view, the misfit function could have a global minimum and more than one local minimum. Thus, the convergence towards the global minimum, when using an iterative inverse method, depends on the selected initial velocity model $\mathbf{v}_0$. Chapter 2 discuss how to find $\mathbf{v}_0$ from the FWI point of view.

The velocity model can be updated using the first two terms of the Taylor series around the starting velocity model $\mathbf{v}^k$ as

$$\mathbf{v}^{k+1} = \mathbf{v}^k - \alpha \cdot [\mathbf{H}(\mathbf{v}^k)]^{-1}\mathbf{g}(\mathbf{v}^k), \tag{3.2}$$

where $\mathbf{g}(\mathbf{v}^k)$ and $[\mathbf{H}(\mathbf{v}^k)]^{-1}$ are the gradient and the inverse of the Hessian matrix evaluated at the point $\mathbf{v}^k$ and α represents the step length. The gradient for the 2D acoustic and isotropic wave equation presented in Equation 2.3 is computed using the first order adjoint state method by Plessix (2006) as

$$\mathbf{g}(\mathbf{v}^k) = \sum_s \frac{-2}{(v^k(\mathbf{x},\mathbf{z}))^3} \int_0^T q_s(\mathbf{x},\mathbf{z},T-t) \frac{\partial^2 p_s(\mathbf{x},\mathbf{z},t)}{\partial t^2} dt, \tag{3.3}$$

where p_s is the normal forward field using a source $src(x,z)$ at the spatial coordinates $x,\ z$; q_s is the backward field obtained having as source the error between the modeled and observed data, in reverse time; T is an specific time and t is the time variable; s represents a specific shot; each shot produces its own gradient, therefore the final gradient per iteration is the summation of all the gradients.

The Hessian matrix is not directly computed because calculating and storing the full matrix take $O(N_x \times N_z)^2$. However, there are some options that avoid the direct computation of this matrix, and instead, they compute the product $[\mathbf{H}(\mathbf{v}^k)]^{-1}\mathbf{g}(\mathbf{v}^k)$. L-BFGS (Liu & Nocedal, 1989) is one of the most common method to do this because it allows to control the quantity of RAM used in the process in contrast to BFGS.

The following sections describe the details of the step forward implementation, the stability and numerical dispersion of the FDTD stencil, the execution time of computing the gradient, and the multi-scale FWI implementation. The discussion and conclusions section close this chapter.

3.1 Step Forward

In this section, the way in which the step forward is computed is described. The first iteration uses linear interpolation and the next ones L-BFGS.

3.1.1 Linear interpolation

The main idea behind linear interpolation (Nocedal & Wright, 2006) is to perturb the gradient by a constant value ε to obtain a change of one percent over the new velocity model as

$$\begin{aligned} \mathbf{v}_{k+1} &= \mathbf{v}_k - \alpha \mathbf{g}_k \\ 0.01\mathbf{v}_k + \mathbf{v}_k &= \mathbf{v}_k - \varepsilon \mathbf{g}_k \\ 0.01\mathbf{v}_k &= -\varepsilon \mathbf{g}_k \\ \varepsilon &= |-\frac{0.01\mathbf{v}_k}{\mathbf{g}_k}|_{minimum}. \end{aligned} \tag{3.4}$$

The cost function at the point ε is smaller than the cost function at the initial point ($\phi(\varepsilon) < \phi(0)$) because the iterative solution moves in the opposite direction of the gradient. According to Figure 3.1 the cost function evaluated at the point α_1 is also smaller than the cost function at the initial point, where α_1 can be computed as

$$\begin{aligned} \phi(\alpha) &= m\alpha + b \\ \phi(\alpha) &= m\alpha + \phi(0) \\ \phi(\alpha_1) &= (\frac{\phi(\varepsilon) - \phi(0)}{\varepsilon - 0})\alpha_1 + \phi(0) \\ 0 &= (\frac{\phi(\varepsilon) - \phi(0)}{\varepsilon - 0})\alpha_1 + \phi(0) \\ \alpha_1 &= \frac{\phi(0)\varepsilon}{\phi(0) - \phi(\varepsilon)}. \end{aligned} \tag{3.5}$$

If $\phi(\alpha_1) < \phi(0)$ then α_1 becomes the first alpha value. Otherwise, ε will be the first α value to guarantee that the cost function always decreases during the first step.

3.1.2 L-BFGS

L-BFGS method was defined by Liu & Nocedal (1989) to calculate an approximation of the product between the inverse of the Hessian matrix and the gradient. This approximation uses the last *m* gradients and velocity models to compute the step forward and it needs at least two

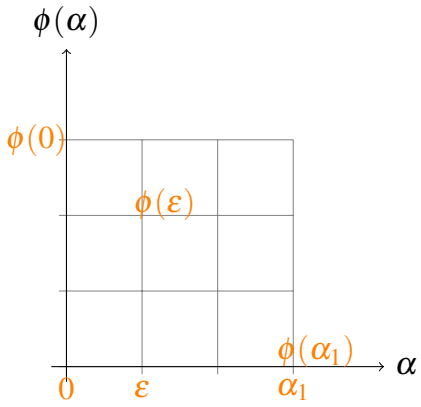


Figure 3.1 Linear interpolation principle.

gradients and two velocity models to obtain the search direction r.

The pseudo-code of this method is presented in Algorithm 4 where $\mathbf{s}_k = \mathbf{v}_{k+1} - \mathbf{v}_k$, $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ and $\sigma_k = 1/\mathbf{y}_k^T \mathbf{s}_k$. The matrix $\mathbf{D}_k^0$ is approximated by a diagonal matrix $\mathbf{D}_k^0 = \gamma_k \mathbf{I}$, with

$$\gamma_k = \frac{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^T \mathbf{y}_{k-1}}. \tag{3.6}$$

Algorithm 4 L-BFGS method

$\mathbf{q} \leftarrow \mathbf{g}_k$ ▷ Save gradient

for $i = k-1, k-2, \cdots, k-m$ **do** ▷ m, gradients history

$\varepsilon_i \leftarrow \sigma_i \mathbf{s}_i^T \mathbf{q}$;

$\mathbf{q} \leftarrow \mathbf{q} - \varepsilon_i \mathbf{y}_i$;

end for

$\mathbf{r} \leftarrow \mathbf{D}_k^0 \mathbf{q}$;

for $i = k-m, k-m+1, \cdots, k-1$ **do**

$\beta \leftarrow \sigma_i \mathbf{y}_i^T \mathbf{r}$;

$\mathbf{r} \leftarrow \mathbf{r} + \mathbf{s}_i(\varepsilon_i - \beta)$;

end for

Algorithm 5 shows how to compute α through attempts (each while-loop is an attempt). L-BFGS usually needs one attempt to obtain a new $\mathbf{v}_{k+1}$ when $\mathbf{v}_k$ is far from the solution and more than one attempt when it is close to converge (Dos Santos & Pestana, 2015).

Algorithm 5 α attempts

$\alpha = 2$

$\mathbf{h}_k = r$; ▷ Input: r comes from L-BFGS.

Compute $\phi_{(\mathbf{v}_k)}$;

$\phi_{(attempt)} = \phi_{(\mathbf{v}_k)} + 100$;

while $\phi_{(attempt)} > \phi_{(\mathbf{v}_k)}$ **do**

$\alpha = \alpha/2$;

$attempt = \mathbf{v}_k - \alpha\mathbf{h}_k$;

Compute $\phi_{(attempt)}$;

end while

$\mathbf{v}_{k+1} = attempt$ ▷ Output: $\mathbf{v}_{k+1}$

3.2 Stability and Numerical Dispersion

This section presents the general expression that allows to measure the stability and numerical dispersion of implementing Equation 2.3 using Finite Differences in Time Domain (FDTD).

3.2.1 FDTD

Before discussing the numerical stability and dispersion, it is important to highlight that each FDTD stencil of Equation 2.3 has its own stability equations. A second order stencil in time and eighth order stencil in space has been used through all this book because this combination keeps the numerical dispersion in the order of 1×10^{-3} when $k\Delta h$ is less than 1.

Let consider a plane wave propagating through a bi-dimensional space in the following way

$$U_{l,m}^{n} = U_0 \exp^{j(n\omega\Delta t - lk_x\Delta x - mk_z\Delta z)} \tag{3.7}$$

where $j = \sqrt{(-1)}$, ω is the angular frequency, k_x, k_z are the components of the wave-number-vector k in the directions x and z, Δx and Δz are the spacial resolution, Δt is the time step and, n, l, m are the discretization in the time, x and z directions, respectively.

The second order stencil in time and eighth order stencil in space of central Finite Difference for the 2 D wave equation is

$$\begin{aligned} U_{l,m}^{n+1} - 2U_{l,m}^{n} + U_{l,m}^{n-1} = \frac{v^2\Delta t^2}{\Delta x^2}[&\frac{-1}{560}U_{l-4,m}^{n} + \frac{8}{315}U_{l-3,m}^{n} - \frac{1}{5}U_{l-2,m}^{n} + \frac{8}{5}U_{l-1,m}^{n} - \frac{205}{72}U_{l,m}^{n} \\ &+ \frac{8}{5}U_{l+1,m}^{n} - \frac{1}{5}U_{l+2,m}^{n} + \frac{8}{315}U_{l+3,m}^{n} - \frac{1}{560}U_{l+4,m}^{n}\frac{-1}{560}U_{l,m-4}^{n} + \frac{8}{315}U_{l,m-3}^{n} \\ &- \frac{1}{5}U_{l,m-2}^{n} + \frac{8}{5}U_{l,m-1}^{n} - \frac{205}{72}U_{l,m}^{n} + \frac{8}{5}U_{l,m+1}^{n} - \frac{1}{5}U_{l,m+2}^{n} + \frac{8}{315}U_{l,m+3}^{n} - \frac{1}{560}U_{l,m+4}^{n}] \end{aligned} \tag{3.8}$$

where v is the velocity of the medium and the stability condition is (Equation A.12)

$$\alpha \leq \frac{1}{\sqrt{3.25}} = 0.55.$$

The numerical dispersion appears when the phase velocity v_p is different from the medium velocity v (Alford, Kelly & Boore, 1974). The phase velocity is defined as

$$v_p = \frac{\omega}{k} \tag{3.9}$$

and the numerical dispersion can be measured as

$$\frac{v_p}{v}. \tag{3.10}$$

According to Equation 3.10, there is numerical dispersion on the implementation when the relation is different from 1. Equation A.14:

$$\frac{v_p}{v} = \frac{2}{\alpha k\Delta h}\sin^{-1}[\alpha\sqrt{\begin{aligned}&-\frac{1}{560}\sin^2(2\cos(\theta)k\Delta h) + \frac{8}{315}\sin^2(3\cos(\theta)k\Delta h/2) - \frac{1}{5}\sin^2(\cos(\theta)k\Delta h) \\ &+\frac{8}{5}\sin^2(\cos(\theta)k\Delta h/2) - \frac{1}{560}\sin^2(2\sin(\theta)k\Delta h) + \frac{8}{315}\sin^2(3\sin(\theta)k\Delta h/2) - \frac{1}{5}\sin^2(\sin(\theta)k\Delta h) \\ &+\frac{8}{5}\sin^2(\sin(\theta)k\Delta h/2)]\end{aligned}}$$

allows to perform an analysis of numerical dispersion where $\alpha = \frac{v\Delta t}{\Delta x}$, $k = \sqrt{k_x^2 + k_z^2}$, $k_x = k\cos(\theta)$, $k_z = k\sin(\theta)$, θ is the angle of propagation, and $\Delta x = \Delta z = \Delta h$ (see Appendix A).

Figures 3.2 and 3.3 show the numerical dispersion in function of θ, α and $k\Delta h$. According to both figures, a wave field traveling through a velocity model will be always jumping between the dispersion curves because α depends of the velocity of the medium and the angle changes with each reflection. For that reason it is always recommended to keep $k\Delta h < 1$.

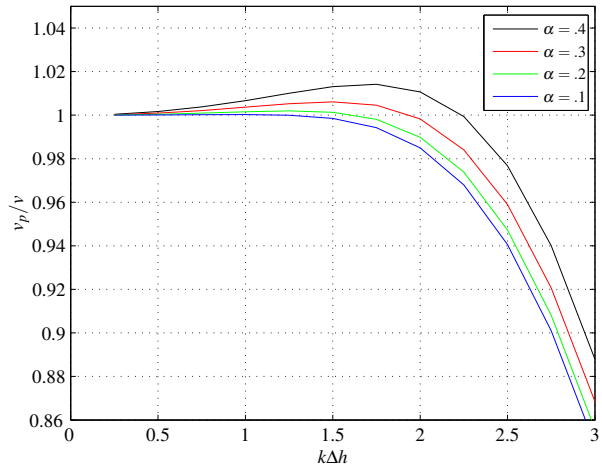


Figure 3.2 Numerical dispersion curves for different α values when $\theta = 0,\ \frac{\pi}{2},\ \pi,\ \frac{3\pi}{2}$, or 2π.

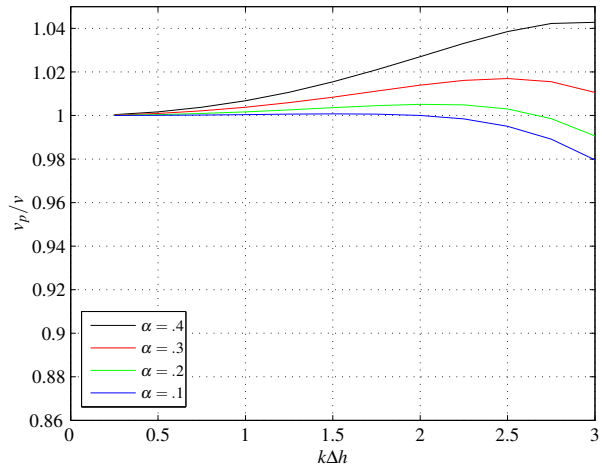


Figure 3.3 Numerical dispersion curves for different α values when $\theta = \frac{\pi}{4},\ \frac{3\pi}{4},\ \frac{5\pi}{4}$, or $\frac{7\pi}{4}$.

3.3 FWI Computational Strategies

This section describes the strategies proposed to take advantage of Graphical Processing Units (GPU) for computing the FWI.

3.3.1 Gradient Computation

According to Equation 3.3, two fields are needed to compute the gradient at the current velocity model ($\mathbf{g}(\mathbf{v}^k)$). The first field is the derivative of the forward field p_s shown in Figure 3.4 and the second field is the backward field q_s shown in Figure 3.5.

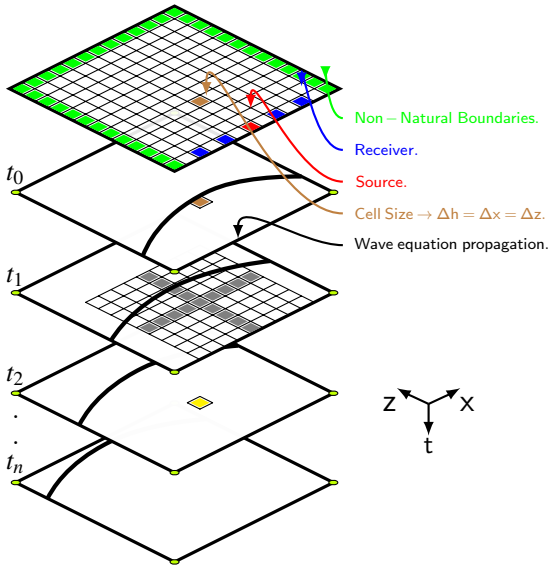


Figure 3.4 Snapshots of the propagation field p_s.

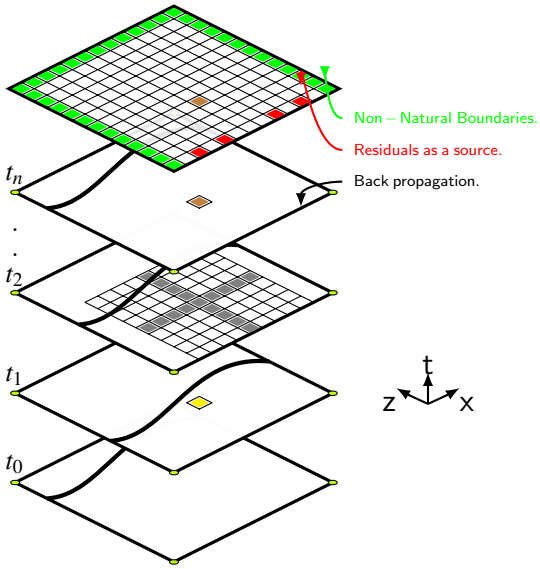


Figure 3.5 Snapshots of the backpropagation field q_s.

Both fields are computed using Equation 3.8 and Convolutional Perfectly Matched Layer (CPML) for the non-natural boundaries (green color in Figures 3.4 and 3.5) (Pasalic & McGarry, 2010). The field p_s is computed injecting a Ricker wavelet with a central frequency f_0 at the source position (red color in Figure 3.4) and it is sampled using the receivers at surface (blue color in Figure 3.4). On the other hand, compute the field q_s implies to remove the original source and injects the minus error between the modeled and observed data, in reverse time,

at the receivers position.

A single point, $U_{l,m}^{n+1}$ (see Equation 3.8), in the future layer of the propagated wave field (yellow color in Figure 3.4), requires information of seventeen points of the present (gray color in Figure 3.4), and one point of the past (brown color in Figure 3.4) wave fields, which are supposed to be known. Since the past and present layers of the wavefield are saved in RAM, every point of the future layer can be computed in parallel exploiting the real capacity of the GPUs.

3.3.2 GPU Implementation

Parallel applications running inside a GPU using functions are called kernel. When a kernel is launched, it uses blocks with a certain number of threads. A thread represents the minimum group of hardware elements required to run a task. All the threads within the same block work in parallel, and the number of blocks working in parallel depends on the number of Streaming Multiprocessors (SMs) inside the GPU. A kernel usually requires various blocks to execute a function and, depending on the number of SMs, the function is partially or completely parallelized.

If Equation 3.8 is assigned to each thread to calculate in parallel every future point of the wavefield (see yellow points in Figures 3.4, and 3.5) using the GPU Nvidia Tesla K40c with 1024 threads per block, two blocks per SMs and fifteen SMs ($1024 \times 2 \times 15$, see Table 3.1), then the total number of future points that can be computed in parallel is 30720.

A pseudo-code of the FWI implementation using two volume allocation in GPU is presented in Appendix B. The execution time of the FWI depends on the number of sources, total propagation time, the FDTD stencil and the number of iterations; however, different strategies can be used to balance the impact of these parameters over the final implementation.

Table 3.1
Nvidia Tesla K40c architecture.

Item	Tesla K40c
Stream Multiprocessors.(SMs)	15
Blocks per SM	2
Threads per Block (Max.)	1024
32-bits registers per Multiprocessor (Max.)	65536
Global memory	12GB
L2 cache size	1.5MB
Shared memory per block	48kB
Clk frequency	745MHz

3.3.3 FWI Strategies

The principal advantage of using GPUs for the FWI implementation is the possibility to compute every snapshot of the wave propagation in parallel. Thus, the pressure wavefield at all spatial positions of one snapshot is computed in a single clock cycle. However, the implementation of Algorithm 11 has an expensive RAM consumption because it needs to store the backward and forward wavefield volumes. Those volumes usually require between 80% to 90% of the total memory reserved for the implementation, limiting the use of GPU architectures in real problems. For this reason it is necessary to find a strategy that makes feasible the use of GPUs for real problems. In the following subsections, four different implementation strategies based on check pointing methods (Anderson, Tan & Wang (2012), Griewank & Walther (2000)) and wavefield reconstruction (Yang, Brossier, Metivier & Virieux, 2016) are discussed to reduce the volume allocation.

Strategy I: Dual volume allocation

The first strategy consists on allocating GPU RAM for all shot-gathers, velocity models, gradient and the backward and forward propagation volumes (Abreo, Ramirez, Abreo, Reyes & Gonzales, 2015). A GPU function performs the propagation and stores the backward and forward wavefields to then compute the gradient vector by first making a dot product of the wavefields, and then summing the resulting volume in the time direction. The main steps of this implementation strategy are given in Algorithm 6.

Algorithm 6 : Double volume allocation

1: **for** $t \leftarrow 1, Nt$ **do**
2: $\quad \frac{1}{v(x,z)^2}\frac{\partial^2 p_s}{\partial t^2} = \frac{\partial^2 p_s}{\partial x^2} + \frac{\partial^2 p_s}{\partial z^2} + S(sj,t)$ ▷ Compute and save forward field
3: **end for**
4: **for** $t \leftarrow 1, Nt$ **do**
5: $\quad \frac{1}{v(x,z)^2}\frac{\partial^2 q_s}{\partial t^2} = \frac{\partial^2 q_s}{\partial x^2} + \frac{\partial^2 q_s}{\partial z^2} + Residual(x, Nt-t)$ ▷ Compute and save backward field
6: **end for**
7: $g(x,z) = g(x,z) - \frac{2}{(v^k(\mathbf{x},\mathbf{z}))^3}\int_0^T q_s(x,z,T-t)\frac{\partial^2 p_s(x,z,t)}{\partial t^2}dt$ ▷ Compute Gradient

Strategy II: Single volume allocation

The second FWI implementation strategy consists on allocating GPU RAM for all matrices, shot-gathers, velocity model and a single propagation volume (Modified check pointing method, Anderson et al. (2012)). One GPU kernel function computes and stores the forward wavefield. A second GPU kernel computes the backward propagated wavefield and overwrites the allocated memory with the dot product between the computed snapshot of the backward wavefield and the stored forward wavefield. Finally, a third GPU kernel adds all the snapshots in order to obtain the gradient vector. The main steps of the second implementation strategy are shown in Algorithm 7.

Algorithm 7 : Single volume allocation

1: **for** $t \leftarrow 1, Nt$ **do**
2: $\quad \frac{1}{v(x,z)^2}\frac{\partial^2 p_s}{\partial t^2} = \frac{\partial^2 p_s}{\partial x^2} + \frac{\partial^2 p_s}{\partial z^2} + S(sj,t)$ ▷ Compute and save forward field
3: **end for**
4: **for** $t \leftarrow 1, Nt$ **do**
5: $\quad \frac{1}{v(x,z)^2}\frac{\partial^2 q_s}{\partial t^2} = \frac{\partial^2 q_s}{\partial x^2} + \frac{\partial^2 q_s}{\partial z^2} - Residual(x, Nt-t)$ ▷ Compute backward field
6: $\quad \frac{\partial^2 p_s(x,z,Nt-t)}{\partial t^2} = \frac{\partial^2 p_s(x,z,Nt-t)}{\partial t^2} * q_s(x,z,Nt-t)$ ▷ Overwrite forward field
7: **end for**
8: $g(x,z) = g(x,z) - \frac{2}{(v^k(\mathbf{x},\mathbf{z}))^3}\int_0^T \frac{\partial^2 p_s(x,z,t)}{\partial t^2}dt$ ▷ Compute Gradient

Strategy III: No volume allocation: version 1

The third FWI implementation strategy requires no GPU RAM allocation for the wavefield volumes (Wavefield reconstuction by reverse propagation (RP), Yang et al. (2016)). First of all, a first kernel computed the forward propagation to obtain the modeled data, calculate the residuals and save the last two snapshots ($p_s(x,z,Nt)$ and $p_s(x,z,Nt-1)$) together with the CPML boundaries at each time step. A second kernel computes the backward field, reconstructs

the forward field from the stored data, and calculates the gradient vector through the summation in time of the dot product between the snapshots. The main steps of this version of the third implementation strategy are given in Algorithm 8. However, this strategy is not feasible because the numerical errors during the reconstruction of the p_s field are amplified with the derivative destroying the gradient reconstruction.

Algorithm 8 : No volume allocation: version 1

1: **for** $t \leftarrow 1, Nt$ **do**
2: $\quad \frac{1}{v(x,z)^2}\frac{\partial^2 p_s}{\partial t^2} = \frac{\partial^2 p_s}{\partial x^2} + \frac{\partial^2 p_s}{\partial z^2} + S(sj,t)$ ▷ Compute forward field
3: **end for**
4: $save \leftarrow p_s(x,z,Nt)$ *and* $p_s(x,z,Nt-1)$ ▷ Final conditions to forward field reconstruction
5: **for** $t \leftarrow Nt, 1$ **do**
6: $\quad \frac{1}{v(x,z)^2}\frac{\partial^2 q_s}{\partial t^2} = \frac{\partial^2 q_s}{\partial x^2} + \frac{\partial^2 q_s}{\partial z^2} - Residual(x,t)$ ▷ Compute Backward Field
7: $\quad \frac{1}{v(x,z)^2}\frac{\partial^2 p_s}{\partial t^2} = \frac{\partial^2 p_s}{\partial x^2} + \frac{\partial^2 p_s}{\partial z^2} + S(sj,t)$ ▷ Re-compute forward field
8: $\quad g(x,z) = g(x,z) - \frac{2}{(v^k(x,\mathbf{z}))^3}\frac{\partial^2 p_s(x,z,t)}{\partial t^2} q_s(x,z,t)$ ▷ Overwrite Gradient
9: **end for**

Strategy III: No volume allocation: version 2

A second version of the third FWI implementation strategy requires no GPU RAM allocation for the wavefields volumes (RP, Yang et al. (2016)). Instead, it is computed the forward propagation to obtain the modeled data and calculate the residual traces. Then, backward propagation is computed saving the snapshots $q_s(x,z,1)$ and $q_s(x,z,2)$, together with the CPML boundaries for each time step. A GPU kernel reconstructs the backward propagation wavefield from the stored data while computing the forward propagation wavefield once again. The same GPU kernel computes gradient matrix by multiplying the forward and backward snapshots and summing in time. The main steps of the strategy III.v2 are given in Algorithm 9.

Table 3.2 shows the number of volume allocations (V.A.) and volume computations (V.C) needed for each strategy, taking into account the quality of the gradient reconstruction (G.R.).

From Table 3.2, strategy III.v2 fails in reaching a G.R with a 100% of quality because the reconstruction of the q_s field produces numerical errors that affect the final gradient.

Algorithm 9 : No volume allocation: version 2

1: **for** $t \leftarrow 1, Nt$ **do**
2: $\frac{1}{v(x,z)^2}\frac{\partial^2 p_s}{\partial t^2} = \frac{\partial^2 p_s}{\partial x^2} + \frac{\partial^2 p_s}{\partial z^2} + S(sj,t)$ ▷ Compute forward field
3: **end for**
4: **for** $t \leftarrow 1, Nt$ **do**
5: $\frac{1}{v(x,z)^2}\frac{\partial^2 q_s}{\partial t^2} = \frac{\partial^2 q_s}{\partial x^2} + \frac{\partial^2 q_s}{\partial z^2} - Residual(x, Nt-t)$ ▷ Compute backward field
6: **end for**
7: $save \leftarrow q_s(x,z,1)$ *and* $q_s(x,z,2)$ ▷ Initial conditions for backward field reconstruction
8: **for** $t \leftarrow 1, Nt$ **do**
9: $\frac{1}{v(x,z)^2}\frac{\partial^2 q_s}{\partial t^2} = \frac{\partial^2 q_s}{\partial x^2} + \frac{\partial^2 q_s}{\partial z^2} - Residual(x,t)$ ▷ Backward field reconstruction
10: $\frac{1}{v(x,z)^2}\frac{\partial^2 p_s}{\partial t^2} = \frac{\partial^2 p_s}{\partial x^2} + \frac{\partial^2 p_s}{\partial z^2} + S(sj,t)$ ▷ Re-compute forward field
11: $g(x,z) = g(x,z) - \frac{2}{(v^k(x,\mathbf{z}))^3}\frac{\partial^2 p_s(x,z,t)}{\partial t^2} q_s(x,z,t)$ ▷ Overwrite Gradient
12: **end for**

Table 3.2
FWI computational strategies.

Strategies	V. A.	V. C.	Store	Load	G. R.
I	2	2	100%	100%	100%
II	1	2	50%.	100%	100%
III.v1	0	3	0%.	150%	Incorrect
III.v2	0	4	0%.	200%	$< 100\%$

3.4 Multi-Scale FWI

The strategies described above are implemented in this section using the GPU Tesla K40c and the Nvidia visual profiling tools (nvvp) are used to measure the computational cost.

3.4.1 Experiment Description

The synthetic velocity model depicted in Figure 2.12 is used to obtain the observed data d_{obs}. The observed and modeled data were produced using a set of 171 receivers placed in line at a depth of 125 m every 25 m, starting from the position 525 m to the position 4775 m. Each receiver recorded 3.5 s at a time step of 1 ms ($\Delta t = 1 \times 10^{-3}\ s$).

The sources were placed at a depth of $S_z = 125\ m$ and their spatial location were obtained using the following equation:

$$S_x = S_R \cdot \lfloor a + \frac{(b-a)}{(n-1)} \cdot k \rceil \quad k = \{0, 1, 2, \cdots, n-1\} \tag{3.11}$$

where a is the grid position of the first source ($a = 21$), b is the grid position of the last source ($b = 191$), n is the number of sources ($n = 1,51$), S_R is the spatial resolution ($S_R = 25\ m$) and the operator $\lfloor\ \rceil$ represents the nearest integer. All the sources are described by

$$g_s = (1 - 2(\pi f_0)^2(t - t_0)^2)\exp(-(\pi f_0)^2(t - t_0)^2), \tag{3.12}$$

where f_0 is the central frequency in hertz, and t_0 is a time delay parameter in seconds. The velocity model depicted in Figure 3.6 with a constant value of 2000 m/s is used as starting point for a multi-scale FWI approach (Bunks, Saleck, Zaleski & Chavent, 1995) where the central frequencies of the source wavelet were set to 3 Hz, 6 Hz and 9 Hz and the time delay to 0.5 seconds.

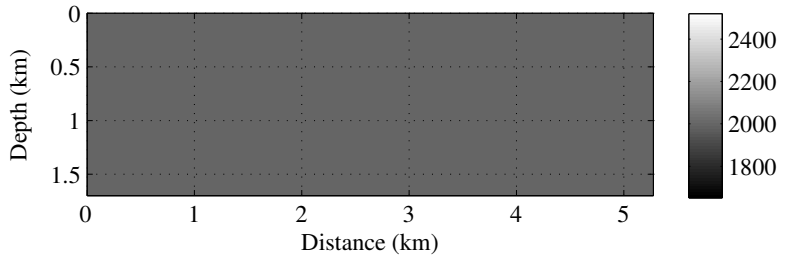
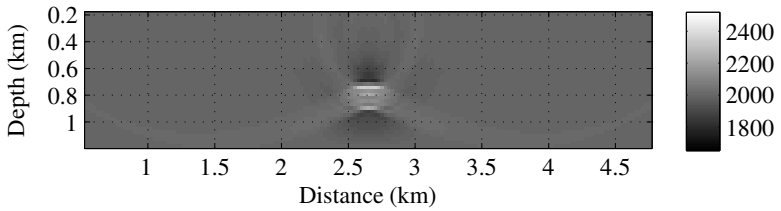


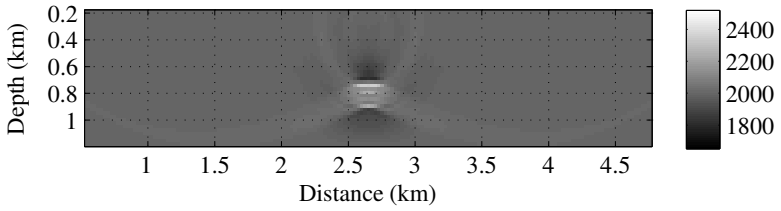
Figure 3.6 Initial velocity model [m/s] for multi-scale at 3 Hz, 6 Hz and 9 Hz.

The implementation uses Equation 3.8 to compute the wavefield propagation and Equation 3.3 to compute the gradient. For the first FWI iteration, linear interpolation is used to compute the step forward, and L-BFGS is used for the subsequent iterations. For this test, the inversion reaches the lowest values of the cost function after Nineteen iterations of L-BFGS. In Figures 3.7 and 3.8 are depicted the estimated velocity models for each strategy using multi-scale FWI with 1 and 51 sources, respectively.

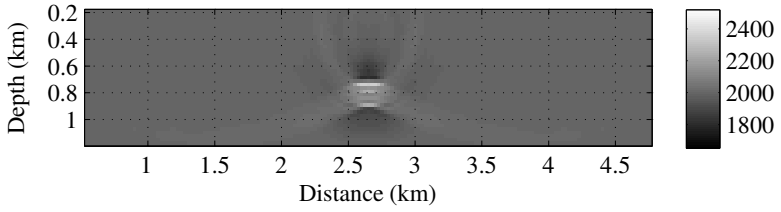
The estimated velocity models are compared with the original velocity model (Figure 2.12) to measure how the strategies affect the final results. The root mean squared error (RMSE) is the metric used for comparison and the results are summarized in Table 3.3. In accordance with Table 3.2, strategies I, II and III.v2 are about the same order of error. Table 3.4 summarizes the execution time and RAM required for all the implementations. The theoretical RAM for each



(a)



(b)



(c)

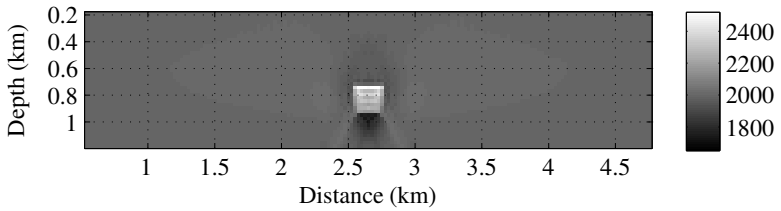
Figure 3.7 Final velocity models [m/s] after a Multi-scale at 3 Hz, 6 Hz and 9 Hz using one source and different strategies. (a) strategy I, (b) strategy II and (c) strategy III.v2.

Table 3.3
RMSE between true and estimated model for all strategies and different number of sources.

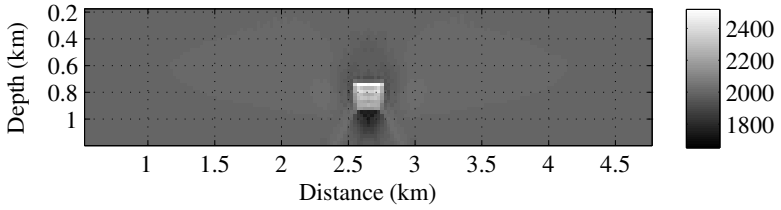
Strategy / No. Sources	1	51
I	46.315	37.223
II	46.256	37.231
III.v2	46.136	37.143

strategy is obtained from

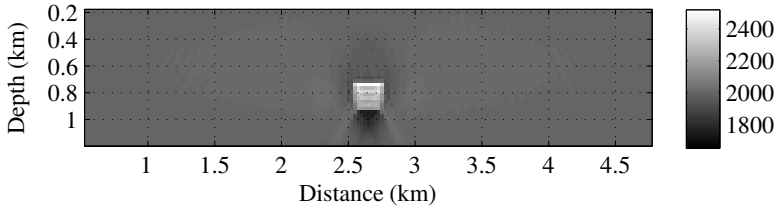
$$\begin{aligned}\text{RAM\_S\_I (MiB)} = RD + 4 * (&13 * Nx * Nz + 3 * Nx * Nt + 2 * Nx * Nz * Nt + Nx * Nt * Ns \\ &+ 2 * Nx + 2 * Nz + 5 * mz + 2 * (nG - 1) * mz + bpg + mt)/(2^{20}),\end{aligned} \tag{3.13}$$



(a)



(b)



(c)

Figure 3.8 Final velocity models [m/s] after a Multi-scale at 3 Hz, 6 Hz and 9 Hz using 51 sources and: (a) strategy I, (b) strategy II and (c) strategy III.v2.

$$\begin{aligned}\text{RAM\_S\_II(MiB)} = RD + 4 * (&13 * Nx * Nz + 3 * Nx * Nt + Nx * Nz * Nt + Nx * Nt * Ns \\ &+2 * Nx + 2 * Nz + 5 * mz + 2 * (nG - 1) * mz + bpg + mt)/(2^{20}),\end{aligned} \tag{3.14}$$

$$\begin{aligned}\text{RAM\_S\_III.v2(MiB)} = RD + 4 * (&22 * Nx * Nz + 4 * Nx * Nt + Nx * Nt * Ns + 2 * Nx + 2 * Nz \\ &+5 * mz + 2 * (nG - 1) * mz + bpg + mt + 2 * order * Nz * Nt + 2 * order * Nx * Nt)/(2^{20}),\end{aligned} \tag{3.15}$$

where $RD = 75$ MiB, $order = 8$, $nG = 19$, $mz = 2^{15}$, $mt = 2^{20}$ and $bpg = 4096$.

According to Table 3.4 the measured and theoretical RAM consumption are close. These theoretical equations allow to predict the RAM consumption of these strategies for different geometries and to determine when the 12 GiB are not enough. As expected, the execution time increases when it is applied a strategy for saving RAM (e.g. strategy II or III.v2) because the

Table 3.4
Theoretical and measured RAM requirements for all strategies including the execution time for different number of shots.

No. of shots	Strategy	Measur.[MiB]	Theor.[MiB]	Time [s]
1	I	478	479.2558	33.7
	II	286	287.6894	37.43
	III.v2	163	159.0335	43.73
51	I	618	620.1135	1649.53
	II	427	428.5471	1832.44
	III.v2	304	299.8912	2867.40

forward field or the backward field or both are recomputed several times. This implies a trade-off between the execution time and RAM. In this way, choosing the implementation strategy strongly depends on the hardware used.

3.5 Discussion

Full Waveform Inversion is a technique that looks for the velocity model that minimizes the difference between the modeled and observed data (seismic traces). FWI needs to reproduce the observed data from a modeling process using a wave equation. Depending on the waves that are going to be inverted, the observed data have to be filtered and the modeling stage must choose the correct wave equation operator considering that the computational cost of the inversion increases as more physical events are taken into account.

Currently the academy and industry use wave equation operators ranging from acoustic to viscoelastic for isotropic or anisotropic media using two and three dimensions. However, they are using clusters of CPUs for their FWI implementations over real data, since RAM requirements of the process cannot be provided by new technologies as GPUs yet.

In this work, the GPU Tesla k40c has been used to perform the FWI over synthetic data using the simplest wave equation operator. The GPU takes advantage of the high level of parallelism inside the modeling process reducing the computational cost when it is compared with a serial CPU implementation (Abreo et al., 2015). The main concern of using the GPU for

FWI is the RAM requirement. From Equations 3.13, 3.14 and 3.15, it is clear that the total RAM needed for a 2D inversion is directly proportional to the geometry of the acquisition and the recorded time of the receivers. For the geophysical experiment studied in this chapter, the RAM requirements were satisfied. Nevertheless, real geophysical surveys can require RAM resources larger than 12 GiB. For example, a 2D elastic or viscoelastic FWI would require more RAM than offered by one GPU; even an acoustic 3D FWI implementation could require more than 12 GiB. For this reason, it is important to explore a domain partition strategy where a group of GPUs are computing at the same time a unique FWI. In this way, the RAM resources and the computational power increases, leaving the main issue to the data transference.

Results in Table 3.3 proves that boundary saving strategies (e.g. Strategies II and III.v2) always introduce errors during the wavefield reconstruction that affects the final results. Sometimes the effect is constructive which increases the root mean squared error and sometimes the effect is destructive which produce the opposite behavior. The results of this chapter show that hypothesis as "*Because we are advocating the wavefield reconstruction method in FWI, the foremost thing is to demonstrate that the boundary saving strategy does not introduce any kind of errors or artifacts for the wavefield to be reconstructed*"[1] are not true and it is important to clarify them thinking on the great impact of these journals in the scientific community.

A second order in time and eighth order in space finite-differences in time domain approximation has been used to find the numerical solution of the 2-D acoustic wave equation. This approximation gives a feasible range of numerical dispersion keeping $k\Delta h < 1$ as it is illustrated in Figure 3.2 and Figure 3.3. The quality of the results, the RAM requirements and the execution time are summarized in Table 3.3 and Table 3.4, respectively. The multi-scale FWI has given the results illustrated in Figure 3.7 and Figure 3.8 for acoustic synthetic noiseless data. However, further analysis must be done on evaluating this stencil on real geophysical data because new issues arise about: the type of filter that must be used over the observed data, the wave equation operator that better emulates the observed data with a feasible computational

[1] Yang, Gao & Wang (2015, Numerical Results section, Exact reconstruction with saved boundaries subsection)

cost and, the stencil that keeps the numerical dispersion in the selected range.

The comparison between the original velocity model of Figure 2.12 with the results after a multi scale process using one source (Figure 3.7) and fifty one sources (Figure 3.8) allow to conclude that as the number of sources increases, the illumination over the subsurface model also increases, producing a velocity model with more details. These details allow to better define the physical characteristics as the size, the shape, the position and the velocity values inside the diffracting square. These characteristics are key to fully focus the reflections during a migration process like Reverse Time Migration (RTM).

The following experiments are focused on gradient computation strategy II because the size of the synthetic models, used to test the proposed methodology, allows storing the forward field.

4. Third Challenge: Uncertainty Quantification

Recently, full waveform inversion (FWI) is becoming more attractive for the estimation of high resolution parameters of the Earth's subsurface. The FWI method is formulated as a local optimization problem, where first-order or pre-conditioned first-order functions of the misfit between observed and modeled data, are used to find the search direction of a global or local minimum (Virieux & Operto, 2009; Wu & Alkhalifah, 2016). The Hessian matrix not only guides the descent direction in the local optimization method, but also allows quantifying the uncertainty of the parameters estimated after an inversion process (Kalmikov & Heimbach, 2014). For uncertainty quantification (UQ), the Hessian matrix is related to the covariance matrix of the estimated set of parameters, when Bayesian inference and Gaussian models are used to model the error between the real and estimated parameters (Tarantola, 2005). In fact, uncertainty of the FWI can be quantified using Gaussian model assumptions since ℓ_2-error norms are used as *data-fitting* term (Zhun, Li, Fomel, Stadler & Ghattas, 2016). However, the second-order information of the Hessian matrix, is usually neglected in FWI because compute this matrix for large scale problems (real inverse problems) is computationally expensive. In consequence, several methods have been proposed to compute approximations of the Hessian matrices, such as the BFGS and its variants (Nocedal, 1980), Newton, Gauss-Newton and Levenberg-Marquardt (Nocedal & Wright, 2006).

The first part of this chapter focuses on obtaining the exact Hessian matrix via the second order adjoint state method (SOASM), which computes Hessian matrix-vector products by using auxiliary variables called *adjoint variables* (Fichtner, 2010; Fichtner & Trampert, 2011; Métivier, Brossier, Operto, Virieux & others, 2012; Métivier, Brossier, Virieux & Operto, 2013; Nocedal & Wright, 2006). For the acoustic case, one column of the Hessian matrix is obtained

using four wave propagations via FDTD. In a FDTD scheme, every spatial point of a pressure field snapshot can be computed independently, which generates an intrinsic parallelism level in the numerical computation of the wave propagations. This thesis takes advantage of this parallelism to make feasible the computation of Hessian matrices using GPU architectures (Abreo et al., 2015).

The main contributions of this chapter are two fold: First, it is provided the formulation of the adjoint wavefields used to compute the Hessian matrix using the SOASM method for the acoustic and isotropic case. Special emphasis is devoted to compute the adjoint fields using parallel programming such that a feasible implementation on GPUs is proposed. Second, it is provided the link between the Hessian matrix and the uncertainty quantification of the estimated velocity model obtained with the acoustic FWI, as a function of the uncertainty of the *a priori* velocity model.

4.1 Hessian Matrix Formulation

The adjoint wavefields used to obtain Hessian-vector-products are presented in this section from an optimization point of view highlighting their relation with the Lagrange multipliers.

4.1.1 First Order Adjoint State Method

FWI can be formulated as an optimization problem, in the canonical form (Virieux & Operto, 2009), using Equation 2.4 and rewriting Equation 3.1 as

$$\min_{\mathbf{v}} \frac{1}{2}\|R\{L(\mathbf{v})[p]\} - \mathbf{d}^{obs}\|_2^2, \text{ subject to } L(\mathbf{v})[p] = f \tag{4.1}$$

where $L(\mathbf{v})[\cdot] = \nabla^2[\cdot] - \frac{1}{\mathbf{v}^2(x,z)}\frac{\partial^2[\cdot]}{\partial t^2}$ is the d'Alembertian operator that maps $\mathbf{v}$ from the model space $\mathfrak{M}$ to the wavefield space $\mathfrak{W}$ and, $R\{\cdot\}$ maps a wavefield from $\mathfrak{W}$ to the data space $\mathfrak{D}$. The Lagrangian $\mathscr{L}$ is applied over Equation 4.1 to include the constraint in the cost function

(Plessix, 2006) as

$$\mathscr{L}(\mathbf{v},p,\lambda)=\frac{1}{2}\langle(R\{L(\mathbf{v})[p]\}-\mathbf{d}^{obs}),(R\{L(\mathbf{v})[p]\}-\mathbf{d}^{obs})\rangle_d+\langle\ L(\mathbf{v})[p]-f,\lambda\ \rangle_w \tag{4.2}$$

where $\langle\cdot,\cdot\rangle_d$ is the inner product in $\mathfrak{D}$, $\langle\cdot,\cdot\rangle_w$ is the inner product in $\mathfrak{W}$, and the Lagrange multiplier $\lambda\in\mathfrak{W}^*$ the dual space of $\mathfrak{W}$. If the solution to the forward problem $\widehat{p}$ is replaced in Equation 4.2, such that

$$\mathscr{L}(\mathbf{v},\widehat{p},\lambda)=\Phi(\mathbf{v}) \tag{4.3}$$

then

$$\frac{\partial\mathscr{L}(\mathbf{v},\widehat{p},\lambda)}{\partial\mathbf{v}}=\frac{\partial\Phi(\mathbf{v})}{\partial\mathbf{v}}=\mathbf{g}(\mathbf{v}), \tag{4.4}$$

which is the same gradient of Equation 3.3. The derivative can be solved by the chain rule for multi-variable functions as

$$\frac{\partial\mathscr{L}(\mathbf{v},p,\lambda)}{\partial\mathbf{v}}=\frac{\partial\mathscr{L}(\mathbf{v},p,\lambda)}{\partial\mathbf{v}}\frac{\partial\mathbf{v}}{\partial\mathbf{v}}+\frac{\partial\mathscr{L}(\mathbf{v},p,\lambda)}{\partial p}\frac{\partial p}{\partial\mathbf{v}}+\frac{\partial\mathscr{L}(\mathbf{v},p,\lambda)}{\partial\lambda}\frac{\partial\lambda}{\partial\mathbf{v}} \tag{4.5}$$

where $\frac{\partial\lambda}{\partial\mathbf{v}}=0$ because λ does not depend on $\mathbf{v}$ and, $\frac{\partial\mathbf{v}}{\partial\mathbf{v}}=\mathbb{I}$ (identity matrix). Equation 4.5 is simplified to

$$\frac{\partial\mathscr{L}(\mathbf{v},p,\lambda)}{\partial\mathbf{v}}=\frac{\partial\mathscr{L}(\mathbf{v},p,\lambda)}{\partial\mathbf{v}}+\frac{\partial\mathscr{L}(\mathbf{v},p,\lambda)}{\partial p}\frac{\partial p}{\partial\mathbf{v}}, \tag{4.6}$$

therefore

$$\frac{\partial\mathscr{L}(\mathbf{v},p,\lambda)}{\partial p}\frac{\partial p}{\partial\mathbf{v}}=0. \tag{4.7}$$

After performing the derivatives of the first term of Equation 4.6 and replacing the forward problem solution $\widehat{p}$, it becomes

$$\frac{\partial\mathscr{L}(\mathbf{v},\widehat{p},\lambda)}{\partial\mathbf{v}}=\langle\frac{\partial L(\mathbf{v})[\,\widehat{p}\,]}{\partial\mathbf{v}},\lambda\rangle_w. \tag{4.8}$$

The Lagrange multiplier λ is obtained from Equation 4.7 as

$$L^*(\mathbf{v})[\lambda]=-R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\}-\mathbf{d}^{obs}\} \tag{4.9}$$

where $L^*(\mathbf{v})[\cdot]$ is the adjoint operator of $L(\mathbf{v})[\cdot]$ and the operator $R^T\{\cdot\}$ is the transpose of $R\{\cdot\}$ (Métivier et al., 2012). Equation 4.9 resembles a wave equation where the source is $-R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\}$ and λ is the computed wavefield. If Lagrange multiplier λ is replaced by the wavefield solution $\widehat{\lambda}$ in Equation 4.9 the equality of Equation 4.7 is reached and Equation 4.8 becomes Equation 4.4. In other words, the Lagrange multiplier λ becomes a wavefield that is commonly known as the adjoint field which coincide with the description given in Chapter 3.

4.1.2 Second Order Adjoint State Method

The gradient of the cost function can be rewriting as

$$\mathbf{g}(\mathbf{v}) = \mathbf{J}^T(R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\}) \tag{4.10}$$

where $\mathbf{J}(\cdot)$ is the Jacobian matrix and $\mathbf{g}(\mathbf{v}) \in \mathfrak{M}$ (Métivier et al., 2013, Equation 2.1). The functional

$$h_\upsilon(\mathbf{v}) = \langle \mathbf{g}(\mathbf{v}), \upsilon \rangle_m, \tag{4.11}$$

represents the inner product in $\mathfrak{M}$, $\langle \cdot, \cdot \rangle_m$, between the gradient of the cost function and the vector $\upsilon \in \mathfrak{M}^*$, the dual space of $\mathfrak{M}$. By definition, the gradient of the functional is

$$\nabla h_\upsilon(\mathbf{v}) = H(\mathbf{v})\upsilon, \tag{4.12}$$

where $H(\mathbf{v})$ is the Hessian matrix of the cost function $\Phi(\mathbf{v})$. If Equation 4.10 is replaced in Equation 4.11

$$h_\upsilon(\mathbf{v}) = \langle \mathbf{J}^T(R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\}), \upsilon \rangle_m, \tag{4.13}$$

with the Jacobian matrix operator moved using the transpose property

$$h_\upsilon(\mathbf{v}) = \langle R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\}, \mathbf{J}(\upsilon) \rangle_w, \tag{4.14}$$

and $\alpha \triangleq \mathbf{J}(\upsilon) \in \mathfrak{W}^*$

$$h_\upsilon(\mathbf{v}) = \langle R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\}, \alpha\rangle_w \tag{4.15}$$

then the canonical form of the optimization problem that allows to compute $H(\mathbf{v})\upsilon$ is

$$\min_{\mathbf{v}} \langle R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\}, \alpha\rangle_w, \text{ subject to } L(\mathbf{v})[p] = f,\ L(\mathbf{v})[\alpha] = \Phi_w. \tag{4.16}$$

The Lagrangian $\mathscr{L}$ is applied over Equation 4.16 to include the constraints in the cost function as

$$\mathscr{L}(\mathbf{v}, p, \lambda, \alpha, \mu) = \langle R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\}, \alpha\rangle_w + \langle\, L(\mathbf{v})[p] - f, \lambda\,\rangle_w + \langle\, L(\mathbf{v})[\alpha] - \Phi_w, \mu\,\rangle_w \tag{4.17}$$

where $\mu \in \mathfrak{W}^*$. If the solutions to the forward problems $\widehat{p}$ and $\widehat{\alpha}$ are replaced in Equation 4.17

$$\mathscr{L}(\mathbf{v}, \widehat{p}, \lambda, \widehat{\alpha}, \mu) = \langle R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\}, \widehat{\alpha}\rangle_w = h_\upsilon(\mathbf{v}) \tag{4.18}$$

then

$$\frac{\partial \mathscr{L}(\mathbf{v}, \widehat{p}, \lambda, \widehat{\alpha}, \mu)}{\partial \mathbf{v}} = H(\mathbf{v})\upsilon. \tag{4.19}$$

The derivative can be solved by the chain rule for multi-variable functions as

$$\begin{aligned}\frac{\partial \mathscr{L}(\mathbf{v}, p, \lambda, \alpha, \mu)}{\partial \mathbf{v}} = &\frac{\partial \mathscr{L}(\mathbf{v}, p, \lambda, \alpha, \mu)}{\partial \mathbf{v}}\frac{\partial \mathbf{v}}{\partial \mathbf{v}} + \frac{\partial \mathscr{L}(\mathbf{v}, p, \lambda, \alpha, \mu)}{\partial p}\frac{\partial p}{\partial \mathbf{v}} \\ &+ \frac{\partial \mathscr{L}(\mathbf{v}, p, \lambda, \alpha, \mu)}{\partial \lambda}\frac{\partial \lambda}{\partial \mathbf{v}} + \frac{\partial \mathscr{L}(\mathbf{v}, p, \lambda, \alpha, \mu)}{\partial \alpha}\frac{\partial \alpha}{\partial \mathbf{v}} \\ &+ \frac{\partial \mathscr{L}(\mathbf{v}, p, \lambda, \alpha, \mu)}{\partial \mu}\frac{\partial \mu}{\partial \mathbf{v}}\end{aligned} \tag{4.20}$$

where $\frac{\partial \lambda}{\partial \mathbf{v}} = 0$, $\frac{\partial \mu}{\partial \mathbf{v}} = 0$ and $\frac{\partial \mathbf{v}}{\partial \mathbf{v}} = \mathbb{I}$ because λ and μ do not depend of $\mathbf{v}$. Similarly to Equation 4.7

$$\frac{\partial \mathscr{L}(\mathbf{v}, p, \widehat{\lambda}, \alpha, \mu)}{\partial p}\frac{\partial p}{\partial \mathbf{v}} = 0 \tag{4.21}$$

and,

$$\frac{\partial \mathscr{L}(\mathbf{v}, p, \lambda, \alpha, \widehat{\mu})}{\partial \alpha}\frac{\partial \alpha}{\partial \mathbf{v}} = 0. \tag{4.22}$$

After performing the derivative of the first term of Equation 4.20 and replacing the forward problem solutions $\widehat{p}$ and $\widehat{\alpha}$ it becomes

$$\begin{aligned} H(\mathbf{v})\upsilon_j = \langle\, \widehat{\mu}, \frac{\partial(L(\mathbf{v})[\,\widehat{p}\,]-f)}{\partial \mathbf{v}} \,\rangle_w + \langle\, \widehat{\lambda}, \frac{\partial(L(\mathbf{v})[\widehat{\alpha}]-\Phi_w)}{\partial \mathbf{v}} \,\rangle_w \\ + \sum_{j=1}^{k} \upsilon_j \langle\, \widehat{\lambda}, \frac{\partial(\frac{\partial(L(\mathbf{v})[\,\widehat{p}\,]-f)}{\partial \mathbf{v}})}{\partial \mathbf{v}_j} \,\rangle_w, \end{aligned} \tag{4.23}$$

where $k = N_x \times N_z$ represents the model size and, υ_j is the vector used to select the column of the Hessian matrix (Métivier et al., 2012). The Lagrange multipliers λ and μ are obtained from Equation 4.21 and Equation 4.22 as

$$L^*(\mathbf{v})[\,\widehat{\lambda}\,] = -R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\} \tag{4.24}$$

and

$$L^*(\mathbf{v})[\widehat{\mu}] = -R^T\{R\{L(\mathbf{v})[\widehat{\alpha}]\}\} - \sum_{j=1}^{m} \upsilon^T \frac{\partial L(\mathbf{v})[\widehat{\lambda}]}{\partial \mathbf{v}_j}. \tag{4.25}$$

Equations 4.9 and 4.24 are the same, and again the Lagrange multiplier μ resembles a wavefield when it is used the right hand side term of Equation 4.25 as a source.

4.2 Hessian-vector-products

In this section, the adjoint fields are formulated for the acoustic and constant density model of a seismic source wave propagation. Furthermore, the pseudo-code of the adjoint field implementations and the results for a synthetic velocity model are also presented.

4.2.1 Computing the derivative of the p field

The wave field p is produced by Equation 2.4 with the initial conditions given by $p(x,z,0) = 0$ and $\frac{\partial p}{\partial t}(x,z,0) = 0$. The non-natural boundaries of the Finite Differences in Time Domain (FDTD) numerical implementation are handled using Convolutional Perfectly Matched Layer (CPML). Equation 2.4 can be re-written in a general form as

$$L(\mathbf{v})[p] = f, \tag{4.26}$$

where L represents the acoustic wave equation. The derivative of Equation 4.26 with respect to the model parameters is given by

$$\frac{\partial(L(\mathbf{v})[p]-f)}{\partial \mathbf{v}} = \frac{-2}{\mathbf{v}^3(x,z)}\frac{\partial^2 p}{\partial t^2}. \tag{4.27}$$

Once the auxiliary field $\frac{\partial^2 p}{\partial t^2}$ has been computed, the solution for Equation 4.27 is easily obtained by multiplying the auxiliary field by the scale factor $\frac{-2}{\mathbf{v}^3(x,z)}$. This is shown in steps 1 and 2 of Algorithm 10.

4.2.2 Computing the adjoint field λ

Let the adjoint field λ, in operator form, be given by

$$L(\mathbf{v})[\lambda] = f, \tag{4.28}$$

where f is defined in Equation 4.30. The derivative of previous operator with respect to the model parameters $\mathbf{v}$ is then

$$\frac{\partial(L(\mathbf{v})[\lambda]-f)}{\partial \mathbf{v}} = \frac{-2}{\mathbf{v}^3(x,z)}\frac{\partial^2 \lambda}{\partial t^2}. \tag{4.29}$$

When the acoustic and isotropic wave equation (Eq. 2.4) is used to compute the forward field p, its adjoint field can be computed using the same wave equation, because the system is self-adjoint ($L^*(\mathbf{v})[\cdot] = L(\mathbf{v})[\cdot]$). However, the propagation direction changes and the adjoint field λ represents a back-propagation, meaning that the final conditions of the problem are defined as $\lambda(x,z,T) = 0$ and $\frac{\partial \lambda}{\partial t}(x,z,T) = 0$. In order to compute the adjoint field λ, the receivers become sources and the real source disappears. The mathematical expression for the new set of sources is given by

$$f = -R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\}, \tag{4.30}$$

where d_{obs} represents the pressure field at the surface level (seismic traces), $R\{L(\mathbf{v})[\,\widehat{p}\,]\}$ are the modeled seismic traces, and $R^T\{\cdot\}$ represents the wave equation operator, reverse in time.

Note that the residuals, used as sources, are injected from the final sample to the first sample. In others words, the field λ is the same field q_s shown in Figure 3.5. Steps 3 and 4 of Algorithm 10 summarize the process.

4.2.3 Computing the derivative of α

Before obtaining $\frac{\partial(L(\mathbf{v})[\alpha]-\Phi_w)}{\partial \mathbf{v}}$, the auxiliary source Φ_w, and field α should be defined. Let Φ_w be an auxiliary source defined by (Métivier et al., 2013, Equation 2.16) as

$$\begin{aligned}\Phi_w &= -\sum_{j=1}^{k} \upsilon_j \frac{\partial(L(\mathbf{v})[p]-f)}{\partial \mathbf{v}},\\ \Phi_w &= -\sum_{j=1}^{N_x \times N_z} \upsilon_j \left(-\frac{2}{\mathbf{v}^3(x,z)} \frac{\partial^2 p}{\partial t^2} \right),\end{aligned} \tag{4.31}$$

where N_x is the number of grid points in the x direction, N_z is the number of points in the z direction, and υ_j is a perturbation vector having zero-value in all but the j^{th} coordinate. Introducing a non-zero value selects the location for the perturbation, which corresponds to one column vector of the Hessian matrix. If more than one perturbation is selected (*i.e.,* more than one element in the vector υ_j is different from zero) then the Hessian-vector-products compute a column that represents a linear combination of the selected Hessian matrix columns. The auxiliary field α is a forward field computed when the source is Φ_w, and the initial conditions are $\alpha(x,z,0) = 0$ and $\frac{\partial \alpha(x,z,0)}{\partial t} = 0$.

The derivative of the field α with respect to the model parameters $\mathbf{v}$, is given by

$$\frac{\partial(L(\mathbf{v})[\alpha]-\Phi_w)}{\partial \mathbf{v}} = \frac{-2}{\mathbf{v}^3(x,z)} \frac{\partial^2 \alpha}{\partial t^2} + \sum_{j=1}^{N_x \times N_z} \upsilon_j \left(\frac{6}{\mathbf{v}^4(x,z)} \frac{\partial^2 p}{\partial t^2} \right). \tag{4.32}$$

Equation 4.32 represents the interaction between the forward field α and the perturbation of the geophysical model. This is shown in steps 5, 6 and 7 of Algorithm 10.

4.2.4 Computing the adjoint field μ

The adjoint field μ is computed using as source

$$-R^T\{R\{L(\mathbf{v})[\widehat{\alpha}]\}\} - \sum_{j=1}^{N_x \times N_z} \upsilon_j \left(-\frac{2}{\mathbf{v}^3(x,z)} \frac{\partial^2 \tilde{\lambda}}{\partial t^2} \right), \tag{4.33}$$

where $R\{L(\mathbf{v})[\widehat{\alpha}]\}$ is the pressure field at the surface, obtained with the auxiliary field $\widehat{\alpha}$. This means that the adjoint field μ uses as sources the number of receivers and the perturbations of the geophysical model. Besides, the final conditions are $\mu(x,z,T) = 0$ and $\frac{\partial \mu}{\partial t}(x,z,T) = 0$. Note that $\frac{\partial^2 \lambda}{\partial t^2}$ was previously obtained. The computation of the adjoint field μ is shown in step 8 of Algorithm 10.

4.2.5 Computing the second derivate of the p field

The second derivative of the forward field p for the acoustic and isotropic case is given by

$$\frac{\partial^2 (L(\mathbf{v})[p] - f)}{\partial \mathbf{v}^2} = \frac{6}{\mathbf{v}^4(x,z)} \frac{\partial^2 p}{\partial t^2}. \tag{4.34}$$

This is shown in step 9 of Algorithm 10. At this point, the functions in Equation 4.23 are known and the columns of the Hessian matrix can be obtained. Fichtner & Trampert (2011) define the three main elements of the Hessian matrix-vector products as kernels.

4.2.6 Hessian Kernels

Each kernel represents the interaction of the source, receivers and perturbations during an acquisition. The source field propagates forward and the receiver and perturbation fields propagate backward in these kernels. To better explain this interaction, a simple case with one shot is reviewed, as shown in Figure 4.1. Each one of these three elements has an action radius that is depicted with dotted line and that allows to graphically analyze how each kernel is generated.

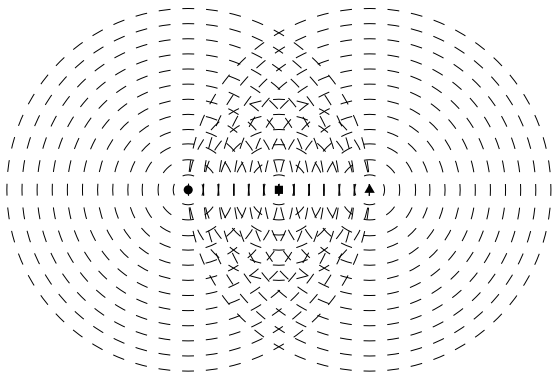


Figure 4.1 Interaction area of the source, perturbation and receiver. From left to right: Source •, perturbation ■ and receiver ▲.

Algorithm 10 Recipe to compute $H(\mathbf{v})\upsilon_j$

Require: Compute $L(\mathbf{v})[p] = f$. ▷ $f =$ Normal source.

1: $\frac{1}{\mathbf{v}^2(x,z)}\frac{\partial^2 p}{\partial t^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial z^2} + f(x,z)$ ▷ $p(x,z,0) = 0;\ \frac{\partial p}{\partial t}(x,z,0) = 0$

Require: Compute $\frac{\partial(L(\mathbf{v})[p]-f)}{\partial \mathbf{v}}$

2: $\frac{-2}{\mathbf{v}^3(x,z)}\frac{\partial^2 p}{\partial t^2}$

Require: Compute $L^*(\mathbf{v})[\lambda] = -R^T\{R\{L(\mathbf{v})[\,\widehat{p}\,]\} - \mathbf{d}^{obs}\}$ ▷ R^T =flip up to down the array.

3: $\frac{1}{\mathbf{v}^2(x,z)}\frac{\partial^2 \lambda}{\partial t^2} = \frac{\partial^2 \lambda}{\partial x^2} + \frac{\partial^2 \lambda}{\partial z^2} + flipud(d_{syn} - d_{obs})$ ▷ $\lambda(x,z,T) = 0;\ \frac{\partial \lambda}{\partial t}(x,z,T) = 0$

Require: Compute $\frac{\partial(L^*(\mathbf{v})[\lambda] + R^T\{R\{L(\mathbf{v})[p]\} - d_{obs}\}}{\partial \mathbf{v}}$

4: $\frac{-2}{\mathbf{v}^3(x,z)}\frac{\partial^2 \lambda}{\partial t^2}$

Require: Compute $\Phi_w = -\sum_{j=1}^{k}\upsilon_j\frac{\partial(L(\mathbf{v})[p]-f)}{\partial \mathbf{v}}$ ▷ $\upsilon_j =$ Perturbation mask.

5: $\Phi_w = -\sum_{j=1}^{N_x\times N_z}\upsilon_j\Big(-\frac{2}{\mathbf{v}^3(x,z)}\frac{\partial^2 p}{\partial t^2}\Big)$

Require: Compute $L(\mathbf{v})[\alpha] = \Phi_w$ ▷ $\alpha(x,z,0) = 0;\ \frac{\partial \alpha}{\partial t}(x,z,0) = 0$

6: $\frac{1}{\mathbf{v}^2(x,z)}\frac{\partial^2 \alpha}{\partial t^2} = \frac{\partial^2 \alpha}{\partial x^2} + \frac{\partial^2 \alpha}{\partial z^2} - \sum_{j=1}^{N_x\times N_z}\upsilon_j(-\frac{2}{\mathbf{v}^3(x,z)}\frac{\partial^2 p}{\partial t^2})$

Require: Compute $\frac{\partial(L(\mathbf{v})[\alpha]-\Phi_w)}{\partial \mathbf{v}}$

7: $\frac{-2}{\mathbf{v}^3(x,z)}\frac{\partial^2 \alpha}{\partial t^2} + \sum_{j=1}^{N_x\times N_z}\upsilon_j(\frac{6}{\mathbf{v}^4(x,z)}\frac{\partial^2 p}{\partial t^2})$

Require: Compute $L(\mathbf{v})[\mu] = -R^T\{R\{L(\mathbf{v})[\widehat{\alpha}]\}\} - \sum_{j=1}^{N_x\times N_z}\upsilon_j\Big(-\frac{2}{\mathbf{v}^3(x,z)}\frac{\partial^2 \tilde{\lambda}}{\partial t^2}\Big)$ ▷ Final conditions.

8: $\frac{1}{\mathbf{v}^2(x,z)}\frac{\partial^2 \mu}{\partial t^2} = \frac{\partial^2 \mu}{\partial x^2} + \frac{\partial^2 \mu}{\partial z^2} - R^T\{R\{L(\mathbf{v})[\widehat{\alpha}]\}\} - \sum_{j=1}^{N_x\times N_z}\upsilon_j\Big(-\frac{2}{\mathbf{v}^3(x,z)}\frac{\partial^2 \tilde{\lambda}}{\partial t^2}\Big)$ ▷ $\mu(x,z,T) = 0.$

▷ $\frac{\partial \mu}{\partial t}(x,z,T) = 0.$

Require: Compute $\frac{\partial^2(L(\mathbf{v})[p]-f)}{\partial \mathbf{v}^2}$

9: $\frac{6}{\mathbf{v}^4(x,z)}\frac{\partial^2 p}{\partial t^2}$

Require: $H(\mathbf{v})\upsilon_j = \langle\,\widehat{\mu}, \frac{\partial(L(\mathbf{v})[\,\widehat{p}\,]-f)}{\partial \mathbf{v}}\,\rangle_w + \langle\,\widehat{\lambda}, \frac{\partial(L(\mathbf{v})[\widehat{\alpha}]-\Phi_w)}{\partial \mathbf{v}}\,\rangle_w + \sum_{j=1}^{k}\upsilon_j\langle\,\widehat{\lambda}, \frac{\partial(\frac{\partial(L(\mathbf{v})[\,\widehat{p}\,]-f)}{\partial \mathbf{v}})}{\partial \mathbf{v}_j}\,\rangle_w$

10: $H(\mathbf{v})\upsilon_j = <Step\ 8, Step\ 2> + <Step\ 3, Step\ 7> + \sum_{j=1}^{N_x\times N_z}\upsilon_j < Step\ 3, Step\ 9>$

I. Kernel $k^{2\to1}$

The first kernel

$$k^{2\to1} \triangleq \langle\, \widehat{\mu}, \frac{\partial(L(\mathbf{v})[\,\widehat{p}\,]-f)}{\partial\mathbf{v}} \,\rangle_w, \tag{4.35}$$

represents the interactions of the source-perturbation and the source-receiver fields. Figure 4.2 shows the intersection between the source field and the perturbation field (red area) and the source-receiver interaction (black area). Fichtner & Trampert (2011) call the source-receiver interaction as a primary influence zone that corresponds to the first-order scattering and the source-perturbation interaction as a secondary influence zone that corresponds to second-order-scattering. From the three kernels, this is the one with more energy.

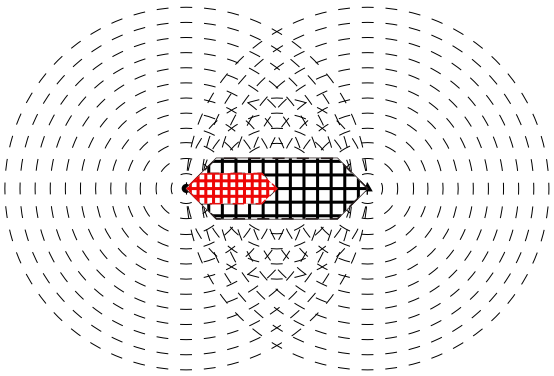


Figure 4.2 Kernel $k^{2\to1}$ for one source, one perturbation and one receiver. The primary influence zone is depicted in black color and the secondary influence zone in red color. From left to right: Source •, perturbation ■ and receiver ▲.

II. Kernel $k^{1\to2}$

The second kernel

$$k^{1\to2} \triangleq \langle\, \widehat{\lambda}, \frac{\partial(L(\mathbf{v})[\widehat{\alpha}]-\Phi_w)}{\partial\mathbf{v}} \,\rangle_w, \tag{4.36}$$

contains the interaction of the perturbation and the receiver fields where the perturbation field goes forward and the receiver field propagates backward. Figure 4.3 shows the intersection between this two fields (red area). This kernel represents the secondary influence zone created between the perturbation and the receiver.

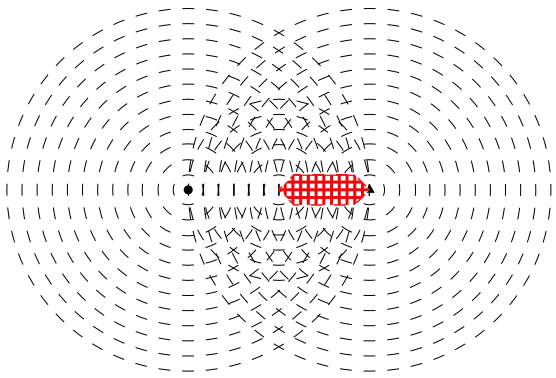


Figure 4.3 Kernel $k^{1\rightarrow 2}$ for one source, one perturbation and one receiver. The secondary influence zone is depicted in red color. From left to right: Source •, perturbation ■ and receiver ▲.

III. Kernel $k^{1\longleftrightarrow 1}$

The third kernel

$$k^{1\longleftrightarrow 1} \triangleq \sum_{j=1}^{k} \upsilon_j \langle\, \widehat{\lambda}, \frac{\partial\left(\frac{\partial(L(\mathbf{v})[\,\widehat{p}\,]-f)}{\partial \mathbf{v}}\right)}{\partial \mathbf{v}_j} \,\rangle_w, \tag{4.37}$$

contains the interaction between the source and the receiver fields but projected over the perturbation position. From the three kernels, this is the one with less energy and with a unique non-zero value at the perturbation positions. Figure 4.4 shows this kernel when there is only one perturbation (red dot).

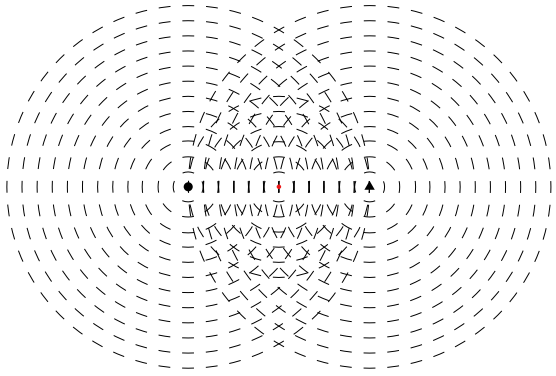


Figure 4.4 Kernel $k^{1\longleftrightarrow 1}$ for one source, one perturbation and one receiver. The secondary influence zone is depicted in red color. From left to right: Source •, perturbation ■ and receiver ▲.

IV. Hessian column

One column of the Hessian matrix is obtained if there is only one perturbation and the three kernels are summed. If this Hessian column of $N_x \times N_z$ rows is seen as an image of N_z rows and N_x columns, the link between the three kernels and the geophysical acquisition is clearer. Figure 4.5 shows the interaction of these three kernels to create the first and second influence zones. In a normal acquisition there is at least one source and more than one receiver activated. If one perturbation is applied to compute one column of the Hessian matrix, the effect shown in Figure 4.5 is replicated N_r times, where N_r is the number of receivers.

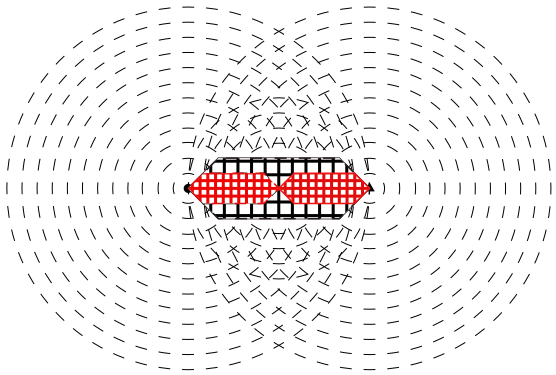


Figure 4.5 $H(\mathbf{v})\upsilon_j$ for one source, one perturbation and one receiver. The primary influence zone is depicted in black color and the secondary influence zone in red color. From left to right: Source •, perturbation ■ and receiver ▲.

4.2.7 Computing the kernels

To illustrate the kernels described above, Algorithm 10 is applied over the velocity models of Figure 2.12 (observed data) and Figure 3.6 (modeled data). The source $g_s(t)$ at the position $x = 2650$ m and $z = 175$ m is defined as

$$g_s(t) = -2\pi f_0(t - t_0)e^{(-2\pi f_0(t-t_0)^2)} \tag{4.38}$$

where $f_0 = 3$ Hz is the central frequency, and $t_0 = 0.5$ s is the time delay parameter. The perturbation is placed at $x = 2650$ m and $z = 850$ m using $N_r = 171$ receivers placed in line at a depth of 125 m every 25 m, starting from the position 525 m to the position 4775 m. Each receiver

recorded 3.5 s at a time step of 1 ms (1×10^{-3} s).

The computed kernels presented in Figures 4.6a, 4.6b and 4.6c are reshaped to the geophysical model dimensions ($N_z \times N_x$) to highlight their geophysical meaning. Figure 4.6a shows the interaction between the source, the perturbation and the 171 receivers. According to the theoretical analysis, this is the kernel with more energy (1×10^{-5}) and it shows a blurred interference in the middle of the model. Figure 4.6b shows the interaction between the perturbation and the 171 receivers. This second kernel shows slightly the shape of the object that produces the interference in the middle of the model (the diffracting square). Figure 4.6c shows the third kernel, which corresponds to a unique non-zero value placed at the perturbation coordinates $x = 2650$ m and $z = 850$ m. Finally, figure 4.7 shows the complete Hessian column obtained in a total time of 18.369 ms.

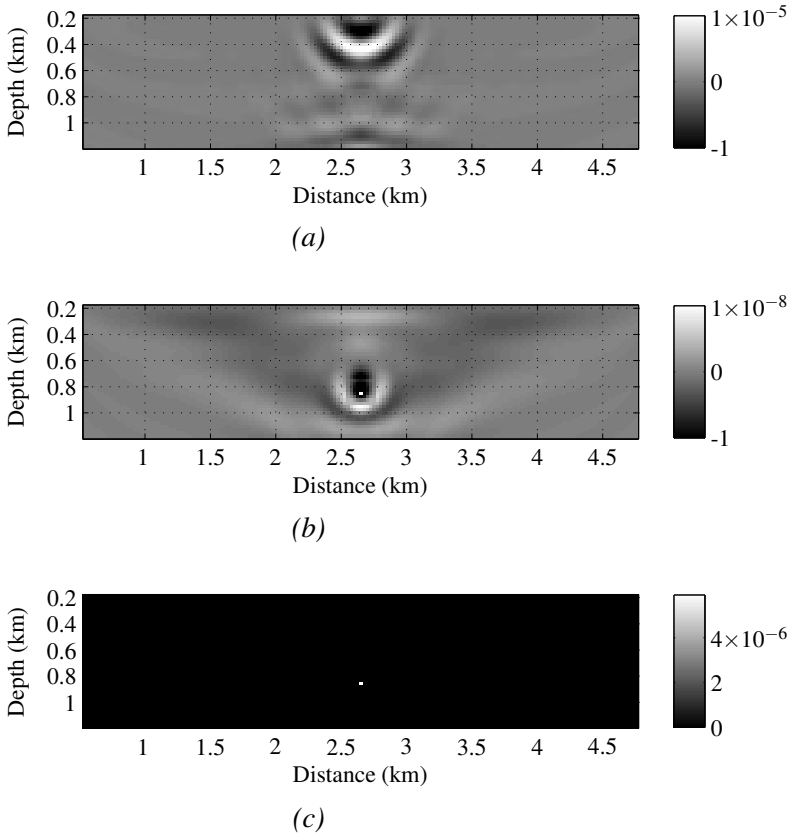


Figure 4.6 Hessian kernels. (a) Kernel $k^{2\rightarrow 1}$, (b) Kernel $k^{1\rightarrow 2}$, and (c) Kernel $k^{1\longleftrightarrow 1}$.

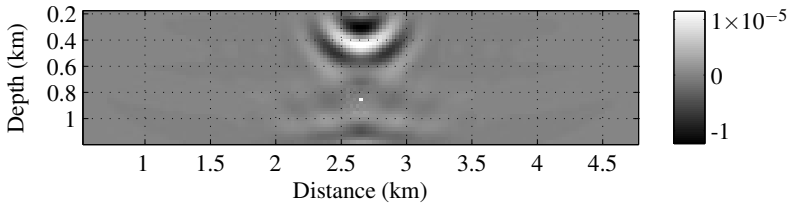


Figure 4.7 One column of the Hessian matrix $= k^{2\rightarrow 1} + k^{1\rightarrow 2} + k^{1\longleftrightarrow 1}$.

4.3 Hessian matrix

This section presents the characteristics, RAM requirements and execution times of the Hessian matrices computed with the SOASM for the multi-scale FWI results.

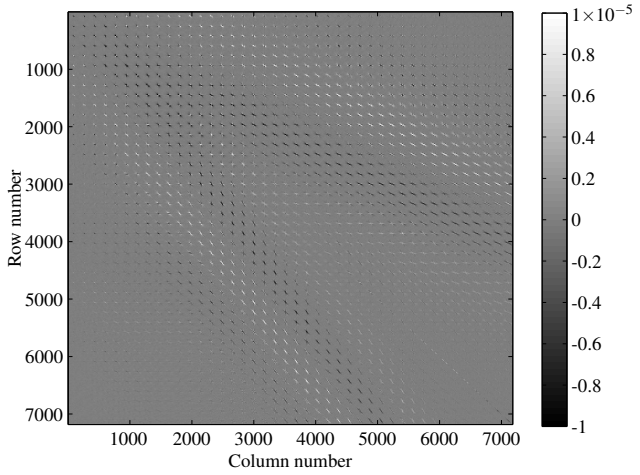
Figure 4.8 illustrates the full Hessian matrices obtained using the second order adjoint state method over the final velocity models after a Multi-scale at 3 Hz for one source (Figure 3.7a) and fifty-one sources (Figure 3.8a).

By definition the Hessian matrix is square, symmetric, full rank, with a condition number close to one and it is positive definite if evaluated over a local minimum (positive eigenvalues). However, the Hessian matrices obtained in this chapter only meet two of the five conditions (square shape and full rank) as summarized in Table 4.1.

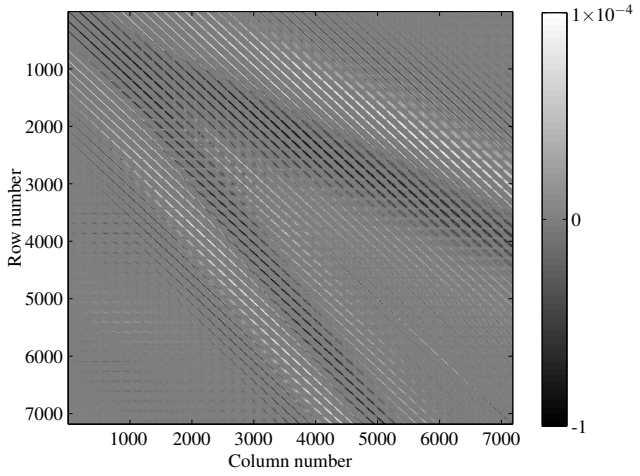
Table 4.1
Characteristics of the theoretical and computed Hessian matrices.

Hessian matrices	Shape	Rank	Symmetric	Eigenvalues	Condition number
Theoretical	Square	Full	Yes	$+$	≈ 1
H_{1s}	Square	Full	No	$+-$	2.12×10^8
H_{51s}	Square	Full	No	$+-$	7.42×10^9

The lack of symmetry is a consequence of using only superficial sources and receivers. To obtain a symmetric Hessian matrix, an array of sources and receivers should be located enclosing the area of interest, as explained by Virieux & Operto (2009). However, this is not feasible in real seismic acquisitions. Therefore, the resulting Hessian matrices are not positive definite (i.e., the eigenvalues are both positive and negative) even if the solution is in a local minimum, since in a surface acquisition, there are not enough observations and it is not possible to construct a high-dimensional Hessian matrix. A condition number and rank



(a)



(b)

Figure 4.8 Hessian matrices using (a) one source and, (b) fifty-one sources with a central frequency of 3 Hz.

measurement can be done over the Hessian matrix through a singular value decomposition (SVD) and analyzing its singular values (ξ) (Aster, Borchers & Thurber, 2005). For example, Figure 4.9 shows the singular values ξ for both Hessian matrices of Figure 4.8 in a logarithmic scale.

The singular values of both Hessian matrices are not zero and their magnitudes are -241.6 dB$\leq \xi_1 \leq -49.84$ dB and -258.3 dB$\leq \xi_{51} \leq -31.07$ dB (see Figure 4.9). This produces a full rank for both matrices (i.e., $\text{rank}(\xi_i) = N_x \times N_z = 7182$) with condition numbers much greater than one because of the ratio of the largest singular value to the smallest singular value is also much greater than one.

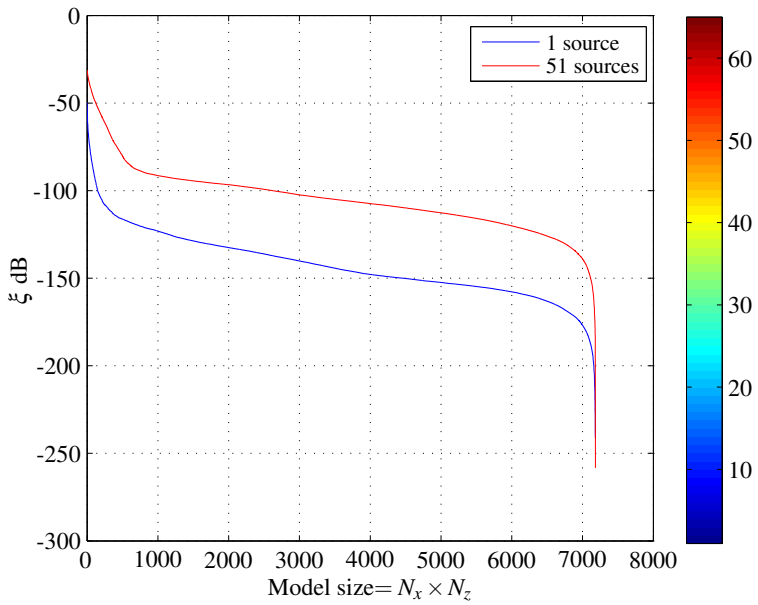


Figure 4.9 Eigenvalues of the Hessian Matrices at 3 Hz for one source and fifty-one sources.

4.3.1 Computational Cost

The cost of computing the full Hessian matrix using FDTD, in terms of execution time and memory resources, is high since four wave field propagations must be obtained for each column. In particular, the cost of computing a single column is evaluated when the FDTD approximation uses a second order stencil for time derivative and an eighth order stencil for the spatial derivatives. Two implementations are considered: (1) A serial ANSI-C implementation on an Intel(R) Xeon(R) E5-2609 and (2) a CUDA-C implementation on a Nvidia(R) GPU Tesla K40c. The execution times of one single column of the Hessian matrix are presented in Table 4.2. Additionally, Table 4.3 summarizes the execution times and RAM size required by the CUDA-C implementation to compute the full Hessian matrix. Note in Table 4.3 that increasing the number of sources does not increase the RAM required because each Hessian matrix is accumulated at the same memory location, but instead the computational cost increases linearly with the number of sources. The execution times of the full Hessian matrices for the serial ANSI-C implementation are not presented in Table 4.3, because their computational cost is over three orders of magnitude compared with the CUDA-C implementation (see Table 4.2) which is computationally infeasible.

Table 4.2
Execution time of the computation of one-column of the Hessian matrix in CPU and GPU.

Architecture	Language	Exec. time [s]
Intel(R) Xeon(R) E5-2609	ANSI-C	14.78
GPU K40c	CUDA- C	0.018

Table 4.3
Execution time and RAM size used to compute the full matrix $\mathbf{H}$ *(*7182×7182 *elements) for different sources using a GPU Tesla K40c.*

Source number	RAM (MB)	Exec. time [HH:MM:SS]
1	206.3	00:46:00
51	206.3	39:06:00
101	206.3	77:26:00

4.4 Covariance and Resolution Matrices

This section describes the link between the posterior covariance and the Hessian matrix using Bayesian inference and additionally presents how to compute the resolution matrix.

The posterior probability function of the model $\mathbf{v}$ given the observations d_{obs} can be expressed as

$$\rho(\mathbf{v}|d_{obs}) \propto \rho(\mathbf{v})\rho(d_{obs}|\mathbf{v}), \tag{4.39}$$

where $\rho(\mathbf{v})$ is the *a priori* density function of the model parameters and $\rho(d_{obs}|\mathbf{v})$ is the likelihood function that relates the observations with the given model parameters. If the likelihood and the *a priori* functions follow Gaussian multivariate distributions, and the relation between the observations and the model parameters is linear, then the posterior probability function $\rho(\mathbf{v}|d_{obs})$ is also Gaussian (Tarantola, 2005), and it is given by

$$\rho(\mathbf{v}|d_{obs}) \sim \mathcal{N}(\tilde{\mathbf{v}}, \Sigma_{post}), \tag{4.40}$$

where $\tilde{\mathbf{v}}$ is the maximum likelihood solution of the model parameters and Σ_{post} is the posterior covariance. The posterior covariance matrix can be obtained as a function of the Hessian matrix

of the cost function H, and the *a priori* covariance matrix $\sum_{prior}$, as follows

$$\sum_{post} = (H + \sum_{prior}^{-1})^{-1}. \tag{4.41}$$

Assuming that the model parameters are uncorrelated and identically distributed, the *a priori* covariance matrix is given by

$$\sum_{prior} = \sigma_{prior}^2 \cdot \mathbb{I}, \tag{4.42}$$

where $\mathbb{I}$ is the identity matrix, and the elements of the diagonal σ_{prior}^2 are the variances. Even though this is not a realistic model for the prior correlation of the model parameters, because usually parameters located at neighboring positions belong to the same class, the computation of the inverse $\sum_{prior}^{-1}$ is simple. Note that despite the assumption of uncorrelated model parameters in the prior density function of the model parameters, the posterior covariance matrix has elements off-diagonal that are nonzero. A posterior correlation different from zero for $i \neq j$ between the parameters $\tilde{v}_i$ and $\tilde{v}_j$, means that they were not independently resolved by the observed data during the full waveform inversion.

Once $\sum_{post}$ and $\sum_{prior}$ have been obtained, the resolution matrix $\mathscr{R}$ can be computed as (Tarantola, 2005, Equation 3.63)

$$\mathscr{R} = \mathbb{I} - \sum_{post} \sum_{prior}^{-1}. \tag{4.43}$$

If the resolution matrix is equal to the identity matrix then the obtained model is perfectly resolved and as Tarantola states "*The farther the resolution operator is from the identity, the worse the resolution is*"[1].

4.5 Uncertainty Quantification

In this section it is described how to use the *a posterior* covariance matrix, the *a priori* covariance matrix, the FWI results and the resolution matrix for uncertainty quantification. Besides,

[1] Tarantola (2005). *Inverse problem theory and methods for model parameter estimation*, page 72.

a variance reduction metric is formulated to measure the new information brought by the FWI.

The posterior covariance matrix can be obtained using Equation 4.41 for the estimated velocity model after a FWI. The Hessian matrix H is obtained via SOASM and the *a priori* covariance is obtained assuming uncorrelated pixels, as expressed in Equation (4.42). Note from line 1 of Algorithm 10 that SOASM needs a source to compute the Hessian matrix. This means that it is possible to compute a Hessian matrix for any source/frequency. However since the estimated velocity model comes from a FWI that uses a specific source then it is recommended to use the same source to compute the Hessian matrix. In the multi-scale case it is possible to compute a Hessian matrix for each frequency step.

For the estimated models using 1 and 51 sources, shown in Figure 3.7a and Figure 3.8a, the uncertainty of the model can be quantified using three different Hessian matrices, each one associated to the multi-scale frequency steps (3, 6 and 9 Hz). The posterior covariance matrices $\sum_{post}$ are obtained by setting the *a priori* standard deviation for all pixels of $\sigma_a =$ 500 m/s. Figures 4.10 and 4.11 depict the posterior standard deviation of each pixel in the model, for 1 and 51 sources, which correspond to the square root of the diagonal elements of the posterior covariance matrix $\sum_{post}$. Note in Figure 4.10 and Figure 4.11 that the posterior variance decreases (black zone) in comparison to the prior variance (white zone), in the area that is correctly illuminated by the sources. For one source, the illumination area is smaller than the one for 51 sources, and thus a small portion of the model is correctly updated during the inverse process. Additionally, note that for the smaller frequency (3 Hz), the uncertainty can only be coarsely quantified, whereas for the larger frequency (9 Hz), the uncertainty can be quantified near the pixel level.

In this case, the uncertainty of the prior velocity model is reduced due to the *new* information brought by the observations in the inversion process. The comparison between posterior uncertainties and prior uncertainties can also be quantified. The uncertainty reduction of the i^{th}

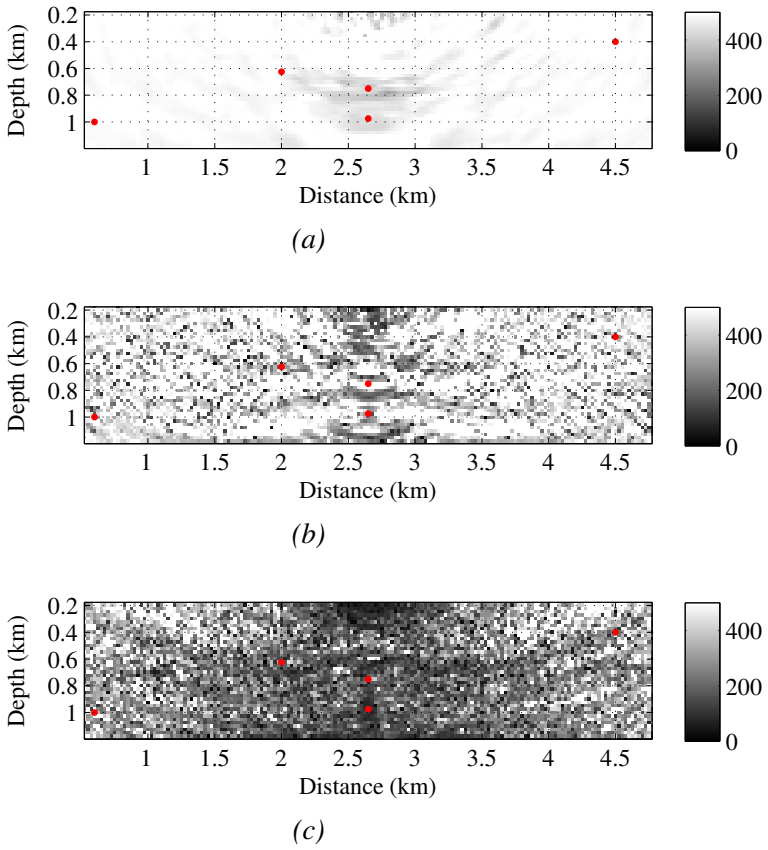


Figure 4.10 Posterior standard deviation of the velocity model estimated in Figure 3.7a using one source for (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz. Five test points are marked with red color.

pixel in the model, after the multi-scale FWI is measured as

$$VR_i = \frac{\sigma^2_{prior} - \sigma^2_{p_i}}{\sigma^2_{prior}}, \tag{4.44}$$

where $\sigma^2_{p_i}$ is the posterior variance of the i^{th} pixel in the model. A value of $VR_i = 0$ or negative mean that no-new information was brought from the observed data on the i^{th} pixel. A positive value of VR_i means that the inversion method added new information on the i^{th} pixel and thus the uncertainty of the posterior variance $\sigma^2_{p_i}$ is smaller than the a priori variance. Figure 4.12 and Figure 4.13 illustrate the variance reduction.

The uncertainty reduction given in Figure 4.12a shows that only a few parameters in the model have been *correctly* resolved. This occurs since the posterior covariance is obtained using a low frequency wavelet as source, which quantifies the uncertainty of a group of neighboring pixels instead of the uncertainty of one single pixel at the time. Figure 4.12b and Fi-

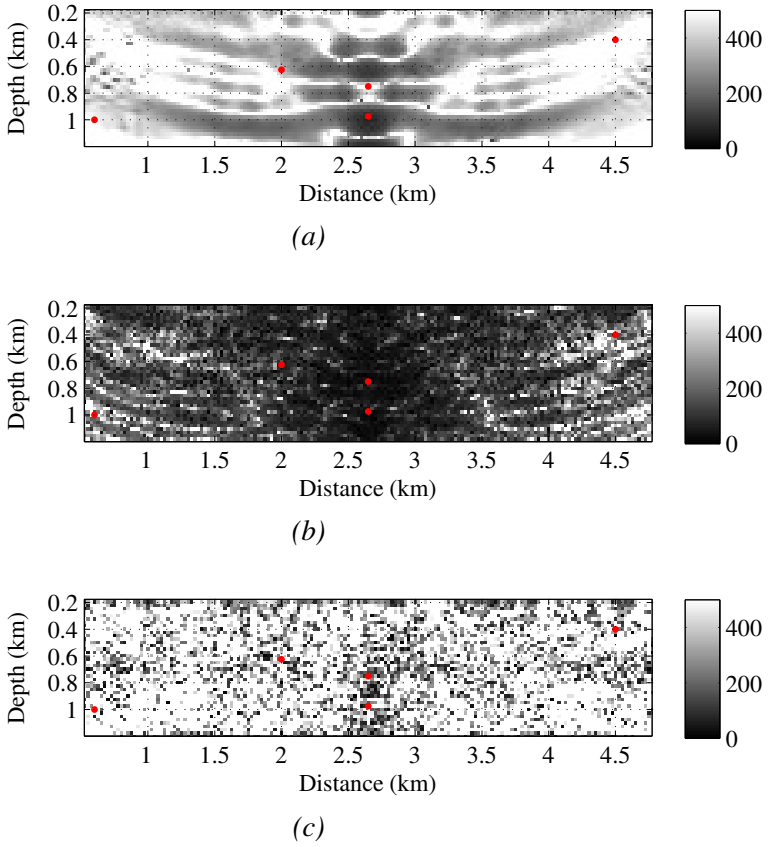


Figure 4.11 Posterior standard deviation of the velocity model estimated in Figure 3.8a using fifty-one sources for (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz. Five test points are marked with red color.

gure 4.12c show the uncertainty reduction using sources with higher frequencies. Particularly, Figure 4.12c shows the reduction in the uncertainty for every single parameter in the model, and a higher reduction is obtained (as expected) in the area underneath the source.

Five pixels inside the velocity model were specially selected to evaluate the uncertainty reduction. The test pixels are located at the following coordinates (in km): $P_1 = (2, 0.625)$, $P_2 = (4.5, 0.4)$, $P_3 = (0.6, 1)$, $P_4 = (2.65, 0.975)$. and $P_5 = (2.65, 0.75)$. The true velocity value for P_1, P_2, P_3 and P_4 is 2000 m/s (outside the diffracting square) and for P_5 is 2500 m/s (inside the diffracting square). The pixels are marked in red color in Figures 4.10, 4.11, 4.12, and 4.13. For all the points the VR values were computed together with a 95% confidence interval through $v_{fwi} \pm 3\sigma_p$. The 95% confidence interval was chosen as an illustrative example of an statistical analysis. The special points were chosen to test if 95% confidence interval always contains the true velocity and the results are presented on Table 4.4.

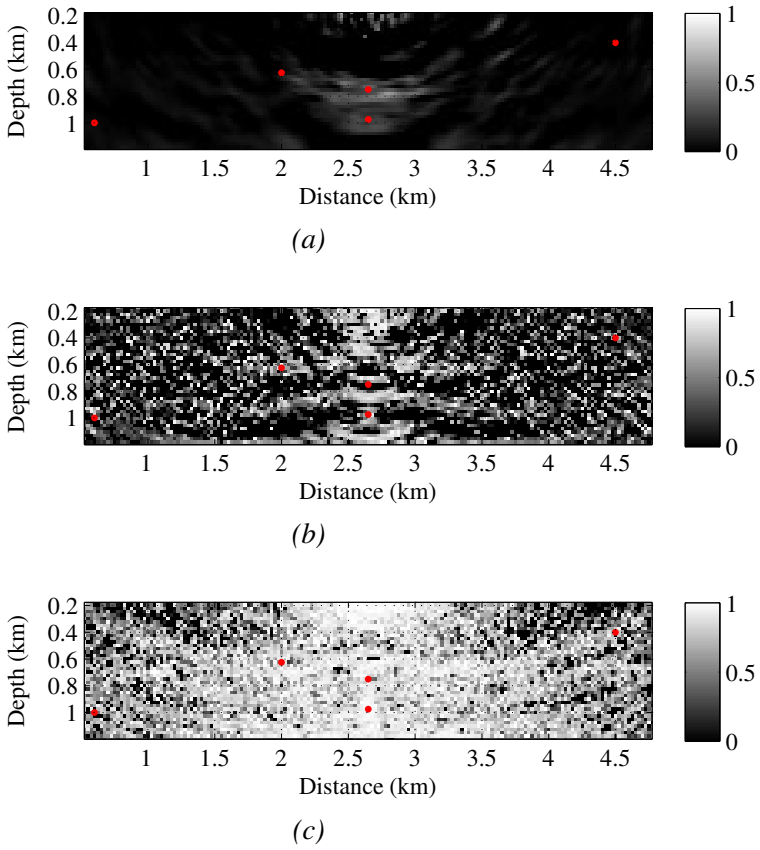


Figure 4.12 Variance reduction (VR) of the posterior standard deviation for the velocity model estimated of Figure 3.7a using one source for (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz. Five test points are marked with red color.

P_4 and P_5 fail for some tests and they are highlighted with boldface in Table 4.4. Figure 4.14 illustrates both a successful and failed case of uncertainty quantification where Figure 4.14a shows $\rho(\mathbf{v}|d_{obs})$ still contains the true velocity with a $VR = 98.2\%$ while in Figure 4.14b $\rho(\mathbf{v}|d_{obs})$ fails.

Finally, the resolution of the FWI results are computed using Equation 4.43. Figure 4.15 and Figure 4.16 illustrate the results, where the pixels with the best resolution are in green color with a value of 1 and the pixels with a resolution value different from 1 are in blue color spectrum if the resolution value is lower than 1 or in red color spectrum if the resolution value is greater than 1. If the number of sources or the central frequency of the source increase, more pixels are correctly resolved, but there are always areas where the resolution is different of one

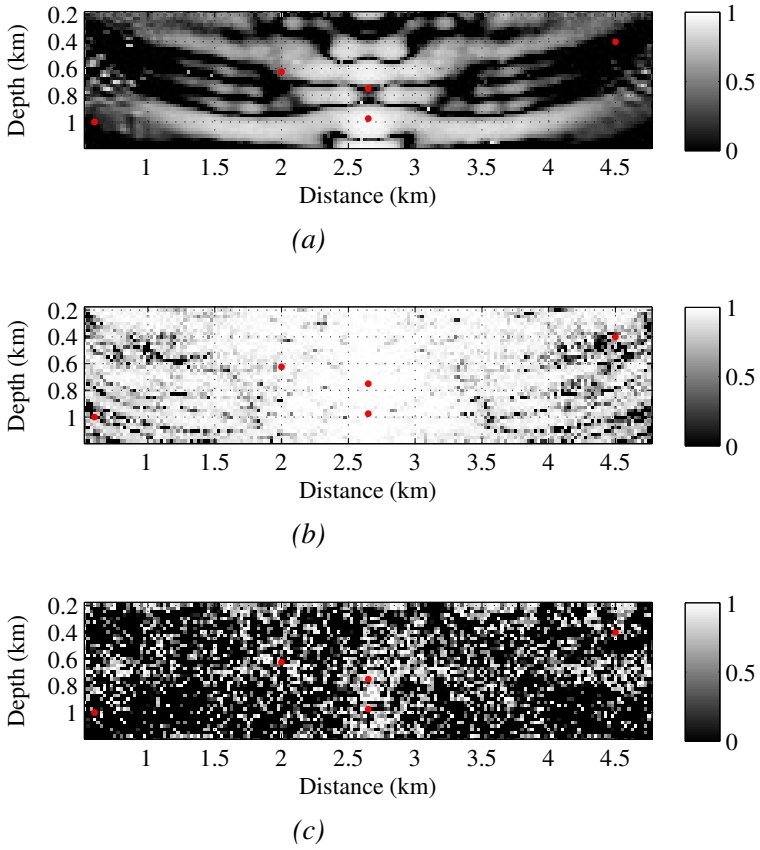


Figure 4.13 Variance reduction (VR) of the posterior standard deviation for the velocity model estimated of Figure 3.8a using fifty-one sources for (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz. Five test points are marked with red color.

due to the type of acquisition (superficial). Last column of Table 4.4 shows the resolution $\mathscr{R}$ on the test points.

From a physical point of view, if the number of sources increases then the illumination area also increases improving the quality of the observations which turn into a better reconstruction of the final velocity model. Besides, the increment in the central frequency of the source allows obtaining a more detailed reconstruction of the underground from the high-frequency information in the observed data. Both effects are illustrated in almost all the results from Figure 4.10 to 4.16.

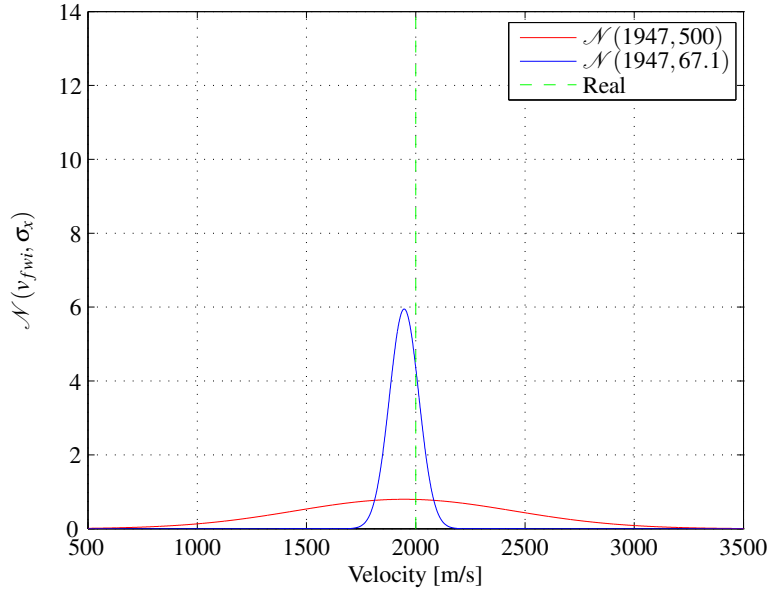
However, Figures 4.11c and 4.13c illustrate a behavior that does not have a physical meaning but can be produced by aspects as: numerical errors during the implementation, wrong selection of the a priori information, the acquisition geometry, among others. Therefore, a

Table 4.4
Special points to test the variance reduction and resolution.

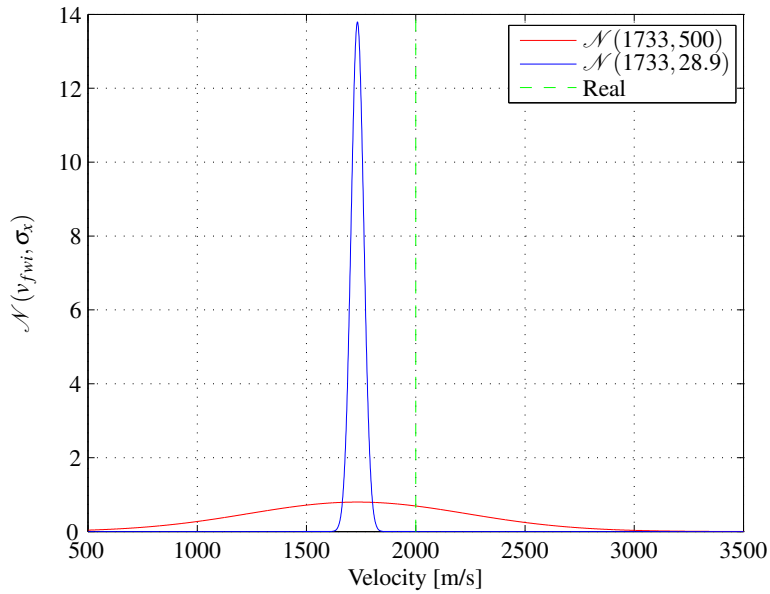
Sources(#)	Freq.(Hz)	Test point	$v_{fwi}(m/s)$	$\sigma_p(m/s)$	$\%VR_i$	$\mathscr{R}$
1	3	P_1	1996	484.3	6.16	-0.24
1	6	P_1	1996	351.5	50.59	1.172
1	9	P_1	1997	203.2	83.48	0.818
51	3	P_1	2009	364	47.02	1.103
51	6	P_1	2009	108.1	95.33	1.015
51	9	P_1	2010	1176	-453.4	1.002
1	3	P_2	1998	499.1	0.36	0.068
1	6	P_2	1998	450.3	18.9	1.075
1	9	P_2	1991	174.9	87.76	0.7511
51	3	P_2	2001	756.4	-128.8	6.457
51	6	P_2	2001	218.5	80.9	0.9813
51	9	P_2	2000	440.8	22.26	0.9875
1	3	P_3	2001	488.5	4.56	0.067
1	6	P_3	2001	638.4	-63.02	0.4593
1	9	P_3	2004	457.1	16.42	2.209
51	3	P_3	1999	472.3	10.78	-0.6634
51	6	P_3	1999	363.2	47.24	0.8937
51	9	P_3	1995	145.4	91.54	0.8892
1	3	P_4	1857	433.7	24.77	2.51
1	6	P_4	1947	67.08	98.2	1.054
1	9	P_4	1855	74.45	97.78	1.012
51	3	P_4	**1732**	**81.23**	97.36	0.984
51	6	P_4	**1733**	**28.92**	99.67	1.001
51	9	P_4	1818	445.3	20.7	1.001
1	3	P_5	2154	421.6	28.91	0.4022
1	6	P_5	2274	1018	-314.5	0.7609
1	9	P_5	2429	375.5	43.61	1.358
51	3	P_5	2248	671.2	-80.22	1.024
51	6	P_5	**2348**	**22.41**	99.8	1.002
51	9	P_5	2480	88.62	96.86	1.001

Note: Both coordinates are in kilometers $P_i = (S_x, S_z)$. $V_{ori} = 2000$ m/s for $P_1 = (2, 0.625)$, $P_2 = (4.5, 0.4)$, $P_3 = (0.6, 1)$ and $P_4 = (2.65, 0.975)$. $V_{ori} = 2500$ m/s for $P_5 = (2.65, 0.75)$.

depth study about the Hessian's sensitivity must be done before it can be correctly used for uncertainty quantification.



(a)



(b)

Figure 4.14 Uncertainty Quantification for the test point P_4 at 6 Hz using one source and fifty-one sources. The a priori uncertainty is built using a normal distribution where the mean is v_{fwi} and the standard deviation is 500 m/s (red color) and the a posteriori uncertainty is built only changing the standard deviation to σ_p (blue color). The one source result (a) represents a successful case while the fifty-one sources result (b) represents a failed case.

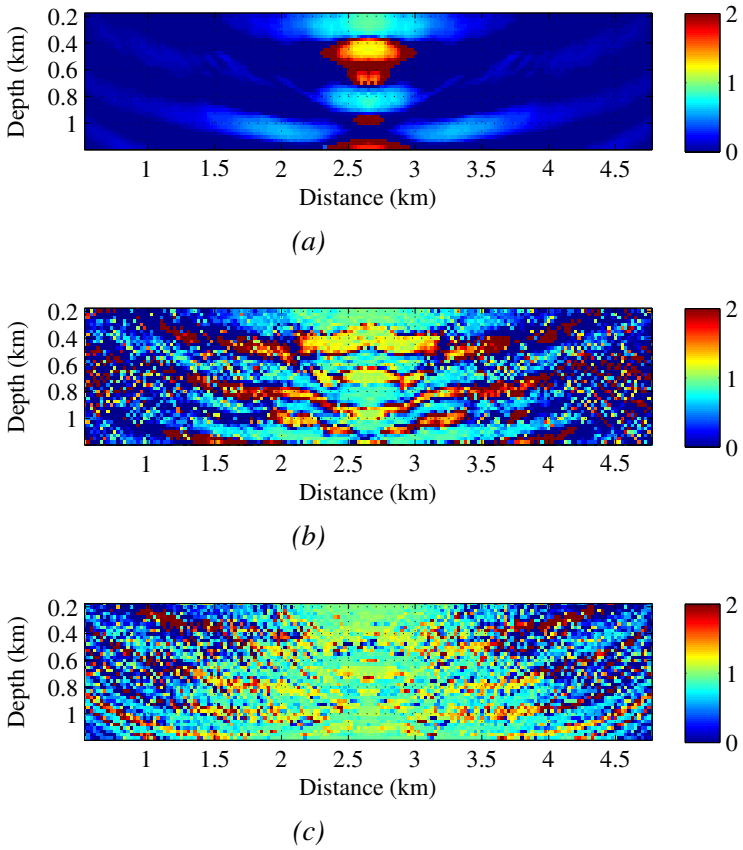


Figure 4.15 Resolution for the velocity model estimated in Figure 3.7a using one source at (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz.

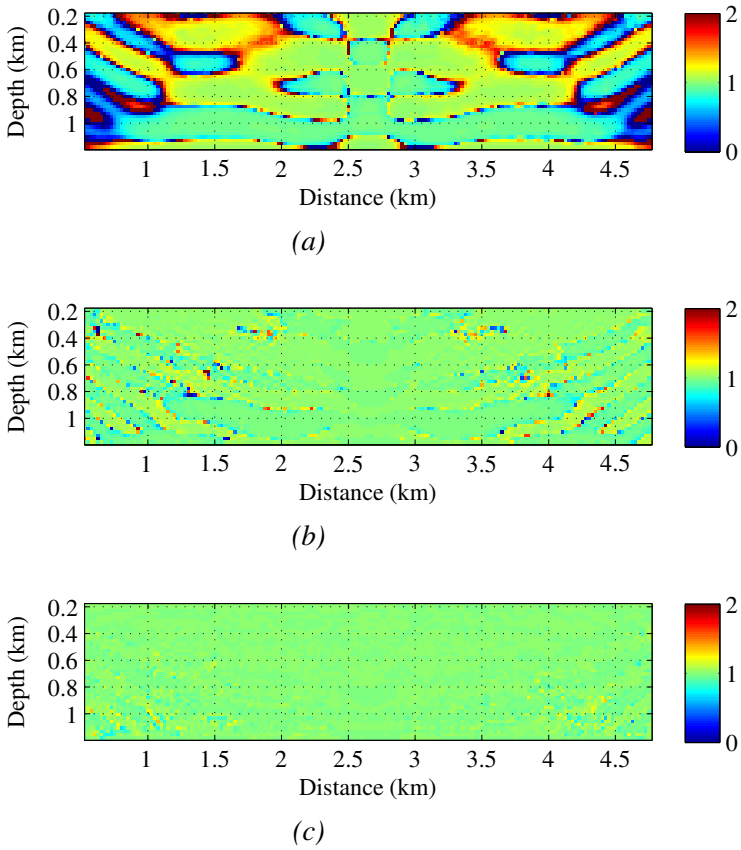


Figure 4.16 Resolution for the velocity model estimated in Figure 3.8a using fifty-one sources at (a) 3 Hz, (b) 6 Hz, and (c) 9 Hz.

4.6 Discussion

The important role of the Hessian matrix in resolution and uncertainty analysis requires to find methods to efficiently compute the exact matrix and its inverse. GPU architectures make feasible the implementation of the second-order adjoint method even for large scale problems.

This chapter presents a parallel implementation of the Hessian matrix-vector products using GPU architectures. This implementation uses the FDTD approximation to compute the wavefield propagations required by the SOASM method.

There is presented a practical method to compute the standard deviation over the final results (σ_p) from the a priori variance (σ_a^2) and $\mathbf{H}_{SOASM}$. Besides, a variance reduction (VR) and the resolution matrices ($\mathscr{R}$) are computed to define the areas with less uncertainty.

It is shown that VR growths and $\mathscr{R}$ improve if the source frequency and source number increase. However, choosing an intermediate value of sources and frequency is enough to build a trustworthy 95% confidence interval.

The 95% confidence interval fails for those points below the square diffracting area because they are not correctly illuminated. In fact, they represent the area with the biggest uncertainty even when the resolution for 6 Hz and 9 Hz is close to one.

According to Table 4.3, the main issue of obtain the Hessian matrices to compute σ_p, $\%VR$ and, $\mathscr{R}$ over real data is the computational cost. One shot over a 2D velocity model of 171×42 pixels spent 46 minutes which becomes a problem when the model size, the number of sources or the model dimensions increase. A possible solution for this issue would be using a cluster of GPUs where the computational load would be shared over all the nodes.

σ_p, $\%VR$ and, $\mathscr{R}$ represent new information that complements the FWI results and becomes useful inside a decision-making process.

5. Testing the Proposed Methodology

This chapter presents four synthetic velocity models used to test the proposed methodology. For all of them, the size, grid space and particular challenges are described. Each synthetic velocity model is discussed in a separate section, where all the technical characteristics and results are specified (i.e., PSO result, FWI result and the Hessian matrix), including final comments about the strengths and weaknesses of the proposed methodology over a specific area of each synthetic model. At the end, the discussion section addresses the topics about full-offset and short-far-offset strategies for the FWI working together with the proposed Quality Controls (QCs) and the use of the Hessian matrix for uncertainty quantification.

One of the geophysical aspects that must be handled carefully to avoid a local minimum convergence during the inversion process is the low frequency information. In this way, the selected bandwidth of the multi-scale approach must be guaranteed at each frequency step to be successful. To this end, an Ormsby filter[1] is applied to the modeled data, the observed data and the source wavelet for the following tests. The first inversion stage uses an Ormsby filter with parameters $f_1 = 0, f_2 = 1, f_3 = 3$ and $f_4 = 4$ to work in the frequency range $0 < f < 4$ Hz; the second inversion stage changes the Ormsby filter parameters $f_3 = 6$ and $f_4 = 7$ for a frequency range $0 < f < 7$ Hz; and in the last inversion stage the frequency range increases to $0 < f < 10$ Hz changing the Ormsby filter parameters $f_3 = 9$ and $f_4 = 10$. As the Ormsby filter produces a ringing effect on the traces, an outer muting is applied to each shot gather to minimize this effect and avoid possible cycle-skipping events.

[1] A filter of trapezoidal shape specified by four corner frequencies, f_1, f_2, f_3, f_4. The filter rejects below f_1 and above f_4, is linear from f_1 to f_2 and from f_3 to f_4, and flat from f_2 to f_3. http://wiki.seg.org/wiki/Dictionary:Ormsby_filter

The complex trace in the Hilbert domain and the Hessian matrix over the FWI results are used during the whole inversion as a supporting tool for quality control (QC) that allows to measure the cycle-skipping events on the traces and the resolution of the estimated models. The first two synthetic velocity models, Marmousi and Canadian Foothills, show the behavior of both QCs when they are applied over smooth velocity models but the next two synthetic velocity models, Hess and BP, show the behavior of both QCs when they are applied over synthetic models with areas of high velocity contrast.

5.1 Experiment Setup

Before testing the proposed methodology over the synthetic velocity models, it is necessary to define the model size that makes feasible the use of PSO to find the initial guess for the FWI. For all the experiments, the global, local and inertia weights of the PSO exploration are set to 1.1, 0.8 and 0.7, respectively (see Table 2.1); while the number of particles and particles step size are adjust to 500 and 1.

In the Marmousi case, as the area of interest has a size of 211×68 elements and five sources (Equation 3.11 is used to define the sources position, where $a = 21$, $b = 191$, $S_R = 25$ and $n = 5$) are used together with 171 receivers then two PSO experiments running 100 iterations per particle spent four days per GPU. Those receivers record 4.5 s with a time step of 4×10^{-3} s. As the CPS cluster has six GPUs, it is possible to run 24 experiments per week and 70 experiments in three weeks, which is the maximum waiting time defined for the PSO exploration.

A second important aspect is to keep controlled the numerical dispersion during the modeling stage. For all the cases, it is desired that the value $k \times \Delta h$ be lower than one during the whole inversion process (see Figure 3.2). This numerical dispersion constraint ($k\Delta h < 1$) is preserved when the Marmousi velocity model is re-sampled from its original grid space to a squared grid space of 25 m $\times$ 25 m producing a numerical dispersion range of $0.25 < k\Delta h < 0.96$. Figure 5.2c illustrates the selected area to test the proposed methodology that meets both the numerical dispersion and the model size requirements. This section ranges from the position

2.325 km to 6.625 km with a depth of 1.2 km.

On the other hand, the Canadian Foothills velocity model has a distance of 25 km and a depth of 10 km on a grid of 15 m $\times$ 10 m which produces a matrix of 1000 elements in depth and 1668 elements in the horizontal direction. Due to the computational cost of the PSO exploration, the original grid is re-sampled to a squared grid of 50 m taking only the area between the kilometer 5 and 20 in the horizontal direction with a depth of 8 km. Figure 5.11c illustrates the area of the Canadian foothills velocity model with reduced dimensions (301 elements in the horizontal direction and 161 elements in depth) used to test the proposed methodology.

For the Canadian, Hess and BP PSO explorations, each particle performs 50 iterations. In these cases, PSO uses 10 sources (Equation 3.11 is used to define the sources position where $a = 21$, $b = 281$, $S_R = 50$ and $n = 10$) and 261 receivers with a time step of 4×10^{-3} seconds and a recording time of 8 seconds to produce the modeled data. For all the cases, the model resolutions are changed to squared grids of 50 m to reduce the models density and keep feasible the PSO exploration. Each PSO exploration uses 440 MiB (3.7% of GPU RAM) with a central frequency of 3 Hz and numerical dispersion factors of 0.24, 0.61 and 0.64 for the Canadian, Hess and BP, respectively. As one PSO exploration with the Hilbert metric spent 13 days per GPU then the number of PSO experiments is reduced to 10 to obtain the initial guess in a period no longer than 3 weeks.

In the case of the Hess velocity model, the first 6.8 km in depth and 13.7 km in the x direction are selected to perform the PSO exploration (see Figure 5.17). But for the BP velocity model, the PSO exploration area is selected from the kilometer 30 to the kilometer 45 in the x direction using the first 8.2 km of depth. Figure 5.25 illustrates the BP zone of interest used to test the proposed methodology with 160×301 elements where the high velocity intrusion represents the 21% of the selected area.

All the PSO tests are computed with the observed data that comes from the original models in the selected areas described above using the wavelet defined in Equation 2.11 with $f = 3$ Hz and $t_0 = 0.5$ s. Besides, the complex trace in the Hilbert domain is the selected metric to measure the cycle-skipping (Algorithm 1.3) for all the experiments.

Finally, the initial models for the FWI (see Figure 5.2a, Figure 5.11a, Figure 5.18a and Figure 5.26a) are obtained from the average in the experiment direction of the best PSO results. Then, the FWI tests are computed using the wavelet defined in Equation 3.12 with $f = 3$ Hz, 6 Hz and 9 Hz for the multi-scale[2] approach, and $t_0 = 0.5$ seconds. During the inversion process, the number of sources increases to 21 (Equation 3.11 with $n = 21$ for the source position) to improve the lighting of the area of interest. The gradient is computed using the strategy II (see subsection 3.3.3) with a constant step for the first FWI iteration and the use of L-BFGS for the next iterations.

5.2 Marmousi Model

The Marmousi velocity model was created in 1988 by the Institut Francais du Pétrole (IFP) to produce complex seismic data that require advanced processing techniques to obtain a correct earth image. The model was designed based on a profile through the North Quenguela Trough in the Cuanza Basin in Angola and it contains many reflectors, steep dips and strong velocity gradients in both vertical and lateral directions (Versteeg, 1994). This dataset was used in the workshop on practical aspects of seismic data inversion at the 52^{nd} EAGE meeting in Copenhagen 1990 with a distance of 8.9 km and a depth of 3 km on a grid of 30 m $\times$ 10 m (see Figure 5.1).

For this experiment, the FWI process spent 15 minutes and 164 MiB (1.4%) of the GPU RAM. Figure 5.2b illustrates the FWI result after the multi-scale approach, and the experiment

[2]In the first step, the FWI uses the PSO result where the modeled data and the observed data are processed with a lowpass Ormsby filter to eliminate the frequencies greater than 3 Hz. Then a number of iterations are run to resolve the details at 3 Hz on the initial guess. The output of the 3 Hz stage is used as an input for the inversion at 6 Hz with the observed and modeled data filtered at 6 Hz. The last step uses the same methodology with both datasets filtered at 9 Hz.

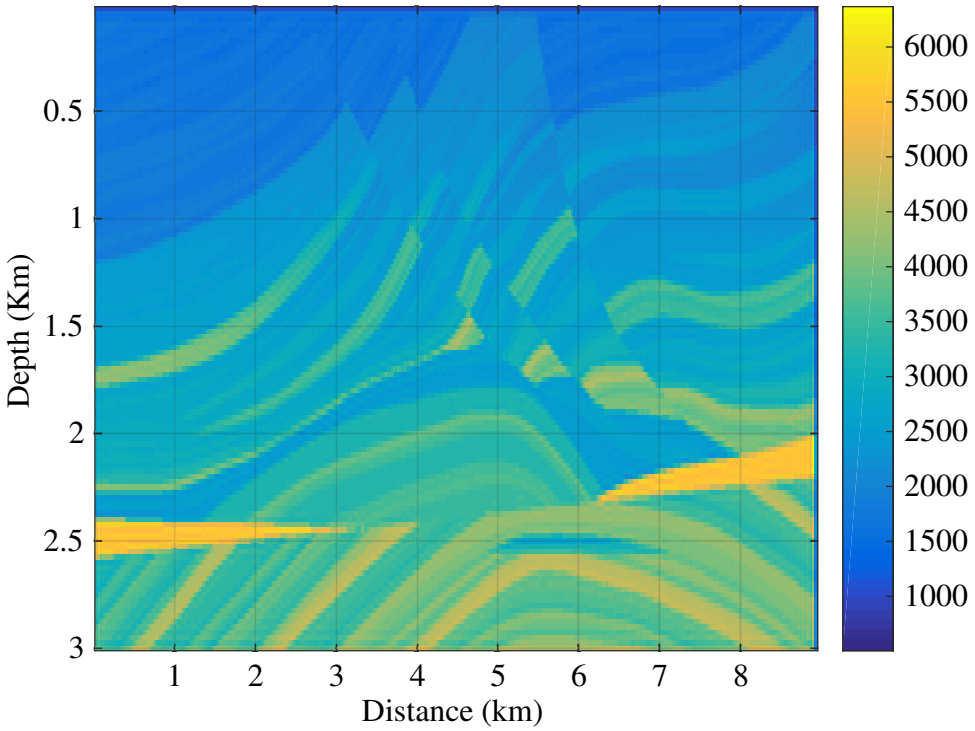


Figure 5.1 Marmousi velocity model [m/s].

parameters are summarized in Table 5.1. If the original section of the Marmousi velocity model (Figure 5.2c) is subtracted from the FWI result (Figure 5.2b), then the area with the best reconstruction is clearly identified (see the values close to zero in Figure 5.3). In this case, it is observed that a better and deeper reconstruction is reached in the middle of the model and the worst areas are the left, right and bottom sides of the model. The left and right sides are affected by the lack of illumination during the acquisition process, and the bottom area is affected by the far-offset that depends on the acquisition geometry.

In real cases, it is not possible to perform this comparison directly, but it can be performed indirectly through the shot gather analysis. If the complex traces of both the modeled data and the observed data are computed using Equation 2.8 then their instantaneous phases can be extracted with Equation 2.9. From the instantaneous phase differences between the observed data ${\theta_{obs}}^{s,r}(t)$ and the modeled data ${\theta_{mod}}^{s,r}(t)$ it is possible to identify the model closest to the solution, from a set of models, counting the number of phases less than $\pi/2$.

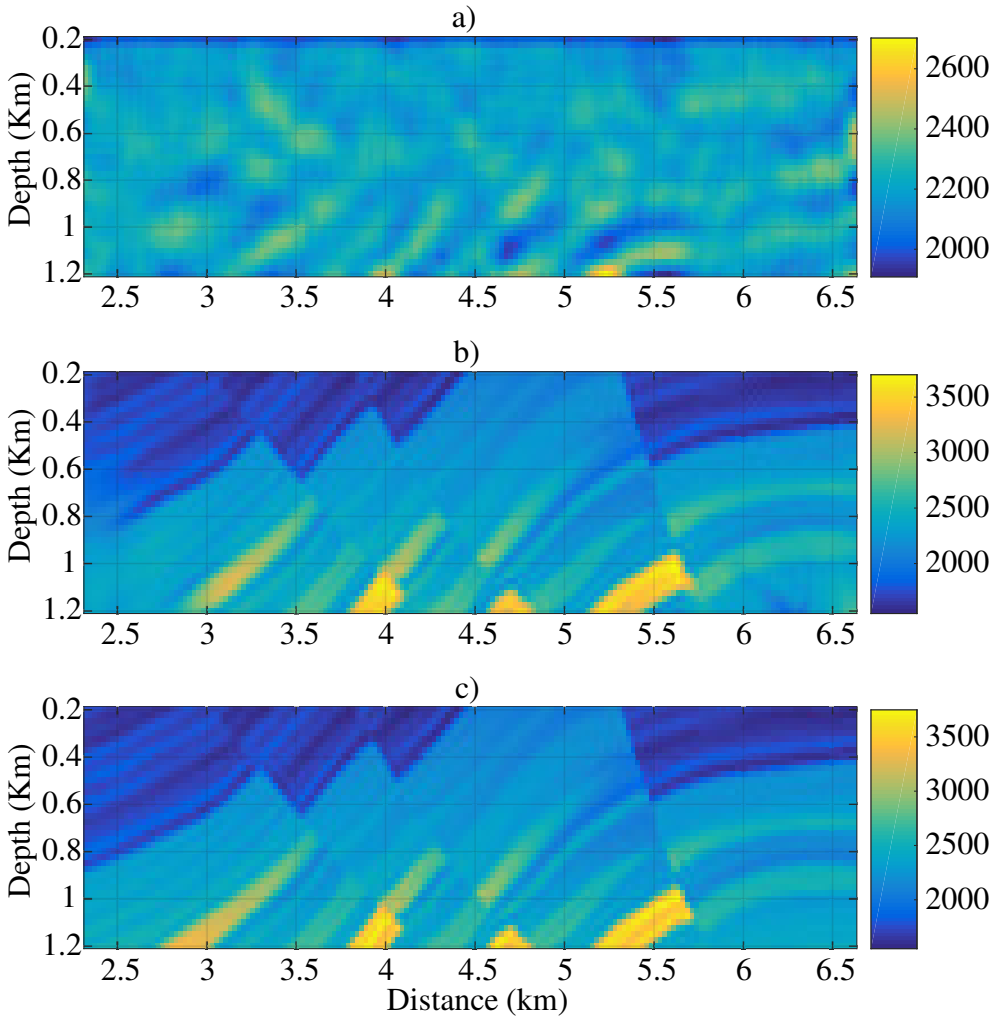


Figure 5.2 Marmousi velocity model section selected to test the proposed methodology. (a) Initial velocity model from the PSO exploration, (b) FWI result after a multi-scale approach and (c) Original section. All of them in [m/s].

The first row of Figure 5.4 illustrates the instantaneous phase difference between the observed data ${\theta_{obs}}^{s,r}(t)$ and the modeled data $\theta^{s,r}_{mod-pso}(t)$ produced from the PSO result; while the second, third and fourth row of Figure 5.4 illustrate the instantaneous phase difference between the observed data ${\theta_{obs}}^{s,r}(t)$ and the modeled data $\theta^{s,r}_{mod-fwi}(t)$ produced from the FWI results at 3 Hz, 6 Hz and 9 Hz, respectively.

The instantaneous phase difference of the shot gathers shows in blue color the areas without cycle-skipping (CS) and in yellow color the areas with a phase difference greater than $\pi/2$ or with CS problems. From the PSO results (first row of Figure 5.4), it is seen that the instanta-

Table 5.1
Parameters and values for the Marmousi experiment.

Parameter	Description	PSO	FWI	Hessian
Δh	Spatial step	25 m	25 m	25 m
D	Model size	211×68	211×68	211×68
Δt	Time step	4×10^{-3} s	4×10^{-3} s	4×10^{-3} s
T_{end}	Acquisition time	4.5 s	4.5 s	4.5 s
S_i	Number of sources	5	21	21
Rec_i	Number of receivers	171	171	171
V_{max}	Maximum velocity	2702 m/s.	3717 m/s	3717 m/s
V_{min}	Minimum velocity	1908 m/s.	1466 m/s	1466 m/s
$k\Delta h$	Numerical dispersion	0.25	0.96	0.96
P	Number of particles	500		
K	Number of iterations	100		
E	Number of experiments	70		
w	Inertia weight	0.7		
c_1	Local search weight	0.8		
c_2	Global search weight	1.1		
dt	Particles step size	1		
$F\{\mathbf{v}_{i,d}^k\}$	Metric domain	Hilbert		
α =ctte	Number of iterations		1	
L-BFGS	Number of iterations		39	
$\mathbf{g}(\mathbf{v}^k)$	Gradient strategy		**II**	
f_{steps}	Frequency steps	3 Hz	3,6,9 Hz	3,6,9 Hz
H_{area}	Hessian matrix area			121 points
H_{ite}	Hessian matrix iterations			360
RAM	Computational cost	131 MiB	164 MiB	627 MiB
Time	Computational cost	4 Days per GPU Experi.	45 minutes	42 Hours

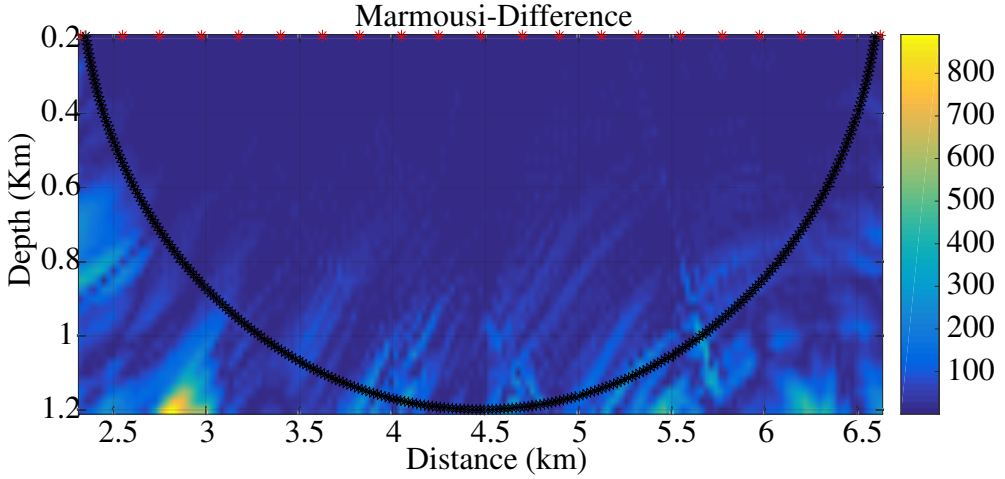


Figure 5.3 Difference in m/s between the original Marmousi velocity model and the FWI reconstruction. The sources used in the experiment are market in red color and the illuminated area is enclosed by a black ellipse.

neous phases with CS problems are between 6.9% and 8.9%. This means that even though the low frequency information is correctly estimated by PSO, the velocity model is still far from the real model. However, if the same shots are analyzed on the FWI results (see second, third and fourth rows of Figure 5.4) the areas with CS problems are strongly reduced at each frequency step, which can be interpreted as the FWI is getting a better estimation of the velocity model.

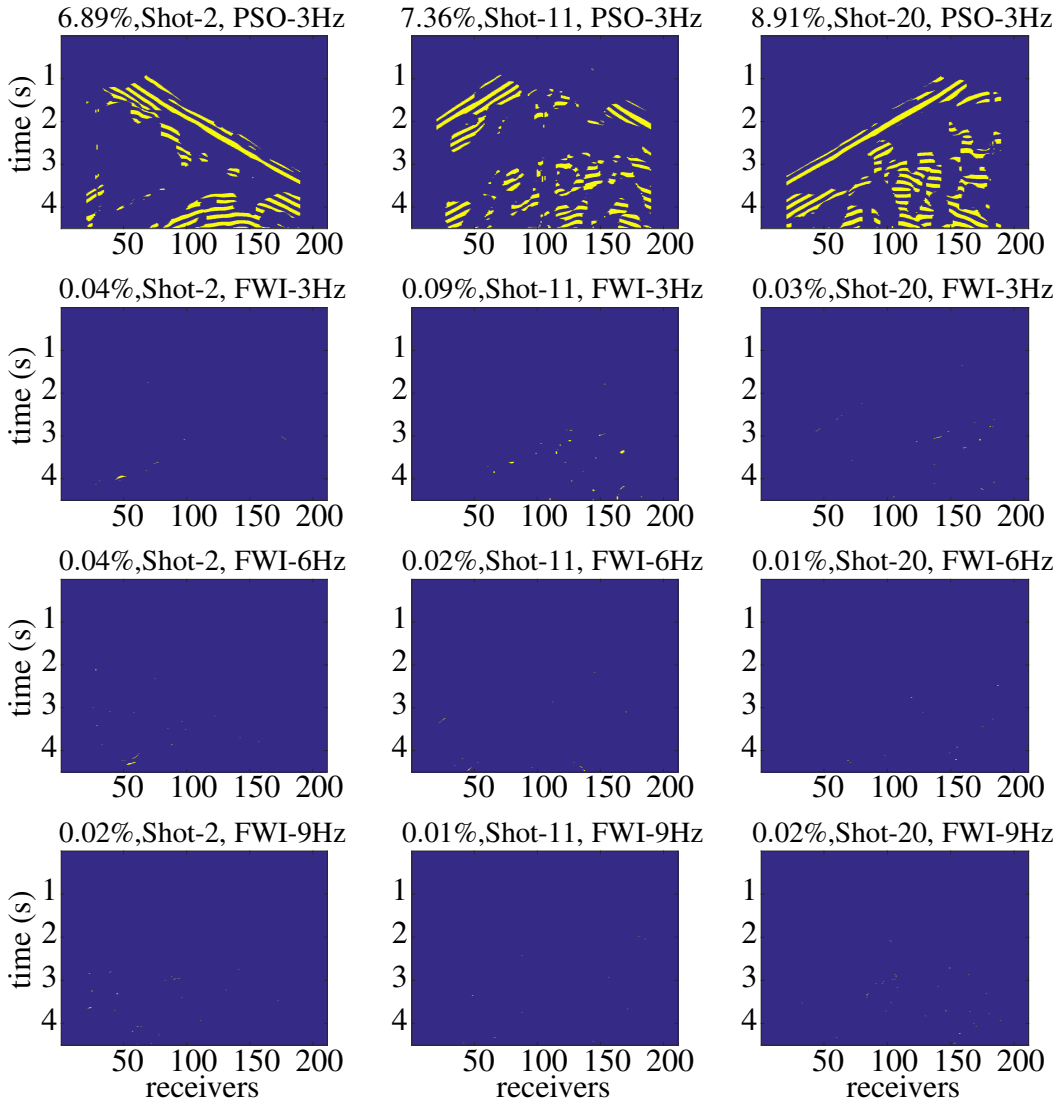


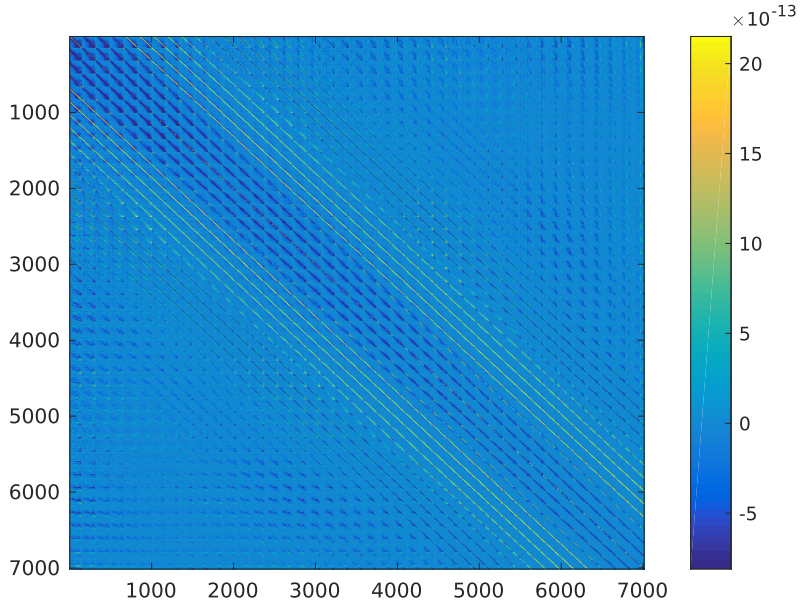
Figure 5.4 Instantaneous phase difference between the observed data and the modeled data over the PSO and full-offset FWI result for the Marmousi velocity model at 3 Hz, 6 Hz and 9 Hz using the shots 2, 11, and 20. The phase difference is greater than $\pi/2$ at the yellow color areas (areas with cycle-skipping problems) and the CS percentage is presented in the upper left corner of each result.

Hessian matrices are computed over the whole models of the PSO and the FWI results to measure the resolution reached after the multi-scale process, using the same 21 sources, 171 receivers and the wavelet of the FWI; but with central frequencies of 3 Hz, 6 Hz and 9 Hz. Figure 5.5a and Figure 5.5b illustrate the full Hessian matrices of the PSO and the FWI result, respectively, where the whole process spent 4.7 hours using 538 MiB (4.5% of GPU RAM).

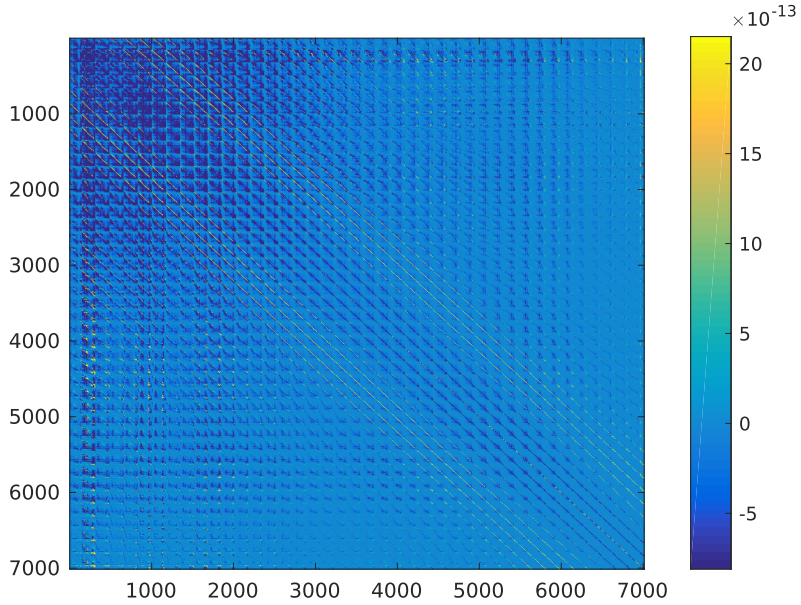
The Hessian matrices of the FWI results (Figure 5.5b is an example for 3 Hz) have most of their energy focused in the upper-left corner while the PSO result (Figure 5.5a) has its energy more spread around its diagonal. This energy distribution produces a visual effect of symmetry over the PSO result.

The eigenvalues of a matrix are useful to identify, classify and compare set of matrices through characteristics as the condition number, the trace and the rank. In the theoretical case, if the velocity model has been correctly resolved then the rank of the Hessian matrix evaluated at that point is equal to the number of model dimensions (i.e., $N_x \times N_z$), the condition number equals 1 because all the eigenvectors have the same energy and the normalized trace is equal to the number of eigenvalues $N_x \times N_z$.

In this case, the theoretical behavior of the Hessian matrix near the solution is used as a reference to define whether the FWI results are better than the PSO result. Figure 5.6 illustrates the normalized eigenvalue magnitudes of the Hessian matrices evaluated at the PSO and FWI results for all the 7011 dimensions. The Hessian matrices are also computed over an area that emulates the illumination produced by the acquisition geometry (yellow area of Figure 5.7a) and over a small squared area in the middle of the model (yellow area of Figure 5.7b). In the first case, the number of dimensions is reduced to 3217, producing the normalized eigenvalue magnitudes illustrated in Figure 5.8a, and, in the second case, the number of dimensions is reduced to 121 with the normalized eigenvalue magnitudes depicted in Figure 5.8b. For all the cases, Table 5.2 summarizes their rank, symmetry, trace and condition numbers.



(a)



(b)

Figure 5.5 Full Hessian matrix evaluated over (a) PSO result at 3 Hz and (b) FWI results at 3 Hz.

In this experiment, when the number of dimensions is 7011 (the whole model), the Hessian matrix computed over the velocity model reached by PSO has the major eigenvalue magnitudes, which produces the greatest trace with the condition number closest to one; indicating that PSO is the best result. When the number of dimensions is reduced to 3217 (avoiding the areas with

poor illumination, blue areas of Figure 5.7a), PSO result keeps the best trace and the major eigenvalue magnitudes, but it is not the best result because the condition number of the FWI results are closer to one. In the last case, when the number of dimensions is reduced to 121 (keeping a good illuminated area, see yellow area in Figure 5.7b); the best results in terms of trace, condition number and eigenvalue magnitudes are reached by the FWI over PSO.

In a theoretical Hessian matrix, a condition number close to one implies that all the eigenvalue magnitudes have the same value as a consequence of a perfect velocity estimation. However from the PSO and FWI results (see Figure 5.6 and Figure 5.8) it is seen that the normalized eigenvalue magnitudes decrease exponentially indicating lack of resolution on the estimated models. As expected, the resolution on the velocity estimation improves after each FWI iteration and it is also expected that the area under the curve of the normalized eigenvalue magnitudes increase after each FWI step. However this behavior it is only seen on the Hessian matrices (Figure 5.8b) computed over the squared area of 121 dimensions (Figure 5.7b). In the cases of Hessian matrices for 7011 (Figure 5.6) and 3217 (Figure 5.8a) dimensions the normalized eigenvalue magnitudes have an erratic behavior as a consequence of data coming from the areas that are not correctly illuminated by the acquisition geometry.

As the Hessian matrices of the squared area of 121 dimensions (Figure 5.8b) are only computed over the final PSO and FWI results, it is necessary to compute the Hessian matrix over each FWI result to verify whether the behavior keeps according to the theoretical case. For this experiment, the FWI computes 40 velocity models per frequency step, which means that it is necessary to compute 120 Hessian matrices to keep the track of the 121 dimensions' squared area.

The trace of a matrix allows to compute the area under the curve of its eigenvalues. In this way, the trace can be used to measure the evolution of the 120 Hessian matrices during the inversion process. Additionally, as the Hessian matrix computation uses a source to obtain the

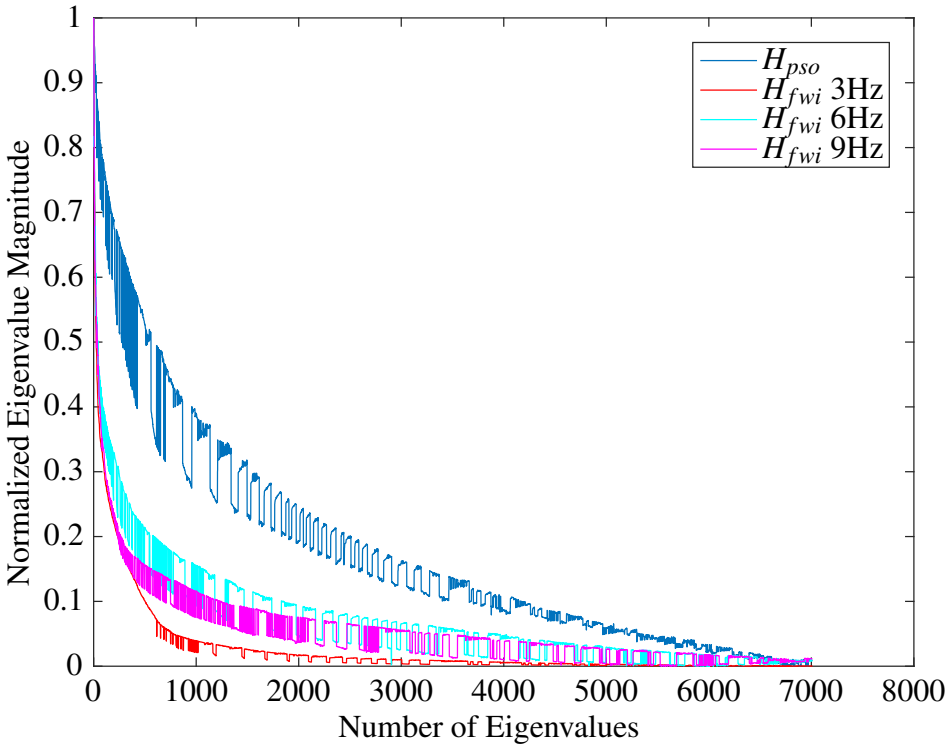


Figure 5.6 Normalized eigenvalue magnitudes of the Hessian matrix evaluated over the PSO and FWI results for 7011 dimensions.

Table 5.2
Characteristics of the Hessian matrices computed over the PSO and FWI results using the whole Marmousi velocity model and the yellow areas defined in Figure 5.7.

Matrices	Size	Shape	Rank	Symmetric	Eigenvalues	Condition #	Trace
Theoretical	$(N_x \times N_z)^2$	Square	Full	Yes	$\mathbb{R}^+$	1	$N_x \times N_z$
$H_{pso(3\ Hz)}$	$(7011)^2$	Square	Full	No	$\mathbb{R}$	1.52×10^5	1240.3
$H_{fwi(3\ Hz)}$	$(7011)^2$	Square	Full	No	$\mathbb{R}$	2.64×10^6	192.1
$H_{fwi(6\ Hz)}$	$(7011)^2$	Square	Full	No	$\mathbb{R}$	2.42×10^5	487.96
$H_{fwi(9\ Hz)}$	$(7011)^2$	Square	Full	No	$\mathbb{R}$	1.78×10^5	390.83
$H_{pso(3\ Hz)}$	$(3217)^2$	Square	Full	No	$\mathbb{R}$	2.60×10^5	702.65
$H_{fwi(3\ Hz)}$	$(3217)^2$	Square	Full	No	$\mathbb{R}$	1.83×10^5	120.59
$H_{fwi(6\ Hz)}$	$(3217)^2$	Square	Full	No	$\mathbb{R}$	2.86×10^4	398.48
$H_{fwi(9\ Hz)}$	$(3217)^2$	Square	Full	No	$\mathbb{R}$	3.66×10^4	498.10
$H_{pso(3\ Hz)}$	$(121)^2$	Square	Full	No	$\mathbb{R}^+$	509.19	32.5
$H_{fwi(3\ Hz)}$	$(121)^2$	Square	Full	No	$\mathbb{R}^+$	6.7687	52.44
$H_{fwi(6\ Hz)}$	$(121)^2$	Square	Full	No	$\mathbb{R}^+$	7.5225	72.51
$H_{fwi(9\ Hz)}$	$(121)^2$	Square	Full	No	$\mathbb{R}^+$	6.3524	81.88

kernels, it is possible to observe the Hessian matrix evolution at different frequencies.

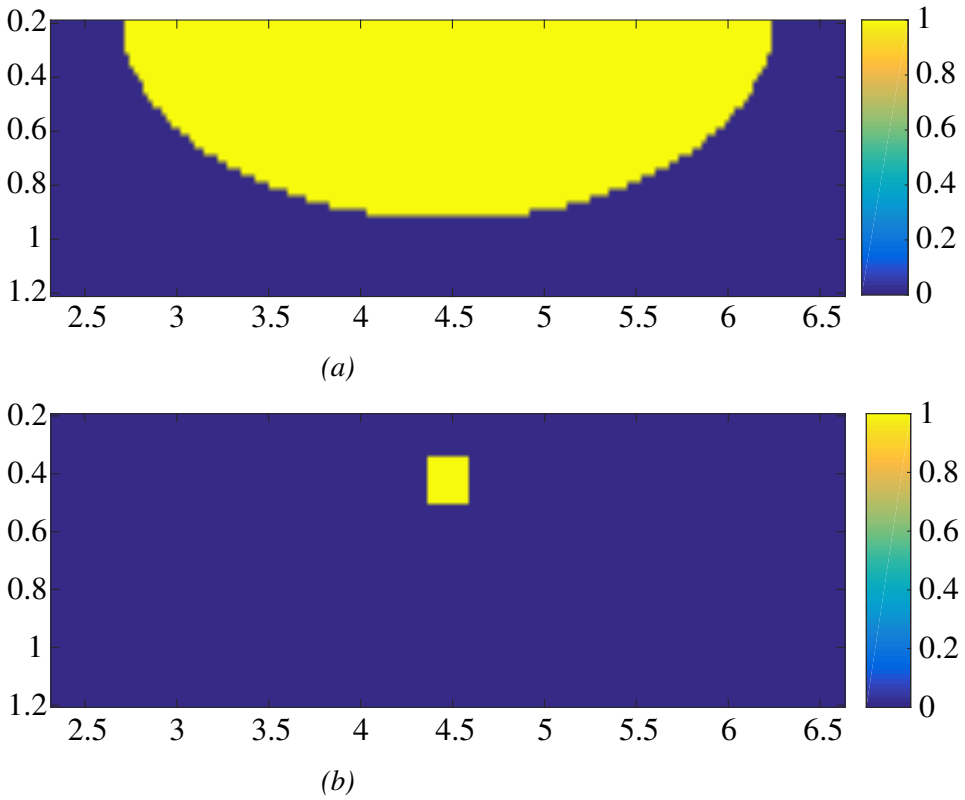


Figure 5.7 Illumination areas where the Hessian matrix is evaluated for the PSO and the FWI results. (a) The Hessian matrix is evaluated over 3271 pixels that emulates the illumination area produced by the seismic acquisition. (b) The Hessian matrix is evaluated over 121 pixels that emulates a square of 11 pixels centered at the position x= 4.475 km with a depth of 0.423 km.

For this experiment, 120 Hessian matrices are computed for 3 Hz, 6 Hz and 9 Hz as illustrated in Figure 5.9. Figure 5.9b, Figure 5.9c, and Figure 5.9d show the eigenvalues behavior on each step of the multi scale FWI.

At 3 Hz the Hessian matrix evolves as expected theoretically, because the normalized trace value has a strong increase at each FWI frequency step (blue line of Figure 5.9a). At 6 Hz the normalized trace value increases with a low rate in the first 80 iterations and has a big increase only at the last FWI frequency step (red line of Figure 5.9a). At 9 Hz, the normalized trace value never has a big increases keeping a low increment rate for the 120 iterations.

If the source frequency used to compute the Hessian matrix varies, then more or less physical features are detected by the Hessian kernels producing a change on the final characteristics

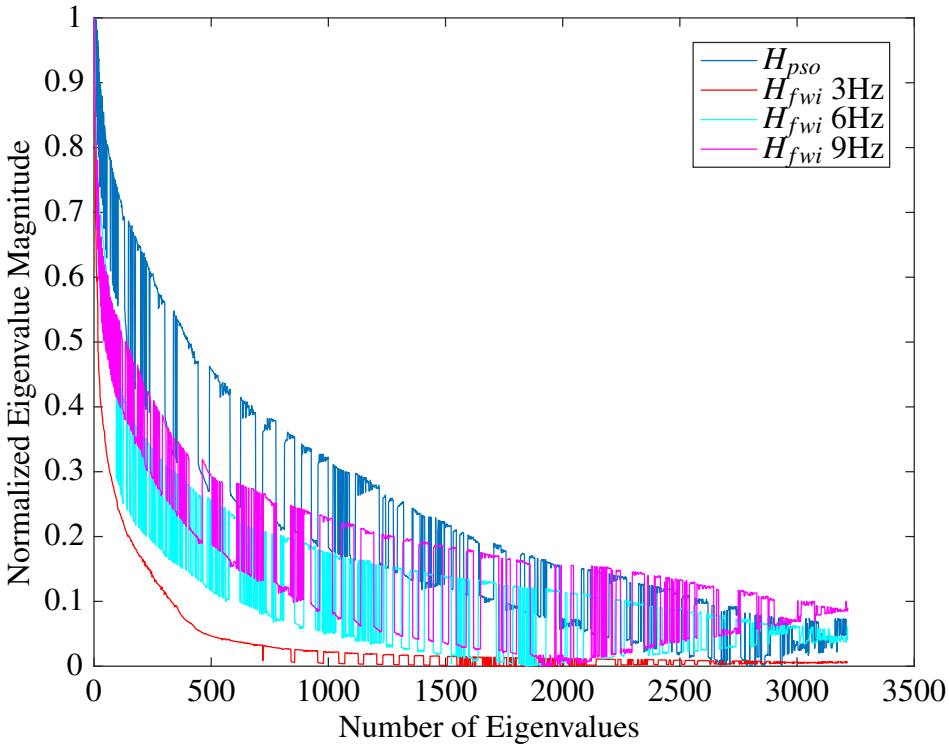
of the Hessian matrix. This effect explains the three different behaviors described on Figure 5.9 and it can be used to indirectly measure the velocity model resolution of a specific area or it must be taken into account for uncertainty quantification purposes.

5.2.1 Final Comments

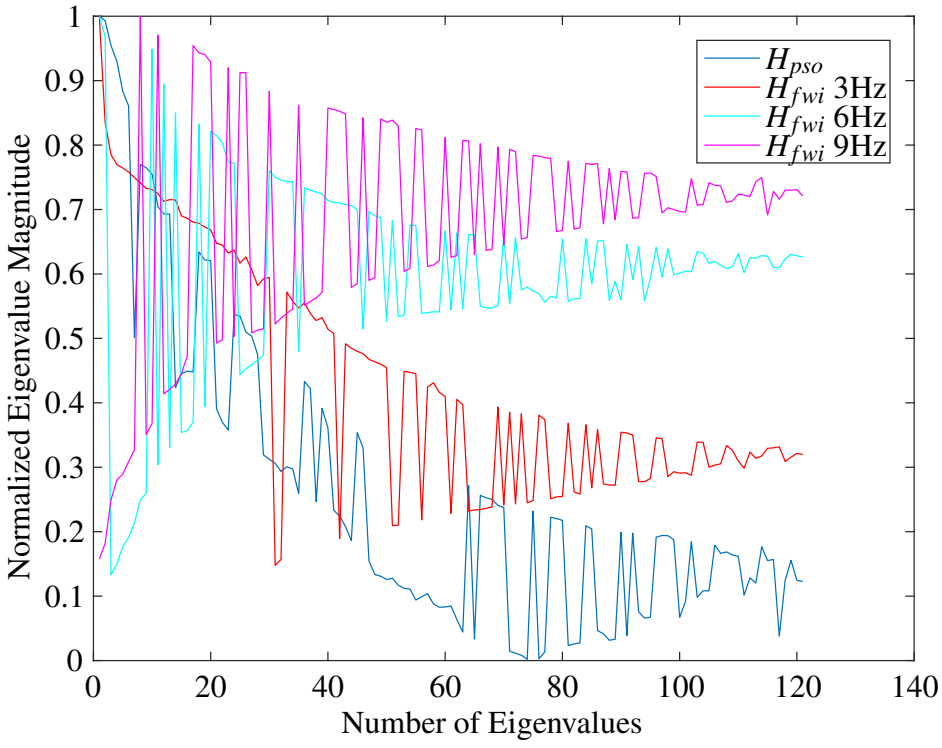
In this experiment, PSO has proved to be successful in the cost function exploration because it reached a final velocity model with enough low frequency information to fulfill the FWI requirements. In other words, PSO found a point in the global minimum neighborhood that allows the FWI reaches the global minimum of the cost function. But the main drawback of this exploration is that it spent 92.2% of the time of the whole process.

The FWI successfully reaches a velocity estimation that is closer to the real velocity model because: the initial guess is in the global minimum neighborhood; the frequency information has been controlled with the Ormsby filter during the multi-scale inversion; and the Marmousi velocity model has a smooth velocity variation. The whole FWI process just spent 45 minutes because it exploits the capabilities of the GPU technology as it is described in Abreo et al. (2015). However, the main drawback of using GPUs for FWI is associated with their low RAM capacity. For this experiment, a GPU Tesla k40c with 12 GiB of RAM is used for the Marmousi inversion. In this case, the RAM size is enough because the inversion only spent 164 MiB (1.4% of RAM). But if the model size increases then other strategies must be used to save RAM and avoid to leave the GPUs without opportunity (Noriega, Ramirez, Abreo & Arce, 2017).

The instantaneous phase difference using the Hilbert transform has proved to be a successful quality control (QC) during the inversion process because it is possible to identify whether the new velocity estimation is better than the previous one from the CS reduction on the traces. Its main advantages are the phase extraction in time and the trace amplitude independence.



(a)



(b)

Figure 5.8 Normalized eigenvalue magnitudes of the Hessian matrix evaluated over the PSO and FWI results for (a) the illumination area of Figure 5.7a and (b) the squared area illustrated in Figure 5.7b.

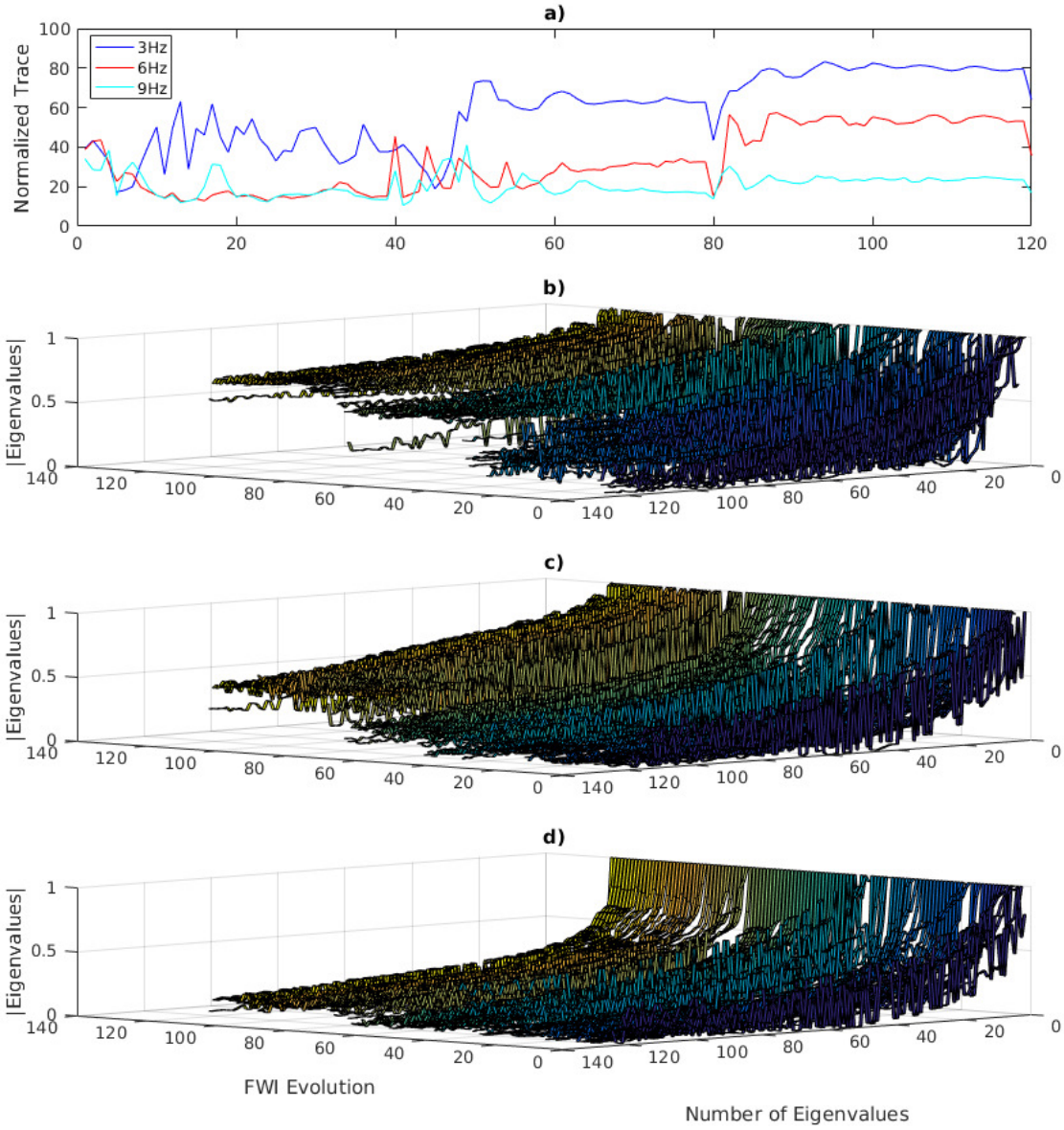


Figure 5.9 Marmousi's Hessian matrix evolution over 120 FWI's iterations for the squared area of Figure 5.7b. a) Normalized trace for 3 Hz, 6 Hz, and 9 Hz; Eigenvalues magnitude evolution for b) 3 Hz, c) 6 Hz and d) 9 Hz.

In a seismic case, the Hessian matrix characteristics are affected by: the illumination area produced by sources and receivers; the source frequency used for the Hessian kernels computation; and the resolution of the estimated velocity model. In this way, the Hessian matrix must be computed only over areas correctly illuminated and using a source wavelet that allows to identify the physical characteristics of interest.

5.3 Canadian Foothill Model

A Canadian overthrust synthetic dataset was created for the CSEG paper by Gray & Marfurt (1995) (see Figure 5.10). The top layer is air with the surface of the earth defined by the first

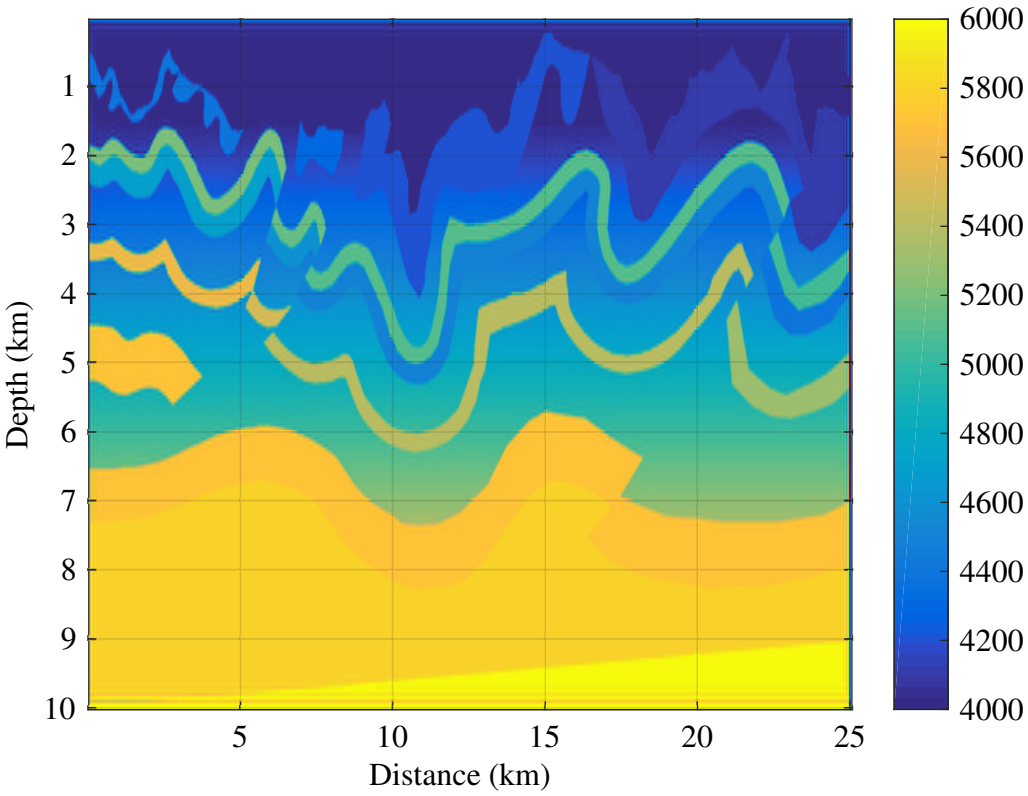


Figure 5.10 Cross section through the foothills of the Canadian rockies [m/s].

irregular interface. The model is compound of velocity gradients, faults, a complex topography and mixed areas of low and high velocities. The velocity model was widely circulated among Canadian contractor companies in the mid 1990's.

For this model, the FWI spent 90 minutes using 3.3 GiB (27.5%) of the GPU RAM of one GPU Tesla k40c, and the result after three frequency steps (3 Hz, 6 Hz and 9 Hz) is illustrated in Figure 5.11b. Table 5.3 summarizes the parameters used during this experiment. Figure 5.12 illustrates the difference between the original Canadian foothills model (Figure 5.11c) and the velocity model estimated with the FWI (Figure 5.11b). The main differences are presented at the left, right and bottom sides of the model as a consequence of the lighting area produced by the acquisition geometry. The best reconstruction is reached inside the lighting area where the velocities lower than 5000 m/s are correctly resolved.

Table 5.3
Parameters and values for the Canadian foothills experiment.

Parameter	Description	PSO	FWI	Hessian
Δh	Spatial step	50 m	50 m	50 m
D	Model size	301×161	301×161	301×161
Δt	Time step	4×10^{-3} s	2×10^{-3} s	4×10^{-3} s
T_{end}	Acquisition time	8 s	8 s	8 s
S_i	Number of sources	10	21	21
Rec_i	Number of receivers	261	261	261
V_{max}	Maximum velocity	5512 m/s.	5789 m/s	5789 m/s
V_{min}	Minimum velocity	4000 m/s.	3549 m/s	3549 m/s
$k\Delta h$	Numerical dispersion	0.24	0.8	0.8
P	Number of particles	500		
K	Number of iterations	50		
E	Number of experiments	10		
w	Inertia weight	0.7		
c_1	Local search weight	0.8		
c_2	Global search weight	1.1		
dt	Particles step size	1		
$F\{\mathbf{v}_{i,d}^k\}$	Metric domain	Hilbert		
α =ctte	Number of iterations		1	
L-BFGS	Number of iterations		200	
$\mathbf{g}(\mathbf{v}^k)$	Gradient strategy		**II**	
f_{steps}	Frequency steps	3 Hz	$3,6,9$ Hz	$3,6,9$ Hz
H_{area}	Hessian matrix area			121 points
H_{ite}	Hessian matrix iterations			189
RAM	Computational cost	440 MiB	3300 MiB	5.5 GiB
Time	Computational cost	13 Days per GPU Experi.	90 minutes	72.5 Hours

The cost function evolution and the instantaneous phase difference between the observed data and the modeled data are used as a quality control (QC) during the inversion process.

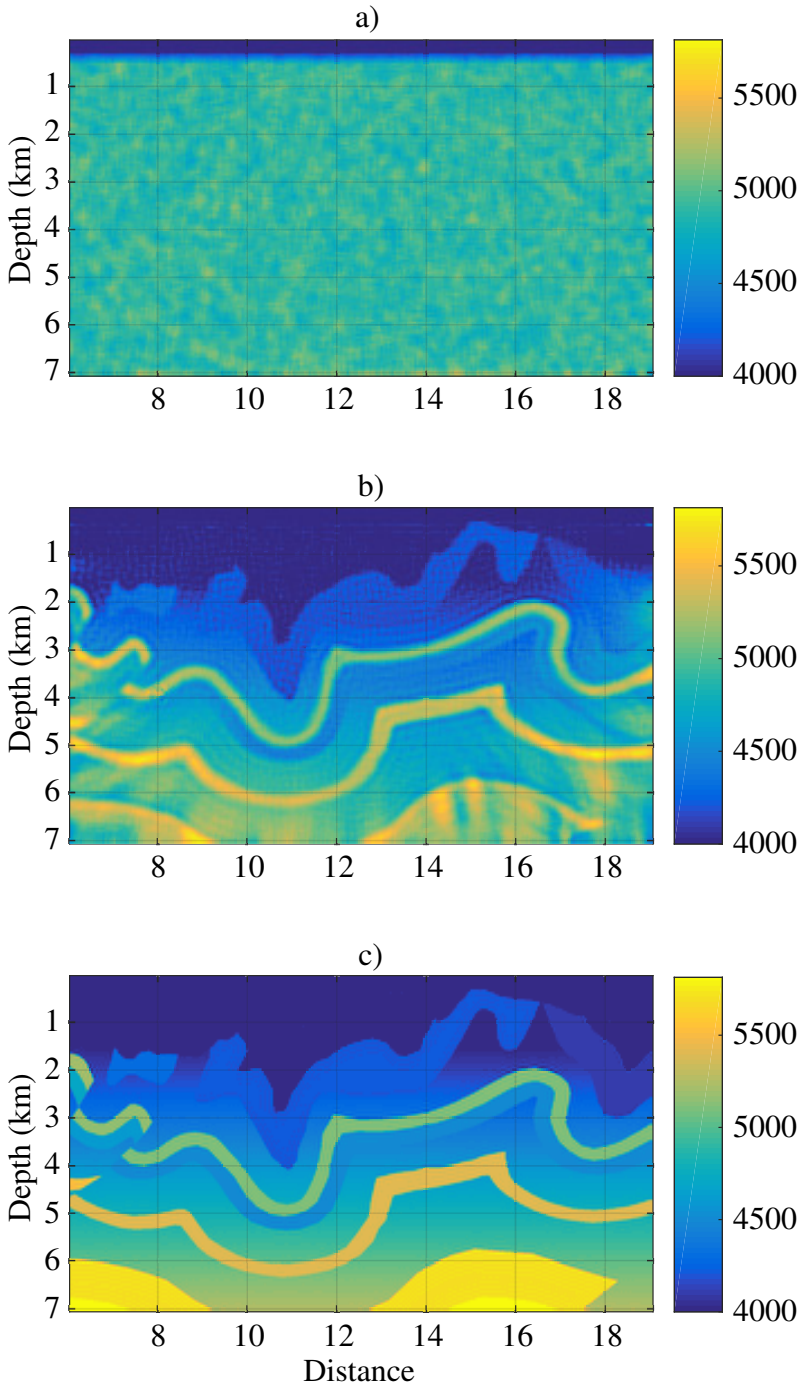


Figure 5.11 Canadian foothills velocity model section selected to test the proposed methodology. (a) Initial velocity model from the PSO exploration, (b) FWI result after a multi-scale approach and (c) Original section. All of them in [m/s].

Therefore, if both the cost function value and the phase difference decrease then the inversion process is going in the right direction. First row of Figure 5.13 illustrates the instantaneous

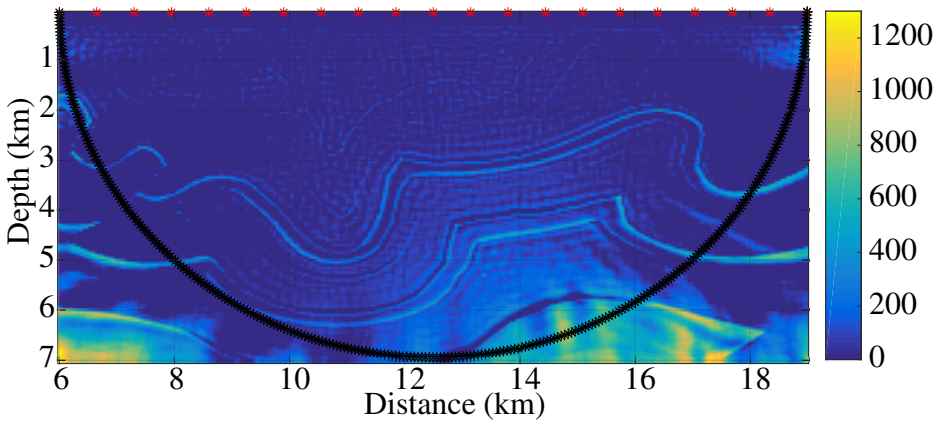


Figure 5.12 Difference in m/s between the original Canadian foothills velocity model and the FWI reconstruction. The sources used in the experiment are market in red color and the illuminated area is enclosed by a black ellipse.

phase difference between the observed data and the modeled data produce by the PSO result. For the three shots, the area with CS problems (yellow color) represents between 15% and 18% of the instantaneous phase differences which agree with the PSO results for the Hess and BP models (see first row of Figure 5.19, Figure 5.20, Figure 5.27 and, Figure 5.28). This behavior possibly means that the estimated model is still far from the correct value, even though the initial guess has the correct low frequency information.

When the instantaneous phase difference is computed over the FWI result at 3 Hz, the areas with CS are strongly reduced (yellow areas on second row of Figure 5.13). The yellow spots represent less than 0.7% of the total area while the first four seconds are better resolved. This behavior possibly means that the velocity model estimated by the FWI at 3 Hz is closer to the real velocity model than the initial guess, similarly as observed in Marmousi results (see Figure 5.4).

The instantaneous phase difference over the FWI result at 6 Hz reveals a CS decreasing of 0.2% compared with the FWI result at 3 Hz (yellow areas on third row of Figure 5.13). But at 9 Hz the instantaneous phase difference shows a slight increase in CS compared with the result at 6 Hz (see fourth row on Figure 5.13). This behavior can be explained because at 9 Hz the wavelet detects more details than at 6 Hz which can produce more CS. As the starting point for

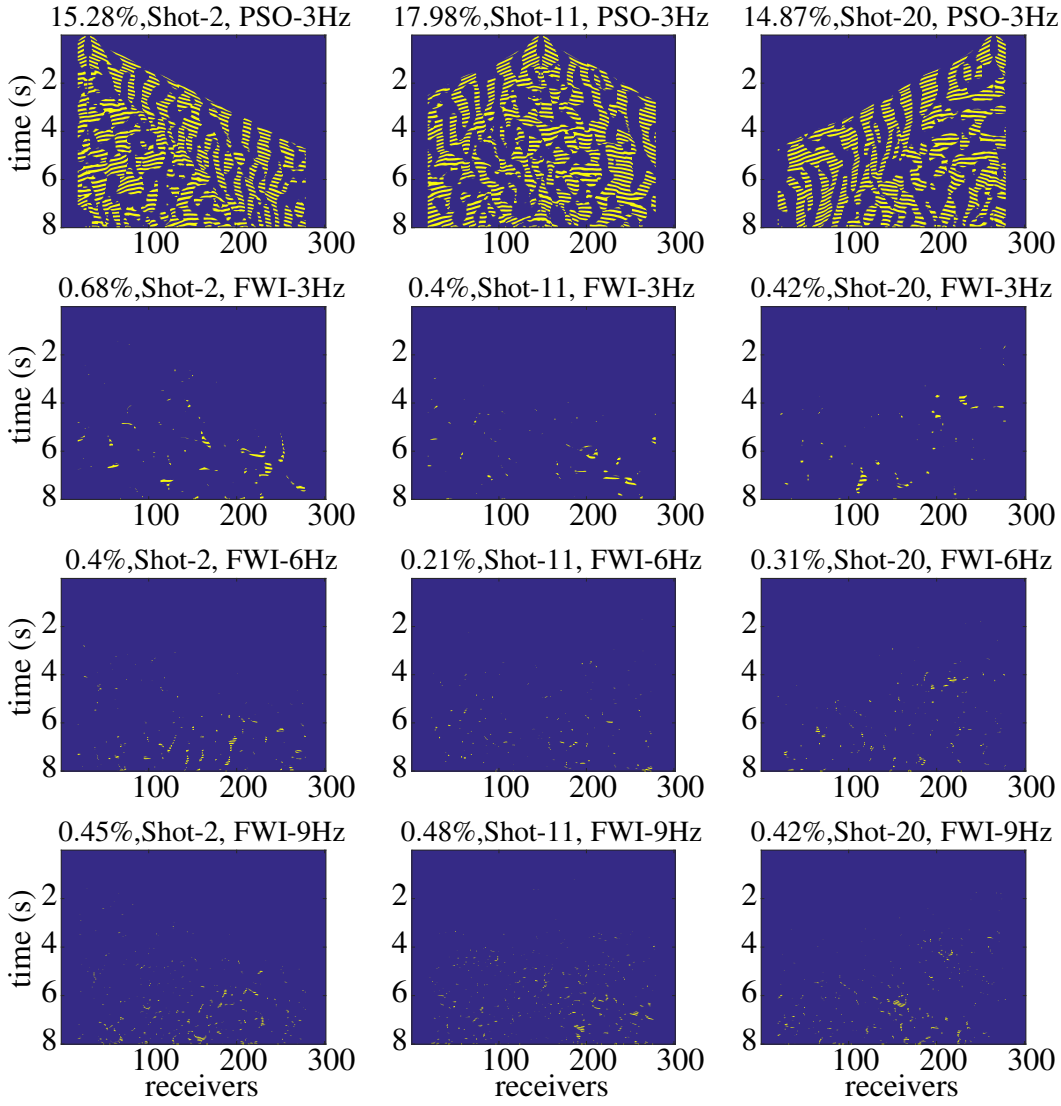


Figure 5.13 Instantaneous phase difference between the observed data and the modeled data over the PSO and full-offset FWI result for the Canadian foothills velocity model at 3 Hz, 6 Hz and 9 Hz using the shots 2, 11, and 20. The phase difference is greater than $\pi/2$ at the yellow color areas (areas with cycle-skipping problems) and the CS percentage is presented in the upper left corner of each result.

the inversion at 9 Hz is the result of the previous inversions steps at 3 Hz and 6 Hz then the main update at 9 Hz is focused on small details of the velocity model.

The use of the Hessian matrix to measure resolutions implies that it must be applied on areas correctly illuminated. In this way, the Hessian matrix computation out of the illumination

area (see Figure 5.12) does not make much sense because those places are not updated by the FWI. In fact, the best zone to compute the Hessian matrix should be inside the area with the higher illumination level. For this experiment, that area corresponds to the middle of the model, preferably in the upper side.

A squared area of 11 points in both directions centered at the position x=12.05 km with a depth of 0.75 km has been selected to compute the Hessian matrix (see yellow square in Figure 5.14). A total of 189 Hessian matrices have been computed to identify the behavior of 121 points over the FWI evolution at 3, 6 and 9 Hz. The process spent 72.5 hours using 5.5 GiB (45.8% of RAM) of one GPU Tesla k40c to reach the results. The same 21 sources and 261 receivers have been used to keep the illumination area of the FWI.

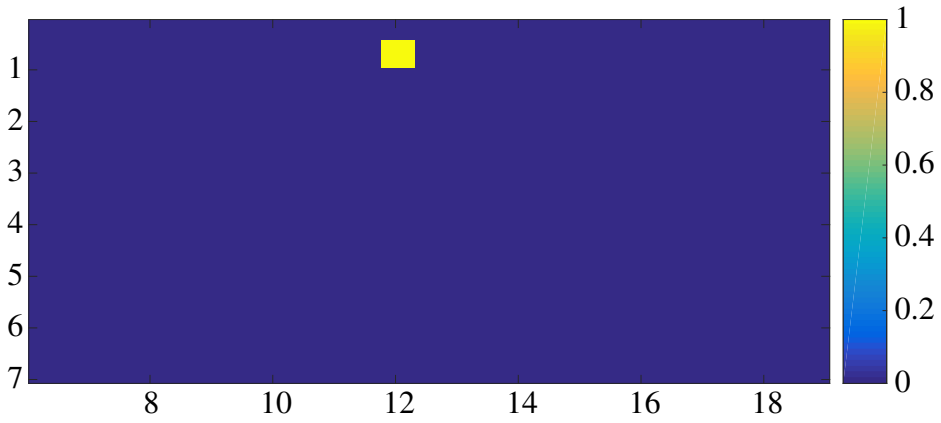


Figure 5.14 Illumination area where the Hessian matrix is evaluated for the PSO and the FWI results. The Hessian matrix is evaluated over 121 pixels that emulates a square of 11 pixels centered at the position x= 12.05 km with a depth of 0.75 km.

Twenty-one Hessian matrices have been computed per each frequency step of the multi-scale FWI using Equation 3.12 as a source. As the FWI is computed over 3 frequency stages then 63 Hessian matrices are produced to described the whole inversion process. The central frequency of Equation 3.12 allows to select the resolution that will be measured with the Hessian matrices. For this experiment, the evolution of the estimated velocity models have been measured at 3 Hz, 6 Hz, and 9 Hz and Figure 5.15 illustrates the results.

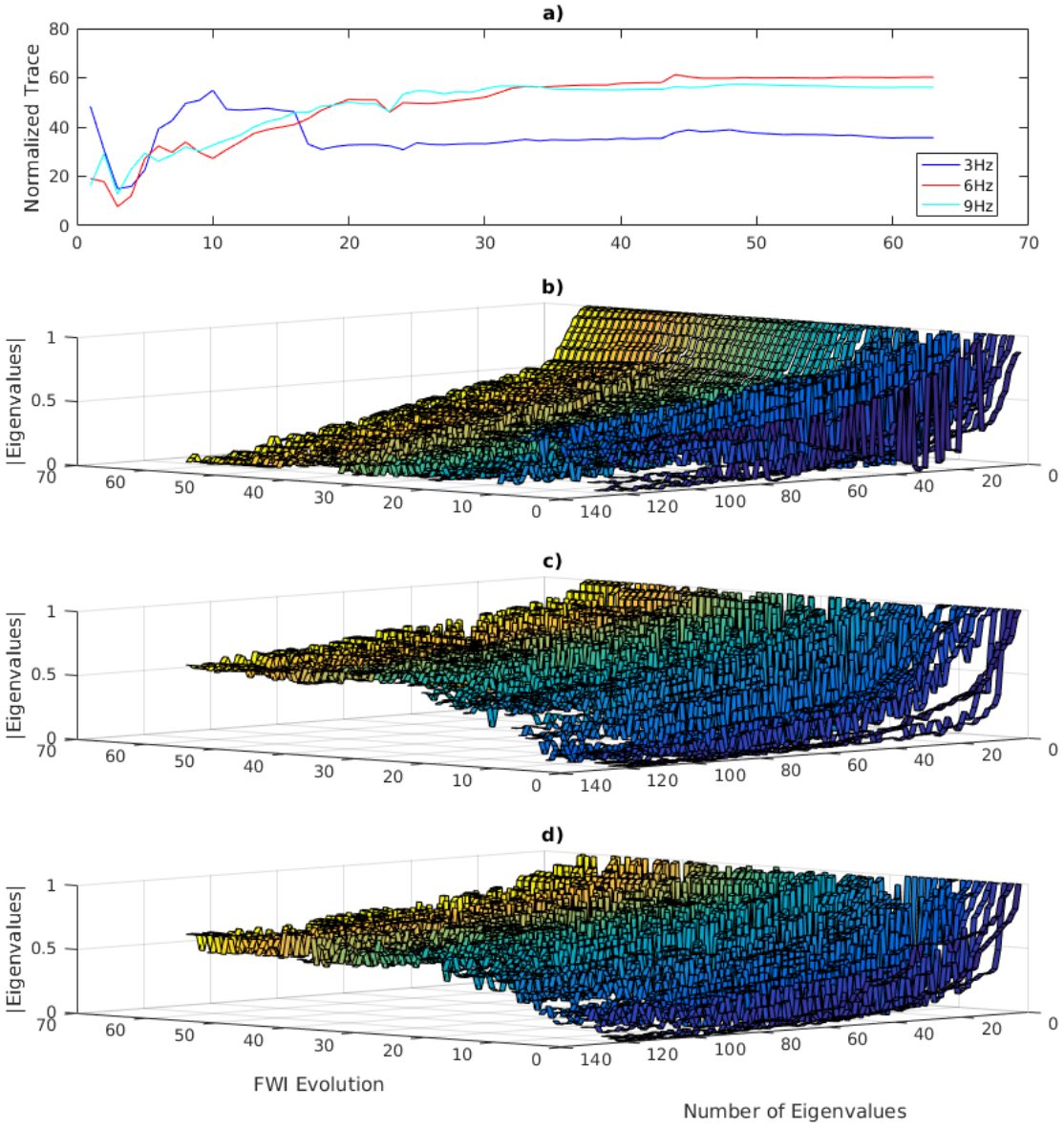


Figure 5.15 Canadian Foothills' matrix evolution over 63 FWI's iterations for the squared area of Figure 5.14. a) Normalized trace for 3 Hz, 6 Hz, and 9 Hz; Eigenvalues magnitude evolution for b) 3 Hz, c) 6 Hz and d) 9 Hz.

For all the cases, the Hessian matrices are squared and full rank which corresponds with the theoretical behavior but they neither are symmetric nor have condition numbers close to one, similarly as observed in the behavior of the Marmousi velocity inversion. The evolution of the eigenvalue magnitudes allows to identify the improvement on the initial guess while the normalized trace of the Hessian matrices allows to see the improvement on the eigenvalue magnitudes.

In this experiment, it is seen that at 6 Hz (Figure 5.15c and red line on Figure 5.15a) and at 9 Hz (Figure 5.15d and cyan line on Figure 5.15a) the estimated velocity model is reaching a better resolution with each FWI step on the squared yellow area (Figure 5.14). However at 3 Hz (Figure 5.15b and blue line on Figure 5.15a), there is an erratic behavior in the first 18 Hessian matrices, but then the estimated velocity model slightly improves with each FWI step.

As the original model has thin layers with alternated high and low velocity values and the layers have shapes with strong lateral velocity variations then the best velocity estimation is reached when the frequency increases because the wavefield can detect much finer contrasts. This physical effect is detected by the Hessian matrix and it explains the results illustrated in Figure 5.15.

5.3.1 Final Comments

A time constraint of three weeks has been imposed to the PSO exploration to reach the initial guess for all the four experiments. If it is compared with the Marmousi PSO exploration (5 source with 70 experiments), the number of sources have been increased to 10 but with only 10 experiments to respect the time constraint. In this example, the illumination area is improved due to the number of sources but each PSO particle has less time (only 10 experiments) to reach the global minimum neighborhood. Therefore, PSO has shown to get a faster convergence to the global minimum neighborhood when each PSO particle has more observations of the area of interest.

FWI spent 90 minutes and 3.3 GiB (27.5% of RAM) to reaches the final result using only one GPU Tesla k40c. Once more the FWI result reaches a velocity model estimation that is closer to the real one, validating the PSO result and the multi-scale strategy that preserve the correct bandwidth through the filtering stage.

The instantaneous phase difference between the modeled data and the observed data using the Hilbert transform and the Hessian matrix evolution measured on the estimated velocity

models through its normalized trace have proved to be complementary QCs. This is because while Hilbert transform indirectly measures the correct evolution from the data the Hessian matrix directly measures the correct evolution on the estimated models.

5.4 Hess Model

The Hess VTI model was generated at Hess Corporation with a distance of 22 km and a depth of 9 km on a grid of 6.096 m $\times$ 6.096 m (see Figure 5.16). The model has a salt body intrusion

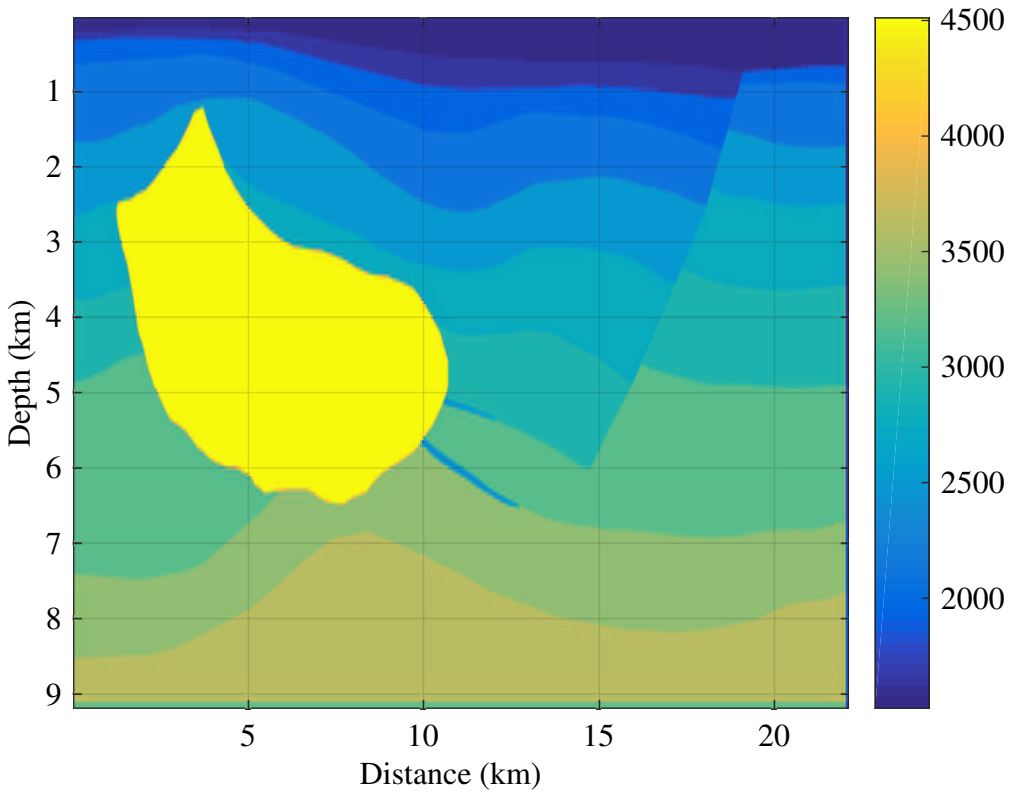


Figure 5.16 Hess velocity model [m/s].

in the left side and a geologic fault in the right side. The background velocity is formed by a layered model where the velocity increases with the depth. The salt body reaches the maximum velocity of 4500 m/s and, it has two slow velocity intrusions at its southeast side. One of the big challenges that offers this model is the high velocity contrast between the background velocity and the salt intrusion with a difference near 2000 m/s in 13.8% of the total area. Figure 5.17 illustrates the area of interest used to test the proposed methodology with 160 $\times$ 301 elements.

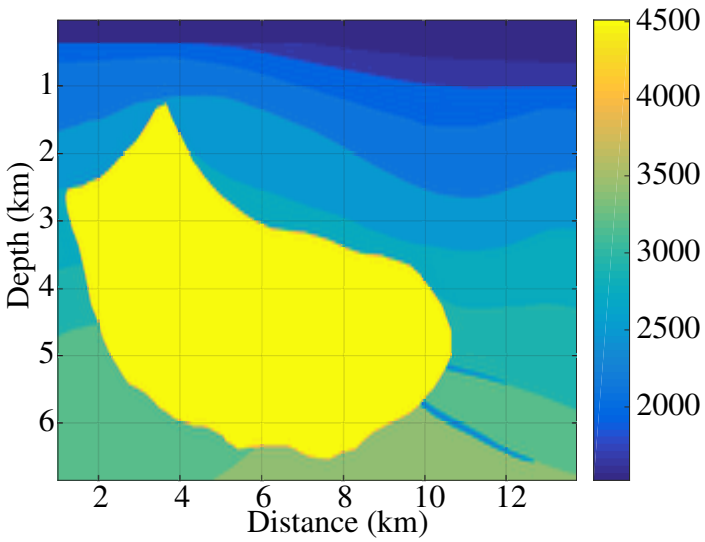


Figure 5.17 Section of the Hess velocity model [m/s].

For this model, the FWI spent 90 minutes using 3.3 GiB (27.5%) of GPU RAM and the result is illustrated in Figure 5.18b. Table 5.4 summarizes the parameters used during this experiment where two hundred iterations are run to resolve the details at 3 Hz, 6 Hz and 9 Hz.

For this experiment, the multi-scale FWI result (Figure 5.18b) could resolve neither the background layered velocities nor the salt intrusion (Figure 5.18c illustrates the difference between the original Hess velocity model and the FWI reconstruction using full-offset receivers). Instead, the FWI spreads high velocity areas (yellow color areas) throughout the model and it only resolves the northwest flank border of the salt intrusion. In this experiment, it seems as if the FWI reaches a local minimum that has some characteristics of the real model (global minimum). When the wavefields reach the high velocity intrusion area (see yellow area of Figure 5.17), they can travel through a larger area in a shorter time than the background wavefields (i.e., the wavefields that do not travel through the high velocity intrusion). This physical characteristic produces a mixed arrival on time of the background wavefields and the combined wavefields at the receivers position which divert the FWI to a local minimum.

This effect can be reduced by increasing the acquisition geometry because if the background area increases then the mixed arrivals on time of the combined wavefields are diminished due to the “relative” reduction of the high velocity intrusion area. A second option is to perform

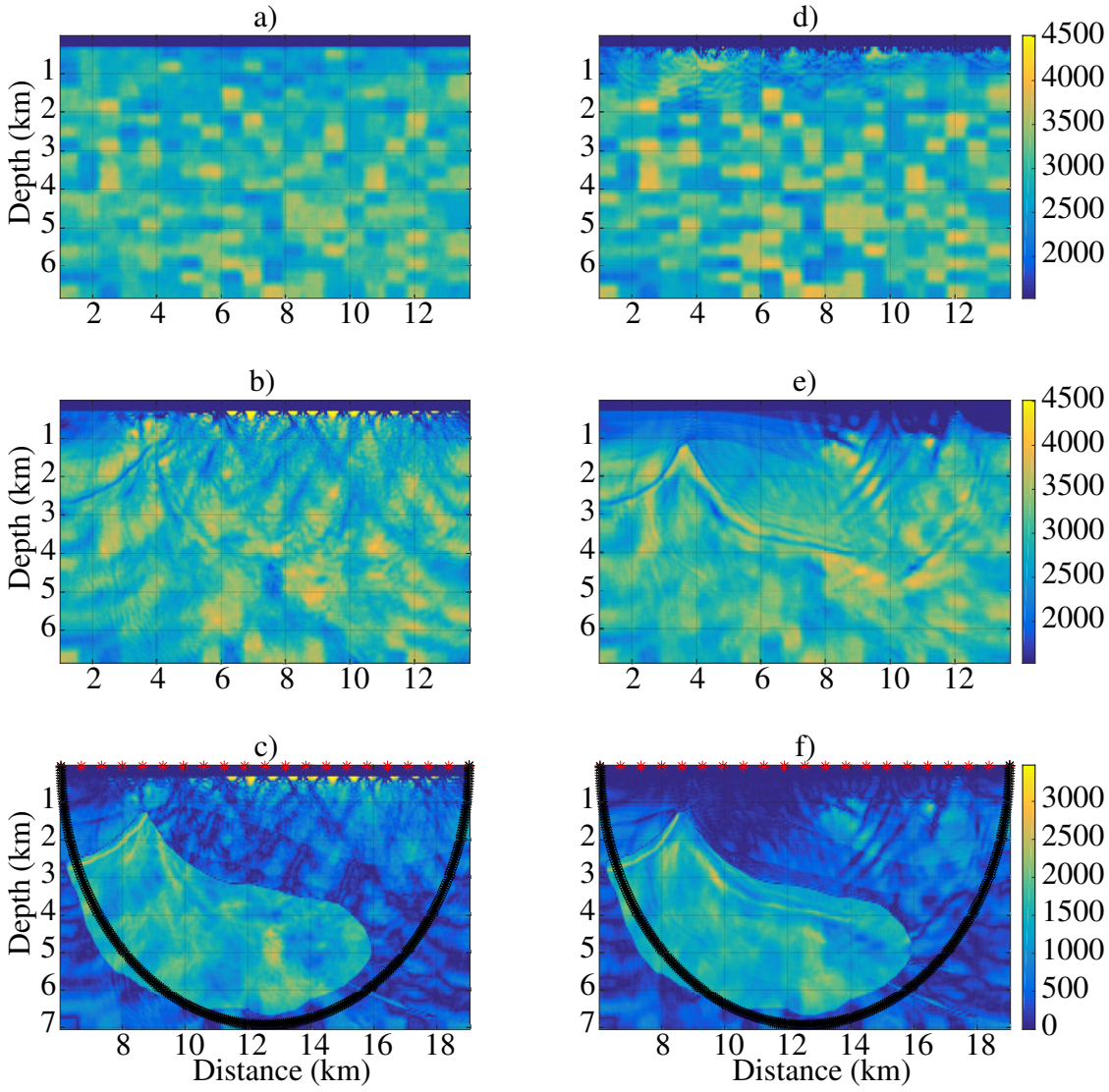


Figure 5.18 Hess experiment results for the full-offset and the short-far-offset strategies: a) PSO result, b) Full-offset FWI result, c) Difference between the Hess original section an the full-offset FWI result, d) short-offset FWI result, e) short-far-offset FWI result, and f) the difference between the Hess original section an the short-far-offset FWI result. All of them in [m/s]. On c) and f), the sources used in the experiment are market in red color and the illuminated area is enclosed by a black ellipse.

a short-offset inversion using seismic traces without depth events and once the first velocity layers have been resolved then the far-offset and the deepest events are taking into account to resolve the deepest layers.

Table 5.4
Parameters and values for the Hess experiment.

Parameter	Description	PSO	FWI	Hessian
Δh	Spatial step	50 m	50 m	50 m
D	Model size	301×160	301×160	301×160
Δt	Time step	4×10^{-3} s	2×10^{-3} s	4×10^{-3} s
T_{end}	Acquisition time	8 s	8 s	8 s
S_i	Number of sources	10	21	21
Rec_i	Number of receivers	261	261	261
V_{max}	Maximum velocity	3900 m/s.	5000 m/s	5000 m/s
V_{min}	Minimum velocity	1524 m/s.	1951 m/s	1951 m/s
$k\Delta h$	Numerical dispersion	0.61	0.99	0.99
P	Number of particles	500		
K	Number of iterations	50		
E	Number of experiments	10		
w	Inertia weight	0.7		
c_1	Local search weight	0.8		
c_2	Global search weight	1.1		
dt	Particles step size	1		
$F\{\mathbf{v}_{i,d}^k\}$	Metric domain	Hilbert		
$\alpha =$ctte	Number of iterations		1	
L-BFGS	Number of iterations		200	
$\mathbf{g}(\mathbf{v}^k)$	Gradient strategy		**II**	
f_{steps}	Frequency steps	3 Hz	$3,6,9$ Hz	$3,6,9$ Hz
H_{area}	Hessian matrix area			121 points
H_{ite}	Hessian matrix iterations			189
RAM	Computational cost	440 MiB	3300 MiB	5.5 GiB
Time	Computational cost	13 Days per GPU Experi.	90 minutes	72.5 Hours

Figure 5.18d illustrates the result after a multi-scale short-offset (2 km) FWI with shallow events over the PSO result while Figure 5.18e illustrates the result after a multi-scale short-far-offset FWI including deepest events. Both tests use the same FWI configuration described on Table 5.4.

As expected, the short-offset FWI (Figure 5.18d) only resolves the first 2 km of depth of the PSO initial guess. However, if this result is used as an input for the far-offset FWI then the upper border of the high velocity intrusion together with the first velocity layers can be identified (Figure 5.18e). But the right side of the estimated velocity model and the lower border of the high velocity intrusion are still missing. All this without mentioning that there are still high velocity areas in wrong places (Figure 5.18f illustrates the difference between the original Hess

velocity model and the FWI reconstruction using the short-far-offset strategy).

The instantaneous phase difference between the observed data and the modeled data over the PSO and both full-offset and short-far-offset FWI results are illustrated in Figure 5.19 and Figure 5.20, respectively. From the evolution of the shots 2, 11 and 20 it is seen that at 3 Hz the CS decreases under 12.6% and 8.7% for the full-offset and short-far-offset strategies, respectively. Then the CS increases slightly (at least 0.45%) for the three shots at 6 Hz and 9 Hz using both strategies; indicating that at those frequencies the velocity estimations are going in a local minimum direction.

A squared area of 11 points in both directions centered at the position x=6.876 km with a depth of 0.7315 km has been selected to compute the Hessian matrix (see yellow square in Figure 5.21). A total of 189 Hessian matrices have been computed to identify the behavior of 121 points over the FWI evolution at 3, 6 and 9 Hz. The process spent 72.5 hours using 5.5 GiB (45.8% of RAM) of one GPU Tesla k40c to reach the results. The same 21 sources and 261 receivers have been used to keep the illumination area of the FWI.

Twenty one Hessian matrices has been computed per each frequency step of the multi-scale FWI using Equation 3.12 as a source. As the FWI is computed over 3 frequency stages then 63 Hessian matrices are produced to described the whole inversion process. The central frequency of Equation 3.12 allows to select the resolution that will be measured with the Hessian matrices. For this experiment, the evolution of the estimated velocity models have been measured at 3 Hz, 6 Hz, and 9 Hz for the full-offset inversion and the short-far-offset inversion. Figure 5.22 and Figure 5.23 illustrate the results.

For all the cases, the Hessian matrices are squared and full rank which agree with the theoretical behavior but they neither are symmetric nor have condition numbers close to one, similarly as observed in the behavior of the Marmousi and Canadian foothills inversions. The evolution of the eigenvalue magnitudes allows to identified the improvement on the initial guess while

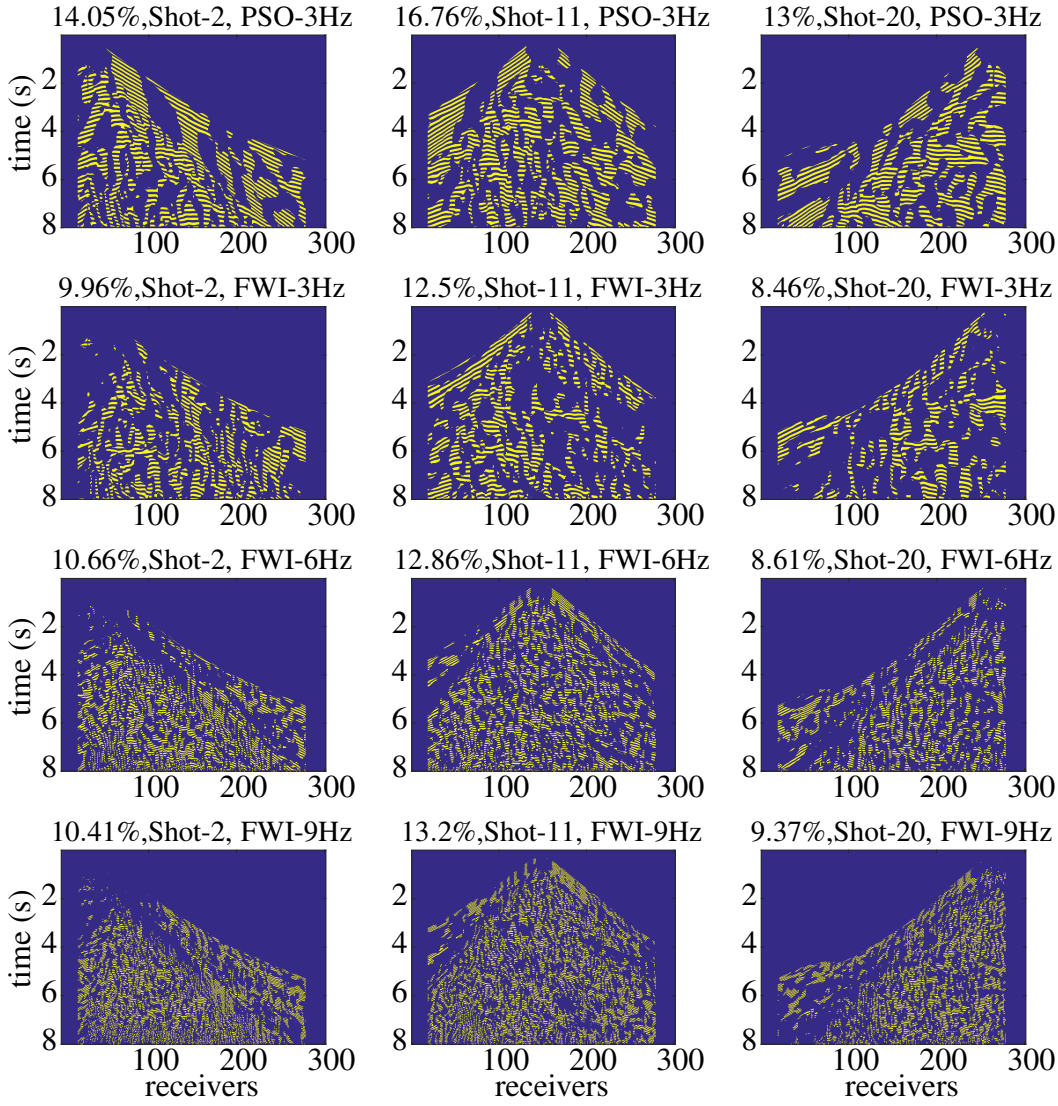


Figure 5.19 Instantaneous phase difference between the observed data and the modeled data over the PSO and the full-offset FWI results for the Hess velocity model at 3 Hz, 6 Hz and 9 Hz using the shots 2, 11 and 20. The phase difference is greater than $\pi/2$ at the yellow color areas (areas with cycle-skipping problems) and the CS percentage is presented in the upper left corner of each result.

the normalized trace of the Hessian matrices allows to see the improvement on the eigenvalue magnitudes.

From the beginning, the full-offset FWI goes in the wrong direction, therefore, the eigenvalue characteristics are getting far from their theoretical behavior with each FWI iteration step

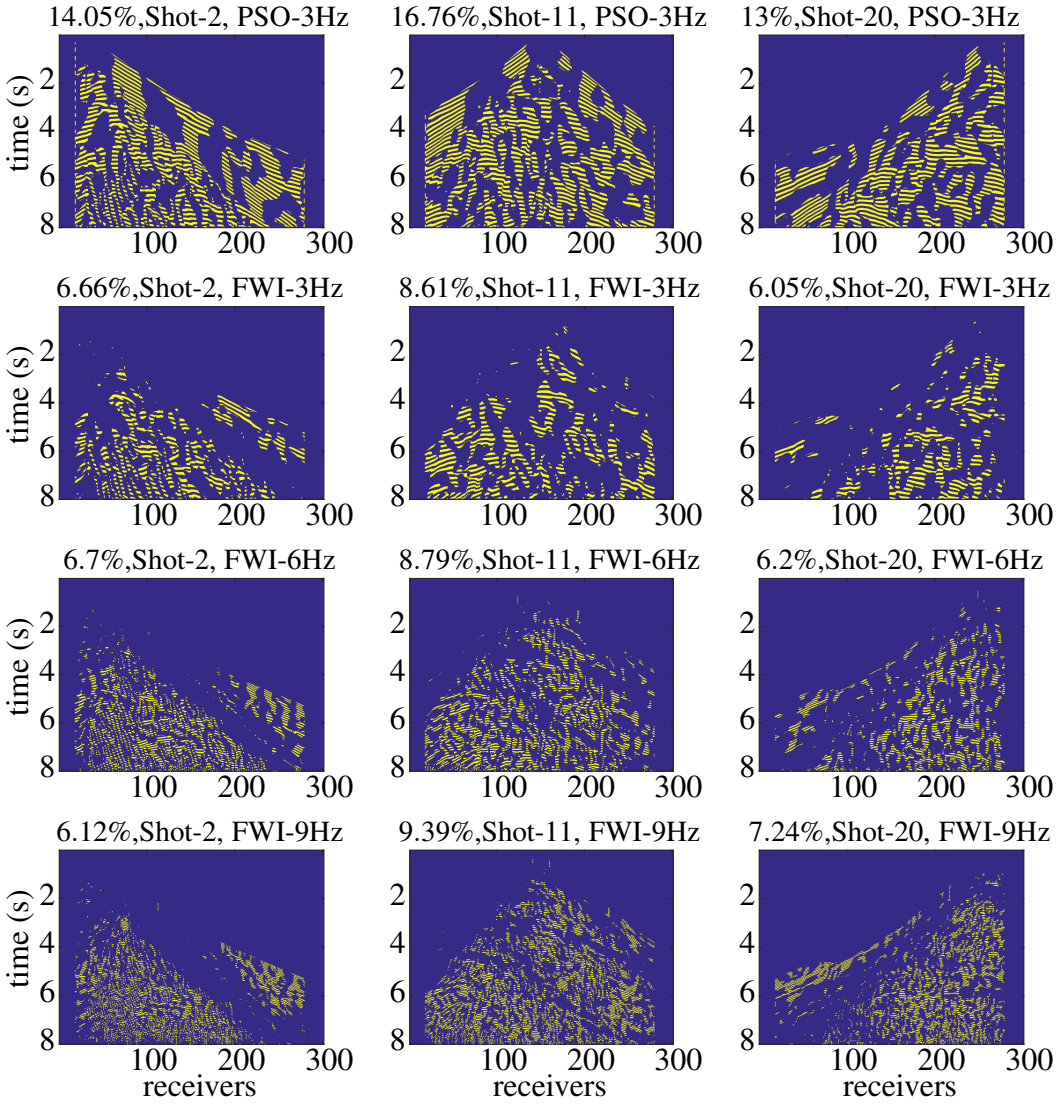


Figure 5.20 Instantaneous phase difference between the observed data and the modeled data over the PSO and the short-far-offset FWI results for the Hess velocity model at 3 Hz, 6 Hz and 9 Hz using the shots 2, 11 and 20. The phase difference is greater than $\pi/2$ at the yellow color areas (areas with cycle-skipping problems) and the CS percentage is presented in the upper left corner of each result.

in all the three frequencies. Figure 5.22a illustrates that in the first 21 iterations the normalized trace decreases for the three frequencies. From iteration 22 to iteration 42 the normalized trace keeps a stable value and in the last 21 iterations at 6 Hz and 9 Hz there is a slight improvement. The eigenvalue magnitudes evolution does not allow to observe this behavior easily

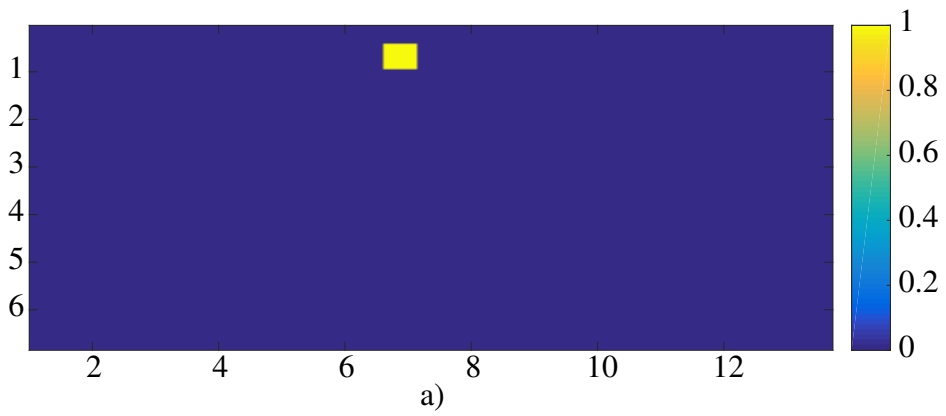


Figure 5.21 Illumination area where the Hessian matrix is evaluated for the PSO and the FWI results. The Hessian matrix is evaluated over 121 pixels that emulates a square of 11 pixels centered at the position x= 6.876 km with a depth of 0.7315 km.

(Figure 5.22b, Figure 5.22c and, Figure 5.22d).

The short-far-offset FWI loses quality on the eigenvalue characteristics in the first 10 iterations because the normalized trace magnitude decreases with each iteration step (Figure 5.23a). However at 3 Hz, the process starts to recover the eigenvalue qualities from the iteration 12 to the iteration 55 without reaching its best value. At 6 Hz and 9 Hz the eigenvalue magnitudes evolution have an erratic behavior with some ups and downs indicating that in average there is not an improvement during the whole process. Figure 5.23b, Figure 5.23c and Figure 5.23d show the details on the eigenvalues evolution but according to the results only at 3 Hz the estimated velocity model reaches a significant improvement.

5.4.1 Final Comments

In this experiment, the PSO cost function exploration is performed using 500 particles, 10 experiments, 10 sources and 261 receivers. The initial guess for the FWI is obtained after running the PSO exploration during three weeks using the GPU cluster of the CPS research group. And, it has a mixture of low, intermediate and high velocity areas spread throughout the velocity model. When the PSO result is compared with the original velocity model (Figure 5.17), it looks like PSO could find the correct velocity values but it could not place them in the right

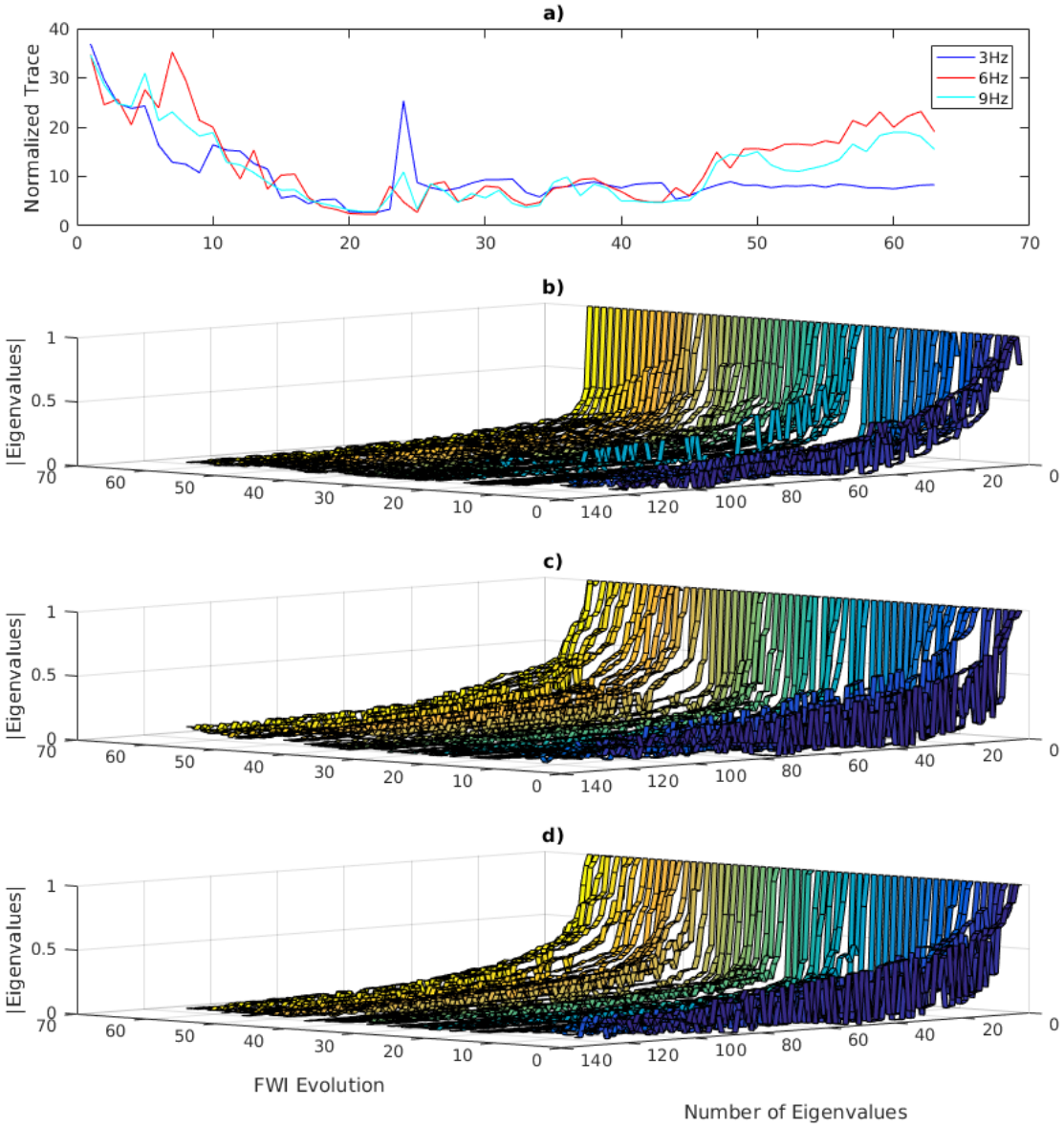


Figure 5.22 Hessian matrix evolution over 63 full-offset FWI's iterations for the squared area of Figure 5.21. a) Normalized trace for 3 Hz, 6 Hz, and 9 Hz; Eigenvalues magnitude evolution for b) 3 Hz, c) 6 Hz and d) 9 Hz.

place.

Keeping that same train of thoughts, the FWI mission is to correctly assemble this puzzle. Two different strategies are applied to perform the FWI: the first one directly uses the full-offset multi-scale FWI over the PSO result; and the second one first run a short-offset multi-scale FWI without depth events over the PSO result and then the far-offset multi-scale FWI is applied over

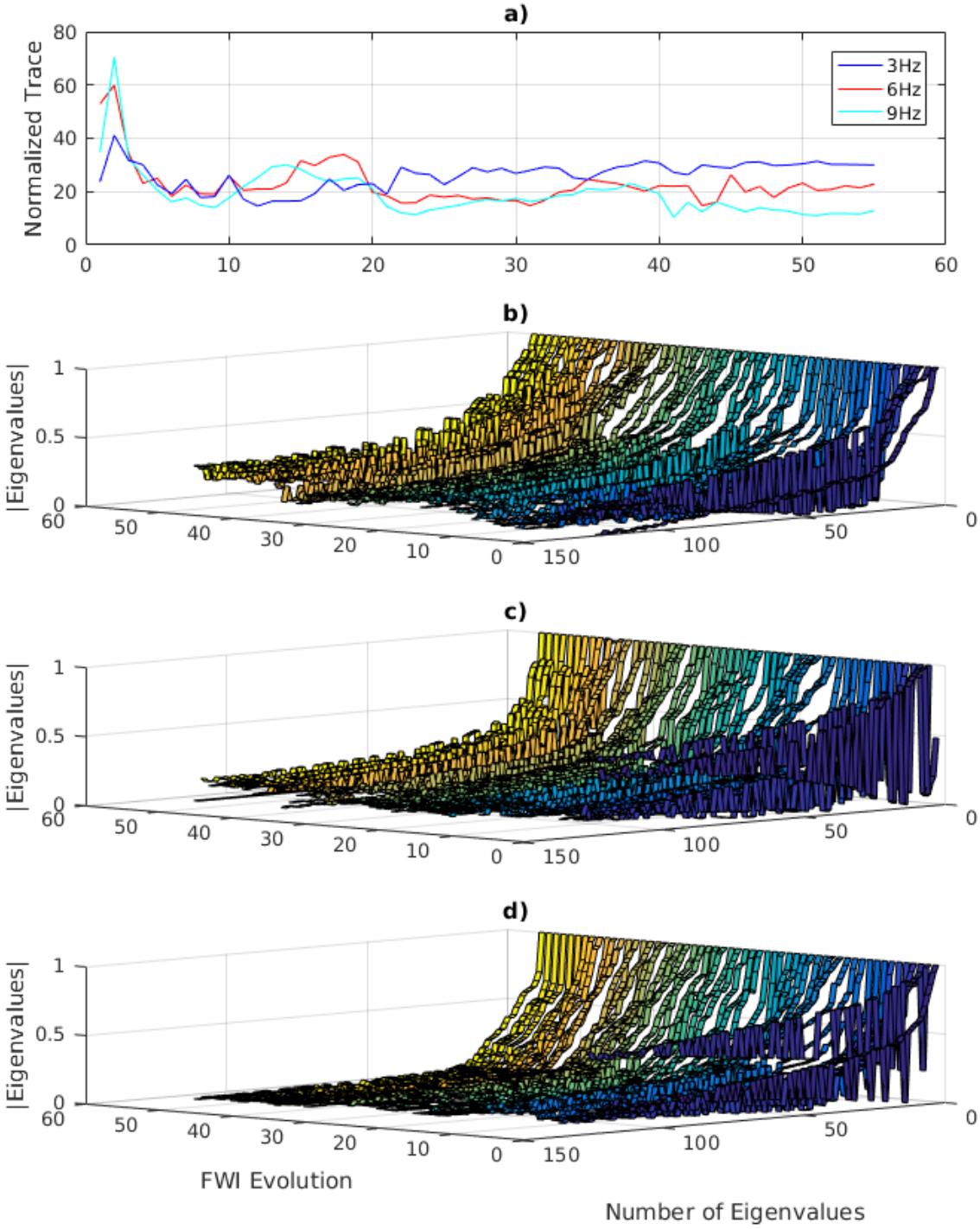


Figure 5.23 Hessian matrix evolution over 55 short-far-offset FWI's iterations for the squared area of Figure 5.21. a) Normalized trace for 3 Hz, 6 Hz, and 9 Hz; Eigenvalues magnitude evolution for b) 3 Hz, c) 6 Hz and d) 9 Hz.

the short-offset result. The best results are obtained from the second strategy because with the short-offset information and without the deepest events the FWI resolves the first layers. Once the first layers are correct then the far-offset and depth events resolve the deepest layers. The first strategy fails because the FWI resolve the background velocity and the high velocity intrusion at the same time using the whole offsets and events. Probably a finer management of seismic events and offsets could produce better velocity estimations when high velocity intrusions are involved.

The instantaneous phase difference between the modeled data and the observed data using the Hilbert transform and the Hessian matrix evolution measured on the estimated velocity models through its normalized trace have proved to be successful identifying the correct evolution of the FWI. In the first strategy, both QCs show that there is not an improvement over the estimated velocity model after the inversion. In fact, the Hessian matrix evolution shows a decay of quality during the process. In the second strategy, both CS on the shots and the Hessian matrix evolution show an improvement on the velocity estimation specifically at 3 Hz.

5.5 BP 2004 Model

The model was developed as a velocity analysis benchmark dataset for the workshop "Estimation of Accurate Velocity Macro Models in Complex Structures" at the 2004 EAGE meeting in Paris, France (Billette & Brandsberg-Dahl, 2005). In the left part of the model (Figure 5.24) there is a complex coarse multi-valued salt body representative of geology in deep water Gulf of Mexico. At the center there is a deeply rooted salt body representative for the Gulf of Mexico and West Africa. And the right part represents a geology with shallow gas and localized shallow anomalies which can be found in areas as the Caspian Sea, offshore Trinidad and the North Sea. The velocity model is 67 km long and 12 km deep on a squared grid of 6.25 m with 10720×1920 elements.

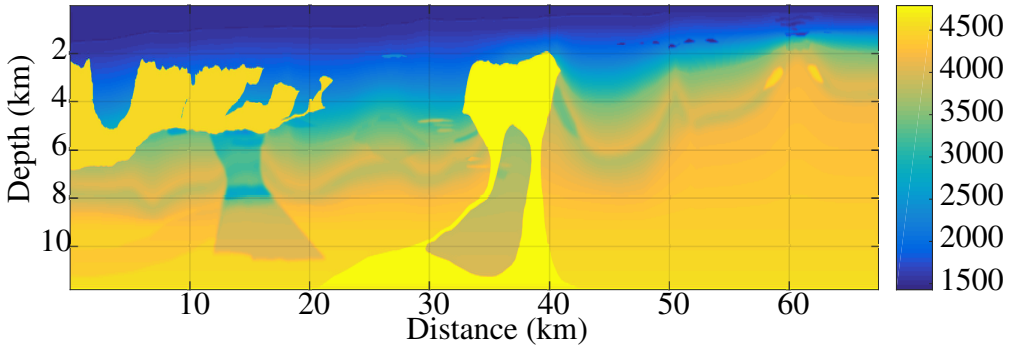


Figure 5.24 BP 2004 velocity model [m/s].

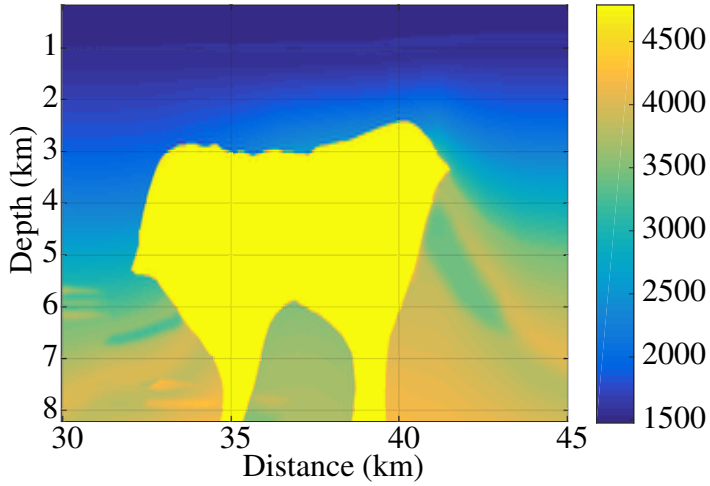


Figure 5.25 BP velocity model section selected to test the proposed methodology in [m/s].

For this experiment, the FWI spent 5 hours using 3.3 GiB (27.5%) of GPU RAM and the result is illustrated in Figure 5.26b. Table 5.5 summarizes the parameters used during this experiment.

Table 5.5
Parameters and values for the BP experiment.

Parameter	Description	PSO	FWI	Hessian
Δh	Spatial step	50 m	50 m	50 m
D	Model size	301×160	301×160	301×160
Δt	Time step	4×10^{-3} s	2×10^{-3} s	4×10^{-3} s
T_{end}	Acquisition time	8 s	11 s	11 s
S_i	Number of sources	10	21	21
Rec_i	Number of receivers	261	261	261
V_{max}	Maximum velocity	3656 m/s.	5000 m/s	5000 m/s
V_{min}	Minimum velocity	1483 m/s.	1483 m/s	1483 m/s
$k\Delta h$	Numerical dispersion	0.64	0.99	0.99
P	Number of particles	500		
K	Number of iterations	50		
E	Number of experiments	10		
w	Inertia weight	0.7		
c_1	Local search weight	0.8		
c_2	Global search weight	1.1		
dt	Particles step size	1		
$F\{\mathbf{v}_{i,d}^k\}$	Metric domain	Hilbert		
α =ctte	Number of iterations		1	
L-BFGS	Number of iterations		200	
$\mathbf{g}(\mathbf{v}^k)$	Gradient strategy		**II**	
f_{steps}	Frequency steps	3 Hz	$3, 6, 9$ Hz	$3, 6, 9$ Hz
H_{area}	Hessian matrix area			121 points
H_{ite}	Hessian matrix iterations			144
RAM	Computational cost	440 MiB	3.3 GiB	7.6 GiB
Time	Computational cost	13 Days per GPU Experi.	5 Hours	62.4 Hours

Wavefields that reach the high velocity intrusion area (see yellow area of Figure 5.25), can travel a greater distance in a shorter time than the wavefields that just propagate over the background velocity. This physical characteristic produces a mixed arrival on time of the background wavefields and the combined wavefields at the receivers position. The multi-scale full-offset FWI result (Figure 5.26b) could resolves neither the background gradient velocity nor the salt intrusion (Figure 5.26c illustrates the difference between the original section and the result after the multi-scale full-offset FWI). Instead, the FWI creates high velocity areas (yellow triangles in Figure 5.26b) in the shallow part of the model with a background mixture of inter-

mediate velocities. In this experiment, it seems as the FWI gets trapped in a local minimum when both the full-offset and all the seismic events are used at the same time.

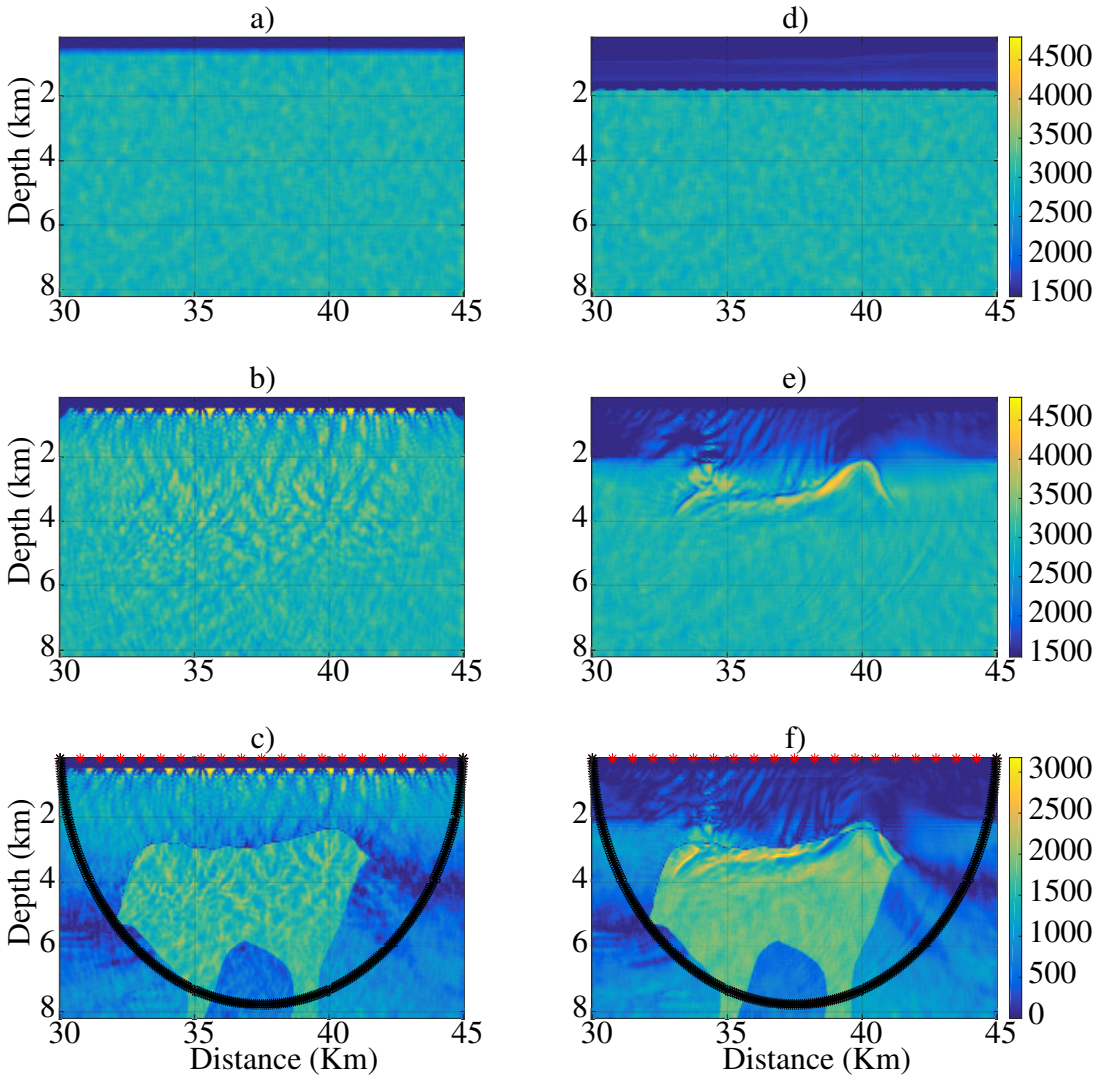


Figure 5.26 BP experiment results for the full-offset and the short-far-offset strategies: a) PSO result, b) Full-offset FWI result, c) Difference between the Hess original section an the full-offset FWI result, d) short-offset FWI result, e) short-far-offset FWI result, and f) the Difference between the Hess original section an the short-far-offset FWI result. All of them in [m/s]. On c) and f), the sources used in the experiment are market in red color and the illuminated area is enclosed by a black ellipse.

If a multi-scale short-offset FWI is performed using seismic traces without depth events then the first velocity layers can be correctly resolve due to the absence of the high velocity

intrusion effects. Once the first velocity layers have been resolved then the far-offset and the deepest events can be taking into account to resolve the deepest layers. Therefore, a short-far-offset multi-scale FWI can help to resolve this type of velocity models from top to bottom reducing the probability of get trapped in a local minimum.

Figure 5.26d illustrates the result after a multi-scale short-offset (1.8 km) FWI with shallow events over the PSO result while Figure 5.26e illustrates the result after a multi-scale short-far-offset FWI. Both tests use the same FWI configuration described on Table 5.5.

As it is expected, the short-offset FWI result (Figure 5.26d) only resolves the first 1.8 km of depth of the PSO initial guess. But, when this result is used as an input for the short-far-offset FWI then the upper border of the high velocity intrusion can be identified (Figure 5.26e). However, the background velocity below 1.8 km in depth and the other flanks of the high velocity intrusion are still missing (Figure 5.26f illustrates the difference between the original section and the result after a multi-scale short-far-offset FWI).

The instantaneous phase difference between the observed data and the modeled data over the PSO and both full-offset and short-far-offset FWI results are illustrated in Figure 5.27 and Figure 5.28, respectively. From the evolution of the shots 2, 11 and 20 it is seen that at 3 Hz the CS oscillates from -0.35% to 1.26% around the PSO result and decreases under 5.7% for the full-offset and short-far-offset strategies, respectively. Then the CS increases at least 3.57% for the three shots at 6 Hz and 9 Hz with the short-far-offset strategy but decreases under 2% for the full-offset strategy. From the experiment results, the short-far-offset strategy at 3 Hz reaches the lowest CS percentages indicating that this result is the closest to the global minimum.

A square area of 11 elements in both sides centered at the position x=36.92 km with a depth of 1 km has been selected to compute the Hessian matrix (see yellow square in Figure 5.29). A total of 144 Hessian matrices have been computed to identify the behavior of 121 points over the FWI evolution at 3, 6 and 9 Hz. The process spent 62.4 hours using 7.6 GiB (63.3% of RAM) of one GPU Tesla k40c to reach the results. The same 21 sources and 261 receivers

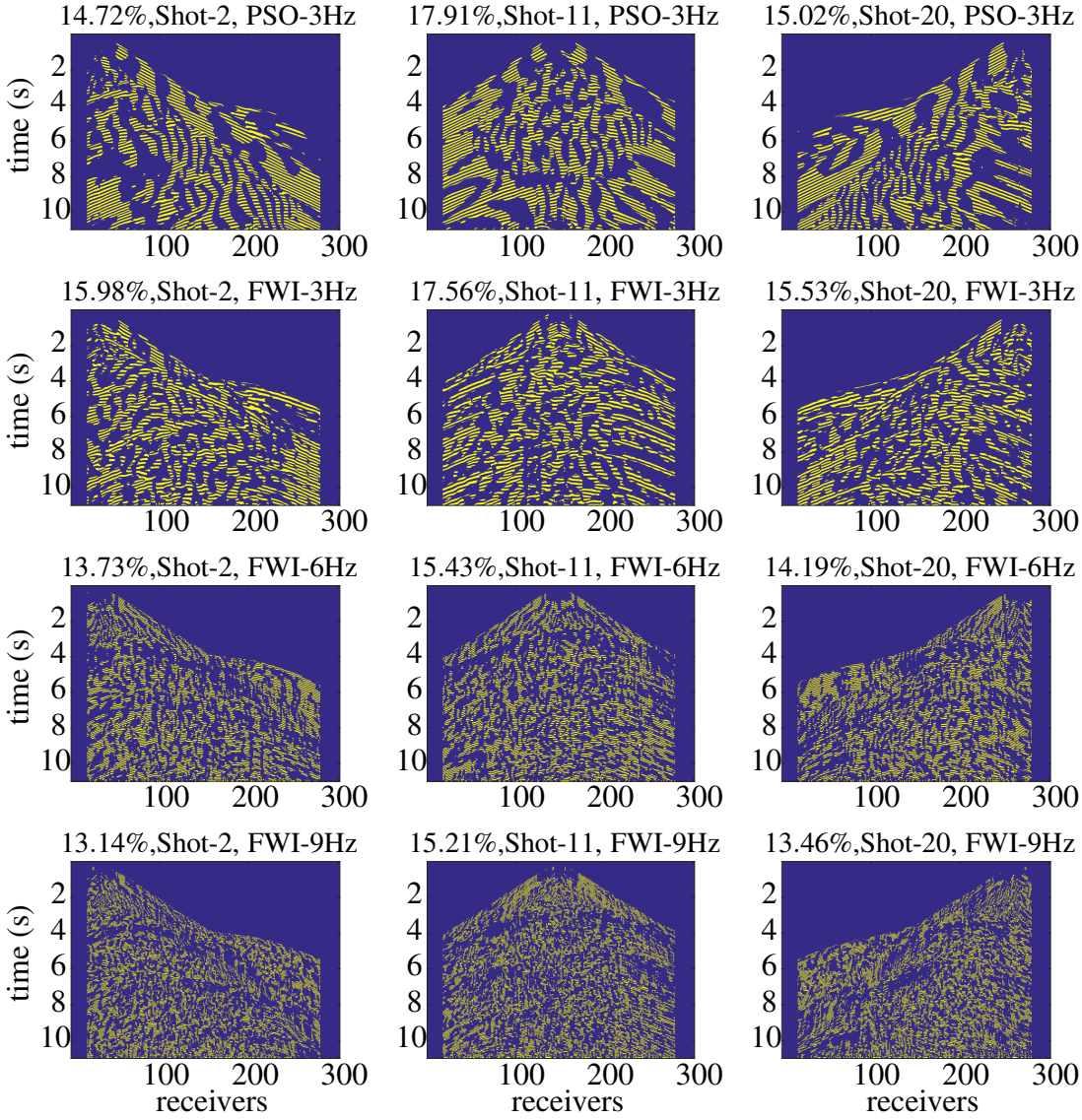


Figure 5.27 Instantaneous phase difference between the observed data and the modeled data over the PSO and the full-offset FWI results for the BP velocity model at 3 Hz, 6 Hz and 9 Hz using the shots 2, 11 and 20. The phase difference is greater than $\pi/2$ at the yellow color areas (areas with cycle-skipping problems) and the CS percentage is presented in the upper left corner of each result.

recording 11 seconds with a time step of 4×10^{-3} s have been used to keep the same illuminated area of the FWI.

Sixteen Hessian matrices has been computed per each frequency step of the multi-scale FWI using Equation 3.12 as a source. As the FWI is computed over 3 frequency stages then 48

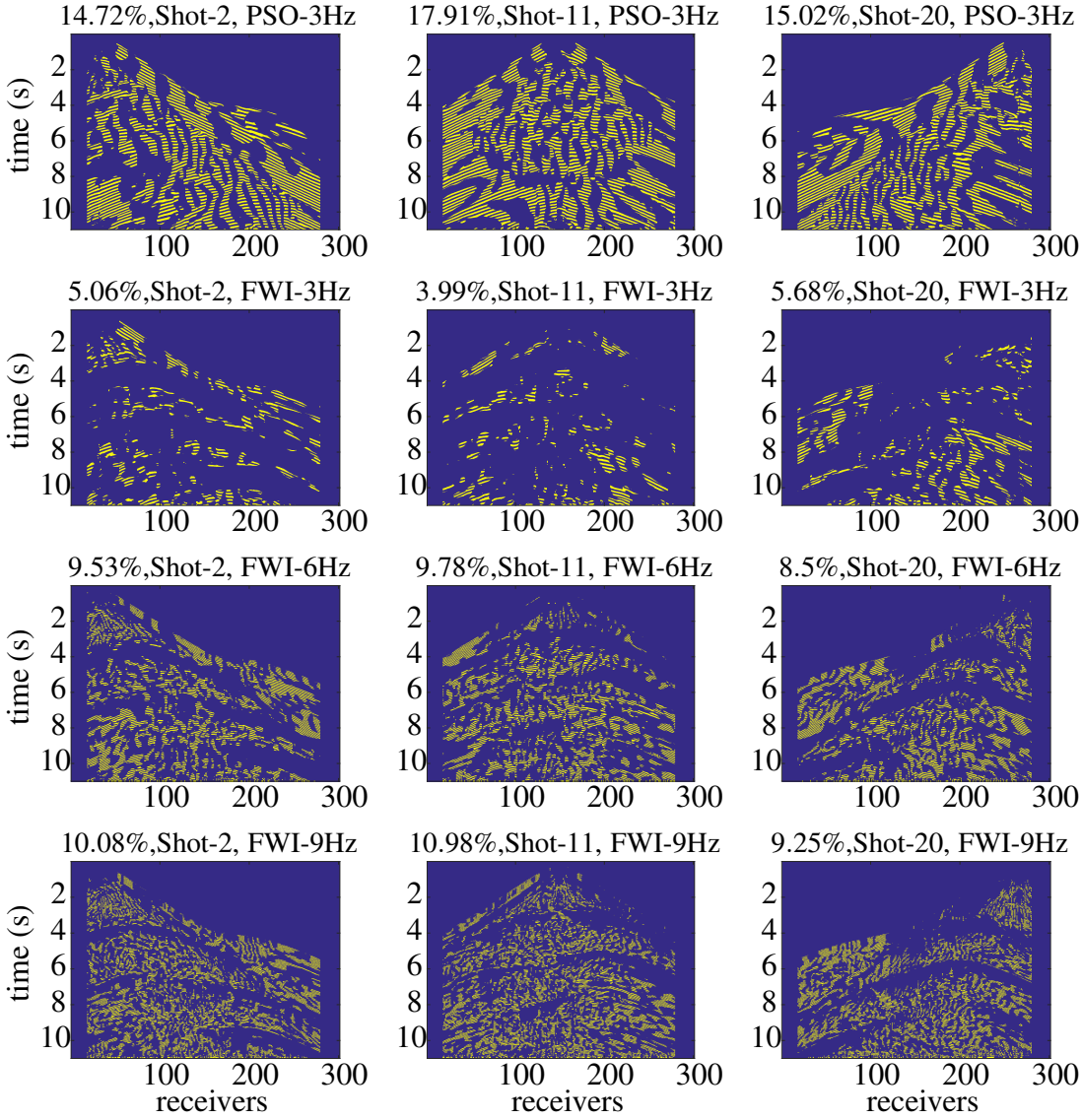


Figure 5.28 Instantaneous phase difference between the observed data and the modeled data over the PSO and the short-far-offset FWI results for the BP velocity model at 3 Hz, 6 Hz and 9 Hz using the shots 2, 11 and 20. The phase difference is greater than $\pi/2$ at the yellow color areas (areas with cycle-skipping problems) and the CS percentage is presented in the upper left corner of each result.

Hessian matrices are produced to described the whole inversion process. The central frequency of Equation 3.12 allows to select the resolution that will be measured with the Hessian matrices. For this experiment, the evolution of the estimated velocity models have been measured at 3 Hz, 6 Hz, and 9 Hz for the full-offset inversion and the short-far-offset inversion. Figure 5.30 and

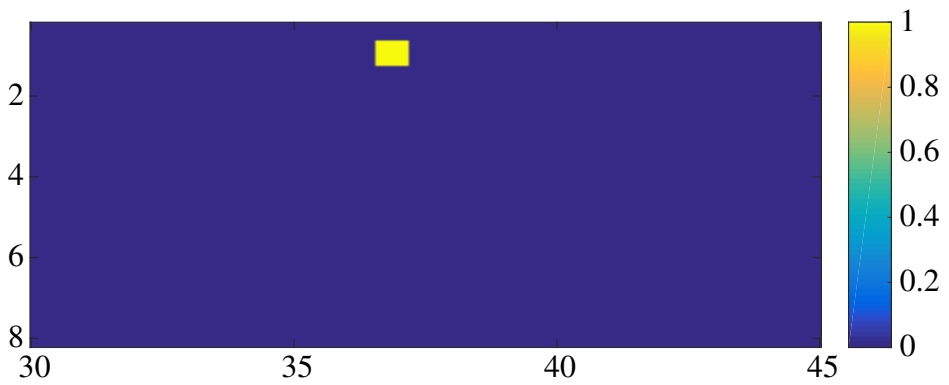


Figure 5.29 Illumination area where the Hessian matrix is evaluated for the PSO and the FWI results. The Hessian matrix is evaluated over 121 pixels that emulates a square of 11 pixels centered at the position x= 36.92 km with a depth of 1 km.

Figure 5.31 illustrate the results.

For all the cases, the Hessian matrices are squared and full rank which agree with the theoretical behavior but they neither are symmetric nor have condition numbers close to one similarly to the behavior of the other three models. The evolution of the eigenvalue magnitudes allows to identified the improvement on the initial guess while the normalized trace of the Hessian matrices allows to see the improvement on the eigenvalue magnitudes.

For this model, the full-offset FWI goes in the wrong direction from the beginning and, the eigenvalue characteristics prove it, because they are getting far from their theoretical behavior with each FWI iteration step in all three frequencies. Figure 5.30a illustrates that the normalized trace has an oscillatory behavior in the first 12 iterations and from the iteration 29 to 40 for the three frequencies. From iteration 13 to iteration 28 the normalized trace loses quality on the eigenvalue characteristics at 3 Hz but at 6 Hz and 9 Hz there is a slight improvement. In the last 8 iterations the normalized trace has a stable behavior on the eigenvalue characteristics. Figure 5.30b, Figure 5.30c and, Figure 5.30d show the details on the eigenvalues evolution but none of them exhibits an important improvement.

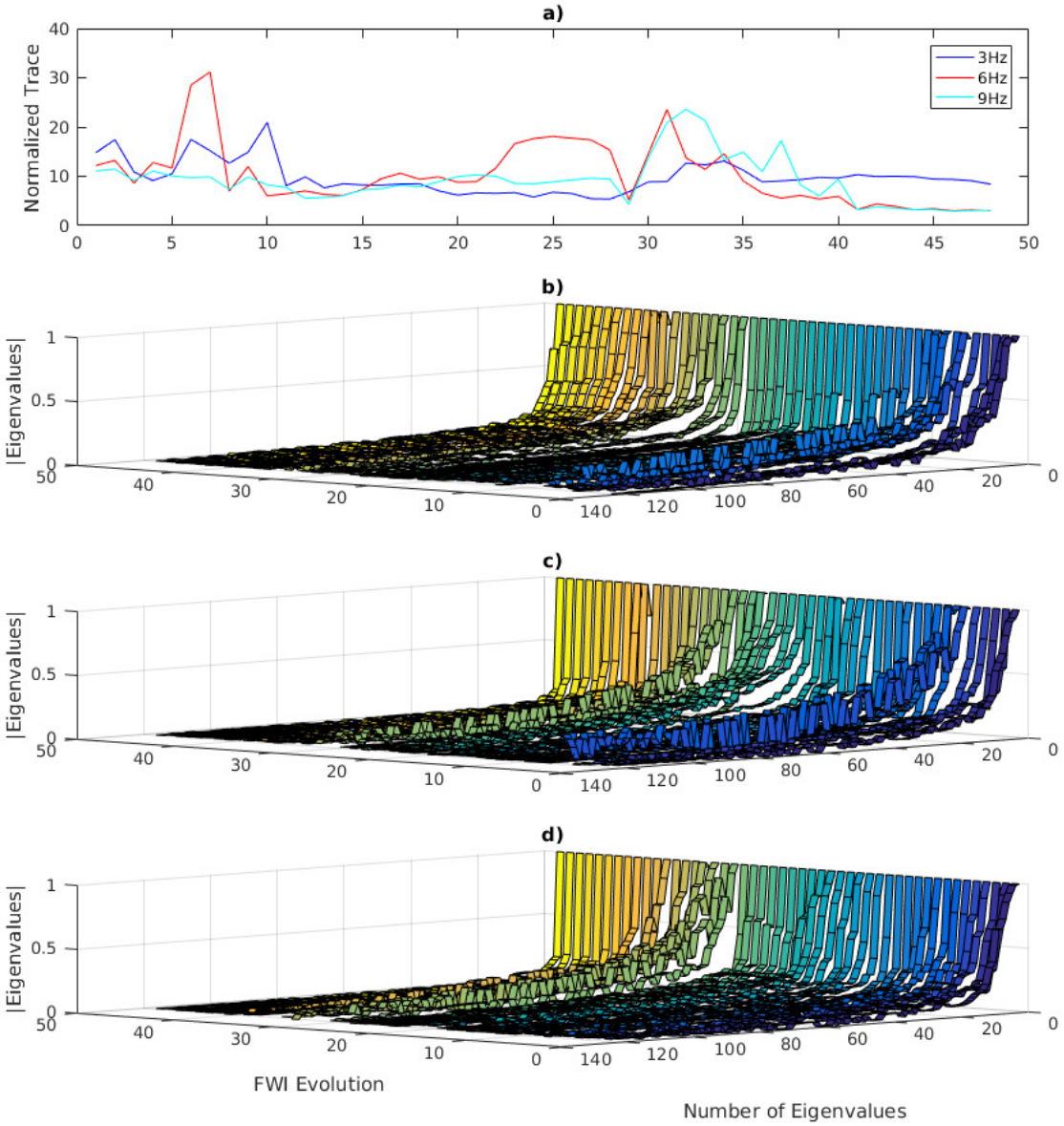


Figure 5.30 Hessian matrix evolution over 48 far-offset FWI's iterations for the squared area of Figure 5.29: a) Normalized trace for 3 Hz, 6 Hz, and 9 Hz; Eigenvalues magnitude evolution for b) 3 Hz, c) 6 Hz and d) 9 Hz.

The short-far-offset FWI loses quality on the eigenvalue characteristics in the first 12 iterations therefore the normalized trace magnitude decreases with each iteration step (Figure 5.31a). However in the range from iteration 13 to 40 the process has an oscillatory behavior with an improvement in the eigenvalue characteristics. From iteration 41 to iteration 53 the eigenvalue quality decreases only for 6 and 9 Hz. And in the last 26 iterations there is always an improvement on the eigenvalue quality. Figure 5.31b, Figure 5.31c and Figure 5.31d show the details on

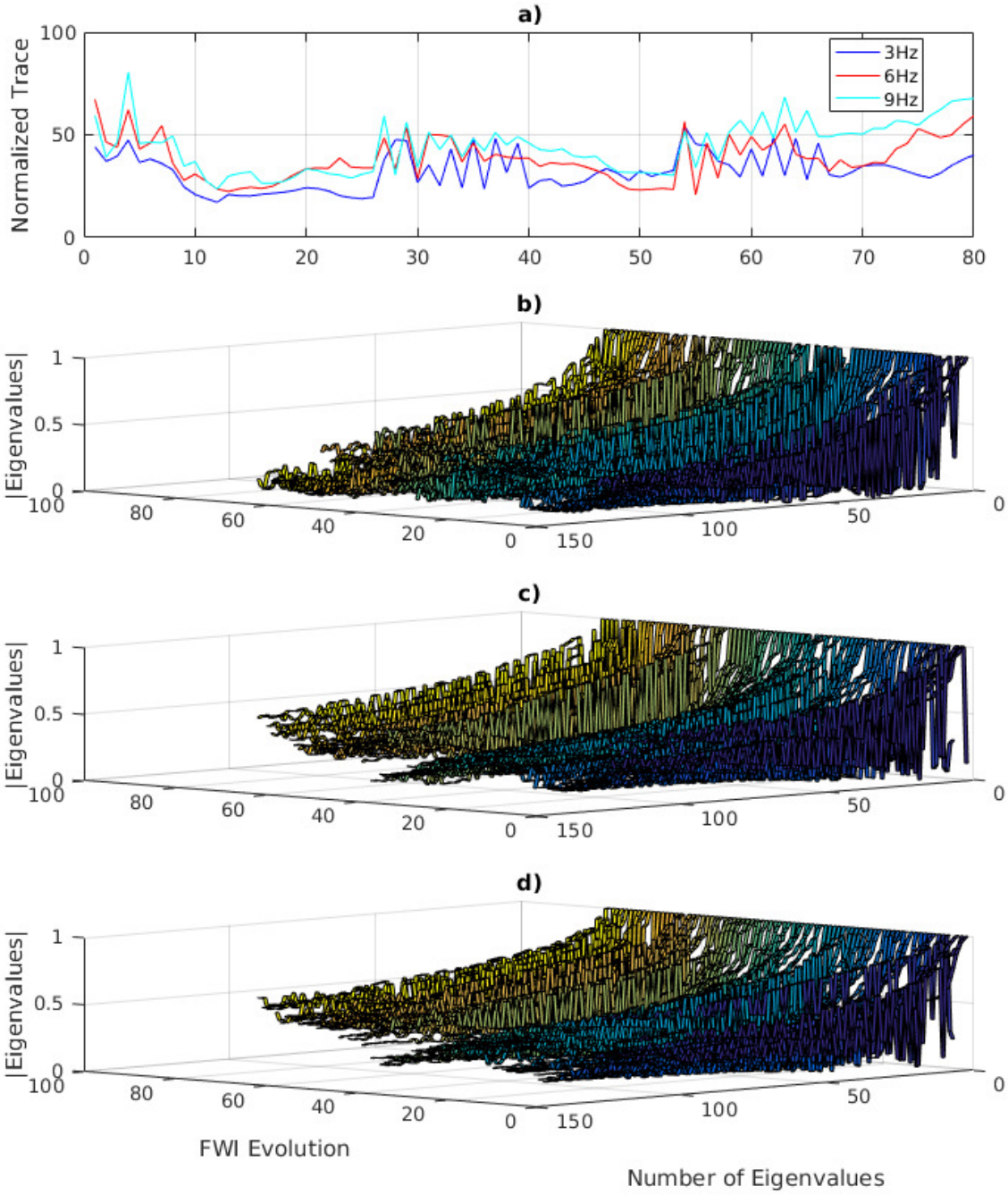


Figure 5.31 Hessian matrix evolution over 80 short-far-offset FWI's iterations for the squared area of Figure 5.29: a) Normalized trace for 3 Hz, 6 Hz, and 9 Hz; Eigenvalues magnitude evolution for b) 3 Hz, c) 6 Hz and d) 9 Hz.

the eigenvalue evolution and according to the results at 6 Hz and at 9 Hz the estimated velocity model reaches an important improvement inside the yellow squared area of Figure 5.29.

5.5.1 Final Comments

In this experiment, the PSO cost function exploration is performed using 500 particles, 10 experiments, 10 sources and 261 receivers. The initial guess for the FWI is obtained after running the PSO exploration during three weeks using the GPU cluster of the CPS research group. As in the others experiments, the PSO result have not any characteristic of the original velocity model (Figure 5.25). Instead, it only has an intermediate velocity background with the low frequency information needed by the FWI.

Two different strategies are applied to perform the FWI: the first one directly uses the full-offset multi-scale FWI over the PSO result (Figure 5.26c); and the second one first run a short-offset multi-scale FWI without depth events over the PSO result and then a far-offset multi-scale FWI is applied over the short-offset result (Figure 5.26d). The best results are obtained from the second strategy because with the short-offset information and without the deepest events the FWI correctly resolves the first layers. And once the first layers are correct then the full-offset and depth events resolve the deepest layers. The first strategy fails because the FWI resolves the background velocity and the high velocity intrusion at the same time using the whole offsets and events. For a FWI of real data that comes from an unknown area, it is highly recommended to perform a finer management of seismic events and offsets to avoid get trapped in a local minimum as a consequence of the presence of high velocity intrusions.

The instantaneous phase difference between the modeled data and the observed data using the Hilbert transform and the Hessian matrix evolution measured on the estimated velocity models through its normalized trace have proved to be successful identifying the correct evolution of the FWI. In the first strategy both QCs show that there is not an improvement over the estimated velocity model after the inversion. In fact, the Hessian matrix evolution shows a decay of quality on the eigenvalue characteristics during the process, while in the second strategy either CS on the shots or the Hessian matrix evolution show an improvement on the velocity estimation specifically at 3 Hz.

5.6 Discussion

The proposed methodology was tested using four synthetic velocity models with complex structures but with different types of velocity contrast. For example, the Marmousi velocity model can be described as a smooth velocity model because its velocity variation between two adjacent layers is not greater than 1000 m/s. But the Canadian foothills, the Hess and the BP velocity models can be classified as rough models because all of them have at least one area with a strong velocity contrast (greater than 1000 m/s).

The complex rough models can be more accurately classified according to the area of the strong velocity variation (svv_{area}). For example, the Canadian foothills can be identified as a model with a small size velocity contrast ($svv_{area} = 6\%$), while the Hess and the BP can be classified as medium size velocity contrast models with $svv_{area} = 22\%$ and $svv_{area} = 21\%$, respectively.

From these experiments, the proposed methodology reached the estimated velocity model closest to the real velocity values when the full-offset strategy is applied over complex smooth velocity models or over complex velocity models with strong velocity variations on small areas. And the best result for the medium size strong velocity variations cases are reached after applied a short-far-offset strategy together with a mute process; where the short-offset mute keeps only the diving and refracted waves, while the full-offset mute only removes the Ormsby filter effects. Given the evidences, a multi-scale offset-mute strategy that has a *surgical* management of the seismic events could produce better FWI results.

The instantaneous phase difference measurement through the Hilbert transform and the eigenvalue characteristics of Hessian matrix evolution have proved to be successful QCs to identify the correct FWI evolution on synthetic data. Therefore, these two QCs could be used as complementary mechanisms to the traditional QCs used on real data (e.g. cross correlation

analysis) and be tested on real scenarios to validate their adequate performances.

If there are two possible computational scenarios to perform the PSO exploration: one of them with a big cluster compound of several nodes but with a low computational capacity per node; and the other one with a small cluster with a few number of nodes but with a high computational capacity per node then, the strategy of increase the number of experiments but reducing the number of sources (e.g. the Marmousi PSO exploration) would be the more suitable for the big cluster, while the strategy of reduce the number of experiments but increasing the number of sources (e.g. the Canadian foothills PSO exploration) should be the indicated for the small cluster.

A final aspect to be discussed is the use of the Hessian matrix for uncertainty quantification (UQ). Unfortunately for all the four experiments the computed Hessian matrix never reaches its theoretical characteristics (see first row on Table 5.2). For all the cases, a condition number close to one and symmetry were never reached, probably as a consequence of: the lack of illumination, the resolution of the estimated model, the source used to compute the Hessian matrix and the numerical error. Therefore, it was not possible to apply Equation 4.41 or Equation 4.44 for UQ. A further discussion about this topic is addressed in Chapter 6.

6. Conclusions and Future work

Throughout this thesis, a new methodology has been proposed to increase the reliability in the results of a geophysical inversion and, numerical strategies to decrease the computational cost of the proposed methodology. This methodology introduces a novelty interaction between optimization techniques and geophysical concepts to increase the reliability on the nonlinear inversion. The results obtained in this work allow to collect conclusions and open new challenging problems that should be further explored for each step of the proposed methodology.

6.1 PSO Exploration

The methodology proposed in this thesis uses the meta-heuristic nature of PSO to explore the cost function of the nonlinear geophysical inversion problem and find a velocity model (initial guess) that belongs to the global minimum neighborhood (i.e., the real velocity model). This initial guess must achieve the low cycle-skipping (CS) requirement of the FWI to avoid get trapped in a local minimum. For all the tests, PSO is configured to perform a low frequency exploration over the cost function that allows to find a velocity model with a low CS level. Therefore a low frequency source, which in this case was 3 Hz, is used together with the best metric, the Hilbert transform, to reach the FWI requirements. In the four experiments presented in Chapter 5, PSO always found a low frequency velocity model with a CS level lower than 18%. In other words, PSO always resolved approximately the 82% of the CS problem leaving the remaining 18% to the FWI.

The set of parameters defined in Chapter 2 are used for all the PSO explorations performed in this thesis (see Table 5.1, Table 5.3, Table 5.4 and Table 5.5). However, a deeper study fo-

cused on new sets of parameters interacting with different types of mute and offsets should be done to exploit the maximal PSO potentiality on this geophysical context. In fact, new metrics, meta-heuristics, or combinations of both can be formulated to find new insights about the cost function behavior on nonlinear inversion problems.

In this thesis the particles movement, the modeling of the wave equation and the filtering module are implemented inside the GPU Tesla k40c; however, the Hilbert transform and the PSO algorithm use a CPU Intel(R) Xeon(R) E5-2620. In the four experiments, the PSO exploration spent between 87% and 92% of the time of the whole process. Therefore, an alternative to reduce the computational cost is to move the Hilbert transform from the CPU to the GPU and use MPI for the CPU execution of the PSO algorithm to exploit the maximal capacity of the cluster.

Two different computational scenarios of the PSO exploration have been used on the Marmousi and the Canadian foothills velocity models to analyze the relationship between the number of experiments and the number of sources. The first scenario focused on many nodes with low computational capacity (more experiments with less sources); and the other scenario with few nodes of high computational capacity (less experiments with more sources). In both cases, the PSO exploration was successful in reaching a velocity model with a low CS level because a poor lighting of the model was compensated with more exploration time. However, it is recommended to analyze more relationships as: the number of particles versus dimensions; and the order of the wave equation operator versus the time step used during the modeling, to identify new ways to reduce the computational cost and to make feasible the use of PSO on real data (see Table 2.4).

6.2 FWI

For all the tests presented in this thesis, the FWI uses: linear interpolation in the first iteration, L-BFGS for the next iterations, a multi-scale approach to resolve the details from low to high frequency, and different strategies to compute the gradient (see chapter 3 for the details). The

RAM requirements are between 1.4% and 27.5% with a computational cost between 0.14% and 0.88% of the whole process.

The full-offset inversions using all the seismic events yield good results for the Marmousi and Canadian foothills velocity models because they are smooth models with a $svv_{area} \leq 6\%$. But for the Hess and BP velocity models, where the $svv_{area} > 20\%$, a short-far-offset strategy together with a muting process were necessary to reach better estimated velocity models.

From the experiments results, it is concluded that besides the importance of an initial guess with enough low frequency information, the mute process definitely do play a key role during the inversion process. In this case, the muting applied keeps only the diving and refracted waves. It is highly recommended to design an offset-mute strategy that has a surgical management of the seismic events, and allows to perform an inversion from the shallow to deepest layers avoiding get trapped in a local minimum.

Nowadays, it is possible to perform a 2D FWI on real data using clusters of GPUs when the gradient computation uses a No-volume allocation strategy (see section 2.3.3 or Noriega et al. (2017)). But a 3D FWI on real data using clusters of GPUs is still far to be feasible due to the RAM requirements. In fact, today it is possible to perform a 3D FWI in GPUs using only small synthetic datasets (Abreo, Abreo & Ramirez (2017)).

In real cases, more parameters must be taken into account to avoid get trapped in a local minimum during an FWI. For example: the source wavelet must be carefully estimated to reproduce the observed data; the selection of a wave equation operator that includes more physical events is also an important issue that must be discussed because the greater the number of physical events, the greater the computational requirements; the noise on the observed data is another big topic because the data must be filtered without eliminating the low frequency information needed by the FWI; and finally the data management is another concern because

for 2D or 3D real acquisitions it is necessary to store and move hundreds of Gigabytes several times.

6.3 Uncertainty Quantification

For uncertainty quantification (UQ) on nonlinear inversions, it is necessary to be able to compute QCs during the whole process that allows obtaining a numeric value of the improvement. For example, the cross correlation between the observed data and the modeled data is one of the QCs used in the Oil and Gas industry that is affected by the data amplitudes. In this thesis, the methodology proposes the use of two novelty QCs that can work together with traditional methods.

On the one hand, the instantaneous phase difference between the observed data and the modeled data allows to measure the CS using the Hilbert transform but without been affected by the data amplitudes (this is a big issue that must be handled carefully with the cross correlation). On the other hand, the eigenvalue characteristics of the Hessian matrices computed over the FWI results allow to measure the reconstruction quality. Therefore, both physical concepts can be used as complementary QCs to measure the improvement: indirectly using the data or directly using the estimated velocities.

From the experiment results, the first QC shows at least a 99.5% of CS reduction in the inversion of the Marmousi and Canadian foothills velocity models using the full-offset strategy and, for the Hess and BP velocity inversions, the greatest improvement is a 91% of CS reduction at 3 Hz using the short-far-offset strategy together with a muting process. A further study must be done about the relationship between the CS percentage and the real improvement over the estimated velocity model. The second QC is applied over a squared section of 121 pixels (see Figure 5.7b, Figure 5.14, Figure 5.21 and Figure 5.29) located inside the lighting area. For the Marmousi and Canadian foothills inversions, the normalized trace evolution always shows an improvement on the area under the curve. However, for the Hess and BP inversions there is an erratic behavior with the full-offset inversion and only a slight improvement of the area

under the curve with the short-far-offset strategy. Both QCs show the best results for: the Hess and BP inversions when the short-far-offset strategy is applied and; the Marmousi and Canadian foothills inversions when the full-offset strategy is applied, presenting the same sensibility to identify if the process is going in the wrong way.

The Hessian matrix QC is affected by: the illumination area produced by sources and receivers; the source frequency used for the Hessian kernels computation; the resolution of the estimated velocity model; the numerical error of the implementation (see Figure 5.9, Figure 5.15, Figure 5.22, Figure 5.23, Figure 5.30, and Figure 5.31 from b to d); and it spent between 7% and 13% of the whole computational process using a maximum GPU RAM of 63.3% for a small area of 121 pixels in the four experiments. Therefore, while the use of the instantaneous phase difference QC over real data is straightforward, a depth study must be done about the proportion in which the above parameters affect the Hessian matrix QC and computational strategies to reduce the RAM requirements before it can be correctly applied on real data.

References

Abreo, D. L., Abreo, S. A., & Ramirez, A. B. (2017). A hybrid methodology for a 3d full waveform inversion in time domain using gpus. In *15th International Congress of the Brazilian Geophysical Society & EXPOGEF, Rio de Janeiro, Brazil, 31 July–03 August 2017*. Society of Exploration Geophysicists and Brazilian Geophysical Society.

Abreo, S., Ramirez, A., Abreo, D., Reyes, O., & Gonzales, H. (2015). A practical implementation of acoustic full waveform inversion on graphical processing units. *Ciencia, Tecnología y Futuro*, *6*(2), 5–16.

Alford, R., Kelly, K., & Boore, D. M. (1974). Accuracy of finite-difference modeling of the acoustic wave equation. *Geophysics*, *39*(6), 834–842.

Anderson, J. E., Tan, L., & Wang, D. (2012). Time-reversal checkpointing methods for rtm and fwi. *Geophysics*.

Aster, R., Borchers, B., & Thurber, C. (2005). *Parameter estimation and inverse problems. 2005*. Elsevier Academic Press: Burlington.

Beskos, A. & Stuart, A. (2009). Mcmc methods for sampling function space. In *ICIAM 07—6th International Congress on Industrial and Applied Mathematics*, (pp. 337–364).

Beydoun, W. B. & Alkhalifah, T. (2015). Full-waveform inversion: Filling the gaps. *The Leading Edge*, *34*(9), 1106–1107.

Billette, F. & Brandsberg-Dahl, S. (2005). The 2004 bp velocity benchmark. In *67th EAGE Conference & Exhibition*.

Bunks, C., Saleck, F. M., Zaleski, S., & Chavent, G. (1995). Multiscale seismic waveform inversion. *Geophysics*, *60*(5), 1457–1473.

Carlisle, A. & Dozier, G. (2001). An off-the-shelf pso. In *Proceeding of Workshop on Particle Swarm Optimization*.

Cichocki, A., Rutkowski, T., Barros, A. K., & Oh, S.-H. (2000). A blind extraction of temporally correlated but statistically dependent acoustic signals. In *Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, volume 1, (pp. 455–464). IEEE.

Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE.

Dos Santos, A. W. G. & Pestana, R. C. (2015). Time-domain multiscale full-waveform inversion using the rapid expansion method and efficient step-length estimation. *Geophysics*, *80*(4), R203–R216.

Fichtner, A. (2010). *Full seismic waveform modelling and inversion*. Springer Science & Business Media.

Fichtner, A. & Trampert, J. (2011). Hessian kernels of seismic data functionals based upon adjoint techniques. *Geophysical Journal International*, *185*(2), 775–798.

Gray, S. H. & Marfurt, K. J. (1995). Migration from topography: Improving the near-surface image. *Can. J. Expl. Geophys.*, *31*(1/2), 18–24.

Griewank, A. & Walther, A. (2000). Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software (TOMS)*, *26*(1), 19–45.

James, K. & Russell, E. (1995). Particle swarm optimization. In *Proceedings of 1995 IEEE International Conference on Neural Networks*, (pp. 1942–1948).

Kalmikov, A. G. & Heimbach, P. (2014). A hessian-based method for uncertainty quantification in global ocean state estimation. *SIAM Journal on Scientific Computing*, *36*(5), S267–S295.

Liu, D. C. & Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, *45*(1-3), 503–528.

Lopes, R. H., Reid, I., & Hobson, P. R. (2007). The two-dimensional kolmogorov-smirnov test.

Métivier, L., Brossier, R., Operto, S., Virieux, J., et al. (2012). Second-order adjoint state methods for full waveform inversion. In *EAGE 2012-74th European Association of Geoscientists and Engineers Conference and Exhibition*.

Métivier, L., Brossier, R., Virieux, J., & Operto, S. (2013). Full waveform inversion and the truncated newton method. *SIAM Journal on Scientific Computing*, *35*(2), B401–B437.

Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, *35*(151), 773–782.

Nocedal, J. & Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

Noriega, R. F., Ramirez, A. B., Abreo, S. A., & Arce, G. R. (2017). Implementation strategies of the seismic full waveform inversion. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, (pp. 1567–1571). IEEE.

Paganoni, A. M., Secchi, P., et al. (2015). *Advances in Complex Data Modeling and Computational Methods in Statistics*. Springer.

Pasalic, D. & McGarry, R. (2010). Convolutional perfectly matched layer for isotropic and anisotropic acoustic wave equations. In *2010 SEG Annual Meeting*. Denver, Colorado, SEG-2010-2925.

Plessix, R. E. (2006). A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophys. J. Int.*, *167*(2), 495–503.

Rao, S. S. (2009). *Engineering Optimization: Theory and Practice, 4th Edition*. Wiley-Interscience.

Sercel (2016). Marine and land seismic acquisition. http://www.sercel.com/about/Pages/what-is-geophysics.aspx. Accessed: 2016-11-04.

Shaw, R. & Srivastava, S. (2007). Particle swarm optimization: A new tool to invert geophysical data. *Geophysics*, *72*(2), F75–F83.

Shi, Y. & Eberhart, R. C. (1998). Parameter selection in particle swarm optimization. In *International Conference on Evolutionary Programming*, (pp. 591–600). Springer.

Snieder, R. (1998). The role of nonlinearity in inverse problems. *Inverse Problems*, *14*(3), 387.

Taherkhani, M. & Safabakhsh, R. (2016). A novel stability-based adaptive inertia weight for particle swarm optimization. *Applied Soft Computing*, *38*, 281–295.

Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*. siam.

Versteeg, R. (1994). The marmousi experience: Velocity model determination on a synthetic complex data set. *The Leading Edge*, *13*(9), 927–936.

Virieux, J. & Operto, S. (2009). An overview of full-waveform inversion in exploration geophysics. *Geophysics*, *74*(6), 1–26.

Wilcox, R. (2005). Kolmogorov–smirnov test. *Encyclopedia of biostatistics*.

Wu, Z. & Alkhalifah, T. (2016). The optimized gradient method for full waveform inversion and its spectral implementation. *Geophysical Journal International*, *205*(3), 1823–1831.

Yang, P., Brossier, R., Metivier, L., & Virieux, J. (2016). Checkpointing-assisted reverse-forward simulation: An optimal recomputation method for fwi and rtm. In *SEG Technical Program Expanded Abstracts 2016* (pp. 1089–1093). Society of Exploration Geophysicists.

Yang, P., Gao, J., & Wang, B. (2015). A graphics processing unit implementation of time-domain full-waveform inversion. *Geophysics*, *80*(3), F31–F39.

Zhang, X., Song, L., Gretton, A., & Smola, A. J. (2009). Kernel measures of independence for non-iid data. In *Advances in neural information processing systems*, (pp. 1937–1944).

Zhun, H., Li, S., Fomel, S., Stadler, G., & Ghattas, O. (2016). A bayesian approach to estimate uncertainty for full-waveform inversion using a priori information from depth migration. *Geophysics*, *81*(5), 307–323.

A. Stability and Numerical Dispersion

Let's consider a plane wave propagating through a bi-dimensional space in the following way

$$U_{l,m}^{n} = U_0 e^{j(n\omega\Delta t - lk_x\Delta x - mk_z\Delta z)} \tag{A.1}$$

where $j = \sqrt{(}-1)$, ω is the angular frequency, k_x, k_z are the components of the wave-number-vector $\overrightarrow{k}$ in the directions x and z, Δx and Δz are the spacial resolution, Δt is the time step and, n, l, m are the positions of the discrete implementation.

The second order stencil in time and eighth order stencil in space of Finite Difference for the 2 D wave equation is

$$\begin{aligned} U_l^{n+1} - 2U_l^n + U_l^{n-1} = \frac{v^2\Delta t^2}{\Delta x^2}[&\frac{-1}{560}U_{l-4,m}^n + \frac{8}{315}U_{l-3,m}^n - \frac{1}{5}U_{l-2,m}^n + \frac{8}{5}U_{l-1,m}^n \\ -\frac{205}{72}U_{l,m}^n + \frac{8}{5}U_{l+1,m}^n - \frac{1}{5}U_{l+2,m}^n + \frac{8}{315}U_{l+3,m}^n - \frac{1}{560}U_{l+4,m}^n &- \frac{1}{560}U_{l,m-4}^n + \frac{8}{315}U_{l,m-3}^n \\ -\frac{1}{5}U_{l,m-2}^n + \frac{8}{5}U_{l,m-1}^n - \frac{205}{72}U_{l,m}^n + \frac{8}{5}U_{l,m+1}^n - \frac{1}{5}U_{l,m+2}^n + \frac{8}{315}U_{l,m+3}^n &- \frac{1}{560}U_{l,m+4}^n] \end{aligned} \tag{A.2}$$

where l is used to represent the spatial discretization in the x direction and, m in the z direction and n is used for the time discretization. If Equation A.1 is replaced in Equation A.2 we obtain

$$\begin{aligned} (e^{i(n\omega\Delta t - lk_x\Delta x - mk_z\Delta z)})(e^{i(\omega\Delta t)} - 2 + e^{i(-\omega\Delta t)}) = \frac{v^2\Delta t^2}{\Delta x^2}(e^{i(n\omega\Delta t - lk_x\Delta x - mk_z\Delta z)})[&\frac{-1}{560}e^{i4k_x\Delta x} \\ +\frac{8}{315}e^{i3k_x\Delta x} - \frac{1}{5}e^{i2k_x\Delta x} + \frac{8}{5}e^{ik_x\Delta x} - \frac{205}{72} + \frac{8}{5}e^{-ik_x\Delta x} - \frac{1}{5}e^{-i2k_x\Delta x} + \frac{8}{315}e^{-i3k_x\Delta x} &- \frac{1}{560}e^{-i4k_x\Delta x} \\ -\frac{1}{560}e^{i4k_z\Delta z} + \frac{8}{315}e^{i3k_z\Delta z} - \frac{1}{5}e^{i2k_z\Delta z} + \frac{8}{5}e^{ik_z\Delta z} - \frac{205}{72} + \frac{8}{5}e^{-ik_z\Delta z} - \frac{1}{5}e^{-i2k_z\Delta z} + \frac{8}{315}e^{-i3k_z\Delta z} & \\ -\frac{1}{560}e^{-i4k_z\Delta z}] & \end{aligned} \tag{A.3}$$

simplifying $e^{i(n\omega\Delta t - lk_x\Delta x - mk_z\Delta z)}$ in both sides we get

$$(e^{i(\omega\Delta t)} - 2 + e^{i(-\omega\Delta t)}) = \frac{v^2\Delta t^2}{\Delta x^2}[\frac{-1}{560}e^{i4k_x\Delta x} + \frac{8}{315}e^{i3k_x\Delta x} - \frac{1}{5}e^{i2k_x\Delta x} + \frac{8}{5}e^{ik_x\Delta x} - \frac{205}{72} + \frac{8}{5}e^{-ik_x\Delta x} - \frac{1}{5}e^{-i2k_x\Delta x} + \frac{8}{315}e^{-i3k_x\Delta x} - \frac{1}{560}e^{-i4k_x\Delta x} - \frac{1}{560}e^{i4k_z\Delta z} + \frac{8}{315}e^{i3k_z\Delta z} - \frac{1}{5}e^{i2k_z\Delta z} + \frac{8}{5}e^{ik_z\Delta z} - \frac{205}{72} + \frac{8}{5}e^{-ik_z\Delta z} - \frac{1}{5}e^{-i2k_z\Delta z} + \frac{8}{315}e^{-i3k_z\Delta z} - \frac{1}{560}e^{-i4k_z\Delta z}] \tag{A.4}$$

Remembering that $e^{ix} + e^{-ix} = 2\cos(x)$ we can rewrite Equation A.4 as

$$(2\cos(\omega\Delta t) - 2) = \frac{v^2\Delta t^2}{\Delta x^2}[\frac{-1}{560}(2\cos(4k_x\Delta x)) + \frac{8}{315}(2\cos(3k_x\Delta x)) - \frac{1}{5}(2\cos(2k_x\Delta x)) + \frac{8}{5}(2\cos(k_x\Delta x)) - \frac{205}{72} - \frac{1}{560}(2\cos(4k_z\Delta z)) + \frac{8}{315}(2\cos(3k_z\Delta z)) - \frac{1}{5}(2\cos(2k_z\Delta z)) + \frac{8}{5}(2\cos(k_z\Delta z)) - \frac{205}{72}] \tag{A.5}$$

Now remembering that $\cos(2\theta) = 1 - 2\sin^2(\theta)$ we can rewrite Equation A.5 as

$$(2(1 - 2\sin^2(\omega\Delta t/2)) - 2) = \frac{v^2\Delta t^2}{\Delta x^2}[\frac{-1}{560}(2(1 - 2\sin^2(2k_x\Delta x))) + \frac{8}{315}(2(1 - 2\sin^2(3k_x\Delta x/2))) - \frac{1}{5}(2(1 - 2\sin^2(k_x\Delta x))) + \frac{8}{5}(2(1 - 2\sin^2(k_x\Delta x/2))) - \frac{410}{72} - \frac{1}{560}(2(1 - 2\sin^2(2k_z\Delta z))) + \frac{8}{315}(2(1 - 2\sin^2(3k_z\Delta z/2))) - \frac{1}{5}(2(1 - 2\sin^2(k_z\Delta z))) + \frac{8}{5}(2(1 - 2\sin^2(k_z\Delta z/2)))] \tag{A.6}$$

reducing some terms we get

$$-4\sin^2(\omega\Delta t/2)) = \frac{v^2\Delta t^2}{\Delta x^2}[\frac{4}{560}\sin^2(2k_x\Delta x) - \frac{32}{315}\sin^2(3k_x\Delta x/2) + \frac{4}{5}\sin^2(k_x\Delta x) - \frac{32}{5}\sin^2(k_x\Delta x/2) + \frac{4}{560}\sin^2(2k_z\Delta z) - \frac{32}{315}\sin^2(3k_z\Delta z/2) + \frac{4}{5}\sin^2(k_z\Delta z) - \frac{32}{5}\sin^2(k_z\Delta z/2)] \tag{A.7}$$

reducing more we get

$$\sin^2(\omega\Delta t/2)) = \frac{v^2\Delta t^2}{\Delta x^2}[-\frac{1}{560}\sin^2(2k_x\Delta x) + \frac{8}{315}\sin^2(3k_x\Delta x/2) - \frac{1}{5}\sin^2(k_x\Delta x) + \frac{8}{5}\sin^2(k_x\Delta x/2) - \frac{1}{560}\sin^2(2k_z\Delta z) + \frac{8}{315}\sin^2(3k_z\Delta z/2) - \frac{1}{5}\sin^2(k_z\Delta z) + \frac{8}{5}\sin^2(k_z\Delta z/2)] \tag{A.8}$$

clearing ω from Equation A.8

$$\omega = \frac{2}{\Delta t}\sin^{-1}[\alpha\sqrt{\frac{-\frac{1}{560}\sin^2(2k_x\Delta x) + \frac{8}{315}\sin^2(3k_x\Delta x/2) - \frac{1}{5}\sin^2(k_x\Delta x) + \frac{8}{5}\sin^2(k_x\Delta x/2)}{-\frac{1}{560}\sin^2(2k_z\Delta z) + \frac{8}{315}\sin^2(3k_z\Delta z/2) - \frac{1}{5}\sin^2(k_z\Delta z) + \frac{8}{5}\sin^2(k_z\Delta z/2)}}] \tag{A.9}$$

where $\alpha = \frac{v\Delta t}{\Delta x}$. If $k_x = k\cos(\theta)$, $k_z = k\sin(\theta)$, $k = \sqrt{k_x^2 + k_z^2}$ and $\Delta x = \Delta z = \Delta h$, we can rewrite Equation A.9 as

$$\omega = \frac{2}{\Delta t}\sin^{-1}[\alpha\sqrt{\frac{-\frac{1}{560}\sin^2(2k\cos(\theta)\Delta h) + \frac{8}{315}\sin^2(3k\cos(\theta)\Delta h/2) - \frac{1}{5}\sin^2(k\cos(\theta)\Delta h) + \frac{8}{5}\sin^2(k\cos(\theta)\Delta h/2) - \frac{1}{560}\frac{\sin^2(2k\sin(\theta)\Delta h) + \frac{8}{315}\sin^2(3k\sin(\theta)\Delta h/2)}{-\frac{1}{5}\sin^2(k\sin(\theta)\Delta h) + \frac{8}{5}\sin^2(k\sin(\theta)\Delta h/2)}}}] \tag{A.10}$$

the stability condition is obtained when $\sin()^2$ functions gets their maximum value, that is

$$\omega = \frac{2}{\Delta t}\sin^{-1}[\alpha\sqrt{3.25}] \tag{A.11}$$

which means $\alpha\sqrt{3.25} \leq 1$ to keep a real wave field. Clearing α we get

$$\alpha \leq \frac{1}{\sqrt{3.25}} = 0.55 \tag{A.12}$$

There is not numerical dispersion when $\frac{v_p}{v} = 1$. It must be remembered that $v_p = \frac{\omega}{k}$. For the second order stencil in time and eighth order stencil in space we get

$$\frac{v_p}{v} = \frac{\omega}{kv} = \frac{2}{kv\Delta t}\sin^{-1}[\alpha\sqrt{-\frac{1}{560}\sin^2(2k\cos(\theta)\Delta h) + \frac{8}{315}\sin^2(3k\cos(\theta)\Delta h/2)}$$
$$\overline{-\frac{1}{5}\sin^2(k\cos(\theta)\Delta h) + \frac{8}{5}\sin^2(k\cos(\theta)\Delta h/2) - \frac{1}{560}\overline{\sin^2(2k\sin(\theta)\Delta h) + \frac{8}{315}\sin^2(3k\sin(\theta)\Delta h/2)}}$$
$$\overline{-\frac{1}{5}\sin^2(k\sin(\theta)\Delta h) + \frac{8}{5}\sin^2(k\sin(\theta)\Delta h/2)]}$$

(A.13)

and remembering that $\alpha = \frac{v\Delta t}{\Delta h}$ we can rewrite Equation A.13 as

$$\frac{v_p}{v} = \frac{2}{\alpha k\Delta h}\sin^{-1}[\alpha\sqrt{-\frac{1}{560}\sin^2(2\cos(\theta)k\Delta h) + \frac{8}{315}\sin^2(3\cos(\theta)k\Delta h/2)}$$
$$\overline{-\frac{1}{5}\sin^2(\cos(\theta)k\Delta h) + \frac{8}{5}\sin^2(\cos(\theta)k\Delta h/2) - \frac{1}{560}\overline{\sin^2(2\sin(\theta)k\Delta h) + \frac{8}{315}\sin^2(3\sin(\theta)k\Delta h/2)}}$$
$$\overline{-\frac{1}{5}\sin^2(\sin(\theta)k\Delta h) + \frac{8}{5}\sin^2(\sin(\theta)k\Delta h/2)]}$$

(A.14)

It must be remembered that $k = \frac{2\pi}{\lambda}$ and $\lambda = \frac{v}{f}$ then $k = \frac{2\pi f}{v}$.

B. FWI implementation using Strategy I

Algorithm 11 FWI in GPU: Two volume allocation

1: $FWI(v, Obs, s, inf, \alpha)$ ▷ FWI inputs
2: v, Obs ▷ Starting velocity model and Observed traces
3: s, α ▷ Source Wavelet and Alpha value
4: inf ▷ Location and offset of each shot
5: **for** $i \leftarrow 1, iG$ **do** ▷ iG, Number of FWI iterations
6: $acum \leftarrow 0$
7: $g(x,z) = 0$
8: **for** $j \leftarrow 1, Ns$ **do** ▷ Ns, Number of sources
9: **for** $t \leftarrow 1, Nt$ **do** ▷ Nt, Number of time steps
10: $\frac{1}{v(x,z)^2}\frac{\partial^2 p_s}{\partial t^2} = \frac{\partial^2 p_s}{\partial x^2} + \frac{\partial^2 p_s}{\partial z^2} + S(sj,t)$
11: $Mod(x,t) \leftarrow p_s(x,0,t)$
12: **end for**
13: $Obs_{dev} \leftarrow Obs_{j,host}$ ▷ Observed traces from host to device
14: $Residual(x,t) \leftarrow Mod(x,t) - Obs_{dev}(x,t)$
15: $norm \leftarrow \|Residual(x,t)\|_2^2$ ▷ (L2 Norm)^2
16: $acum \leftarrow \frac{1}{2} norm + acum$
17: **for** $t \leftarrow 1, Nt$ **do**
18: $\frac{1}{v(x,z)^2}\frac{\partial^2 q_s}{\partial t^2} = \frac{\partial^2 q_s}{\partial x^2} + \frac{\partial^2 q_s}{\partial z^2} - Residual(x, Nt - t)$
19: **end for**
20: $g(x,z) = g(x,z) - \frac{2}{(v^k(\mathbf{x},\mathbf{z}))^3}\int_0^T q_s(x,z,T-t)\frac{\partial^2 p_s(x,z,t)}{\partial t^2}dt$ ▷ Equation 3.3
21: **end for**
22: $\phi(i) \leftarrow acum$
23: **if** $i == 1$ **then**
24: $\alpha \leftarrow Linear\ interpolation$
25: $v(x,z) \leftarrow v(x,z) - \alpha \cdot g(x,z)$
26: **else**
27: $r \leftarrow Algorithm\ 4$
28: $v(x,z) \leftarrow Algorithm\ 5$
29: **end if**