



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Engenharia Elétrica e de Computação

TIAGO TAVARES LEITE BARROS

CONTRIBUTIONS TO THE INITIALIZATION OF SEISMIC SLOPE  
TOMOGRAPHY

CONTRIBUIÇÕES PARA A INICIALIZAÇÃO DA TOMOGRAFIA SÍSMICA  
DE INCLINAÇÃO

CAMPINAS  
2018

TIAGO TAVARES LEITE BARROS

CONTRIBUTIONS TO THE INITIALIZATION OF SEISMIC SLOPE  
TOMOGRAPHY

CONTRIBUIÇÕES PARA A INICIALIZAÇÃO DA TOMOGRAFIA SÍSMICA  
DE INCLINAÇÃO

Thesis presented to the Faculty of Electrical and  
Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering, in the area of Telecommunications and Telematics.

*Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Doutor em Engenharia Elétrica, na Área de Telecomunicações e Telemática.*

*Supervisor/Orientador:* Prof. Dr. Renato da Rocha Lopes

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE DEFENDIDA PELO ALUNO TIAGO TAVARES LEITE BARROS, E ORIENTADA PELO PROF. DR. RENATO DA ROCHA LOPES

CAMPINAS  
2018

**Agência(s) de fomento e número(s) de processo(s):** CNPq, 164764/2013-5

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Luciana Pietrosanto Milla - CRB 8/8129

B278c Barros, Tiago Tavares Leite, 1983-  
Contributions to the initialization of seismic slope tomography / Tiago Tavares Leite Barros. - Campinas, SP : [s.n.], 2018.

Orientador: Renato da Rocha Lopes.  
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Tomografia sísmica. 2. Método sísmico de reflexão. 3. Otimização. 4. Processamento de sinais. 5. Algoritmos evolutivos. I. Lopes, Renato da Rocha, 1972-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Contribuições para a inicialização da tomografia sísmica de inclinação

**Palavras-chave em inglês:**

Seismic tomography

Seismic reflection method

Optimization

Signal processing

Evolutionary algorithms

**Área de concentração:** Telecomunicações e Telemática

**Titulação:** Doutor em Engenharia Elétrica

**Banca Examinadora:**

Renato da Rocha Lopes [Orientador]

Lucio Tunes dos Santos

Leonardo Tomazeli Duarte

Rafael Krummenauer

André Kazuo Takahata

**Data da defesa:** 16-02-2018

**Programa de Pós-Graduação:** Engenharia Elétrica

## **COMISSÃO JULGADORA - TESE DE DOUTORADO**

**Candidato:** Tiago Tavares Leite Barros. RA 036310

**Data da defesa:** 16 de fevereiro de 2018

**Título da tese:** Contribuições para a inicialização da tomografia sísmica de inclinação

Prof. Dr. Renato da Rocha Lopes (Presidente, FEEC/UNICAMP)

Prof. Dr. Lucio Tunes dos Santos (IMECC/UNICAMP)

Prof. Dr. Leonardo Tomazeli Duarte (FCA/UNICAMP)

Prof. Dr. Rafael Krummenauer (UEM)

Prof. Dr. André Kazuo Takahata (UFABC)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no processo de vida acadêmica do aluno.

# Agradecimentos

A realização deste trabalho foi possível graças à colaboração de muitas pessoas, às quais devo mais do que apenas agradecimentos. Como é praticamente impossível citar todas aqui, farei um agradecimento, de forma resumida:

Ao meu orientador, Prof. Dr. Renato da Rocha Lopes, por me guiar nesse caminho sempre com muita disposição e compreensão e por me formar como pesquisador.

Ao Prof. Dr. Hervé Chauris, pela supervisão durante meu período de doutorado sanduíche na École des Mines de Paris. Suas contribuições foram fundamentais para a realização deste trabalho.

À comissão julgadora deste trabalho, Prof. Dr. Lucio Tunes dos Santos, Prof. Dr. Leonardo Tomazeli Duarte, Prof. Dr. Rafael Krummenauer e Prof. Dr. André Kazuo Takahata pelas contribuições.

Aos demais colegas e amigos dos laboratórios de pesquisa DSPCom, na Unicamp, e da École des Mines.

À PETROBRAS e ao CNPq pelo financiamento deste trabalho.

À minha família e aos meus amigos.

# Abstract

Stereotomography is a slope-based tomographic method that uses seismic reflection data to estimate the macro-model of velocities in the subsurface. The velocity model obtained by stereotomography relies on the hypothesis that the subsurface presents an acoustic velocity distribution. This acoustic velocity model can be split into two portions: a velocity macro-model that contains long wavelengths (low-frequency) and velocity perturbations that contain short wavelengths (high-frequency). Therefore, the stereotomography is a tool for estimating the low-frequency macro-model of seismic acoustic velocities in the subsurface. Often, the stereotomography results are employed as initialization in high-frequency methods, such as migration or full-waveform inversion. This is an interesting approach, since several high-frequency methods are modeled as inverse problems, which are mostly solved with the use of local optimization techniques, which are known to be strongly dependent of a good initialization. In that context, we must think of the initialization by the picking of slopes as a crucial part of the stereotomography problem. In this thesis we address the problem of slopes estimation for a better initialization of stereotomography. The central proposal of this work is to use the attributes obtained as outputs of the common-offset common-reflection-surface method as input data for stereotomography. Another proposal of this work is to use the global optimization method known as differential evolution to estimate the common-offset common-reflection-surface attributes. This strategy presents reliable estimates in data sets highly corrupted by noise. We also propose an automatic strategy to extract the slopes used in stereotomography, from the common-offset common-reflection-surface attributes gathers. In this thesis, we discuss the aspects of our stereotomography implementation and we illustrate with synthetic examples the benefits of using the global estimates of common-reflection-surface slopes as stereotomography initialization, for seismic data sets highly corrupted by noise.

**Keywords:** Seismic tomography, seismic reflection method, optimization, signal processing, evolutionary algorithms.

# Resumo

Estéreo-tomografia é um método tomográfico baseado em inclinações que faz uso de reflexões sísmicas para estimar o macro-modelo de velocidades. A obtenção deste modelo de velocidades pela estéreo-tomografia baseia-se na hipótese de que a subsuperfície terrestre apresenta uma distribuição acústica de velocidades. Este modelo de velocidades acústicas pode ser dividido em duas porções: um macro-modelo de velocidades, contendo longos comprimentos de onda (baixas frequências) e perturbações de velocidade, contendo curtos comprimentos de onda (altas frequências). Portanto, a estéreo-tomografia é uma ferramenta adequada para estimar o macro-modelo de baixas frequências de velocidades acústicas na subsuperfície. Frequentemente, os resultados obtidos com a estéreo-tomografia são empregados como inicialização para métodos de alta frequência, como migração e inversão completa de onda. Esta é uma abordagem interessante, uma vez que muitos destes métodos são desenvolvidos sob a ótica de problemas inversos, os quais são em sua maioria resolvidos com o emprego de técnicas de otimização local. Por outro lado, estes métodos de otimização local são fortemente dependentes de uma boa inicialização. Neste contexto, é prudente considerar como crucial a inicialização da estéreo-tomografia, a qual é feita através de técnicas de seleção de inclinações, ou ângulos, extraídas do dado sísmico. Nesta tese, aborda-se o problema de estimação de ângulos para inicializar o método de estéreo-tomografia. A proposta central deste trabalho é a de utilizar os atributos obtidos como saída no método de superfície de reflexão comum para deslocamentos comuns (CO-CRS, do inglês *common-offset common-reflection-surface*) como entrada para a estéreo-tomografia. Outra proposta deste trabalho é a de utilizar o algoritmo de otimização global evolução diferencial (DE, do inglês *differential evolution*) para estimar os atributos do método CO-CRS. Esta estratégia apresenta boas estimativas em dados sísmicos altamente corrompidos por ruído. Também é proposta uma estratégia para seleção automática dos ângulos obtidos com o método CO-CRS. Nesta tese, também são apresentados alguns aspectos práticos da implementação da estéreo-tomografia. As propostas são discutidas e analisadas em experimentos com dados sísmicos sintéticos altamente corrompidos por ruído, onde é possível observar-se os benefícios obtidos com o emprego da seleção automática dos ângulos obtidos com o método CO-CRS na inicialização da estéreo-tomografia.

**Palavras-chave:** Tomografia sísmica, método sísmico de reflexão, otimização, processamento de sinais, algoritmos evolutivos.

# List of Figures

2.1	Illustration of data and model for stereotomography. The model is represented by the vector $\mathbf{m}$ , the observed data is represented by the vector $\mathbf{d}$ and the calculated data is represented by the vector $\mathbf{d}^c$ .	21
3.1	Simplified model for gradient computation.	28
3.2	Velocity model with perfect ray (black) and rays traced with perturbed initial angle (red).	38
3.3	Finite difference test with the cost functions $J_1$ (top left), $J_2$ (top right), $J_3$ (bottom left) and $J_4$ (bottom right).	39
3.4	Velocity model with perfect ray (black) and rays traced with perturbed initial $x$ coordinate (red).	39
3.5	Finite difference test with the cost functions $J_1$ (top left), $J_2$ (top right), $J_3$ (bottom left) and $J_4$ (bottom right).	40
3.6	Velocity model with perfect ray (black) and rays traced with perturbed initial $z$ coordinate (red).	40
3.7	Finite difference test with the cost functions $J_1$ (top left), $J_2$ (top right), $J_3$ (bottom left) and $J_4$ (bottom right).	41
3.8	Velocity model with perfect ray (black) and rays traced with perturbed ray propagation time (red).	41
3.9	Finite difference test with the cost functions $J_1$ (top left), $J_2$ (top right), $J_3$ (middle left) and $J_4$ (middle right) and $J_5$ (bottom left).	42
3.10	Velocity model with perfect ray (black) and rays traced with perturbed B-spline coefficient, away from the source point (red).	42
3.11	Finite difference test with the cost functions $J_1$ (top left), $J_2$ (top right), $J_3$ (bottom left) and $J_4$ (bottom right).	43
3.12	Velocity model with perfect ray (black) and rays traced with perturbed B-spline coefficient, close to the source point (red).	43

3.13	Finite difference test with the cost functions $J_1$ (top left), $J_2$ (top right), $J_3$ (bottom left) and $J_4$ (bottom right). . . . .	44
4.1	Equivalent model. $\theta$ is the specular angle. . . . .	51
4.2	Rays traced in the original synthetic velocity model. . . . .	54
4.3	Difference between the initial and the final velocity models, in the attempt of estimating the perturbation without (a) and with (b) preconditioning. . . . .	54
4.4	Cost function after 50 iterations of the nonlinear conjugate gradient with and without the use of preconditioning. . . . .	55
4.5	Rays traced in the perfect velocity model. . . . .	56
4.6	Original (a) and initial (b) velocity models. Velocity models recovered with the nonlinear conjugate gradient (c) and l-BFGS-b (d) methods. . . . .	56
4.7	Cost function for the nonlinear CG and l-BFGS-b methods. The l-BFGS-b converged in 300 iterations, while the nonlinear CG took 5000 iterations to converge. . . . .	57
5.1	CO-CRS surface defined by its traveltimes. . . . .	60
5.2	CO-CRS sequential search diagram. . . . .	62
5.3	Marmousi data set with offset of $-1000$ m. Raw (a) and noisy (b) data used for the tests. . . . .	68
5.4	Sequential CO-CRS: CMP stacking (a) and coherence (b) gathers. CS stacking (c) and coherence (d) gathers. CO stacking (e) and coherence (f) gathers. . . . .	69
5.5	Sequential CO-CRS final stacked (a) and coherence (b) sections. Global CO-CRS stacked (c) and coherence (d) sections. . . . .	70
5.6	Marmousi CO gather. Raw data (left) enhanced sequential (middle) and global (right) CO-CRS. . . . .	71
5.7	Raw CMP data (a); sequential enhanced CMP data (b); global enhanced CMP data (c). . . . .	72
5.8	Field data: Raw (a) and global enhanced (b) CO gather with $h = -2000$ m. . . . .	73
5.9	Field data: Five raw (a) and global enhanced (b) CMP gathers. . . . .	73
5.10	Field data: Coherence map computed from the raw data set (a) and from the CO-CRS enhanced version of the same data set (b). . . . .	74
6.1	CS gather for $SNR = 5$ and the respective semblance panel. The trace in red corresponds to the reference trace used for the semblance computation. . . . .	79
6.2	CR gather for $SNR = 5$ and the respective semblance panel. The trace in red corresponds to the reference trace used for the semblance computation. . . . .	79
6.3	CO gather for $SNR = 5$ and the respective semblance panel. The trace in red corresponds to the reference trace used for the semblance computation. . . . .	80

6.4	CMP gather for SNR = 5 and the respective semblance panel. The trace in red corresponds to the reference trace used for the semblance computation. . . . .	80
6.5	Values of $\theta$ , $\theta_s$ and $\theta_r$ obtained from local slant stack in data with SNR = 5 (a), 2 (b) and 1 (c). $\theta$ is the specular angle and $\theta_s$ and $\theta_r$ are the angles from the scatterer to source and receiver positions, respectively. . . . .	82
6.6	Coherence gathers obtained in the sequential CO-CRS search for the data with SNR = 5. These gathers come from: CMP search (a); first (b) and second (c) stages of CO search and CS search (d). Final coherence gather obtained with the estimated parameters used in the stacking (e). . . . .	83
6.7	Coherence gathers obtained in CS search and used for the angles pick, in the sequential CO-CRS, for the data with SNR = 5 (a), 2 (b) and 1 (c). . . . .	84
6.8	$\theta_s$ obtained with the sequential CO-CRS for the data with SNR = 5 (a), 2 (b) and 1 (c).	84
6.9	$\theta_r$ obtained with the sequential CO-CRS for the data with SNR = 5 (a), 2 (b) and 1 (c).	85
6.10	Values of $\theta$ , $\theta_s$ and $\theta_r$ obtained from the sequential CO-CRS in data with SNR = 5 (a), 2 (b) and 1 (c). $\theta$ is the specular angle and $\theta_s$ and $\theta_r$ are the angles from the scatterer to source and receiver positions, respectively. . . . .	86
6.11	Coherence gathers obtained with the global CO-CRS, for the data with SNR = 5 (a), 2 (b) and 1 (c). . . . .	87
6.12	$\theta_s$ for the global CO-CRS with SNR = 5 (a), 2 (b) and 1 (c). . . . .	87
6.13	$\theta_r$ for the global CO-CRS with SNR = 5 (a), 2 (b) and 1 (c). . . . .	88
6.14	Values of $\theta$ , $\theta_s$ and $\theta_r$ obtained from the global CO-CRS in data with SNR = 5 (a), 2 (b) and 1 (c). $\theta$ is the specular angle and $\theta_s$ and $\theta_r$ are the angles from the scatterer to source and receiver positions, respectively. . . . .	89
6.15	Equivalent model initialization obtained with picks from the local slant stack in data with SNR = 5 (a), 2 (b) and 1 (c). . . . .	90
6.16	Localization and joint inversion results obtained with picks from local slant stack in data with SNR = 5 (a) and (d), 2 (b) and (e) and 1 (c) and (f). . . . .	91
6.17	Velocity model slice for the picks obtained with local slant stack in data with SNR = 5 (a), 2 (b) and 1 (c). . . . .	92
6.18	Cost functions for the picks obtained with local slant stack in data with SNR = 5 (a), 2 (b) and 1 (c). . . . .	93
6.19	Equivalent model for the picks obtained with sequential CO-CRS in data with SNR = 5 (a) and 1 (b). . . . .	94
6.20	Localization (a) and joint inversion (b) results for the picks obtained with sequential CO-CRS in data with SNR = 5. . . . .	94

6.21	Velocity model slice for the picks obtained with sequential CO-CRS in data with SNR = 5 (a) and 1 (b). . . . .	95
6.22	Cost functions for the picks obtained with sequential CO-CRS in data with SNR = 5 (a) and 1 (b). . . . .	95
6.23	Equivalent model for the picks obtained with global CO-CRS in data with SNR = 5 (a) and 1 (b). . . . .	96
6.24	Localization and joint inversion results for the picks obtained with global CO-CRS in data with SNR = 5 (a) and 1 (b). . . . .	97
6.25	Velocity model slice for the picks obtained with global CO-CRS in data with SNR = 5 (a) and 1 (b). . . . .	98
6.26	Cost functions for the picks obtained with global CO-CRS in data with SNR = 5 (a) and 1 (b). . . . .	98
6.27	Velocity model used to generate the synthetic data. . . . .	100
6.28	Raw CO gather data with SNR = 20 (a) and the same data enhanced with CO-CRS (b). $\theta_s$ (c) and $\theta_r$ (d) estimated attributes. . . . .	102
6.29	Coherence gather superposed by black markers indicating the picking positions (a). Constant gradient $v(z) = 1.5 + z$ Km/s velocity model, used as initialization to the stereotomography, superposed by the dip bars obtained by the equivalent model evaluation in the picked data (b). . . . .	103
6.30	Velocity model obtained in the localization (a) and in each one of the multi-step stages (b)-(f), superposed by the respective equivalent model dip bars. . . . .	105
6.31	True velocity model reconstructed with B-splines coefficients using the same sampling as the one used in the five multi-step inversion stages (a)-(e). . . . .	106
6.32	Cost function for the synthetic data example. The vertical lines correspond to the divisions in the inversion stages. . . . .	107
A.1	Filters for B-Spline coefficients computation. . . . .	119
D.1	DE algorithm. . . . .	129

# Summary

<b>List of Figures</b>	<b>8</b>
<b>1 Introduction</b>	<b>15</b>
1.1 Thesis organization . . . . .	17
1.2 Publications . . . . .	18
1.2.1 Articles published in journals . . . . .	18
1.2.2 Articles published in conferences . . . . .	18
1.2.3 Book chapter . . . . .	19
<b>2 Stereotomography</b>	<b>20</b>
2.1 Stereotomography data and model vectors . . . . .	20
2.2 The direct problem: ray tracing . . . . .	23
2.3 Optimization as a way of solving the inverse problem . . . . .	24
<b>3 Gradient computation with the adjoint-state method</b>	<b>26</b>
3.1 Adjoint-state method . . . . .	26
3.2 Single-ray approximation . . . . .	27
3.2.1 Derivation of the gradient . . . . .	29
3.3 Gradient with two rays . . . . .	34
3.4 Generalization of the gradient . . . . .	35
3.5 Stereotomography gradient verification . . . . .	36
<b>4 Stereotomography implementation</b>	<b>45</b>
4.1 Practical aspects . . . . .	45
4.2 Optimization . . . . .	46
4.2.1 l-BFGS-b implementation . . . . .	47
4.2.2 Preconditioning . . . . .	48
4.3 Regularization . . . . .	49

4.4	Inversion approach . . . . .	50
4.4.1	Initialization: equivalent model . . . . .	50
4.4.2	Localization . . . . .	51
4.4.3	Joint inversion with multi-step optimization . . . . .	51
4.5	Implementation tests . . . . .	52
4.5.1	Constant gradient velocity model . . . . .	52
4.5.2	Perturbation in the velocity model . . . . .	53
<b>5</b>	<b>Common-offset common-reflection-surface</b>	<b>58</b>
5.1	The method . . . . .	59
5.2	Sequential search . . . . .	61
5.2.1	CMP search . . . . .	62
5.2.2	CO search . . . . .	62
5.2.3	CS search . . . . .	63
5.3	Global search . . . . .	63
5.3.1	Differential Evolution . . . . .	64
5.4	Numerical examples . . . . .	65
5.4.1	Synthetic Marmousi data . . . . .	66
5.4.2	Field data . . . . .	67
<b>6</b>	<b>Slopes estimation and picking</b>	<b>75</b>
6.1	Local slant stack . . . . .	75
6.1.1	Stereotomography data picking . . . . .	76
6.1.2	Automatic picking . . . . .	77
6.2	Plane reflector with dip . . . . .	78
6.2.1	Slopes quality analysis . . . . .	78
6.2.2	Inversion comparison . . . . .	85
6.3	Slopes picking with CO-CRS . . . . .	96
6.3.1	Strategy for automatic picking with CO-CRS . . . . .	99
6.3.2	Numerical example: synthetic data . . . . .	99
<b>7</b>	<b>Conclusions</b>	<b>108</b>
	<b>References</b>	<b>110</b>
<b>A</b>	<b>B-Splines</b>	<b>117</b>
A.1	Definition . . . . .	117

A.2	B-splines coefficients . . . . .	118
A.3	2D B-splines . . . . .	119
<b>B</b>	<b>Gradient method</b>	<b>121</b>
B.1	Step size determination . . . . .	122
B.2	Nonlinear conjugate gradient . . . . .	123
<b>C</b>	<b>Broyden-Fletcher-Goldfarb-Shanno method (BFGS)</b>	<b>124</b>
C.1	Limited memory BFGS . . . . .	124
<b>D</b>	<b>Differential evolution (DE)</b>	<b>127</b>

# Chapter 1

## Introduction

Stereotomography is a slope-based tomographic method, proposed in the works of Billette [1998] and Billette and Lambaré [1998], that uses seismic reflection data to estimate the macro-model of velocities in the subsurface. One important improvement in stereotomography, when compared to the travel time tomography [Bishop et al., 1985], is that it uses the slopes information, obtained from locally coherent seismic gathers, in the tomographic approach. This is an interesting approach, since it yields a better-constrained velocity model [Billette, 1998]. Stereotomography is an extension of the controlled directional reception (CDR) tomographic method [Sword, 1987], with some differences, especially regarding the cost function calculation, picking and introduction of the paraxial ray tracing for the calculation of the Fréchet derivatives.

In Billette et al. [2003], some practical aspects regarding the implementation and application of stereotomography were discussed. Later, in Lambaré et al. [2004], some aspects regarding especially the stereotomography picking automation were also discussed. The work of Le Bégat et al. [2004] presents an interesting comparison, in the same framework, between travel time tomography [Bishop et al., 1985, Chiu and Stewart, 1987, Farra and Madariaga, 1988], slope tomography (stereotomography) and migration velocity analysis (MVA) [Chauris et al., 2002]. In Lambaré [2008], the author presents an extensive overview, ten years after the creation of stereotomography. In Alérini et al. [2008], the authors applied a PP/PS version of stereotomography in a 2D ocean bottom cable (OBC) seismic data. The work of Nguyen et al. [2008] investigates how stereotomography would be influenced by making the picking after the data migration, in depth domain. In Costa et al. [2008], several regularization functions are investigated for the stereotomography. Another interesting comparison, this time between stereotomography and normal-incident-point (NIP) wave tomography [Duveneck, 2004], was presented in Dümmonig et al. [2008]. In Gosselet and Le Bégat [2009], the authors combine borehole and surface seismic data in stereotomography, by modeling the borehole seismic data together with the seismic reflection data, in stereotomography formulation. In Prioux et al. [2013],

the authors also perform a combination of seismic reflected data, this time with seismic data from transmitted waves, in the framework of stereotomography. They also proposed to use the macro-velocity model obtained from stereotomography as an initialization to the full waveform inversion (FWI) method [Virieux and Operto, 2009]. In Plessix [2006], a new technique to compute the gradient of the stereotomography cost function with the adjoint-state method is proposed. The authors of Tavakoli et al. [2017] propose to use in the stereotomography framework an eikonal equation solver [Zhao, 2005, Fomel et al., 2009] as forward modeling and the adjoint-state as a way of computing the gradient. They also propose to use the stereotomography results to initialize the FWI method.

The velocity model obtained by stereotomography relies in the hypothesis that the subsurface presents an acoustic velocity distribution. By following the Born approximation, this acoustic velocity model can be split into two portions [Billette, 1998]: a velocity macro-model that contains long wavelengths (low-frequency) and velocity perturbations that contain short wavelengths (high-frequency). Therefore, the stereotomography is a tool for estimating the low-frequency macro-model of seismic acoustic velocities in the subsurface. Often, the stereotomography results are employed as initialization in high-frequency methods, such as migration or FWI [Prioux et al., 2013, Tavakoli et al., 2017]. This is an interesting approach, since several of these methods are modeled in the framework of inverse problems [Tarantola, 2005], which are mostly solved with the use of local optimization techniques. On the other hand, the local optimization methods are known to be strongly dependent of a good initialization. An alternative to the initialization dependency in seismic inverse problems would be the use of global optimization strategies [Sen and Stoffa, 1995, Datta and Sen, 2016, Sajeve et al., 2016]. But these strategies still present several challenges to be dealt with when applied to the estimation of hundreds to thousands of parameters in seismic tomography and FWI and are not in the scope of this thesis.

In that context, we must think of the initialization by the picking of slopes as a crucial part of the stereotomography problem. In this thesis we address the problem of slopes estimation for a better initialization of stereotomography.

The common-offset common-reflection-surface (CO-CRS) [Zhang et al., 2001] is a method mostly used to increase the signal-to-noise ratio (SNR) of seismic data by the use of locally coherent information. This method is a generalization of the zero-offset common-reflection-surface (ZO-CRS) method [Mann et al., 1999, Jäger et al., 2001], which is used as an alternative to the common-midpoint (CMP) stacking [Mayne, 1962]. The CO-CRS method stacks traces in a surface built with seismic attributes, which are estimated from the seismic data. Among these attributes are the same slopes used in stereotomography. However, the estimation of these attributes is not simple, especially in environments with strong presence of noise. We propose to estimate the CO-CRS attributes with

the bio-inspired global optimization algorithm Differential Evolution (DE) [Rainer, 1995, Storn and Price, 1997]. This is a promising approach, since we already had good results in estimating the ZO-CRS attributes with DE in the work published in Barros et al. [2015].

Another important proposal of this work is to use the attributes obtained as outputs of the CO-CRS method with the global optimization method DE [Barros et al., 2016] as input data for stereotomography. The strategy of using the CO-CRS attributes as input for the stereotomography was briefly signaled as a possibility in the introductory CO-CRS work of Zhang et al. [2001]. Recently, the same discussion came to light in Schwarz et al. [2015], where it is suggested the utilization in stereotomography of the CO-CRS parameters, obtained with the ZO-CRS approximation proposed in Baykulov and Gajewski [2009]. Differently, in this work, we use the parameters obtained directly by the CO-CRS method.

In the next section we outline the contents in the remaining chapters of this thesis.

## 1.1 Thesis organization

In this thesis we focus in the stereotomography initialization. We propose a methodology to obtain more robust results in the presence of noise, by using the CO-CRS method to estimate the slopes used in the stereotomography initialization. First, we present our contributions to the global estimation of the CO-CRS method. We also propose an automatic picking strategy to use the slopes estimated with the CO-CRS method as a way of initialization to the stereotomography. We illustrate the benefits of our proposals in numerical experiments. The remaining chapters of this thesis are organized as follow:

- **Chapter 2:** In this chapter we present the details of stereotomography, as it was proposed in Billette [1998].
- **Chapter 3:** In this work we follow the idea of using the adjoint-state method to compute the gradient used in the optimization stage of our stereotomography implementation. In this chapter we explicitly compute the derivatives of stereotomography cost function with respect to the model parameters, using the adjoint-state method, in the stereotomography framework. We also suggest a way of testing if the implementation of these derivatives with the adjoint-state method is correct.
- **Chapter 4:** In this chapter we explicitly discuss the practical aspects in our implementation of stereotomography.

- **Chapter 5:** In this chapter we explain the CO-CRS method. We begin by illustrating the method itself and then we discuss the sequential and global parameter estimation strategies. We then illustrate the behaviour of both these search strategies in synthetic and field data examples.
- **Chapter 6:** In this chapter we illustrate how we use the slopes estimated with CO-CRS in our stereotomography implementation. We present an illustrative discussion about the behaviour of the slopes estimated with the slant stack and the sequential and global versions of the CO-CRS method, for a synthetic single plane reflector in the presence of noise. We present an automatic picking proposal for the slopes estimated with the CO-CRS method. We illustrate this framework in a slightly more complicate synthetic data.
- **Chapter 7:** We present the general conclusions and prospects involving our work in this chapter.

## 1.2 Publications

Next, we present a complete list of the published work within the doctorate period.

### 1.2.1 Articles published in journals

- T. Barros, R. Lopes, and M. Tygel. Implementation aspects of eigendecomposition-based high-resolution velocity spectra. *Geophysical Prospecting*, 63(1) : 99115, 2014.
- T. Barros, R. Ferrari, R. Krummenauer, and R. Lopes (2015). Differential evolution-based optimization procedure for automatic estimation of the common-reflection surface traveltimes parameters. *Geophysics*, 80(6), WD189-WD200.

### 1.2.2 Articles published in conferences

- T. Barros, R. Ferrari, R. Krummenauer, R. Lopes, and M. Tygel. The impact of the parameter estimation strategy in the CRS method. *In SBGf 13th International Congress of the Brazilian Geophysical Society*, 2013.
- M. R. Covre, T. Barros, A. K. Takahata, and R. R. Lopes. Imageamento de difrações sísmicas baseado em métodos de alta-resolução. *Anais do XXXI Simpósio Brasileiro de Telecomunicações - SBrT2013*, 2013. (In portuguese)

- T. Barros, R. Krummenauer, R. Ferrari, and R. Lopes. Global optimization of the parameters of the Common Reflection Surface traveltimes using Differential Evolution. *In 76th EAGE Conference & Exhibition*, 2014.
- M. R. Covre, T. Barros, R. Lopes. High Resolution Stacking of Seismic Data. In: 22nd European Signal Processing Conference (EUSIPCO 2014), 2014, Lisboa. *Proceedings of 22nd European Signal Processing Conference*, 2014.
- T. Barros and R. Lopes. Eigenstructure analysis for the seismic analytical signal in stacking velocity estimation. *In SBGf 14th International Congress of the Brazilian Geophysical Society*, 2015.
- T. Barros, M. Covre, A. K. Takahata, and R. Lopes. Normal moveout with phase equalization. *In SBGf 14th International Congress of the Brazilian Geophysical Society*, 2015.
- T. Barros, R. Krummenauer, R. Lopes, H. Chauris. Pre-stack Data Recovery through Common Offset CRS Stack with Differential Evolution. In: 78th EAGE Conference and Exhibition 2016, 2016, Viena.

### 1.2.3 Book chapter

- R. Krummenauer, A. Takahata, T. Barros, M. Covre, R. Lopes. High-Resolution Techniques for Seismic Signal Prospecting. *Signals and Images*. 1ed.: CRC Press, 2015, v. , p. 533-566.

## Chapter 2

# Stereotomography

Stereotomography is a tomographic method that uses seismic reflection data to estimate the so-called macro-model of acoustic velocities in the subsurface [Billette, 1998, Billette and Lambaré, 1998]. In a very simple way, stereotomography is employed to obtain the velocity model by the inversion of a set of observed seismic data. The vector that contains this set of data is defined as  $\mathbf{d}$ . A vector with the set of parameters, defined by  $\mathbf{m}$ , forms the model for the stereotomography. The parameters from the model can be used in a ray-tracing procedure to generate a data vector defined as  $\mathbf{d}^c$ . In the context of inverse problems, this is known as direct problem [Tarantola, 2005]. The velocity model is chosen as the one that generates the "best" match between the simulated data set,  $\mathbf{d}^c$ , and the actual seismic data,  $\mathbf{d}$ . This problem is modelled as an inversion problem [Tarantola, 1987, 2005] and the match is usually measured as the mean square error between  $\mathbf{d}^c$  and  $\mathbf{d}$  to find its solution. It is common to add a regularization term to the cost function, because the problem is ill-posed [Tarantola, 2005]. In Figure 2.1 we illustrate the model and the data for stereotomography. In this chapter we present the details of stereotomography, as it was proposed in Billette [1998].

### 2.1 Stereotomography data and model vectors

The seismic data,  $\mathbf{d}$ , used in the 2D stereotomography is formed by:

- Two vectors,  $\mathbf{s} = [x^s, z^s]$  and  $\mathbf{r} = [x^r, z^r]$ , with source and receiver horizontal and vertical positions.
- Two slope values,  $p_x^s$  and  $p_x^r$ , which are part of the two vectors,  $\mathbf{p}_s = [p_x^s, p_z^s]$  and  $\mathbf{p}_r = [p_x^r, p_z^r]$ , that indicate the slowness of the seismic wave-front at the source and receiver. For both source and receiver we have that  $\mathbf{p}_s = [\sin \theta_s/v_s, \cos \theta_s/v_s]$  and  $\mathbf{p}_r = [\sin \theta_r/v_r, \cos \theta_r/v_r]$ , where  $v_s$

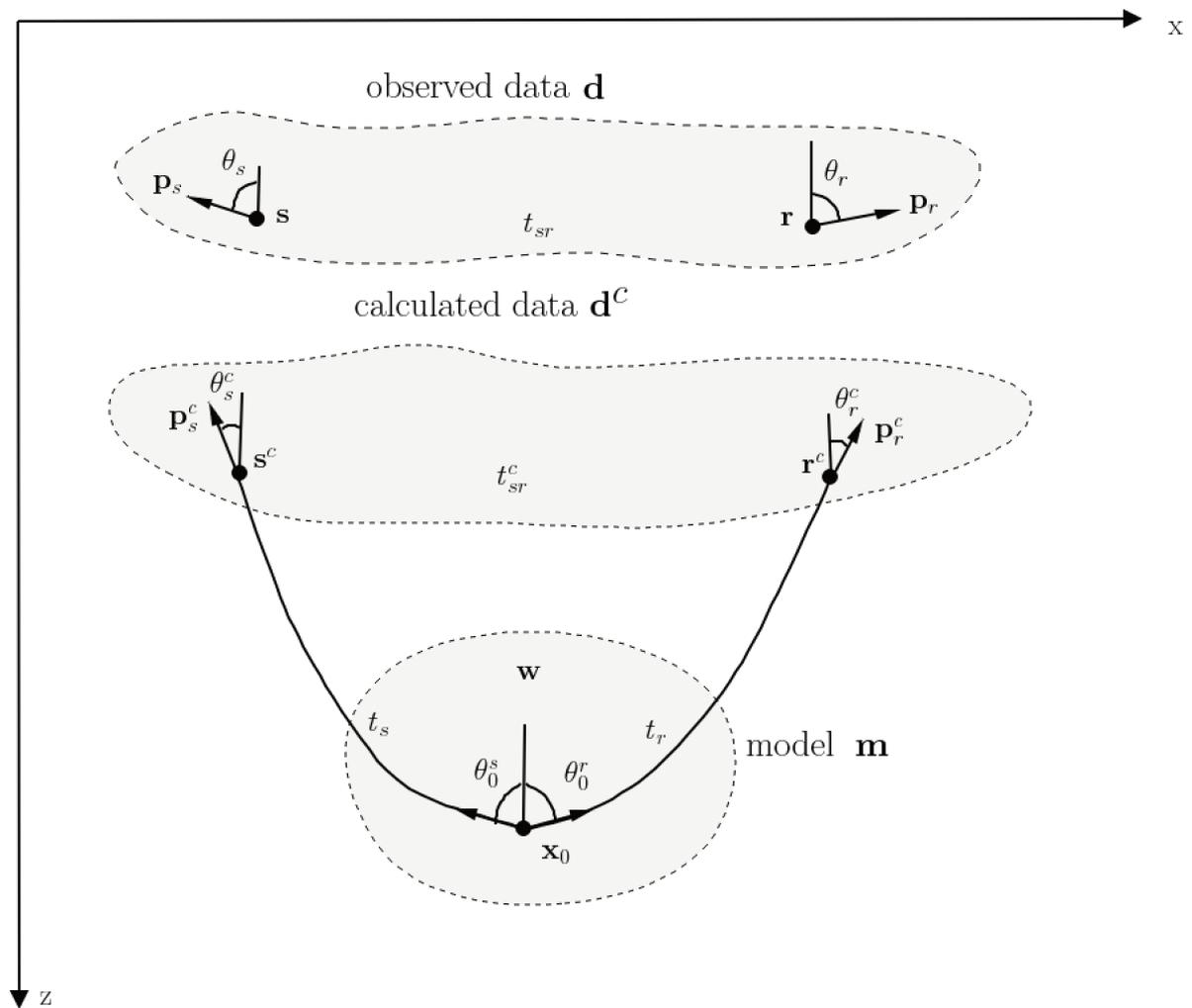


Figure 2.1: Illustration of data and model for stereotomography. The model is represented by the vector  $\mathbf{m}$ , the observed data is represented by the vector  $\mathbf{d}$  and the calculated data is represented by the vector  $\mathbf{d}^c$ .

and  $v_r$  are the near surface velocities in source and receiver positions and  $\theta_s$  and  $\theta_r$  are the angles in Figure 2.1.

- The two-way travel-time,  $t_{sr}$ .

The data are provided to the stereotomography by picking methods, which are used to compute the local slopes from the seismic data [Billette et al., 2003]. A local slope at a given time is obtained from a data gather by the use of the local slant stack method [Schultz and Claerbout, 1978, Ottolini, 1983, Schleicher et al., 2009], where, for a given reference trace, two associated semblance panels are computed from common-receiver (CR) and common-shot (CS) gathers. The slopes and the times are chosen according to the points of maxima in the semblance panels. For more information about the picking process used in the stereotomography, see [Billette et al., 2003]. The vector  $\mathbf{d}$ , containing all the picked parameters for  $N$  data points, is defined as:

$$\mathbf{d} = [(\mathbf{s}, \mathbf{r}, p_x^s, p_x^r, t_{sr})_k]_{k=0}^{N-1}, \quad (2.1)$$

where  $N$  is the number of points where the data parameters were picked.

The velocity model can be described as

$$v(x, z) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_z-1} w_{ij} B_i(x) B_j(z) \quad (2.2)$$

where the  $B_i(x)$  and  $B_j(z)$  represent a basis for the expansion of the velocity using, for instance, B-splines [Tarantola, 2005] or wavelets [Gholami and Siahkoohi, 2010],  $w_{ij}$  are the coefficients of this expansion and  $N_x$  and  $N_z$  are the number of coefficients.

In order to make the notation more clear, when referring to the data and model vectors, we will group the velocities coefficients,  $w_{ij}$ , as a vector with  $M = N_x N_z$  coefficients,  $w_l$ . We define, now, the vector,  $\mathbf{m}$ , which gathers all parameters for the  $N + M$  model points. We represent the model as:

$$\mathbf{m} = [((\mathbf{x}_0, \theta_0^s, \theta_0^r, t_s, t_r)_k)_{k=0}^{N-1}, (w_l)_{l=0}^{M-1}]. \quad (2.3)$$

The parameters that represent the model are:

- A pair of coordinates,  $\mathbf{x}_0 = [x_0, z_0]$ , which represent points of origin of rays travelling to the source and receiver positions.
- Two angles,  $\theta_0^s$  and  $\theta_0^r$ , for the rays which travel from the starting point to the source and receiver positions.

- Two one-way travel-times,  $t_s$  and  $t_r$ , which measure the time the rays take to travel from  $\mathbf{x}_0$  to the source and receiver positions. We have the restriction that  $t_{sr} = t_s + t_r$ .
- A discrete set of  $M = N_x N_z$  coefficients of the expansion which describes the velocity model,  $\mathbf{w} = [w_l]_{l=0}^{M-1}$ .  $N_x$  is the total number of points in the vertical direction and  $N_z$  is the number of points in the horizontal.

Note that each picked point that forms the data has its own values of  $\mathbf{s}$ ,  $\mathbf{r}$ ,  $p_x^s$ ,  $p_x^r$  and  $t_{sr}$ . For each data point, we will determine the parameters  $\mathbf{x}_0$ ,  $\theta_0^s$ ,  $\theta_0^r$ ,  $t_s$  and  $t_r$ , that, along with the velocity model, provide a best match to the data points.

The stereotomography is formulated as an inverse problem [Tarantola, 1987]. In that sort of problem, from the observation of the available data we try to obtain information about the stereotomography model. This problem is solved by the minimization of a cost function between the observed data and the data calculated from the model itself. The direct problem consists in mapping the model into the data and is solved by the use of the *ray tracing* technique. In the next two sections we give more details about the direct and the inverse problems and also about the optimization techniques used to minimize the cost function in stereotomography.

## 2.2 The direct problem: ray tracing

In order to obtain the velocity model with the stereotomography, we employ the ray tracing as forward method to obtain the solution to the direct problem of computing a set of data parameters, represented by the vector  $\mathbf{d}^c$ , from the set of model parameters, represented by the vector  $\mathbf{m}$ . The model  $\mathbf{m}$  which results in a vector  $\mathbf{d}^c$  closest to the picked data  $\mathbf{d}$  provides the estimated velocity model. Ray tracing is a non-linear function of  $\mathbf{m}$ , defined as:

$$\mathbf{g}(\mathbf{m}) = \mathbf{d}^c. \quad (2.4)$$

The ray tracing consists into propagating trajectories defined by positions and directions along a given ray equation [Červený and Ravindra, 1971, Červený, 2005]. In the framework of stereotomography, the ray tracing is described in the context of the Hamiltonian formulation [Farra and Madariaga, 1987, Lambaré et al., 1996, Billette and Lambaré, 1998]. The Hamiltonian equation is defined as

$$H(\mathbf{x}, \mathbf{p}, t) = \frac{1}{2}[\mathbf{p}^2 v^2(\mathbf{x}) - 1], \quad (2.5)$$

The quantities  $t$ ,  $\mathbf{x}$  and  $\mathbf{p}$  are defined along the ray. They are, respectively, related to the time,  $t$ , positions vector,  $\mathbf{x} = (x, z)$ , and slowness vector for the ray trajectories,  $\mathbf{p} = \nabla t(\mathbf{x})$ . The quantities

$\mathbf{x}$  and  $\mathbf{p}$  are also time-dependent, but we do not indicate that in our notation, for simplicity.

The initial condition to the solution of equation (2.5) is given by  $H(\mathbf{x}, \mathbf{p}, t) = 0$ , resulting in  $\mathbf{p}^2 = \frac{1}{v^2(\mathbf{x})}$ , also known as the eikonal equation [Margrave, 2001]. The solution of equation (2.5) can be obtained from the following system of first-order equations [Margrave, 2001]:

$$\begin{cases} \frac{\partial \mathbf{x}}{\partial t} = v^2(\mathbf{x})\mathbf{p} \\ \frac{\partial \mathbf{p}}{\partial t} = -\frac{1}{v(\mathbf{x})}\nabla v(\mathbf{x}) \end{cases} \quad (2.6)$$

The ray trajectories

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{bmatrix} \quad (2.7)$$

are, then, obtained by the integration of the equations in (2.6) along the ray. In this work we used for that a second-order Runge-Kutta method [Butcher, 1987, Press et al., 1997, Butcher, 2007]. The time dependence in equation (2.7) exists, but is usually made implicitly.

## 2.3 Optimization as a way of solving the inverse problem

To obtain data parameters for both source and receiver, each ray is traced starting from  $\mathbf{x}_0$ , with initial angle,  $\theta_0^s$  or  $\theta_0^r$ , until the final travel-times  $t_s$  and  $t_r$  are reached. Rays are traced using the velocity model  $v(\mathbf{x})$ , parametrized as  $\mathbf{w}$ . The final positions at times  $t_s$  and  $t_r$  are  $\mathbf{s}$  or  $\mathbf{r}$ .

The stereotomography employs a cost function [Billette and Lambaré, 1998, Tarantola, 2005] to measure how close  $\mathbf{d}$  and  $\mathbf{d}^c$  are. This cost function is the mean square error plus a regularization term, which is minimized to yield the solution. The cost function,  $S(\mathbf{m})$ , is defined as:

$$S(\mathbf{m}) = \frac{1}{2}[\mathbf{g}(\mathbf{m}) - \mathbf{d}]^T \mathbf{C}_D^{-1}[\mathbf{g}(\mathbf{m}) - \mathbf{d}] + \lambda_r r(\mathbf{m}), \quad (2.8)$$

where, as seen before,  $\mathbf{g}(\mathbf{m}) = \mathbf{d}^c$  is the modeled data,  $\mathbf{C}_D^{-1}$  is a diagonal matrix, used to scale different amplitudes from the parameters in the cost function,  $r(\mathbf{m})$  is the regularization term used to choose a velocity model with a desired feature and  $\lambda_r$  is the parameter used to balance the regularization contribution.

To solve the stereotomography optimization problem, an iterative method is employed, which is based in the local linearization of the operator  $\mathbf{g}(\mathbf{m})$ :

$$\mathbf{g}(\mathbf{m}_k + \Delta\mathbf{m}) \approx \mathbf{g}(\mathbf{m}_k) + \mathbf{G}_k \Delta\mathbf{m}, \quad (2.9)$$

where  $\mathbf{m}_k$  represents the estimated model at the  $k^{th}$  tomography iteration. The matrix  $\mathbf{G}_k$  is formed

by the partial derivatives of the computed data,  $\mathbf{g}(\mathbf{m}_k)$ , with respect to the model parameters,  $\mathbf{m}_k$ :

$$\mathbf{G}_k = \frac{\partial \mathbf{g}(\mathbf{m}_k)}{\partial \mathbf{m}_k}. \quad (2.10)$$

The approach proposed in Billette and Lambaré [1998] to iteratively obtain the linearized minimizer of equation (2.8) makes use of the Newton method and is given by:

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \left( \frac{\partial^2 S}{\partial \mathbf{m}^2}(\mathbf{m}_k) \right)^{-1} \frac{\partial S}{\partial \mathbf{m}}(\mathbf{m}_k). \quad (2.11)$$

If we take the derivatives of equation (2.8) and use the definition of matrix  $\mathbf{G}_k$  in equation (2.10), the update for the model parameters vector becomes:

$$\begin{aligned} \mathbf{m}_{k+1} = \mathbf{m}_k - & \left( \mathbf{G}_k^T \mathbf{C}_D^{-1} \mathbf{G}_k + \lambda_r \frac{\partial^2 r}{\partial \mathbf{m}^2}(\mathbf{m}_k) \right)^{-1} \\ & \times \left( \mathbf{G}_k^T \mathbf{C}_D^{-1} (\mathbf{g}(\mathbf{m}_k) - \mathbf{d}) + \lambda_r \frac{\partial r}{\partial \mathbf{m}}(\mathbf{m}_k) \right). \end{aligned} \quad (2.12)$$

The matrix  $\mathbf{G}_k$  is usually obtained with kinematic ray tracing [Billette and Lambaré, 1998]. In that approach, the partial derivatives of  $\mathbf{G}_k$  are known as Fréchet derivatives. In this work, we used a different approach to compute the gradient, which is explained in the next chapter.

## Chapter 3

# Gradient computation with the adjoint-state method

In Plessix [2006], a new technique to compute the gradient of the stereotomography cost function,  $S(\mathbf{m})$ , with the adjoint-state method is proposed. Plessix [2006] presents a review of the adjoint-state method and then writes stereotomography in the framework of this technique. We follow the idea of using the adjoint-state method to compute the gradient, used for the model optimization, in our stereotomography implementation. In this chapter we explicitly compute the derivatives of  $S(\mathbf{m})$  with respect to the model, using the adjoint-state method, in the stereotomography framework. In Plessix [2006], these derivatives are not explicitly evaluated. We also suggest a method to test if the implementation of these derivatives with the adjoint-state technique is correct.

### 3.1 Adjoint-state method

The adjoint-state method is used to compute the derivatives of an augmented version of a given cost function, which is called the augmented functional. These augmented functional uses the relations from equation (2.8) as restrictions, in a way that is similar to the use of Lagrange multipliers. It is shown by Plessix [2006] that for rays satisfying the ray equations in (2.6), the derivatives of the augmented functional,  $\tilde{J}$ , with respect to the model parameters are the same as the derivatives of the cost function,  $J$ , with respect to the model parameters:

$$\frac{\partial \tilde{J}}{\partial m} = \frac{\partial J}{\partial m}. \quad (3.1)$$

If we define

$$J(\mathbf{m}) = \frac{1}{2}[\mathbf{g}(\mathbf{m}) - \mathbf{d}]^T \mathbf{C}_D^{-1} [\mathbf{g}(\mathbf{m}) - \mathbf{d}] \quad (3.2)$$

and  $r(\mathbf{m})$  as the regularization operator being employed, the stereotomography cost function can be written in terms of the functions  $J(\mathbf{m})$  and  $r(\mathbf{m})$  as

$$S(\mathbf{m}) = J(\mathbf{m}) + \lambda_r r(\mathbf{m}), \quad (3.3)$$

where the parameter  $\lambda_r$  balances the penalization given by the regularization term [Engl et al., 1996].

The derivatives of  $J(\mathbf{m})$  can be obtained with the derivatives of the augmented functional,  $\tilde{J}(\mathbf{m})$ , computed with the adjoint-state method. In this case, the derivatives of  $S(\mathbf{m})$  with respect to the model parameters are:

$$\frac{\partial S(\mathbf{m})}{\partial \mathbf{m}} = \frac{\partial \tilde{J}(\mathbf{m})}{\partial \mathbf{m}} + \lambda_r \frac{\partial r(\mathbf{m})}{\partial \mathbf{m}}. \quad (3.4)$$

The update of the model parameters, with the gradient method, at the iteration  $k + 1$ , is:

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \mathbf{f}_k, \quad (3.5)$$

with the update direction  $\mathbf{f}_k$  being given by the expression

$$\mathbf{f}_k = -\alpha_k \frac{\partial S}{\partial \mathbf{m}}(\mathbf{m}_k), \quad (3.6)$$

where  $\alpha_k$  is the step length taken in the gradient direction, computed at each iteration. In appendix B we discuss how to estimate the parameter  $\alpha_k$ .

## 3.2 Single-ray approximation

In order to simplify the calculations, we first consider a single ray, illustrated in Figure 3.1, starting at a point  $\mathbf{x}_0 = [x_0, z_0]$ , with a starting angle  $\theta_0$ . The ray is propagated in a medium with velocity  $v(\mathbf{x})$ , parametrized as  $\mathbf{w} = [w_l]_{l=0}^{M-1}$ , where  $M$  is the number of coefficients. The ray<sup>1</sup> is propagated from  $t = 0$  until  $t = t_f$ , where it reaches the point  $\mathbf{x}(t = t_f) = [x(t = t_f), z(t = t_f)]$ , with slopes  $\mathbf{p}(t = t_f) = [p_x(t = t_f), p_z(t = t_f)]$ . In this simplified problem, we assume that  $t_f$  is given as part of the picked data-set.

We can define new model and data sets and apply the same concept of stereotomography described in chapter 2, in order to obtain the velocity model  $\mathbf{w}$ . We define, then, the new model vector as:

$$\mathbf{m} = [x_0, z_0, \theta_0, t_f, \mathbf{w}]. \quad (3.7)$$

<sup>1</sup>Note that the coordinates  $\mathbf{x}$  along the ray depend on the integration variable variable  $t$ . For simplicity, we disregard this dependence on the notation.

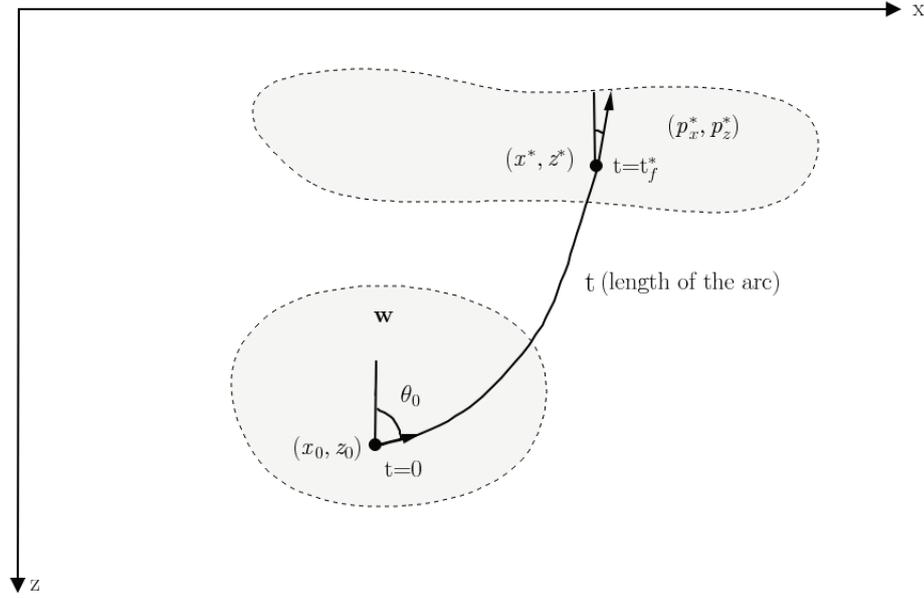


Figure 3.1: Simplified model for gradient computation.

The new vector with the measured data from the picking process is defined as:

$$\mathbf{d} = [x^*, z^*, p_x^*, p_z^*, t_f^*]. \quad (3.8)$$

For the single-ray scheme, the stereotomography cost function is:

$$J(\mathbf{m}) = \frac{C_1}{2}[x(t = t_f) - x^*]^2 + \frac{C_1}{2}[z(t = t_f) - z^*]^2 + \frac{C_2}{2}[p_x(t = t_f) - p_x^*]^2 + \frac{C_2}{2}[p_z(t = t_f) - p_z^*]^2 + \frac{C_3}{2}[t_f - t_f^*]^2, \quad (3.9)$$

where the parameters  $C_1$ ,  $C_2$  and  $C_3$  follow the idea discussed in Billette et al. [2003], where they are chosen to properly scale the amplitude values in the cost function. These parameters can be defined as:

$$C_1 = \frac{1}{\sigma_x^2} = \frac{1}{\sigma_z^2}. \quad (3.10)$$

$$C_2 = \frac{1}{\sigma_{p_x}^2} = \frac{1}{\sigma_{p_z}^2}. \quad (3.11)$$

$$C_3 = \frac{1}{\sigma_t^2}. \quad (3.12)$$

The quantities  $\sigma_x^2$ ,  $\sigma_z^2$ ,  $\sigma_{p_x}^2$ ,  $\sigma_{p_z}^2$  and  $\sigma_t^2$  are elements of the diagonal matrix  $\mathbf{C}_D^{-1}$  and they are used to balance the contribution of each parameter in the minimization of the cost function [Tavakoli et al.,

2017]. For choosing the parameters in  $C_D^{-1}$ , we suggest the reader to see [Billette et al., 2003].

We denote the cost function in equation (3.9) as  $J(\mathbf{m})$  to distinguish from  $S(\mathbf{m})$  in equation (2.8), where a regularization term is involved.

### 3.2.1 Derivation of the gradient

We now compute the gradient of the cost function in equation (3.9) with the use of the adjoint-state method. In order to use the adjoint-state method, we define the vectors with the adjoint-state variables:

$$\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3, \mu_4], \quad (3.13)$$

and

$$\boldsymbol{\lambda}(t) = [\lambda_1(t), \lambda_2(t), \lambda_3(t), \lambda_4(t)]. \quad (3.14)$$

These adjoint-state variables are related to the constraints:

$$\left\{ \begin{array}{ll} x(t=0) = x_0 & (\mu_1) \\ z(t=0) = z_0 & (\mu_2) \\ p_x(t=0) = \frac{\sin \theta_0}{v_0} & (\mu_3) \\ p_z(t=0) = \frac{\cos \theta_0}{v_0} & (\mu_4) \end{array} \right. \quad (3.15)$$

and with the ray-tracing equations

$$\left\{ \begin{array}{ll} \frac{\partial x(t)}{\partial t} = p_x(t) & (\lambda_1(t)) \\ \frac{\partial z(t)}{\partial t} = p_z(t) & (\lambda_2(t)) \\ \frac{\partial p_x(t)}{\partial t} = \frac{1}{v(\mathbf{x})} \frac{\partial}{\partial x(t)} v(\mathbf{x}) & (\lambda_3(t)) \\ \frac{\partial p_z(t)}{\partial t} = \frac{1}{v(\mathbf{x})} \frac{\partial}{\partial z(t)} v(\mathbf{x}) & (\lambda_4(t)) \end{array} \right. \quad (3.16)$$

where  $v_0 = v(x_0, z_0)$ . Note that the vector  $\boldsymbol{\mu}$  is the vector with the constraints related with the initial conditions, while  $\boldsymbol{\lambda}(t)$  represents the constraints related to the differential equations. The vector with the state variables at each  $t$  is

$$\mathbf{y}(t) = [x(t), z(t), p_x(t), p_z(t)]. \quad (3.17)$$

We define the augmented functional for one single ray as  $\tilde{J}$ , which is given by:

$$\begin{aligned}
\tilde{J} = & \frac{C_1}{2}[x(t_f) - x^*]^2 + \frac{C_1}{2}[z(t_f) - z^*]^2 \\
& + \frac{C_2}{2}[p_x(t_f) - p_x^*]^2 + \frac{C_2}{2}[p_z(t_f) - p_z^*]^2 + \frac{C_3}{2}[t_f - t_f^*]^2 \\
& + \int_0^{t_f} dt \lambda_1(t) \left[ \frac{\partial x(t)}{\partial t} - v^2(\mathbf{x})p_x(t) \right] \\
& + \int_0^{t_f} dt \lambda_2(t) \left[ \frac{\partial z(t)}{\partial t} - v^2(\mathbf{x})p_z(t) \right] \\
& + \int_0^{t_f} dt \lambda_3(t) \left[ \frac{\partial p_x(t)}{\partial t} + \frac{1}{v(\mathbf{x})} \frac{\partial}{\partial x(t)} v(\mathbf{x}) \right] \\
& + \int_0^{t_f} dt \lambda_4(t) \left[ \frac{\partial p_z(t)}{\partial t} + \frac{1}{v(\mathbf{x})} \frac{\partial}{\partial z(t)} v(\mathbf{x}) \right] \\
& + \mu_1[x(0) - x_0] + \mu_2[z(0) - z_0] \\
& + \mu_3 \left[ p_x(0) - \frac{\sin \theta_0}{v_0} \right] \\
& + \mu_4 \left[ p_z(0) - \frac{\cos \theta_0}{v_0} \right].
\end{aligned} \tag{3.18}$$

$\tilde{J}$  is a functional of the state variables,  $\mathbf{y}(t)$ , the adjoint-state variables,  $\boldsymbol{\mu}$  and  $\boldsymbol{\lambda}(t)$ , and the model parameters,  $\mathbf{m}$ . The derivatives with respect to the state variables,  $\mathbf{y}(t)$ , can be calculated with the adjoint-state equations.

To rewrite the augmented functional, we first use integration by parts:

$$\begin{aligned}
\int_0^{t_f} dt \lambda_1(t) \frac{\partial x(t)}{\partial t} &= \lambda_1(t)x(t) \Big|_{t=0}^{t=t_f} - \int_0^{t_f} dt \frac{\partial \lambda_1(t)}{\partial t} x(t) \\
\int_0^{t_f} dt \lambda_2(t) \frac{\partial z(t)}{\partial t} &= \lambda_2(t)z(t) \Big|_{t=0}^{t=t_f} - \int_0^{t_f} dt \frac{\partial \lambda_2(t)}{\partial t} z(t) \\
\int_0^{t_f} dt \lambda_3(t) \frac{\partial p_x(t)}{\partial t} &= \lambda_3(t)p_x(t) \Big|_{t=0}^{t=t_f} - \int_0^{t_f} dt \frac{\partial \lambda_3(t)}{\partial t} p_x(t) \\
\int_0^{t_f} dt \lambda_4(t) \frac{\partial p_z(t)}{\partial t} &= \lambda_4(t)p_z(t) \Big|_{t=0}^{t=t_f} - \int_0^{t_f} dt \frac{\partial \lambda_4(t)}{\partial t} p_z(t)
\end{aligned} \tag{3.19}$$

After replacing the integration by parts, the final expression for the augmented functional is:

$$\begin{aligned}
\tilde{J} = & \frac{C_1}{2}[x(t_f) - x^*]^2 + \frac{C_1}{2}[z(t_f) - z^*]^2 \\
& + \frac{C_2}{2}[p_x(t_f) - p_x^*]^2 + \frac{C_2}{2}[p_z(t_f) - p_z^*]^2 + \frac{C_3}{2}[t_f - t_f^*]^2 \\
& - \int_0^{t_f} dt \frac{\partial \lambda_1(t)}{\partial t} x(t) - \int_0^{t_f} dt \lambda_1(t) v^2(\mathbf{x}) p_x(t) \\
& - \int_0^{t_f} dt \frac{\partial \lambda_2(t)}{\partial t} z(t) - \int_0^{t_f} dt \lambda_2(t) v^2(\mathbf{x}) p_z(t) \\
& - \int_0^{t_f} dt \frac{\partial \lambda_3(t)}{\partial t} p_x(t) + \int_0^{t_f} dt \lambda_3(t) \frac{1}{v(\mathbf{x})} \frac{\partial}{\partial x(t)} v(\mathbf{x}) \\
& - \int_0^{t_f} dt \frac{\partial \lambda_4(t)}{\partial t} p_z(t) + \int_0^{t_f} dt \lambda_4(t) \frac{1}{v(\mathbf{x})} \frac{\partial}{\partial z(t)} v(\mathbf{x}) \\
& + \lambda_1(t_f)x(t_f) - \lambda_1(0)x(0) + \lambda_2(t_f)z(t_f) - \lambda_2(0)z(0) \\
& + \lambda_3(t_f)p_x(t_f) - \lambda_3(0)p_x(0) + \lambda_4(t_f)p_z(t_f) - \lambda_4(0)p_z(0) \\
& + \mu_1[x(0) - x_0] + \mu_2[z(0) - z_0] \\
& + \mu_3 \left[ p_x(0) - \frac{\sin \theta_0}{v_0} \right] \\
& + \mu_4 \left[ p_z(0) - \frac{\cos \theta_0}{v_0} \right].
\end{aligned} \tag{3.20}$$

The derivation of  $\tilde{J}$  with respect to  $\theta_0$ ,  $x_0$  and  $z_0$  yields:

$$\begin{aligned}\frac{\partial \tilde{J}}{\partial \theta_0} &= \frac{\partial}{\partial \theta_0} \left[ \mu_3 \frac{\sin \theta_0}{v_0} - \mu_4 \frac{\cos \theta_0}{v_0} \right] \\ &= \mu_4 \frac{\sin \theta_0}{v_0} - \mu_3 \frac{\cos \theta_0}{v_0}.\end{aligned}\quad (3.21)$$

$$\begin{aligned}\frac{\partial \tilde{J}}{\partial x_0} &= \mu_1 \frac{\partial}{\partial x_0} [x(0) - x_0] \\ &= -\mu_1.\end{aligned}\quad (3.22)$$

$$\begin{aligned}\frac{\partial \tilde{J}}{\partial z_0} &= \mu_2 \frac{\partial}{\partial z_0} [z(0) - z_0] \\ &= -\mu_2.\end{aligned}\quad (3.23)$$

The quantities of interest to the gradient method are the equations  $\frac{\partial \tilde{J}}{\partial \theta_0}$ ,  $\frac{\partial \tilde{J}}{\partial x_0}$  and  $\frac{\partial \tilde{J}}{\partial z_0}$  given above. To obtain these quantities we must obtain the adjoint-state variables  $\boldsymbol{\lambda}$  and  $\boldsymbol{\mu}$  by solving the equations  $\frac{\partial \tilde{J}}{\partial x(t)} = 0$ ,  $\frac{\partial \tilde{J}}{\partial z(t)} = 0$ ,  $\frac{\partial \tilde{J}}{\partial p_x(t)} = 0$  and  $\frac{\partial \tilde{J}}{\partial p_z(t)} = 0$ . These derivatives are computed, in different steps, for  $t = 0$ ,  $t = t_f$  and for every value of  $t$  in this interval. We have:

$$\left. \frac{\partial \tilde{J}}{\partial x(t)} \right|_{t=0} = \mu_1 - \lambda_1(0).\quad (3.24)$$

$$\left. \frac{\partial \tilde{J}}{\partial x(t)} \right|_{t=t_f} = C_1[x(t_f) - x^*] + \lambda_1(t_f).\quad (3.25)$$

$$\frac{\partial \tilde{J}}{\partial x(t)} = -2\lambda_1(t)p_x(t)v(\mathbf{x})v_x - 2\lambda_2(t)p_z(t)v(\mathbf{x})v_x\quad (3.26)$$

$$+ \lambda_3(t) \left[ \frac{1}{v(\mathbf{x})} v_{xx} - \frac{1}{v^2(\mathbf{x})} v_x v_x \right]\quad (3.27)$$

$$+ \lambda_4(t) \left[ \frac{1}{v(\mathbf{x})} v_{xz} - \frac{1}{v^2(\mathbf{x})} v_x v_z \right],\quad (3.28)$$

where we used:

$$\begin{cases} v_x &= \frac{\partial}{\partial x(t)} v(\mathbf{x}) \\ v_z &= \frac{\partial}{\partial z(t)} v(\mathbf{x}) \\ v_{xx} &= \frac{\partial^2}{\partial x^2(t)} v(\mathbf{x}) \\ v_{zz} &= \frac{\partial^2}{\partial z^2(t)} v(\mathbf{x}) \\ v_{xz} &= \frac{\partial}{\partial x(t)} \frac{\partial}{\partial z(t)} v(\mathbf{x}). \end{cases} \quad (3.29)$$

To force  $\frac{\partial \tilde{J}}{\partial x(t)} = 0$  for all  $t$  we select:

$$\begin{cases} \lambda_1(t_f) &= -C_1[x(t_f) - x^*] \\ \frac{\partial \lambda_1(t)}{\partial t} &= -2\lambda_1(t)p_x(t)v(\mathbf{x})v_x - 2\lambda_2(t)p_z(t)v(\mathbf{x})v_x \\ &\quad + \lambda_3(t) \left[ \frac{1}{v(\mathbf{x})}v_{xx} - \frac{1}{v^2(\mathbf{x})}v_x v_x \right] + \lambda_4(t) \left[ \frac{1}{v(\mathbf{x})}v_{xz} - \frac{1}{v^2(\mathbf{x})}v_x v_z \right] \\ \lambda_1(0) &= \mu_1 \end{cases} \quad (3.30)$$

The interpretation of equation (3.30) is as follows: to obtain  $\lambda_1(0)$  and, consequently,  $\mu_1$  we start with the initial condition that yields  $\lambda_1(t_f)$  and then solve the differential equation that involves  $\frac{\partial \lambda_1(t)}{\partial t}$  to obtain  $\lambda_1(t)$  until we obtain  $\lambda_1(0)$ . This value is equal to  $\mu_1$ . Furthermore, note that the actual cost function only appears in (3.25), and thus only affects the initial value  $\lambda_1(t_f)$  in the system of equations (3.30).

Analogously, for  $z(t)$ ,  $p_x(t)$  and  $p_z(t)$  we have:

$$\begin{cases} \lambda_2(t_f) &= -C_1[z(t_f) - z^*] \\ \frac{\partial \lambda_2(t)}{\partial t} &= -2\lambda_1(t)p_x(t)v(\mathbf{x})v_z - 2\lambda_2(t)p_z(t)v(\mathbf{x})v_z \\ &\quad + \lambda_3(t) \left[ \frac{1}{v(\mathbf{x})}v_{xz} - \frac{1}{v^2(\mathbf{x})}v_x v_z \right] + \lambda_4(t) \left[ \frac{1}{v(\mathbf{x})}v_{zz} - \frac{1}{v^2(\mathbf{x})}v_z v_z \right] \\ \lambda_2(0) &= \mu_2 \end{cases} \quad (3.31)$$

$$\begin{cases} \lambda_3(t_f) &= -C_2[p_x(t_f) - p_x^*] \\ \frac{\partial \lambda_3(t)}{\partial t} &= -\lambda_1(t)v^2(\mathbf{x}) \\ \lambda_3(0) &= \mu_3 \end{cases} \quad (3.32)$$

$$\begin{cases} \lambda_4(t_f) &= -C_2[p_z(t_f) - p_z^*] \\ \frac{\partial \lambda_4(t)}{\partial t} &= -\lambda_2(t)v^2(\mathbf{x}) \\ \lambda_4(0) &= \mu_4 \end{cases} \quad (3.33)$$

Now we compute the derivatives of the extended cost function,  $\tilde{J}$ , with respect to the velocity,

$v(\mathbf{x})$ . Recalling that the velocity model can be written as

$$v(x, z) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_z-1} w_{ij} B_i(x) B_j(z), \quad (3.34)$$

we need, then, to compute the derivatives of  $\tilde{J}$  with respect to a given coefficient,  $w_{kl}$ . From equation 3.20, we have that

$$\begin{aligned} \frac{\partial \tilde{J}}{\partial w_{kl}} &= \frac{\partial}{\partial w_{kl}} \left[ - \int_0^{t_f} dt \lambda_1(t) v^2(\mathbf{x}) p_x(t) - \int_0^{t_f} dt \lambda_2(t) v^2(\mathbf{x}) p_z(t) \right. \\ &\quad + \int_0^{t_f} dt \lambda_3(t) \frac{1}{v(\mathbf{x})} \frac{\partial}{\partial x} v(\mathbf{x}) + \int_0^{t_f} dt \lambda_4(t) \frac{1}{v(\mathbf{x})} \frac{\partial}{\partial z} v(\mathbf{x}) \\ &\quad \left. - \frac{(\mu_3 \sin \theta_0 + \mu_4 \cos \theta_0)}{v_0} \right]. \end{aligned} \quad (3.35)$$

Considering that  $\frac{\partial \tilde{J}}{\partial w_{kl}} = I_1 + I_2 + I_3 + I_4 + I_0$ , we need to compute each one of these terms at  $k = i$  and  $l = j$ . Considering that  $A = (\mu_3 \sin \theta_0 + \mu_4 \cos \theta_0)$ , we have:

$$\left\{ \begin{array}{l} I_1 = -2 \int_0^{t_f} dt \lambda_1(t) p_x(t) v(\mathbf{x}) B_k(x) B_l(z) \\ I_2 = -2 \int_0^{t_f} dt \lambda_2(t) p_z(t) v(\mathbf{x}) B_k(x) B_l(z) \\ I_3 = \int_0^{t_f} dt \lambda_3(t) \left[ \frac{1}{v(\mathbf{x})} \frac{\partial}{\partial x} B_k(x) B_l(z) - \frac{1}{v^2(\mathbf{x})} v_x B_k(x) B_l(z) \right] \\ I_4 = \int_0^{t_f} dt \lambda_4(t) \left[ \frac{1}{v(\mathbf{x})} B_k(x) \frac{\partial}{\partial z} B_l(z) - \frac{1}{v^2(\mathbf{x})} v_z B_k(x) B_l(z) \right] \\ I_0 = \frac{A}{v_0^2} B_k(x_0) B_l(z_0) \end{array} \right. \quad (3.36)$$

Equation (3.35) becomes, then:

$$\begin{aligned} \frac{\partial \tilde{J}}{\partial w_{kl}} &= -2 \int_0^{t_f} dt \lambda_1(t) p_x(t) v(\mathbf{x}) B_k(x) B_l(z) - 2 \int_0^{t_f} dt \lambda_2(t) p_z(t) v(\mathbf{x}) B_k(x) B_l(z) \\ &\quad + \int_0^{t_f} dt \lambda_3(t) \left[ \frac{1}{v(\mathbf{x})} \frac{\partial}{\partial x} B_k(x) B_l(z) - \frac{1}{v^2(\mathbf{x})} v_x B_k(x) B_l(z) \right] \\ &\quad + \int_0^{t_f} dt \lambda_4(t) \left[ \frac{1}{v(\mathbf{x})} B_k(x) \frac{\partial}{\partial z} B_l(z) - \frac{1}{v^2(\mathbf{x})} v_z B_k(x) B_l(z) \right] \\ &\quad + \frac{A}{v_0^2} B_k(x_0) B_l(z_0). \end{aligned} \quad (3.37)$$

To obtain the derivatives of  $\tilde{J}$  with respect to the elements of  $\mathbf{w}$  we must compute equation (3.37) for every pair  $ij$ . These integrals are computed for values of  $\mathbf{x}$  along the rays.

The derivative of the cost function  $J(\mathbf{m})$ , from equation (3.9), with respect to  $t_f$  is analytically obtained by the expression:

$$\begin{aligned} \frac{\partial J(\mathbf{m})}{\partial t_f} &= C_1 [x(t_f) - x^*] \frac{\partial x(t)}{\partial t} \Big|_{t=t_f} + C_1 [z(t_f) - z^*] \frac{\partial z(t)}{\partial t} \Big|_{t=t_f} \\ &\quad + C_2 [p_x(t_f) - p_x^*] \frac{\partial p_x(t)}{\partial t} \Big|_{t=t_f} + C_2 [p_z(t_f) - p_z^*] \frac{\partial p_z(t)}{\partial t} \Big|_{t=t_f} \\ &\quad + C_3 (t_f - t_f^*) \end{aligned} \quad (3.38)$$

### 3.3 Gradient with two rays

If we consider two different rays emerging from  $\mathbf{x}_0$ , as in Figure 2.1, we can define different model and data vectors:

$$\begin{cases} \mathbf{m}^{\text{sr}} &= [x_0, z_0, \theta_0^{\text{s}}, \theta_0^{\text{r}}, t_{\text{s}}, t_{\text{r}}, (w_l)_{l=0}^{M-1}]. \\ \mathbf{d}^{\text{sr}} &= [\mathbf{s}, \mathbf{r}, p_x^{\text{s}}, p_x^{\text{r}}, t_{\text{sr}}]. \end{cases} \quad (3.39)$$

Now we have that  $\mathbf{x}_0$  is the starting point;  $\theta_0^{\text{s}}$  and  $\theta_0^{\text{r}}$  are the emergence angles for the rays which travel from the starting point to source, at  $\mathbf{s}$ , and receiver, at  $\mathbf{r}$ ;  $p_x^{\text{s}}$  and  $p_x^{\text{r}}$  are the vertical components of the slowness vectors, measured at the source and receiver positions;  $t_{\text{s}}$  and  $t_{\text{r}}$  are the final ray propagation times at source and receiver positions and  $t_{\text{sr}} = t_{\text{s}} + t_{\text{r}}$ . For that case, the general stereotomography cost function may be written as

$$J^{\text{sr}} = J^{\text{s}} + J^{\text{r}} + \frac{C_3}{2}(t_{\text{s}} + t_{\text{r}} - t_{\text{sr}})^2, \quad (3.40)$$

where  $J^{\text{s}}$  and  $J^{\text{r}}$  are cost functions equivalent to equation (3.9), computed for a single ray, which goes from  $\mathbf{x}_0$  to source and receiver positions. The parameter  $C_3$  is used to scale the amplitudes in the cost function.

The derivatives of  $J^{\text{sr}}$  with respect to the model parameters can be obtained with the adjoint-state method if we write the augmented functional as:

$$\tilde{J}^{\text{sr}} = \tilde{J}^{\text{s}} + \tilde{J}^{\text{r}} + \frac{C_3}{2}(t_{\text{s}} + t_{\text{r}} - t_{\text{sr}})^2, \quad (3.41)$$

where  $\tilde{J}^{\text{s}}$  and  $\tilde{J}^{\text{r}}$  are equivalent to equation (3.20), with the rays being traced from the point  $\mathbf{x}_0$  to source and receiver positions. For both these rays, the derivatives of  $\tilde{J}^{\text{sr}}$ , with respect to  $\theta_0$ ,  $x_0$  and  $z_0$  are obtained by equations (3.21), (3.22) and (3.23), respectively. These derivatives are written as

$$\begin{cases} \frac{\partial \tilde{J}^{\text{sr}}}{\partial x_0^{\text{s}}} &= -\mu_1^{\text{s}} \\ \frac{\partial \tilde{J}^{\text{sr}}}{\partial z_0^{\text{s}}} &= -\mu_2^{\text{s}} \\ \frac{\partial \tilde{J}^{\text{sr}}}{\partial \theta_0^{\text{s}}} &= \mu_4^{\text{s}} \frac{\sin \theta_0^{\text{s}}}{v_0} - \mu_3^{\text{s}} \frac{\cos \theta_0^{\text{s}}}{v_0} \end{cases} \quad (3.42)$$

$$\begin{cases} \frac{\partial \tilde{J}^{\text{sr}}}{\partial x_0^{\text{r}}} &= -\mu_1^{\text{r}} \\ \frac{\partial \tilde{J}^{\text{sr}}}{\partial z_0^{\text{r}}} &= -\mu_2^{\text{r}} \\ \frac{\partial \tilde{J}^{\text{sr}}}{\partial \theta_0^{\text{r}}} &= \mu_4^{\text{r}} \frac{\sin \theta_0^{\text{r}}}{v_0} - \mu_3^{\text{r}} \frac{\cos \theta_0^{\text{r}}}{v_0} \end{cases} \quad (3.43)$$

The variables  $\mu_i^{\text{s}}$  are obtained through the variables  $\lambda_i^{\text{s}}(t)$ , making  $t_{\text{f}} = t_{\text{s}}$  in equations (3.30), (3.31),

(3.32) and (3.33), for the ray which travels to the the source. The same procedure is used to obtain the variables  $\mu_i^r$ , for the ray which travels to the the receiver. Note that the superscript in  $x_0^s, z_0^s, x_0^r$  and  $z_0^r$  refers to the ray that ends at the source or receiver positions, since the point of origin is the same for both rays. Also, for both the rays traced from the starting point  $\mathbf{x}_0$  to source and receiver positions, the derivatives of  $\tilde{J}^{sr}$  with respect to the coefficients  $w_{kl}$  are given by equation (3.37), using the adjoint-state variables  $\lambda_1^s(t), \lambda_2^s(t), \lambda_3^s(t)$  and  $\lambda_4^s(t)$  or  $\lambda_1^r(t), \lambda_2^r(t), \lambda_3^r(t)$  and  $\lambda_4^r(t)$ . We have:

$$\frac{\partial \tilde{J}^{sr}}{\partial w_{kl}} = \frac{\partial \tilde{J}^s}{\partial w_{kl}} + \frac{\partial \tilde{J}^r}{\partial w_{kl}}. \quad (3.44)$$

The derivatives of  $\tilde{J}^{sr}$  with respect to  $t_s$  and  $t_r$  are:

$$\begin{aligned} \frac{\partial \tilde{J}^{sr}}{\partial t_s} = & C_1[x(t_s) - x^*] \frac{\partial x(t)}{\partial t} \Big|_{t=t_s} + C_1[z(t_s) - z^*] \frac{\partial z(t)}{\partial t} \Big|_{t=t_s} \\ & + C_2[p_x(t_s) - p_x^*] \frac{\partial p_x(t)}{\partial t} \Big|_{t=t_s} + C_2[p_z(t_s) - p_z^*] \frac{\partial p_z(t)}{\partial t} \Big|_{t=t_s} \\ & + C_3(t_s + t_r - t_{sr}) \end{aligned} \quad (3.45)$$

and

$$\begin{aligned} \frac{\partial \tilde{J}^{sr}}{\partial t_r} = & C_1[x(t_r) - x^*] \frac{\partial x(t)}{\partial t} \Big|_{t=t_r} + C_1[z(t_r) - z^*] \frac{\partial z(t)}{\partial t} \Big|_{t=t_r} \\ & + C_2[p_x(t_r) - p_x^*] \frac{\partial p_x(t)}{\partial t} \Big|_{t=t_r} + C_2[p_z(t_r) - p_z^*] \frac{\partial p_z(t)}{\partial t} \Big|_{t=t_r} \\ & + C_3(t_s + t_r - t_{sr}) \end{aligned} \quad (3.46)$$

The derivatives with respect to  $t$  in equations (3.45) and (3.46) are given by the ray equations.

### 3.4 Generalization of the gradient

For  $N$  data points, the model and data vectors are written as:

$$\begin{cases} \mathbf{m} = [((x_0, z_0, \theta_0^s, \theta_0^r, t_s, t_r)_i)_{i=0}^{N-1}, (w_l)_{l=0}^{M-1}] \\ \mathbf{d} = [(\mathbf{s}, \mathbf{r}, p_x^s, p_x^r, t_{sr})_i]_{i=0}^{N-1}. \end{cases} \quad (3.47)$$

If we compute equation (3.40) for the  $N$  data points, it will be equal to the term of equation (2.8):  $\frac{1}{2}[\mathbf{g}(\mathbf{m}) - \mathbf{d}]^T \mathbf{C}_D^{-1} [\mathbf{g}(\mathbf{m}) - \mathbf{d}]$ .

Now, the matrix  $\mathbf{C}_D^{-1}$  is formed by the coefficients  $C_1, C_2$  and  $C_3$ , for the  $N$  data points. The

derivatives of any regularization term can be computed separately. Recalling that if we add a regularization term to the cost function equation given above, we may rewrite the cost function as

$$S(\mathbf{m}) = J^{\text{sr}}(\mathbf{m}) + \lambda_r r(\mathbf{m}), \quad (3.48)$$

where  $J^{\text{sr}}(\mathbf{m})$  is the cost function from equation (3.40) computed for the  $N$  data points and the operator  $r(\mathbf{m})$  can be any regularization that we use. In a brief way,  $S(\mathbf{m})$  is the complete cost function, involving the summation of the cost from stereotomography, given by  $J(\mathbf{m})$ , and the regularization term,  $r(\mathbf{m})$ .

The derivatives of  $J^{\text{sr}}(\mathbf{m})$  can be obtained with the adjoint-state method, if we use the augmented functional  $\tilde{J}^{\text{sr}}(\mathbf{m})$ , for the  $N$  data points. With the use of the adjoint-state method, the derivatives of  $S(\mathbf{m})$  with respect to the model parameters are:

$$\frac{\partial S(\mathbf{m})}{\partial \mathbf{m}} = \frac{\partial \tilde{J}^{\text{sr}}(\mathbf{m})}{\partial \mathbf{m}} + \lambda_r \frac{\partial r(\mathbf{m})}{\partial \mathbf{m}}. \quad (3.49)$$

The update of the model parameters, obtained with the gradient, at the iteration  $k + 1$ , is:

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \alpha_k \frac{\partial S}{\partial \mathbf{m}}(\mathbf{m}_k). \quad (3.50)$$

## 3.5 Stereotomography gradient verification

A practical way of verifying the gradient implementation with the adjoint-state method is to perform a test that uses a numerical approximation for the derivative of the cost function with respect to the model parameters, commonly known as *finite difference* test. In order to do this test, we compute the derivative of the cost function,  $J(\mathbf{m})$ , with respect of a given model parameter,  $m_i$ , as:

$$\left. \frac{\partial J(\mathbf{m})}{\partial m} \right|_{m=m_i} \approx \frac{J(\mathbf{m} + \delta m \mathbf{e}_i) - J(\mathbf{m} - \delta m \mathbf{e}_i)}{2\delta m}, \quad (3.51)$$

for sufficiently small values of the model parameter perturbation,  $\delta m$ . We have that the vector  $\mathbf{e}_i$  is different than zero only for the  $i$ -th model parameter,  $m_i$ . One of the interpretations of equation (3.51) is that if the model parameter perturbation,  $\delta m$ , is small enough, the straight line given by  $J(\mathbf{m}) + \left. \frac{\partial J}{\partial m} \right|_{m=m_i} \delta m$  should be tangent to the curve given by  $J(\mathbf{m} + \delta m \mathbf{e}_i)$  at  $\delta m = 0$ .

We performed this test in the one-ray configuration, defined in section 3.2. For each variable of the model parameter vector, defined by

$$\mathbf{m} = (x_0, z_0, \theta_0, t_f, \{w_l\}_{l=0}^{M-1}), \quad (3.52)$$

we first computed the following derivatives, with the use of the adjoint-state method:

$$\left\{ \begin{array}{l} G_1 = \frac{\partial \tilde{J}(\mathbf{m})}{\partial \theta_0} \\ G_2 = \frac{\partial \tilde{J}(\mathbf{m})}{\partial x_0} \\ G_3 = \frac{\partial \tilde{J}(\mathbf{m})}{\partial z_0} \\ G_4 = \frac{\partial \tilde{J}(\mathbf{m})}{\partial t_f} \\ G_5 = \frac{\partial \tilde{J}(\mathbf{m})}{\partial w_{kl}} \end{array} \right. . \quad (3.53)$$

After that, we plotted the following curves

$$\left\{ \begin{array}{ll} J_n(\theta_0 + \delta\theta_0) \times \delta\theta_0 & \text{and } [J_n(\theta_0) + G_1\delta\theta_0] \times \delta\theta_0 \\ J_n(x_0 + \delta x_0) \times \delta x_0 & \text{and } [J_n(x_0) + G_2\delta x_0] \times \delta x_0 \\ J_n(z_0 + \delta z_0) \times \delta z_0 & \text{and } [J_n(z_0) + G_3\delta z_0] \times \delta z_0 \\ J_n(t_f + \delta t_f) \times \delta t_f & \text{and } [J_n(t_f) + G_4\delta t_f] \times \delta t_f \\ J_n(w_{kl} + \delta w_{kl}) \times \delta w_{kl} & \text{and } [J_n(w_{kl}) + G_5\delta w_{kl}] \times \delta w_{kl} \end{array} \right. . \quad (3.54)$$

For each model variable, we plotted curves with five different cost functions, by making  $n = 1, \dots, 5$  in equation (3.54). The cost functions used were:

$$\left\{ \begin{array}{l} J_1 = \frac{1}{2}[x(t_f) - x^*]^2 \\ J_2 = \frac{1}{2}[z(t_f) - z^*]^2 \\ J_3 = \frac{1}{2}[p_x(t_f) - p_x^*]^2 \\ J_4 = \frac{1}{2}[p_z(t_f) - p_z^*]^2 \\ J_5 = \frac{1}{2}[t_f - t_f^*]^2 \end{array} \right. . \quad (3.55)$$

The motivation of using the five different cost functions, presented in equation (3.55), was that if some implementation error were to be present in the derivatives of equation (3.53), it should be easily tracked.

In order to perform these tests, we used a velocity model given by the following expression:

$$v(x, z) = 1.5 + z + 0.3x \text{ Km/s}. \quad (3.56)$$

We choose a velocity model with variations in  $x$  and  $z$  because it allows rays with different final data parameters even if we vary only one model parameter. We described this velocity model in a  $(x, z)$  region starting at the point  $(0, 0)$  Km and ending in  $(3.6, 1.8)$  Km. The original velocity model was sampled with  $d_g = 10$  m, resulting in 361 values in  $x$  and 181 in  $z$ . The velocity model was downsampled by a factor of 20, with the use of the B-splines representation, resulting in a velocity

coefficients matrix with dimensions  $10 \times 19$ . We computed a reference ray with the parameters:  $\theta_0 = 30^\circ$ ,  $x_0 = 1.5 \text{ Km}$ ,  $z_0 = 1.5 \text{ Km}$  and  $t_f = 0.4 \text{ s}$ . We used  $d_t = 1 \text{ ms}$ . In order to compute the derivatives  $G_1$ ,  $G_2$ ,  $G_3$  and  $G_4$ , we traced a different ray, for each derivative computation, with a reference perturbation in the initial model parameters of  $\theta_0 + \delta\theta_0^{\text{ref}} = 60^\circ$ ,  $x_0 + \delta x_0^{\text{ref}} = 1.6 \text{ Km}$ ,  $z_0 + \delta z_0^{\text{ref}} = 1.6 \text{ Km}$  and  $t_f + \delta t_f^{\text{ref}} = 0.3 \text{ s}$ , respectively. For the computation of  $G_5$ , we added the reference perturbation of  $\delta w_k l^{\text{ref}} = 1 \text{ Km/s}$  to the coefficients with indexes  $k = 9$  and  $l = 8$ , close to the source, and  $k = 11$  and  $l = 5$ , far from the source. Finally, in order to plot the curves, we added perturbations in the model values used to compute the derivatives. The perturbation ranges used were: from  $-20^\circ$  to  $20^\circ$ , for  $\delta\theta_0$ ; from  $-200 \text{ m}$  to  $200 \text{ m}$ , for  $\delta x_0$ ; from  $-100 \text{ m}$  to  $100 \text{ m}$ , for  $\delta z_0$ ; from  $-0.1 \text{ s}$  to  $0.1 \text{ s}$ , for  $\delta t_f$  and from  $-1 \text{ Km/s}$  to  $1 \text{ Km/s}$ , for  $\delta w_k l$ . We computed the curves in 21 points for each test.

In Figures 3.2, 3.4, 3.6, 3.8, 3.10, and 3.12 we show the velocity model used in these tests and the rays traced with perturbed values for the model parameter variables. In Figures 3.3, 3.5, 3.7, 3.9, 3.11 and 3.13 we show the curves plotted using equations (3.53), (3.54) and (3.55). Each of these figures contains the curves for  $x$ ,  $z$ ,  $p_x$  and  $p_z$ . As expected, these graphics present tangent curves, thus validating the implementation of the adjoint-state method for the gradient computation. We show the cost function  $J_5$ , from equation (3.54), only in Figure 3.9, because only in this test we varied the values of  $t_f$ .

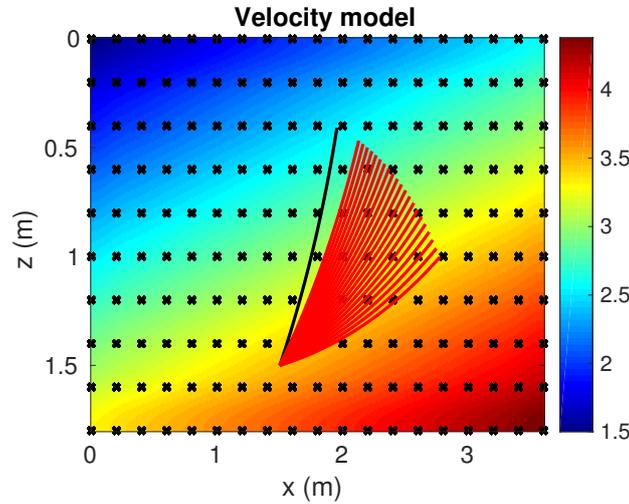


Figure 3.2: Velocity model with perfect ray (black) and rays traced with perturbed initial angle (red).

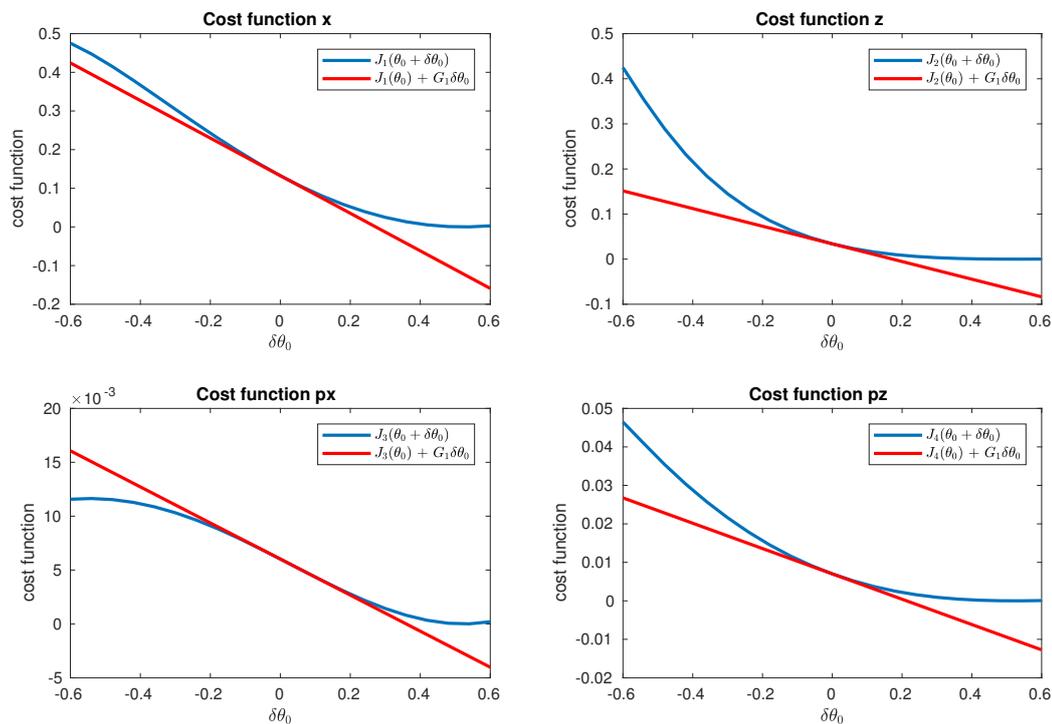


Figure 3.3: Finite difference test with the cost functions  $J_1$  (top left),  $J_2$  (top right),  $J_3$  (bottom left) and  $J_4$  (bottom right).

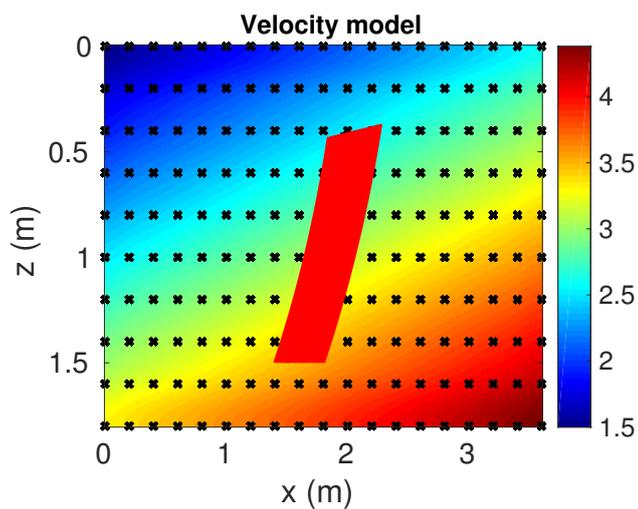


Figure 3.4: Velocity model with perfect ray (black) and rays traced with perturbed initial  $x$  coordinate (red).

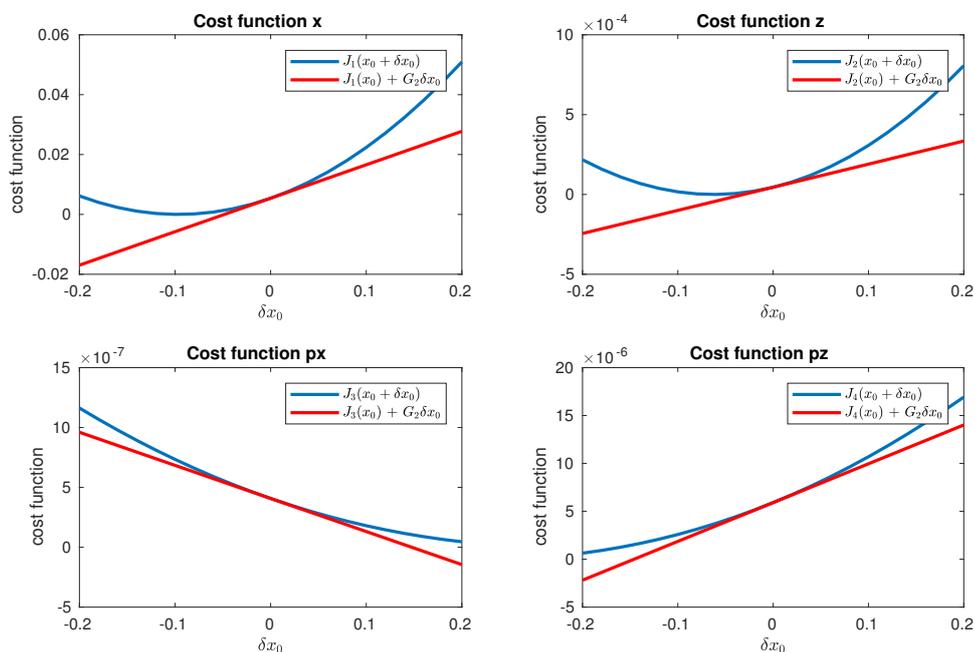


Figure 3.5: Finite difference test with the cost functions  $J_1$  (top left),  $J_2$  (top right),  $J_3$  (bottom left) and  $J_4$  (bottom right).

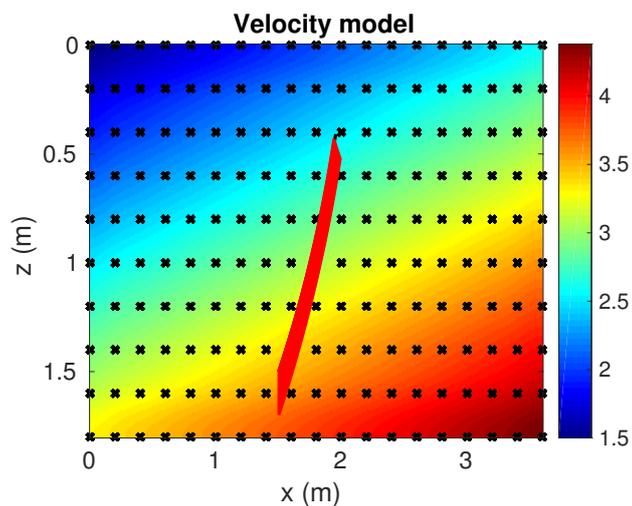


Figure 3.6: Velocity model with perfect ray (black) and rays traced with perturbed initial  $z$  coordinate (red).

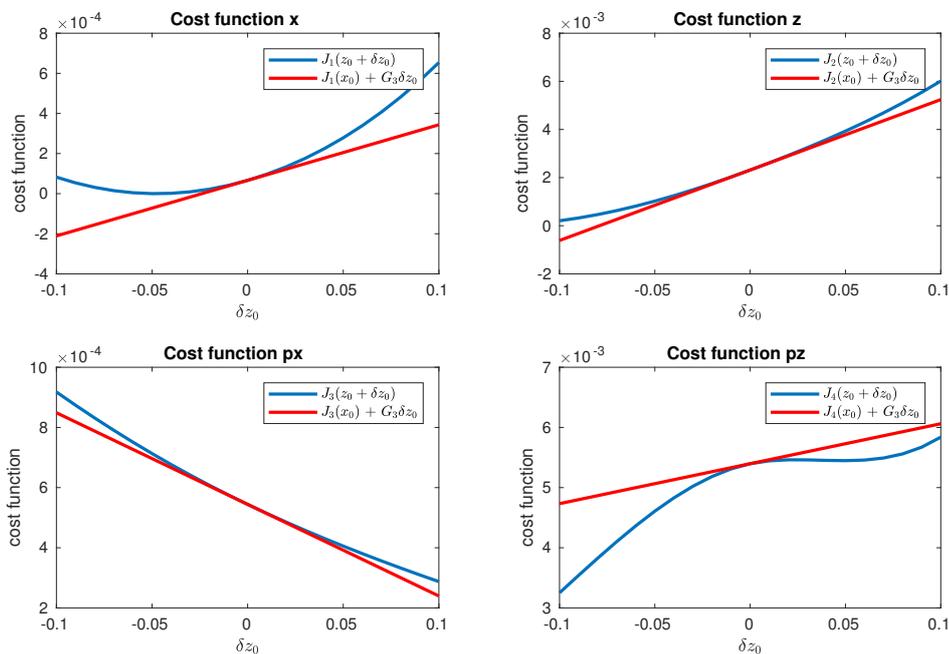


Figure 3.7: Finite difference test with the cost functions  $J_1$  (top left),  $J_2$  (top right),  $J_3$  (bottom left) and  $J_4$  (bottom right).

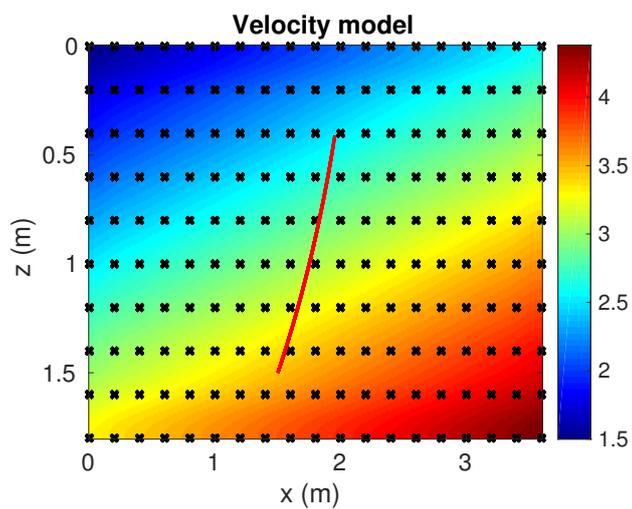


Figure 3.8: Velocity model with perfect ray (black) and rays traced with perturbed ray propagation time (red).

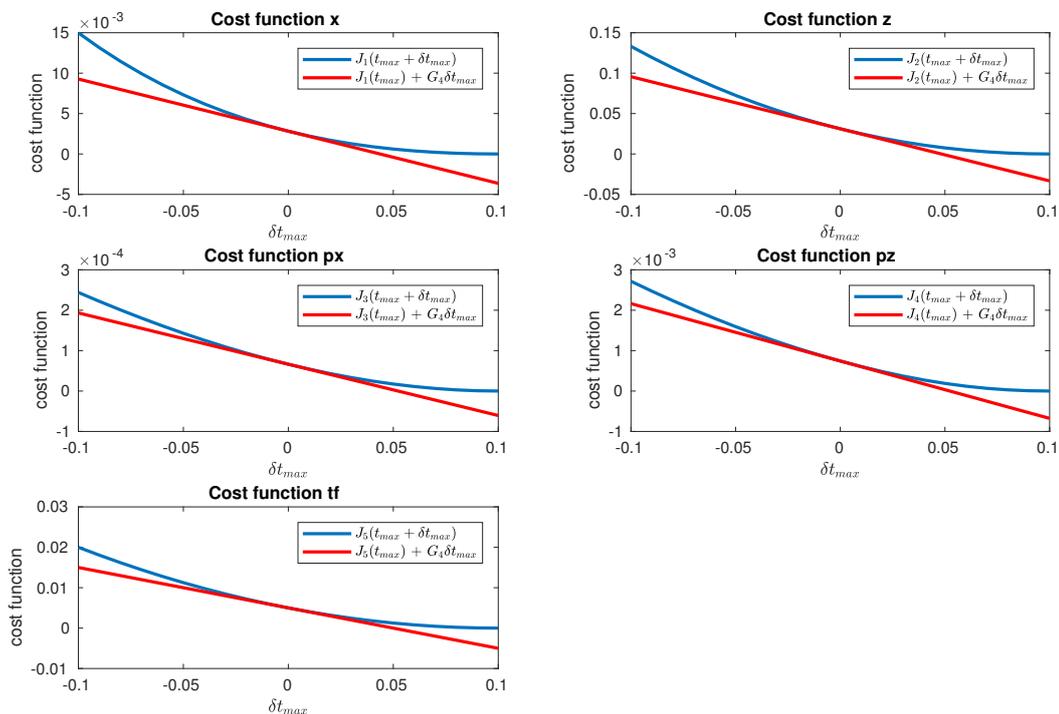


Figure 3.9: Finite difference test with the cost functions  $J_1$  (top left),  $J_2$  (top right),  $J_3$  (middle left) and  $J_4$  (middle right) and  $J_5$  (bottom left).

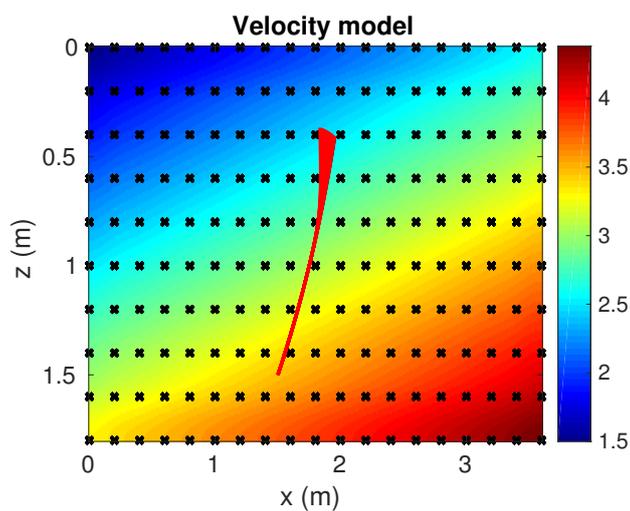


Figure 3.10: Velocity model with perfect ray (black) and rays traced with perturbed B-spline coefficient, away from the source point (red).

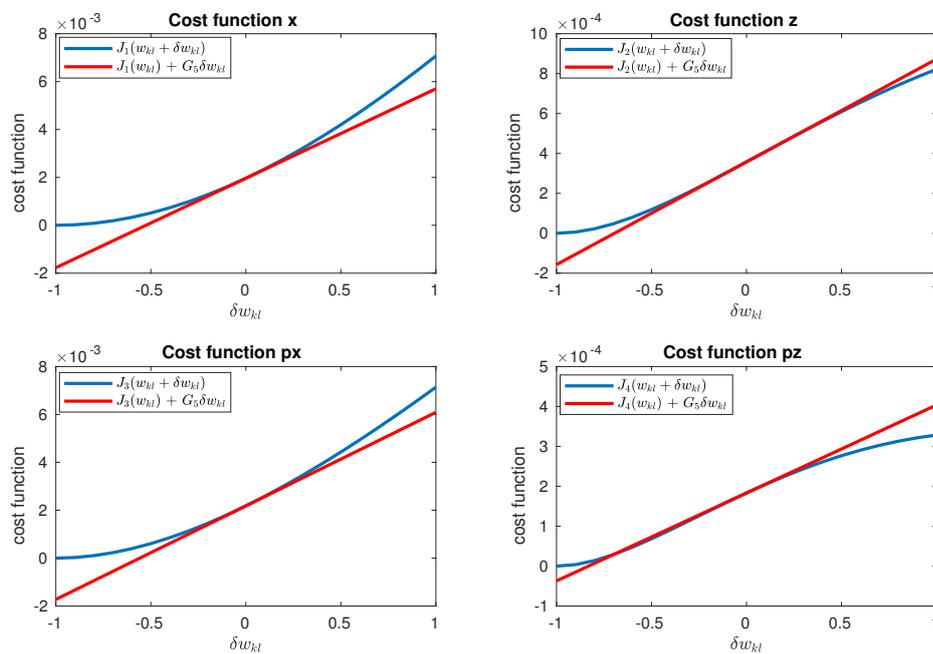


Figure 3.11: Finite difference test with the cost functions  $J_1$  (top left),  $J_2$  (top right),  $J_3$  (bottom left) and  $J_4$  (bottom right).

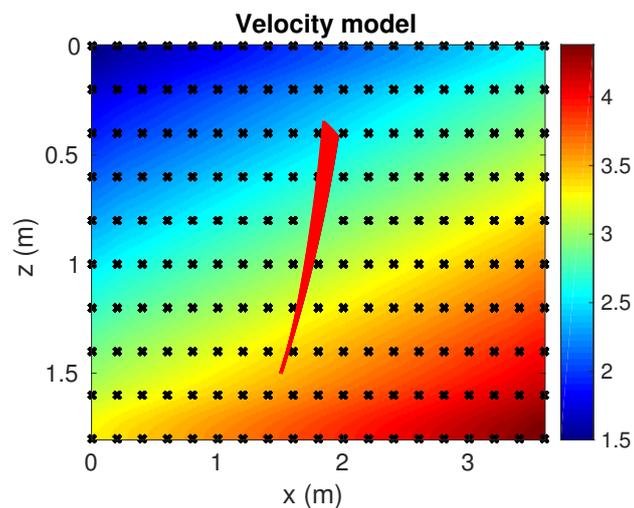


Figure 3.12: Velocity model with perfect ray (black) and rays traced with perturbed B-spline coefficient, close to the source point (red).

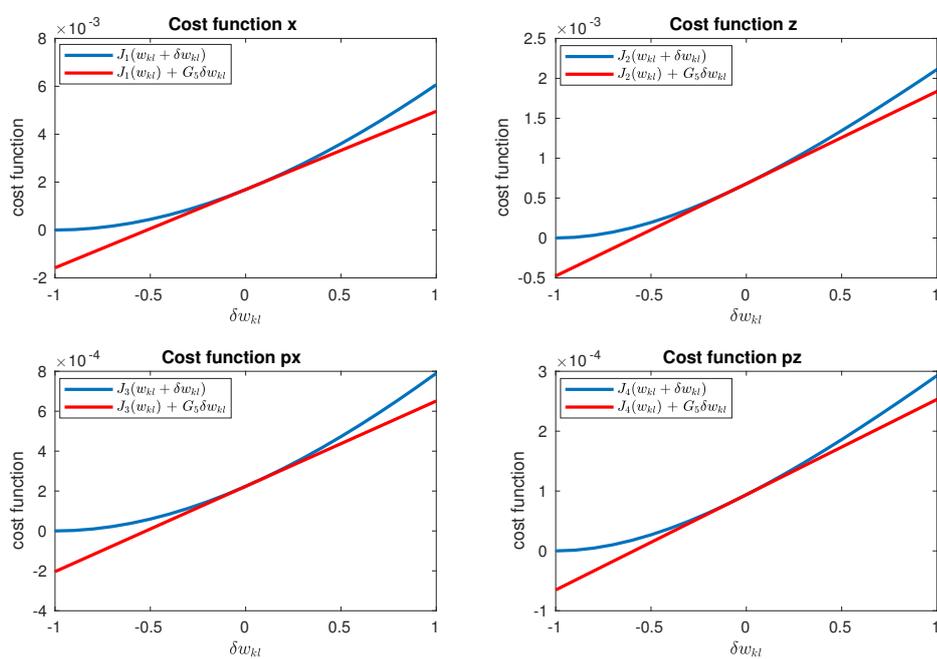


Figure 3.13: Finite difference test with the cost functions  $J_1$  (top left),  $J_2$  (top right),  $J_3$  (bottom left) and  $J_4$  (bottom right).

# Chapter 4

## Stereotomography implementation

Stereotomography may be implemented by a number of different techniques. In this chapter we discuss the practical aspects in our implementation of stereotomography. We present the techniques employed in the optimization and also the strategies adopted in the inversion. We also illustrate our implementation of stereotomography in two synthetic examples.

### 4.1 Practical aspects

The practical aspects of stereotomography implementation and testing may be challenging. Its first approach was in the stereotomography seminal work of Billette [1998]. In that work, stereotomography was first conceived to estimate perturbations in the velocity model. It was later, in the work of Billette et al. [2003], that the strategy of multi-step inversion was employed to estimate the velocity macro-model from a given initial velocity model. This strategy has been extensively employed in tests with synthetic and field data ever since this work [Lambaré et al., 2004, Le Bégat et al., 2004, Lambaré, 2008, Gosselet and Le Bégat, 2009, Prioux et al., 2013, Tavakoli et al., 2017]. The work of Billette et al. [2003] also approached practical aspects related to the picking stage of stereotomography. As previously discussed, the picking plays a crucial role in the inversion process, once that the local optimization adopted requires a good initialization to the problem, in order to avoid local minima. We discuss the picking problem in chapters 5 and 6.

Another particularity of stereotomography is that there are more model variables to determine than available data. This implies in the non-unicity of the solutions, since there might be several different velocity models explaining the same picked data. The way of dealing with this problem is by employing methods of regularization to choose a velocity model. This has been extensively discussed in the work of Costa et al. [2008], but also approached in the works of Gosselet and Le Bégat [2009], Prioux et al. [2013], Tavakoli et al. [2017].

Our stereotomography framework was implemented with the adjoint-state method [Plessix, 2006], explained in chapter 3, for the gradient computation and with the limited memory BFGS bounded (l-BFGS-b) method [Zhu et al., 1994, Byrd et al., 1995], explained in appendix C, as the optimization technique. We also tested other optimization techniques, such as the nonlinear conjugate gradient, explained in appendix B, but we choose to use the l-BFGS-b due to its good results and reasonable computational time. In the second synthetic example shown in this chapter, we present a comparison between the l-BFGS-b and the nonlinear conjugate gradient, where we can see the quality improvement obtained with the use of l-BFGS-b. The stereotomography framework was implemented in FORTRAN and in the next sections we explain our approach to the implementation of its stages. We finish this chapter with two synthetic examples which illustrate the potential of the implemented method.

## 4.2 Optimization

For the update of the stereotomography model parameters we tested methods based on the minimization of a cost function. We computed the gradient of this cost function with the adjoint-state approach and evaluated the performance of some of these methods.

Among the tested methods, the nonlinear conjugate gradient (CG) and the l-BFGS-b presented the more interesting properties. The main difference between them is that the nonlinear CG uses only first-order information given by the gradient, while the l-BFGS-b computes an approximation to the Hessian matrix. The l-BFGS-b seemed to be a better optimization strategy, due to the high-quality in the obtained results and to the reasonable computational time. Moreover, as indicated in Tavakoli et al. [2017], the use of the Hessian in the inversion seems very appropriate when dealing with different classes of parameters in the optimization.

For the nonlinear CG, we used our own implementation, employing the golden ratio to determine the step size and a preconditioner to speed-up the convergence. For the l-BFGS-b we used a well-known implementation, made available by Zhu et al. [1994] and Byrd et al. [1995]. In subsections 4.2.1 and 4.2.2 we briefly explain how to use this l-BFGS-b implementation and the preconditioning strategy adopted for the nonlinear CG implementation.

It is important to clarify that both l-BFGS-b and nonlinear CG are similar algorithms, but very sensitive to its implementations. The focus of this work is not to extensively compare different optimization algorithms for the stereotomography, but to find a reasonable one, which provides good results in a feasible time. In that sense, we tested our own nonlinear CG implementation and the l-BFGS-b implementation from Zhu et al. [1994]. These implementations are difficult to compare, once that in the nonlinear CG case we used, for the step size determination, the golden section line search

with a fixed number of iterations and in the l-BFGS-b implementation from Zhu et al. [1994], the model update and stop criterium are performed differently<sup>1</sup>. After extensive tests, we decided to use the l-BFGS-b implementation, because it presents fast convergence with a reasonable computational time.

### 4.2.1 l-BFGS-b implementation

Regarding the l-BFGS-b implementation, we used the FORTRAN implementation explained in Zhu et al. [1994] and Byrd et al. [1995]. The implementation operates as a black box: at each iteration the user must provide the value of the cost function being minimized and the vector with the gradient of parameters being updated and the software will provide the model vector with the updated parameters for the next iteration. This implementation is sensitive to a proper set of a few control parameters from the algorithm itself. The ones we modified were:

- **factr**: This test is designed to terminate the run when the change in the objective function is sufficiently small. Typical values for **factr** in double precision and in a computer with 15 digits of accuracy are:  $\text{factr} = 1 \times 10^{12}$  for low accuracy;  $\text{factr} = 1 \times 10^7$  for moderate accuracy and  $\text{factr} = 1 \times 10^1$  for extremely high accuracy.
- **pgtol**: This is a built-in stopping test based in the projection of the gradient onto the space tangent to the active bounds. The running program terminates when the infinity norm of the gradient becomes sufficiently small. It is advised by the authors to not set **pgtol** values smaller than the square root of the machine precision.

This algorithm does not stop in a given number of iterations, it runs until one of the stopping criteria is satisfied.

The first stopping test is designed to terminate the run when the change in the cost function being minimized,  $J(\mathbf{m})$ , is sufficiently small. This test finishes the program execution when

$$\frac{(J(\mathbf{m}_k) - J(\mathbf{m}_{k-1}))}{\max(|J(\mathbf{m}_{k-1})|, |J(\mathbf{m}_k)|, 1)} \leq \text{factr} \times \text{epsrch}, \quad (4.1)$$

with **epsrch** being the machine precision automatically generated by the code.

The second stopping test is based on the projected gradient [Bertsekas, 1982]. The test is design to terminate the run when the infinity norm of the projected gradient becomes too small:

$$\|\text{proj} \nabla J(\mathbf{m})\|_{\infty} \leq \text{pgtol}. \quad (4.2)$$

---

<sup>1</sup>We explain the criteria involved in the l-BFGS-b implementation in section 4.2.1 and in appendix C.

More details about the l-BFGS-b algorithm are given in the appendix C.

### 4.2.2 Preconditioning

An interesting approach to speed up the convergence of gradient-based methods is to use a preconditioner [Shin et al., 2001, Métivier et al., 2014]. Remember that  $N$  is the number of scatterer positions<sup>2</sup>,  $M$  is the total number of B-splines coefficients and consider  $N_m = 6N + M$  to be the total number of model parameters, in stereotomography. We define a precondition diagonal matrix  $\mathbf{P} \in \mathbb{R}^{N_m \times N_m}$ , which we try to make as close as possible to the Hessian inverse

$$\mathbf{P} \approx \mathbf{H}^{-1}. \quad (4.3)$$

In the precondition approach, we also define a new model

$$\tilde{\mathbf{m}}_k = \mathbf{P}^{-1} \mathbf{m}_k, \quad (4.4)$$

which we use in the optimization. The update is, then, performed with

$$\tilde{\mathbf{m}}_{k+1} = \tilde{\mathbf{m}}_k - \alpha_k \mathbf{P} \nabla J(\mathbf{m}_k). \quad (4.5)$$

At each iteration  $k + 1$ , the ray tracing is evaluated with the recovered model  $\mathbf{m}_{k+1} = \mathbf{P} \tilde{\mathbf{m}}_{k+1}$ .

#### A preconditioner suggestion

The gradient in the B-splines positions is highly affected by the number of rays crossing in these positions, resulting in gradient values with large variations according to the depth. We may consider the use of a preconditioning method to speed up the convergence, by scaling the gradient to equally update the B-splines parameters. We employed a simple preconditioning approach, which uses the depth information by multiplying the gradient vector for the B-splines coefficients by a reduced preconditioning diagonal matrix  $\mathbf{P}_{\text{red}} \in \mathbb{R}^{M \times M}$ . The elements of this matrix are

$$P_{\text{red},i} = z_i + \epsilon, \quad (4.6)$$

where  $z_i$  is the depth value at each B-spline coefficient and  $\epsilon$  is a small constant value added to avoid instabilities when computing  $\mathbf{P}_{\text{red}}^{-1}$ . Note that this matrix  $\mathbf{P}_{\text{red}}$  is a reduced version of the one defined in equation (4.3), applied only in the B-spline coefficients. In this approach, we do not apply any

<sup>2</sup>The model diffraction points, where the rays from source and receiver meet, are also known as scatterer positions.

preconditioner for the model parameters representing the scatterer positions, slopes and traveltimes. Even that this preconditioning matrix is not a precise approximation to  $\mathbf{H}^{-1}$ , as confirmed in our tests, using it as preconditioner might increase the stereotomography convergence.

The gradient preconditioning is an interesting approach that we used to speed up the convergence in the nonlinear conjugate gradient method. But when we started using the l-BFGS-b method, it did not significantly improve neither the computational time nor the results. We believe that this is mostly because the inverse of the Hessian matrix, estimated in the l-BFGS-b method, already acts as a sort of preconditioner.

### 4.3 Regularization

As previous discussed, the stereotomography may present multiple solutions, since different velocity models might explain the same picked data. In order to overcome this issue, regularization techniques might be used to penalize a given characteristic of the model being estimated. The regularization approach employed in this work is the Tikhonov regularization [Engl et al., 1996, Lopez, 2014]. This approach penalizes the square of the  $l_2$  norm of the velocity model derivatives, therefore prioritizing velocities distributions with smooth interfaces. Furthermore, we are interested in the low-frequency velocity macro-model, which presents low-magnitude derivatives.

The Tikhonov regularization operator penalizes the following equation

$$r(\mathbf{m}) = \frac{1}{2} \|\nabla v(x, z)\|_2^2. \quad (4.7)$$

This equation can be written as

$$r(\mathbf{m}) = \frac{1}{2} \int \int dx dz \left[ \left( \frac{\partial v(x, z)}{\partial x} \right)^2 + \left( \frac{\partial v(x, z)}{\partial z} \right)^2 \right]. \quad (4.8)$$

The velocity model may be written in terms of the B-Spline functions and coefficients as

$$v(x, z) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_z-1} w_{ij} B_i(x) B_j(z) \quad (4.9)$$

Replacing (4.9) in (4.8):

$$r(\mathbf{m}) = \frac{1}{2} \int \int dx dz \left[ \left( \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_z-1} w_{ij} \frac{\partial B_i(x)}{\partial x} B_j(z) \right)^2 + \left( \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_z-1} w_{ij} B_i(x) \frac{\partial B_j(z)}{\partial z} \right)^2 \right]. \quad (4.10)$$

In order to update the model with gradient-based techniques, it is necessary to compute the derivative of the regularization operator with respect to the B-spline coefficients,  $\partial r(\mathbf{m})/\partial w_{kl}$ . This derivative is given by:

$$\begin{aligned} \frac{\partial r(\mathbf{m})}{\partial w_{kl}} = & \int \int dx dz \left( \sum_i \sum_j w_{ij} \frac{\partial B_i(x)}{\partial x} B_j(z) \right) \frac{\partial B_k(x)}{\partial x} B_l(z) \\ & + \int \int dx dz \left( \sum_i \sum_j w_{ij} B_i(x) \frac{\partial B_j(z)}{\partial z} \right) B_k(x) \frac{\partial B_l(z)}{\partial z}. \end{aligned} \quad (4.11)$$

## 4.4 Inversion approach

The inversion approach used as way of solving the stereotomography optimization was the one presented in Billette et al. [2003] and recently discussed in Tavakoli et al. [2017]. We describe this approach in this section. This approach is divided into three major steps. In the first one, known as *initialization*, the picked data is modified in order to form an equivalent model, which is defined in Billette [1998] and is described in more detail in the sequel. This equivalent model presents initial values to the model parameters. In the second step, known as *localization*, only the slopes, positions and travel times of the equivalent model parameters, obtained from the picked data, are updated, while the velocity model is kept unchanged. The localization step assures a better initialization for the joint inversion and in most of the tests that we did involving picked data it was fundamental to avoid issues due to local minima points. The final step is the *joint inversion*, where all the velocities and picked parameters are jointly inverted.

### 4.4.1 Initialization: equivalent model

As illustrated in appendix A of Billette et al. [2003], the initial scatterer positions, slopes and traveltimes, defining part of the stereotomography model vector  $\mathbf{m}$ , can be directly computed from the picked data, by using a velocity model computed from known B-splines coefficients. These scatterer positions are known as *equivalent model* and are a good initial guess to the stereotomography optimization.

Besides that, as discussed in Billette [1998], we can also obtain the specular angle from the angles defined from the scattering positions to the source and receiver. The specular angle  $\theta$ , illustrated in Figure 4.1, is evaluated as

$$\theta = \frac{\theta_0^s + \theta_0^r}{2}. \quad (4.12)$$

We can obtain interesting information about the reflections dip and localization if we overlap the optimized velocity model with dip bars. These dip bars are lines plotted in the scatterer positions, with the specular angle inclination. For more information about the dip bars, please see the appendix A of

Billette et al. [2003].

As discussed in Billette [1998], a simple way to compute the equivalent model positions and angles is to assume a constant velocity model  $v_0$  and compute the scatterer positions as  $x_0 = \frac{x^s + x^r}{2}$  and  $z_0 = \frac{v_0 t_{sr}}{2}$ . In this approach, the initial angles are calculated using the surface slopes and the homogeneous velocity, with the following equations:  $\theta_0^s = \arcsin(p_x^s v_0)$  and  $\theta_0^r = \arcsin(p_x^r v_0)$ . Finally, the traveltimes in the model are computed with  $t_s = t_r = \frac{t_{sr}}{2}$ . This is a rough simplification, but the following model optimization can reallocate the scatterer points to better initial points.

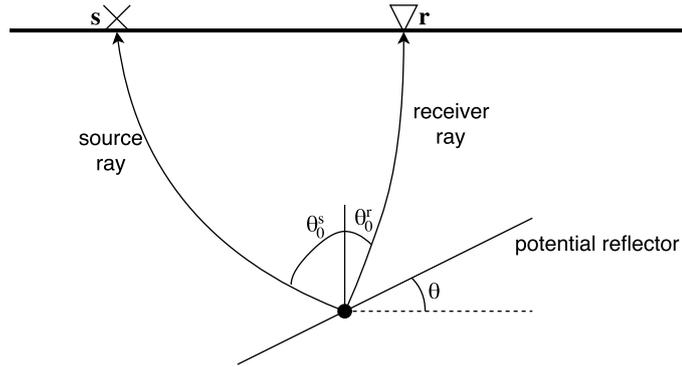


Figure 4.1: Equivalent model.  $\theta$  is the specular angle.

#### 4.4.2 Localization

This step is very important in order to relocate the scatterer positions from the equivalent model to as close as possible to its true positions. In the localization step, only the scatterer positions are updated, while the background velocity model is kept unchanged. We can either use a homogeneous velocity model or a constant gradient velocity model, as indicated in Tavakoli et al. [2017]. The background velocity model is often the initial velocity model that we want to update in the stereotomography.

#### 4.4.3 Joint inversion with multi-step optimization

In this inversion step, all the scatterer positions, slopes and travel times are jointly inverted with the B-splines coefficients representing the velocity model, by the use of a multi-step approach.

The stereotomography, as proposed in Billette [1998], was designed to invert models of the form  $v(x, z) = v_0 + v_1 z + \Delta v_{\text{B-splines}}$  in a optimization procedure with three stages. The successively obtained variables were  $v_0$ , which is a homogeneous velocity;  $v_1$ , which is a constant vertical gradient and  $\Delta v_{\text{B-splines}}$ , which is a velocity perturbation represented by the B-splines coefficients. In this

approach, all the velocity components might be jointly inverted with the scatterer positions, slopes and travel times.

Since the work of Billette et al. [2003], a multi-step scheme has been used for the stereotomography joint inversion. The multi-step approach, first used in the context of FWI [Bunks et al., 1995], is a progressive refinement of the spline parametrization. It consists in starting the inversion procedure with a coarse spacing in the B-splines coefficients, obtaining a velocity model which might be used as an input to a new inversion, performed with a thinner B-splines spacing. This procedure can be performed several times and presents the advantage of firstly obtain a coarse solution, which is refined at every multi-step stage.

## 4.5 Implementation tests

The stereotomography framework was implemented in the FORTRAN programming language, since it is a compiled language with good performance. In this section we describe some of the tests performed to evaluate the behaviour of this implementation.

The main goal of the two numerical examples illustrated here is to perform a sort of sanity check of the implemented stereotomography. We also used these tests to determine which optimization strategy to adopt, that is the reason why we analyzed the implementation behaviour in simple environments. Since the picking is the stage where we concentrate our proposals, we isolate its uncertainties by choosing the scatterer positions, slopes and traveltimes and computing the picked data with ray tracing. To keep the simplicity, we also run these examples only with the joint inversion stage, differently than the approach explained in section 4.4, where the initialization and localization stages are also employed.

In the first example we employed, as optimization tool, the nonlinear conjugate gradient with preconditioning and in the second example we performed a comparison between this method and the l-BFGS-b, in a velocity model slightly more complex. Some of the strategies described in this chapter are used in the more complex synthetic examples presented in chapter 6.

### 4.5.1 Constant gradient velocity model

In order to evaluate our stereotomography implementation, we performed one numerical experiment with a simple velocity model given by  $v(z) = 1.5 + z$  Km/s. We generated this velocity model and, then, we perturbed it, by adding a constant value of  $\delta v = 0.5$  Km/s. The idea is to check if we are able to recover the initial model by starting the stereotomography with the perturbed one. This is a simple test, but still interesting to evaluate our implementation. In this test, we used our first stereoto-

mography implementation, which employed as optimization tools the nonlinear conjugate gradient with the Polak-Ribiere approximation [Polak and Ribiere, 1969] and the golden ratio search method to determinate the step size (these methods are explained in appendix B). We also used the preconditioner explained in section 4.2.2. In this test, we performed the joint inversion of the B-splines coefficients and the scatterer positions, slopes and traveltimes. We did not use the initialization and localization inversion steps.

We described the velocity model with  $9 \times 17$  B-Splines coefficients, resulting in a total of 153 B-Splines coefficients. We assumed that the picking was given: the picked data was obtained from the ray tracing in the correct velocity model of  $v(z) = 1.5 + z$  Km/s, computed with the true scatterer positions, slopes and traveltimes. We computed 256 pairs of rays, which gave 1536 model variables. Thus, the total size of the model vector was 1689. These artificial picked rays resulted in a total of 1792 picked data variables. For the model initialization, we used scatterer positions, angles and traveltimes slightly displaced by constant values, in comparison to the true ones. The initial velocity model was given by the perturbed gradient  $v(z) = 2.0 + z$  Km/s.

In Figure 4.2 we illustrate the rays, used for the picked data computation, traced in the original synthetic velocity model ( $v(z) = 1.5 + z$  Km/s), with the correct model parameters. The B-splines positions are indicated by the black markers. In Figures 4.3(a) and 4.3(b) we show the difference between the initial and the final velocity models without and with preconditioning, respectively. The goal in this plot is to estimate the perturbation of  $\delta v = 0.5$  Km/s. We can see in this figure that the estimate was closer to the perturbation value in the case where we used preconditioning of the gradient, especially in the regions with small depths. In Figure 4.4 we show the cost function after 50 iterations of the nonlinear conjugate gradient with and without the use of preconditioning. It is possible to observe that the convergence was faster when we used the preconditioning

### 4.5.2 Perturbation in the velocity model

In this numerical experiment we test how our stereotomography implementation behaves in the presence of large variations in the velocity model. In order to do that, we superposed a high-velocity perturbation in the simple velocity model given by the depth constant gradient  $v(z) = 1.5 + z$  km/s. Our goal was to see which velocities would be recovered with the initialization given only by the constant gradient velocity model. In this test we did not use the three step inversion approach, discussed in section 4.4. Instead of that, we did a joint inversion of the scatterer parameters and the B-splines coefficients in a single step. We tested this approach with the nonlinear CG, computed with the Polak-Ribiere method, the depth preconditioning and the golden ratio step size search, and the l-BFGS-b optimization strategies. We did not apply any regularization in these tests.

We described the velocity model with  $9 \times 17$  B-splines coefficients, given a total of 153 velocity

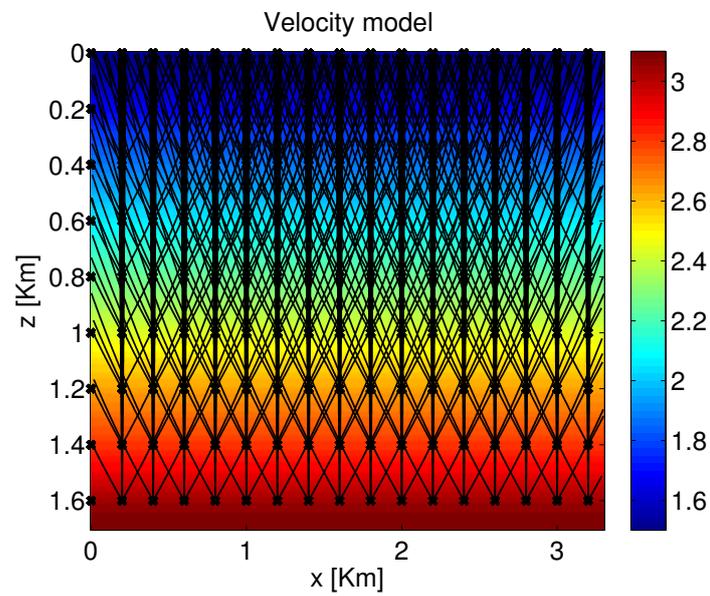


Figure 4.2: Rays traced in the original synthetic velocity model.

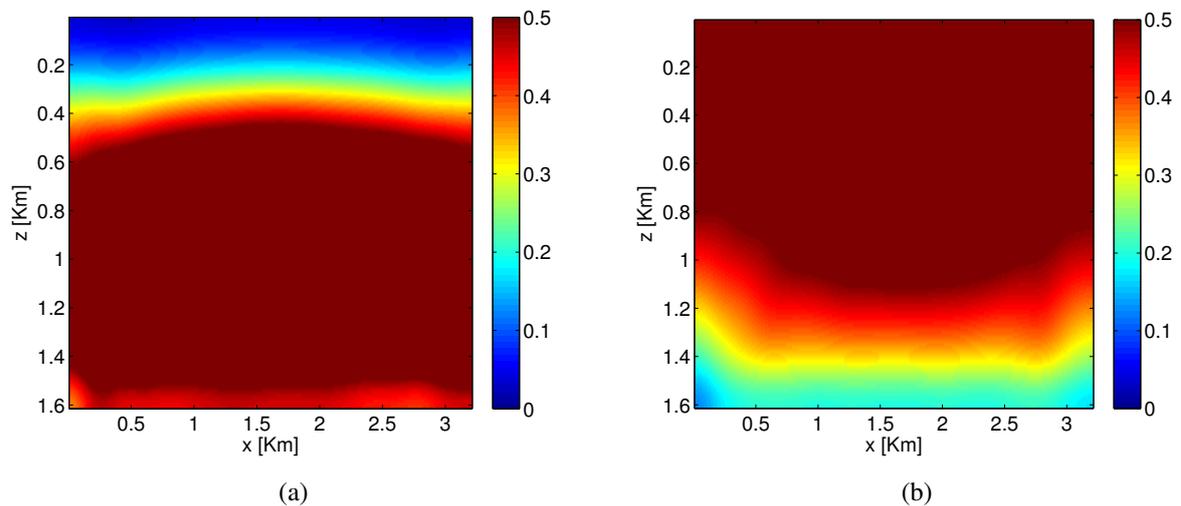


Figure 4.3: Difference between the initial and the final velocity models, in the attempt of estimating the perturbation without (a) and with (b) preconditioning.

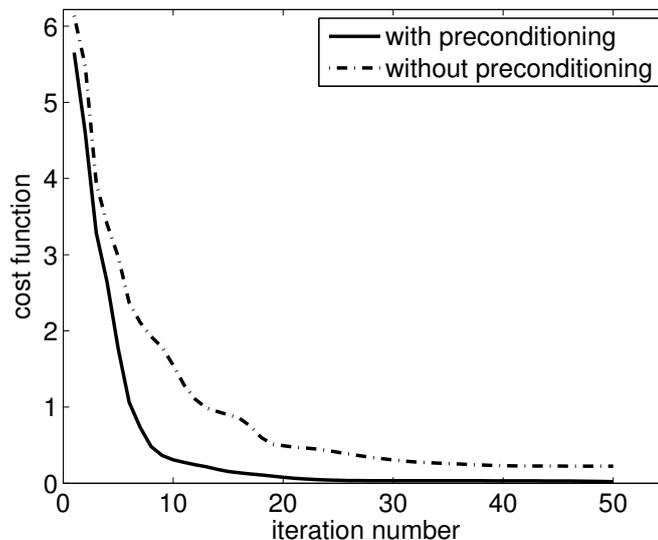


Figure 4.4: Cost function after 50 iterations of the nonlinear conjugate gradient with and without the use of preconditioning.

coefficients. We computed 96 pairs of rays and, again, assumed that the picking was given by the ray tracing evaluated in the velocity model that we wanted to recover (in this case, the perturbed one). The velocity initialization was given by the constant gradient in depth,  $v(z) = 1.5 + z$  km/s. Finally, in order to generate the scatterer positions initialization we also used a small displacement in the true scatterer positions and angles.

In Figure 4.5 we illustrate the rays traced in the perfect velocity model. The B-splines positions are indicated by the black markers. In Figure 4.6(a) we show the original velocity model and in Figure 4.6(b) we show the initial velocity model. In Figures 4.6(c) and 4.6(d) we show the final velocity models obtained, respectively, with the nonlinear CG and l-BFGS-b methods. It is possible to see that both the recovered velocity models are quite close to the original one, especially in the upper parts of the high-velocity perturbation, where there are more rays to compute the B-splines updating derivatives. In Figure 4.7 we show the cost function for the l-BFGS-b method and a zoomed region of the cost function for the nonlinear CG method. The nonlinear CG took 5000 iterations to converge, while the l-BFGS-b took only 300. We can also observe that the misfit is smaller for the l-BFGS-b method in the 300-th iteration.

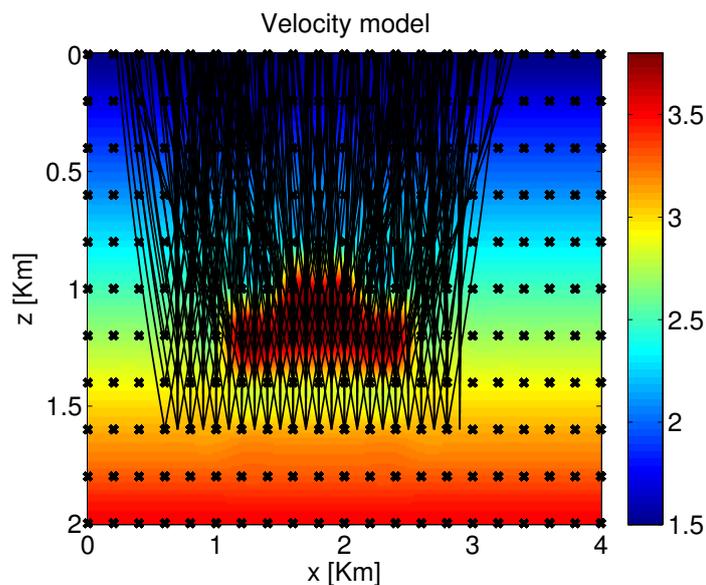


Figure 4.5: Rays traced in the perfect velocity model.

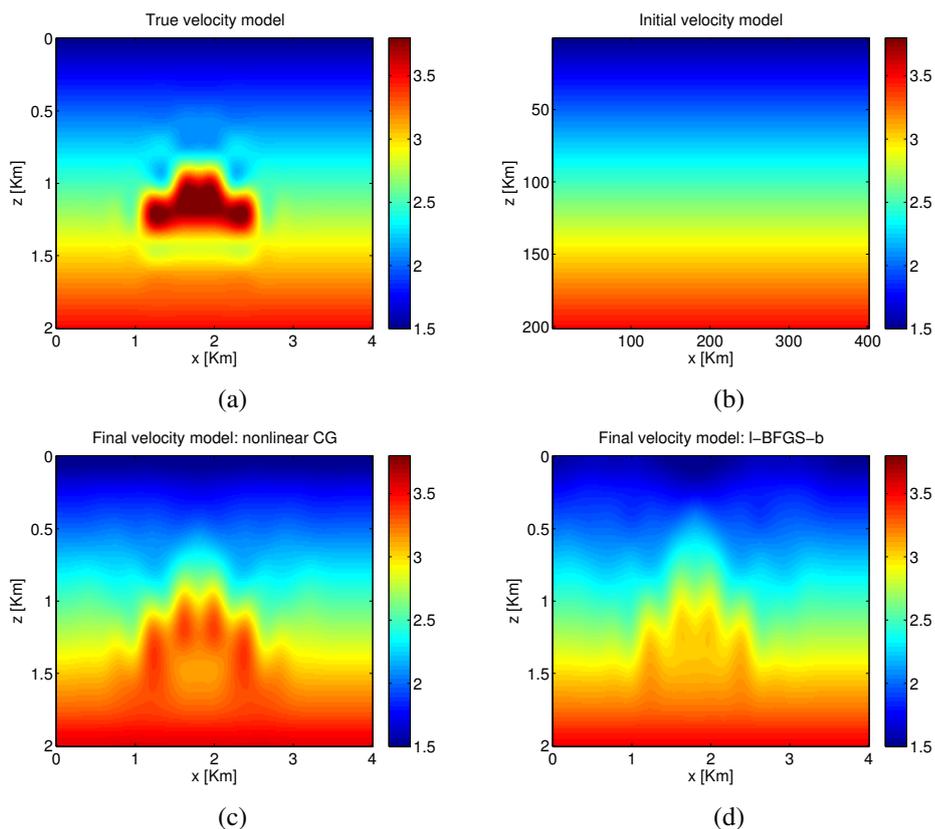


Figure 4.6: Original (a) and initial (b) velocity models. Velocity models recovered with the nonlinear conjugate gradient (c) and l-BFGS-b (d) methods.

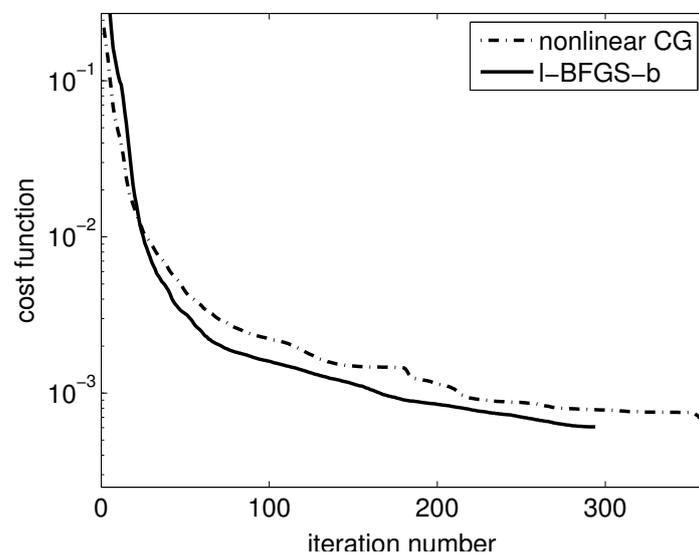


Figure 4.7: Cost function for the nonlinear CG and l-BFGS-b methods. The l-BFGS-b converged in 300 iterations, while the nonlinear CG took 5000 iterations to converge.

## Chapter 5

# Common-offset common-reflection-surface

The stereotomography method [Billette and Lambaré, 1998], employed to estimate the seismic macro-model of velocities in the subsurface, requires, as an input, the two slope values  $p_x^s$  and  $p_x^r$ , which are part of the two vectors,  $\mathbf{p}_s = [p_x^s, p_z^s]$  and  $\mathbf{p}_r = [p_x^r, p_z^r]$ . These vectors indicate the slowness of the seismic wavefront at the source and receiver. For both source and receiver:  $\mathbf{p}_s = [\sin \theta_s/v_s, \cos \theta_s/v_s]$  and  $\mathbf{p}_r = [\sin \theta_r/v_r, \cos \theta_r/v_r]$ , where  $v_s$  and  $v_r$  are the near surface velocities in source and receiver positions and  $\theta_s$  and  $\theta_r$  are the angles that the rays make with the normal in source and receiver positions.

The common-reflection-surface (CRS) method [Mann et al., 1999, Jäger et al., 2001] is an alternative method to the common-midpoint (CMP) stacking [Mayne, 1962], used to provide a simulated zero-offset (ZO) image of the subsurface in time. To stack each ZO trace, the CRS method uses more traces than the CMP, which allows a significant improvement in the signal-to-noise ratio (SNR) of the produced images. The CRS stacking operator approximates ZO reflection events in the vicinity of a ZO central ray [Höcht et al., 1999]. In 2D, the ZO-CRS stacking operator is parametrized by three attributes.

In Zhang et al. [2001], a generalization of the 2D ZO-CRS was introduced, with a new traveltime approximation, parametrized by five attributes. This generalization is known as common-offset (CO) CRS and can be applied in any CO section of the pre-stack data. The CO-CRS stacking operator approximates finite-offset (FO) reflection events in the vicinity of a given FO central ray. This generalization is particularly interesting once it allows to perform CRS stacking in pre-stack data, enabling the benefits of the SNR enhancement in other pre-stack processing flows.

One important proposal of this work is to use the attributes obtained as outputs of CO-CRS method with the global optimization method Differential Evolution [Barros et al., 2016] as input data for stereotomography. The strategy of using the CO-CRS attributes as input for the stereotomography was briefly signalized as a possibility in the introductory CO-CRS work of Zhang et al. [2001]. Re-

cently, the same discussion came to light in Schwarz et al. [2015], where it is suggested the utilization in stereotomography of the CO-CRS parameters, obtained with the ZO-CRS approximation proposed in Baykulov and Gajewski [2009].

Differently, in this work, we use the parameters obtained directly by the CO-CRS method. Moreover, we also indicate that there are cases where it is more appropriate to use the parameters obtained by the CO-CRS method with the global optimization algorithm DE, in order to have more accurate estimates, especially in cases where the signal is highly distorted by the presence of random noise.

In this chapter we explain the CO-CRS method. We begin by illustrating the method itself and then we discuss the sequential and global parameter estimation strategies. We then illustrate the behaviour of both these search strategies in synthetic and field data examples. In chapter 6 we analyze with more details the quality of the slopes estimated with the CO-CRS and the results obtained by the use of these slopes in stereotomography.

## 5.1 The method

Let  $x_0$  be the midpoint coordinate of the central ray. As usual, we associate each trace with a source-receiver pair with coordinates  $s$  and  $r$ , respectively. Alternatively, a trace may be identified by the midpoint  $x_m$  and the half-offset  $h$  of the source-receiver pair. In this case, the coordinates are related as  $s = x_m - h$  and  $r = x_m + h$ . For the midpoint and offset coordinates in the central point given by  $(x_0, h_0)$ , the 2D<sup>1</sup> hyperbolic CO-CRS formula can be defined as

$$\begin{aligned}
T^2(\Delta x_m, \Delta h) &= \left[ t_0 + \left( \frac{\sin(\theta_r)}{v_r} + \frac{\sin(\theta_s)}{v_s} \right) \Delta x_m + \left( \frac{\sin(\theta_r)}{v_r} - \frac{\sin(\theta_s)}{v_s} \right) \Delta h \right]^2 \\
&+ t_0 \left[ (4K_1 - 3K_3) \frac{\cos^2(\theta_r)}{v_r} - K_2 \frac{\cos^2(\theta_s)}{v_s} \right] \Delta x_m^2 \\
&+ t_0 \left[ K_3 \frac{\cos^2(\theta_r)}{v_r} - K_2 \frac{\cos^2(\theta_s)}{v_s} \right] \Delta h^2 \\
&+ 2t_0 \left[ K_3 \frac{\cos^2(\theta_r)}{v_r} + K_2 \frac{\cos^2(\theta_s)}{v_s} \right] \Delta h \Delta x_m
\end{aligned} \tag{5.1}$$

where  $\Delta x_m = x_m - x_0$  and  $\Delta h = h - h_0$  are the midpoint and half-offset displacements, respectively. The quantities  $v_s$  and  $v_r$  are, respectively, the near-surface velocities at the source and receiver positions, which are related to the central ray. Among the five CO-CRS parameters, which are determined from the pre-stack data, are the two angles of incidence between the central ray and the  $z$ -axis at the source,  $\theta_s$ , and receiver,  $\theta_r$ . The other parameters are the three wavefront curvatures  $K_1$ ,  $K_2$  and

<sup>1</sup>There is also a 3D extension of this method [Hoecht et al., 2009, Müller et al., 2013].

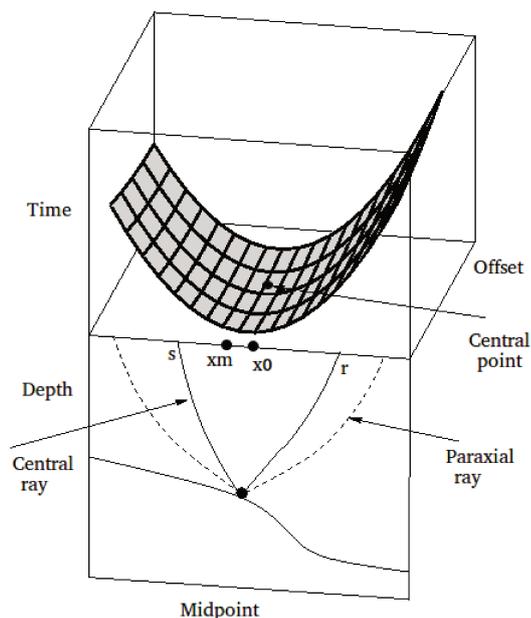


Figure 5.1: CO-CRS surface defined by its traveltime.

$K_3$ , which are associated with the central ray from source to receiver. In this work, we focus in the estimation of the two angles,  $\theta_s$  and  $\theta_r$ , therefore we do not provide details about the parameters  $K_1$ ,  $K_2$  and  $K_3$ . More information about the CO-CRS curvatures can be found in Bergler [2001]. The five CO-CRS parameters are also known as CO-CRS attributes.

In Figure 5.1 we see the surface defined by the CO-CRS traveltime and how it is related with a reflection in depth. This image is similar to the one illustrated in Zhang et al. [2001].

The CO-CRS attributes are estimated from the pre-stack data by means of coherence analysis, being the semblance [Neidell and Taner, 1971] the most used coherence measure. In Zhang et al. [2001], it is proposed to estimate the five CO-CRS attributes in simplified versions of the travel-time equation (5.1), obtained in the common-shot (CS), common-offset (CO) and common-midpoint (CMP) gathers. We refer this search strategy as *sequential CO-CRS search*.

Alternatively, the CO-CRS attribute search may be performed simultaneously. Recently, several studies about using alternative heuristics that speed up the estimation time for the CRS global strategy were presented for the ZO-CRS [Garabito et al., 2012] and CO-CRS [Garabito et al., 2013] methods. The main idea behind the global search is that the coherence is evaluated by the simultaneous variation of the five attributes. This strategy is referred to as the *global CO-CRS search*. The procedure of estimating the CO-CRS attributes is then performed, independently, at each point defined by the time

$t_0$  and the coordinates  $(x_0, h_0)$ .

The simultaneous search for all the five parameters may be too time-consuming. On the other hand, the parameters obtained by the global search may be more accurate, as discussed for the ZO-CRS method in Barros et al. [2013]. In Barros et al. [2015], a bio-inspired heuristic, named differential evolution (DE) [Storn and Price, 1997], was employed to estimate the ZO-CRS attributes. There, it is shown that this heuristic presents a good convergence rate, simple parametrization and fast convergence, when compared to other heuristics.

## 5.2 Sequential search

The sequential search of the CO-CRS parameters is the traditional approach for the CO-CRS stacking [Zhang et al., 2001]. This approach is explained in details in the work of Bergler [2001]. It consists of consecutive searches of a small number of parameters in different seismic gathers. The main idea behind this approach is that, at each gather, some assumptions are made for the traveltimes simplification, allowing the application of the following traveltimes operator:

$$T^2(y) = (t_0 + ay)^2 + by^2. \quad (5.2)$$

This equation can be referred to as *general search equation* for the sequential CO-CRS, since the quantities  $a$ ,  $b$  and  $y$  are related to different parameters at each seismic gather. The variable  $y$  is related to the traces coordinate in the seismic gather where the attributes search takes place and the quantities  $a$  and  $b$  are related to different CO-CRS attributes being searched in the given seismic gather. The search for the parameters consist in testing several parabola, constructed with the pairs of parameters  $a$  and  $b$  being tested, and choosing the one which maximizes a coherence measure, being the semblance [Neidell and Taner, 1971] the most used one. The search for the parameters  $a$  and  $b$  is performed with a brute force method in pre-defined grids of values.

The sequential strategy is employed in common-midpoint (CMP), common-offset (CO) and -shot (CS) gathers, allowing the estimation of:

- 2 parameters in CMP gather.
- 2 parameters in CO gather.
- 1 parameter in CS gather.

In Figure 5.2 we see a diagram with the central point coordinates and how these three gathers are disposed in a *midpoint-offset* seismic coordinates plane. In the next subsections, we can see with more details the sequential search strategy.

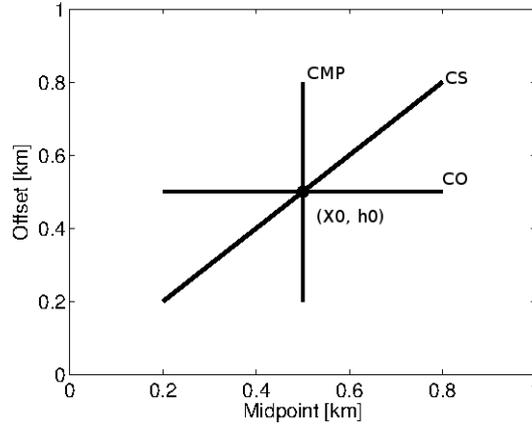


Figure 5.2: CO-CRS sequential search diagram.

### 5.2.1 CMP search

For the parameters search in the CMP gathers, we can use the fact that  $\Delta x_m = 0$ , yielding the traveltimes operator:

$$T^2(y) = (t_0 + a_{\text{CMP}}y)^2 + b_{\text{CMP}}y^2. \quad (5.3)$$

In the CMP gathers we have, then, to search for the  $a_{\text{CMP}}$  and  $b_{\text{CMP}}$  parameters, considering  $y = \Delta h = h - h_0$ . This search for the parameters  $a_{\text{CMP}}$  and  $b_{\text{CMP}}$  can either be done simultaneously or independently. In the latter case, we first search for  $a_{\text{CMP}}$ , by using  $b_{\text{CMP}} = 0$  in equation (5.3), and then search for  $b_{\text{CMP}}$ , by using the whole operator in equation (5.3). In this work we perform the bi-parametric simultaneous search for the CMP parameters. The CMP parameters are related to the CO-CRS parameters as:

$$a_{\text{CMP}} = \frac{\sin \theta_r}{v_r} - \frac{\sin \theta_s}{v_s}, \quad (5.4)$$

and

$$b_{\text{CMP}} = t_0 \left[ K_3 \frac{\cos^2 \theta_r}{v_r} - K_2 \frac{\cos^2 \theta_s}{v_s} \right]. \quad (5.5)$$

### 5.2.2 CO search

In the CO gathers, we have that  $\Delta h = 0$ . The traveltimes operator used in the CO parameters search is, then:

$$T^2(y) = (t_0 + a_{\text{CO}}y)^2 + b_{\text{CO}}y^2, \quad (5.6)$$

with  $y = \Delta x_m = x_m - x_0$ . In this gather, the search for the parameters  $a_{\text{CO}}$  and  $b_{\text{CO}}$  is performed individually, by searching one parameter at each time. First, we determine  $a_{\text{CO}}$  by using  $b_{\text{CO}} = 0$  in equation (5.6) and then we use this estimated value and search for  $b_{\text{CO}}$  by applying the complete

operator of equation (5.6). The CO parameters are related to the CO parameters as:

$$a_{\text{CO}} = \frac{\sin \theta_r}{v_r} + \frac{\sin \theta_s}{v_s}, \quad (5.7)$$

and

$$b_{\text{CO}} = t_0 \left[ (4K_1 - 3K_3) \frac{\cos^2 \theta_r}{v_r} - K_2 \frac{\cos^2 \theta_s}{v_s} \right]. \quad (5.8)$$

### 5.2.3 CS search

The last search for the sequential strategy of estimating the CO-CRS parameters is performed in common-shot gathers. The CS traveltimes operator is defined by

$$T^2(y) = (t_0 + a_{\text{CS}}y)^2 + b_{\text{CS}}y^2, \quad (5.9)$$

where  $y = \Delta x_g = \Delta x_m + \Delta h$ . The CS parameters are given by

$$a_{\text{CS}} = \frac{\sin \theta_r}{v_r} = \frac{a_{\text{CMP}} + a_{\text{CO}}}{2} \quad (5.10)$$

and

$$b_{\text{CS}} = t_0 \left[ K_1 \frac{\cos^2 \theta_r}{v_r} \right]. \quad (5.11)$$

The CS search involves only one-parameter search, once that, as we can see in equation (5.10), the parameter  $a_{\text{CS}}$  is given by the combination of parameters obtained in the CMP and CO searches.

After we obtain the parameters from all the steps of the sequential search we can apply the operator from equation (5.1) to obtain the data enhanced by the CO-CRS technique. We can also obtain a coherence gather with this operator as a way of quality control for the parameters estimation. We also obtain coherence gathers at each different search performed in the sequential strategy.

## 5.3 Global search

As discussed in Barros et al. [2015] for the ZO-CRS case, probably the most obvious drawback of the sequential search is that it does not find the globally optimal estimates of the parameters, in the sense of maximizing the coherence for the CRS traveltimes. This may have some serious consequences, both in the values of the parameters themselves and on the quality of the result. To deal with these problems, different strategies to simultaneously estimate all the ZO-CRS parameters in a single search procedure have been proposed [Garabito et al., 2012, Barros et al., 2015]. As

discussed in these works, the joint estimation of the CRS parameters may lead to better results than those from the sequence of one-parameter searches, with the drawback of higher computational costs.

In Barros et al. [2015], we provide practical insights to guide the implementation of an efficient optimizer for the ZO-CRS global search problem. The tests performed with synthetic and real data surveys have indicated that among the natural computing algorithms applied to optimization [de Castro, 2007], the original DE algorithm is an attractive approach for the ZO-CRS fully automatic global search.

Regarding the CO-CRS case, the global strategy has been discussed by Garabito et al. [2013], where the *very-fast simulated annealing* (VFSA) algorithm [Sen and Stoffa, 1995] is used as global optimization algorithm. In Barros et al. [2016], we have proposed to use the *differential evolution* (DE) global optimization algorithm [Storn and Price, 1997] to estimate the parameters of the CO-CRS traveltime. Next, we present some details and practical insights to the use of DE algorithm in the global CO-CRS parameters search context.

### 5.3.1 Differential Evolution

Differential Evolution (DE) is a parallel direct search method for continuous space variables which utilizes  $N_P$   $D$ -dimensional parameter vectors

$$\mathbf{x}_{i,G}, \quad i = 1, \dots, N_P \quad (5.12)$$

as population, on each generation<sup>2</sup>  $G$ . The initial vector population is chosen randomly and should cover the entire search space to guarantee diversity among individuals of the population. In the absence of knowledge on the search space, one may use an uniform statistical distribution to generate the initial population. DE generates new parameter vectors by adding the weighted difference between two randomly chosen population vectors to a third randomly chosen vector. This operation is known as *mutation*. A *trial vector* is generated by the combination between the mutated parameter vectors and another randomly predetermined vector, the *target vector*. This combination is referred to as *crossover*. It is important that each vector in the population be used once as the target vector, so that  $N_P$  parameter vectors comparisons take place in one generation. The last operation of the DE algorithm is the *selection*, where the target vector is replaced by the trial vector if the value of the cost function for the trial vector is greater than that of the target vector. (This holds for the maximization problem. For the minimization problem the search is for the lower value.) For the CO-CRS global search, we use the semblance measure as cost function and the parameters vector contains the CO-CRS attributes  $\{\theta_s, \theta_t, K_1, K_2, K_3\}$ , with length  $D = 5$ . For the operation of DE algorithm, the user

<sup>2</sup>The generation is an analogy to a given iteration in the optimization algorithm.

must set the algorithm parameters  $G$ ,  $N_P$  and also the parameters related to the crossover ( $CR$ ) and mutation ( $F$ ). We explain with more details these parameters, the DE algorithm and also the mutation, crossover and selection operations in Appendix D.

### Practical insights

In our tests, we noticed that a reasonable value for the number of individuals in the population  $N_P$  is between  $5D$  and  $10D$ , where  $D$  is the dimensionality of the search problem. For two-dimensional data, the number of parameters to be searched is  $D = 5$ , so that the recommended interval is  $25 < N_P < 50$ . The number of generations,  $G$ , is usually employed as a stopping criterion, and limits the DE iterations number. The solution is more likely to be close to the global optimum for large values of  $G$ . We suggest that  $G$  be chosen to be at least 150.

Regarding the mutation in DE, large  $F$  values increase the algorithm capacity to explore parameters in large regions within the search limits. In our experiments we noted that a good choice for  $F$  lies in the interval  $[0.4, 1.3]$ . We also have noted that the DE sensitivity in finding the global optimum with respect to this control parameter is not as significant as for the crossover rate. In other words, the results are robust to the choice of  $F$ , which does not need to be carefully fine-tuned.

As noted by Das and Suganthan [2011], a low  $CR$  value (e.g.,  $CR < 0.1$ ) results in a search that changes each direction of the mutated vectors separately. In practice it benefits functions that are separable in terms of dimensions. This is not the case of the semblance measure as a function of the CRS parameters. Even so, we have noted that small crossover rates may be a good choice for very low SNR data. We recommend the interval  $[0.1, 0.9]$ . Although this range seems to be large, DE is quite robust in finding the global optimum if enough generations  $G$  are allowed to be executed, as illustrated for the ZO-CRS in Barros et al. [2015].

For further information about DE and other evolutionary computing algorithms, please see Rainer [1995] and Storn and Price [1997] as well as the references in the overview article from de Castro [2007].

## 5.4 Numerical examples

In this section we present some results that demonstrate the application of the selected DE global optimization algorithm in the CO-CRS global search problem. We applied this method in both synthetic and field data sets. For the synthetic case we also compared the global optimization strategy with the sequential search strategy explained in previous sections. The field data set used in our tests is from a land acquisition, with very low fold. Due to that, we choose only to use our proposed global

optimization strategy, since this method is quite data-driven and may present good results with simple algorithm parametrizations.

### 5.4.1 Synthetic Marmousi data

In the synthetic tests, we use the Marmousi data set [Bourgeois et al., 1991], which presents interesting complexity. We added random noise to the pre-stack data, in order to obtain a low signal-to-noise ratio (SNR). In the first test, we used the CO gather with the offset coordinate of  $-1000$  m. In Figures 5.3(a) and 5.3(b), we present the raw and noisy versions of this CO gather, respectively. This data set presents a time sample interval of  $dt = 0.004$  s. The offset and midpoint intervals are 50 m and 25 m, respectively.

For the sequential search, we used apertures of  $\Delta h^{\max} = 250$  m and  $\Delta x_m^{\max} = 50$  m both for the searches and stacking operations. The searches in all gathers were performed with time windows of 11 samples and grids of 101 values for the brute parameters search. The search in the CMP gather was bi-parametric, as described in the subsection 5.2.1. For all the tests with the global search, we parametrized the DE algorithm with  $N_P = 50$ ,  $CR = 0.3$ ,  $F = 0.7$  and  $G = 200$  iterations. For the global search, we used apertures of  $\Delta h^{\max} = 150$  m and  $\Delta x_m^{\max} = 150$  m. The apertures were chosen by trial and error and one important difference between the global and the sequential searches is that while, in the global search, the apertures define the limits of a two-dimension surface, in the sequential searches the limits are only in a one-dimension travelttime equation. It is interesting to note that if we use the same aperture values for the sequential and global approaches, in the global approach we would be using a larger number of traces in the search and stacking operations. For instance, in this data set, if in the CMP search we use  $\Delta h^{\max} = 250$  m, it means that we are using 11 traces in this search. In other hand, in the global search, using apertures of  $\Delta h^{\max} = 150$  m and  $\Delta x_m^{\max} = 150$  m, which are even smaller than 250 m, means using 91 traces in both search and stacking operations. At a first glance, the apertures in the sequential search may seem small. It is important to clarify that the Marmousi data presents strong horizontal variations in the velocity model, which caused significant estimation errors in the sequential CO-CRS whenever we attempted to use large apertures. Besides that, the sequential approach presents the drawback of stacking in a surface of parameters given by the travelttime in equation (5.1), which is not the same as the one used to estimate the parameters and given by the travelttime in equation (5.2). Because of that, we tried to minimize errors by using the same apertures in the estimation and stacking procedures.

At each stage of the sequential search we produce a stacking and a coherence section, which are very useful for quality control. Moreover, we also produce gathers with the parameters  $a$  and  $b$ , which are later used for stacking. In Figures 5.4(a), 5.4(b), 5.4(c), 5.4(d), 5.4(e) and 5.4(f) we show the stacking and coherence sections produced in the bi-dimensional CMP search and in the CS and

CO searches. In Figures 5.5(a), 5.5(b), 5.5(c) and 5.5(d) we show the final stacking and coherence sections from the sequential search compared with the same gathers from the global search. It is possible to see the benefits of the global search in terms of better image definition, due to a higher SNR, when compared to the sequential search. In Figure 5.6 we show again, only for comparison purposes, the raw data and the enhanced CO-CRS results for the sequential and global searches. Finally, in Figures 5.7(a), 5.7(b) and 5.7(c) we show the same displays for five different CMP gathers. It is possible to observe from the figures a significant SNR enhancement after the CO-CRS correction. Also, the reflections are better defined in the CO-CRS corrected images.

It is possible to observe that, in the presence of random noise, the CO-CRS method may present better results if we use a global search for the parameters estimation. But this global estimation may present higher computational costs, when compared to the currently used sequential estimation method. We now illustrate how this complexity could be analyzed in the parameters estimation of only 1 central point, for both sequential and global methods. Assume that, for this central point, in a global search, we set the control DE parameters with the values  $N_P = 50$  and  $G = 200$ . As we have to compute the coherence for all the individuals of the population, at each iteration, this implies in computing  $50 \times 200 = 10,000$  coherence functions. For the sequential search, assuming that we set 100 search points at each gather, we have two options: we can either use a bi or mono-parametric search in the CMP search. If we use a bi-parametric search, this implies in  $100 \times 100 = 10,000$  computations. On the other hand, in a mono-parametric search, we have only  $100 + 100 = 200$  coherence evaluations. In the CS search, we have  $100 + 100 = 200$  coherence evaluations and in the CO search we have only 100 coherence evaluations. The total would be, then, 10,300 coherence evaluations if we choose to use the bi-parametric CMP search and only 500 if we choose the mono-parametric. It is important to emphasize that, in our tests with noisy data sets, the mono-parametric search does not present a satisfactory behaviour. Also, if we choose to use the bi-parametric search, the global search would present the same computational costs as the sequential search, possibly with better results and with easier input algorithm variables parametrization.

### 5.4.2 Field data

In these tests with real data sets, we used a seismic line acquired in Amazônia, Brazil, which presents low SNR, reflectors with large dip variations and, at most, 12 traces in each CMP. In this numerical example we only evaluated the global CO-CRS method, in order to discuss the benefits of our proposal.

The global estimation of CRS parameters was performed with the DE algorithm, and we used  $CR = 0.3$ ,  $F = 0.7$ ,  $N_P = 50$ , and  $G = 200$ . In Figure 5.8(a) we show the raw CO with offset  $h = -2000$  m and in Figure 5.8(b) we show the same offset after the global CO-CRS method. It is

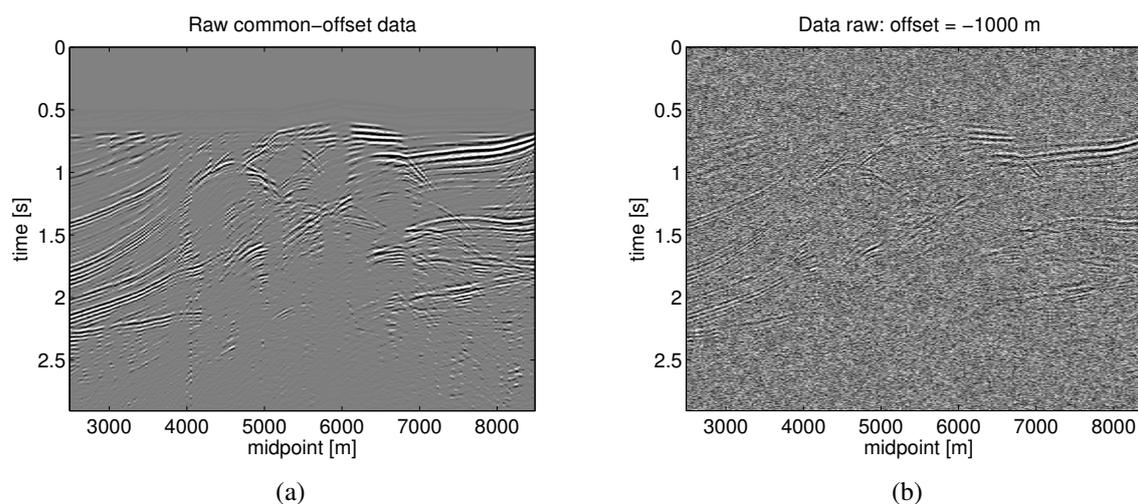


Figure 5.3: Marmousi data set with offset of  $-1000$  m. Raw (a) and noisy (b) data used for the tests.

possible to observe that the global CO-CRS method allows us to see more geologic information in the data, especially in regions with large dips. This is mostly because the parameters in the global CO-CRS search are estimated and stacked with a large number of traces, as discussed for the ZO-CRS case in Barros et al. [2015]. This fact can lead to a significant increase in the quality of the results for data sets with low SNR and low fold. For visualization purposes, we show, in Figure 5.9(a), five raw CMPs from the same data. In Figure 5.9(b) we show the same CMPs enhanced by the application of the DE CO-CRS method. It is also possible to observe that these CMPs present a significant increase in the SNR and that they are also regularized<sup>3</sup>, with traces created where there were no traces before.

For an interesting glimpse of the CO-CRS benefits we show, in Figure 5.10(a), the velocity spectrum from one raw CMP gather. In Figure 5.10(b) we show the same spectrum after the application of the global CO-CRS method. It is possible to observe that the reflection events are much better defined by the semblance function after the application of the global CO-CRS method. It is possible to even follow a velocity trend in that spectrum, which was very hard to do in the spectrum computed directly from the raw data.

<sup>3</sup>As pointed in Hoecht et al. [2009], the CO-CRS method presents the capability of regularize the data, by using the neighbouring information to create traces where there were missing ones, due to irregular acquisition geometries.

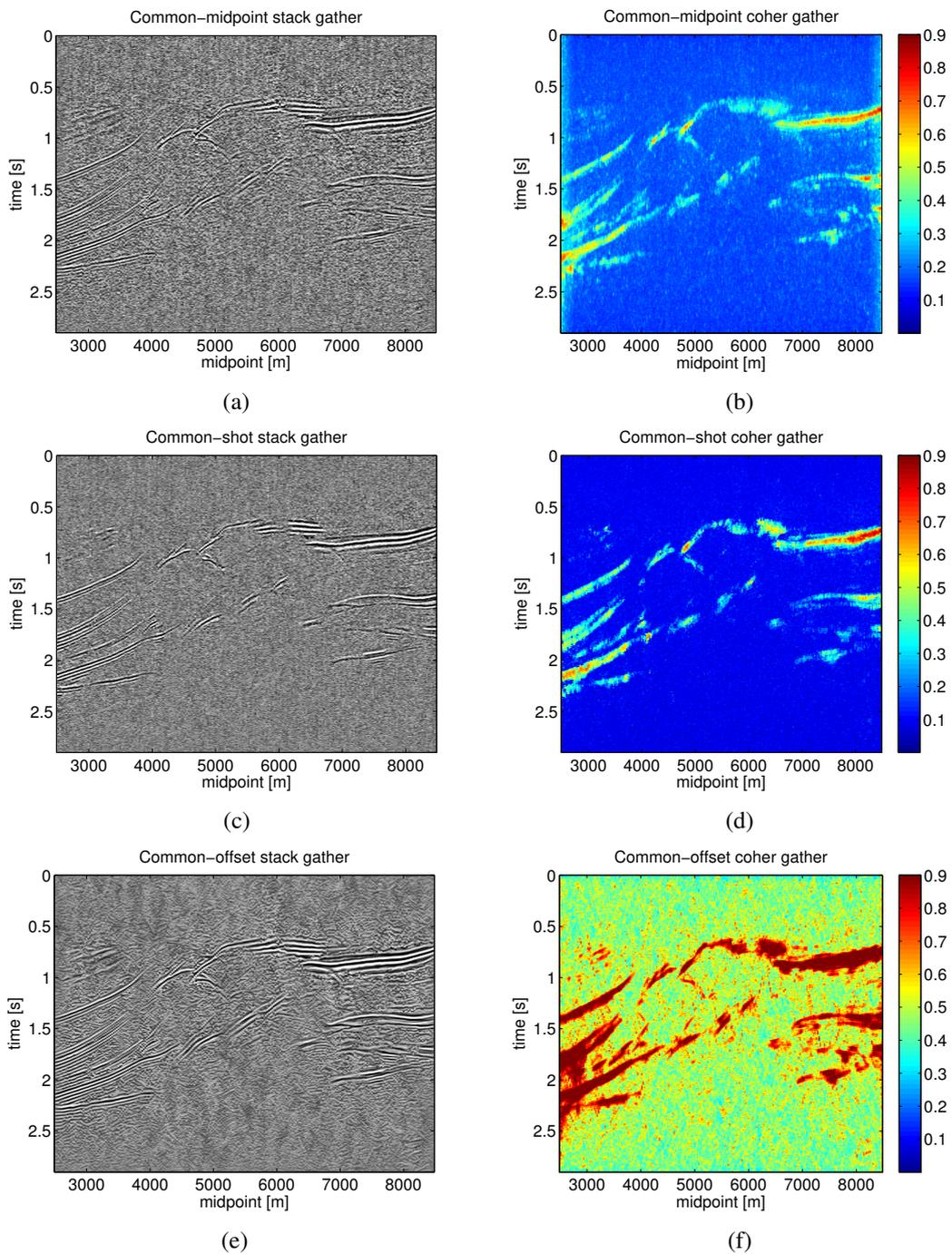


Figure 5.4: Sequential CO-CRS: CMP stacking (a) and coherence (b) gathers. CS stacking (c) and coherence (d) gathers. CO stacking (e) and coherence (f) gathers.

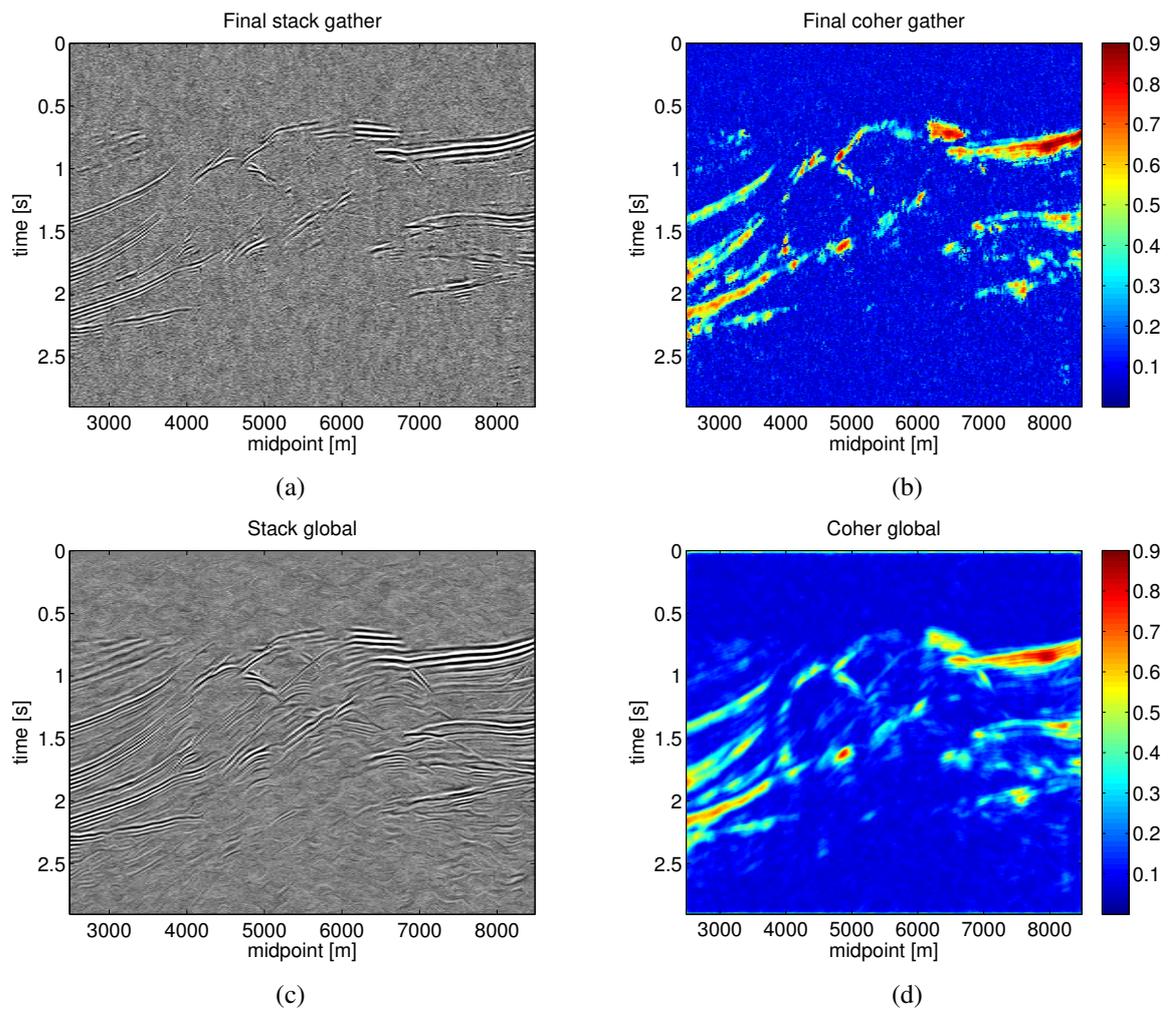


Figure 5.5: Sequential CO-CRS final stacked (a) and coherence (b) sections. Global CO-CRS stacked (c) and coherence (d) sections.

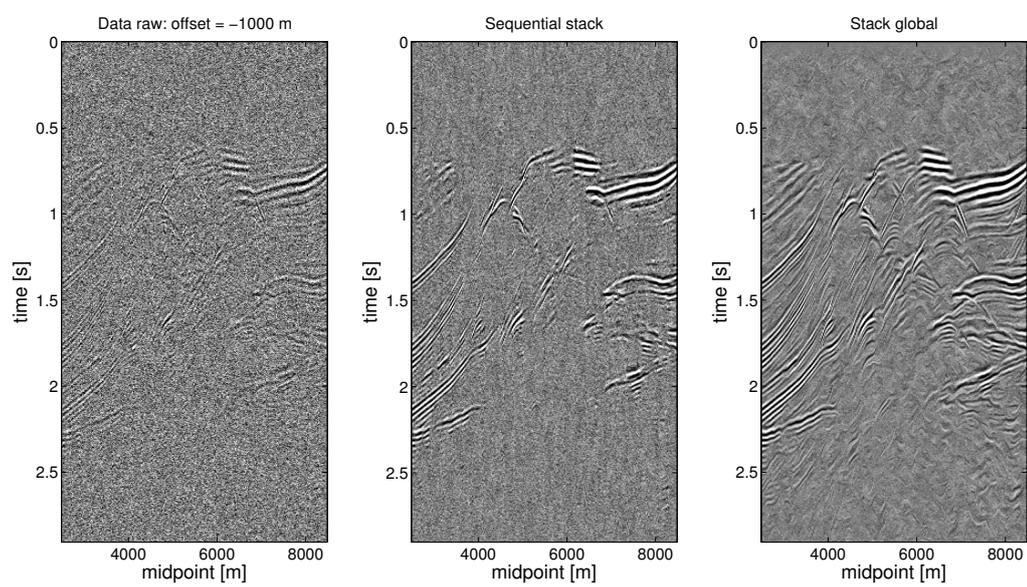


Figure 5.6: Marmousi CO gather. Raw data (left) enhanced sequential (middle) and global (right) CO-CRS.

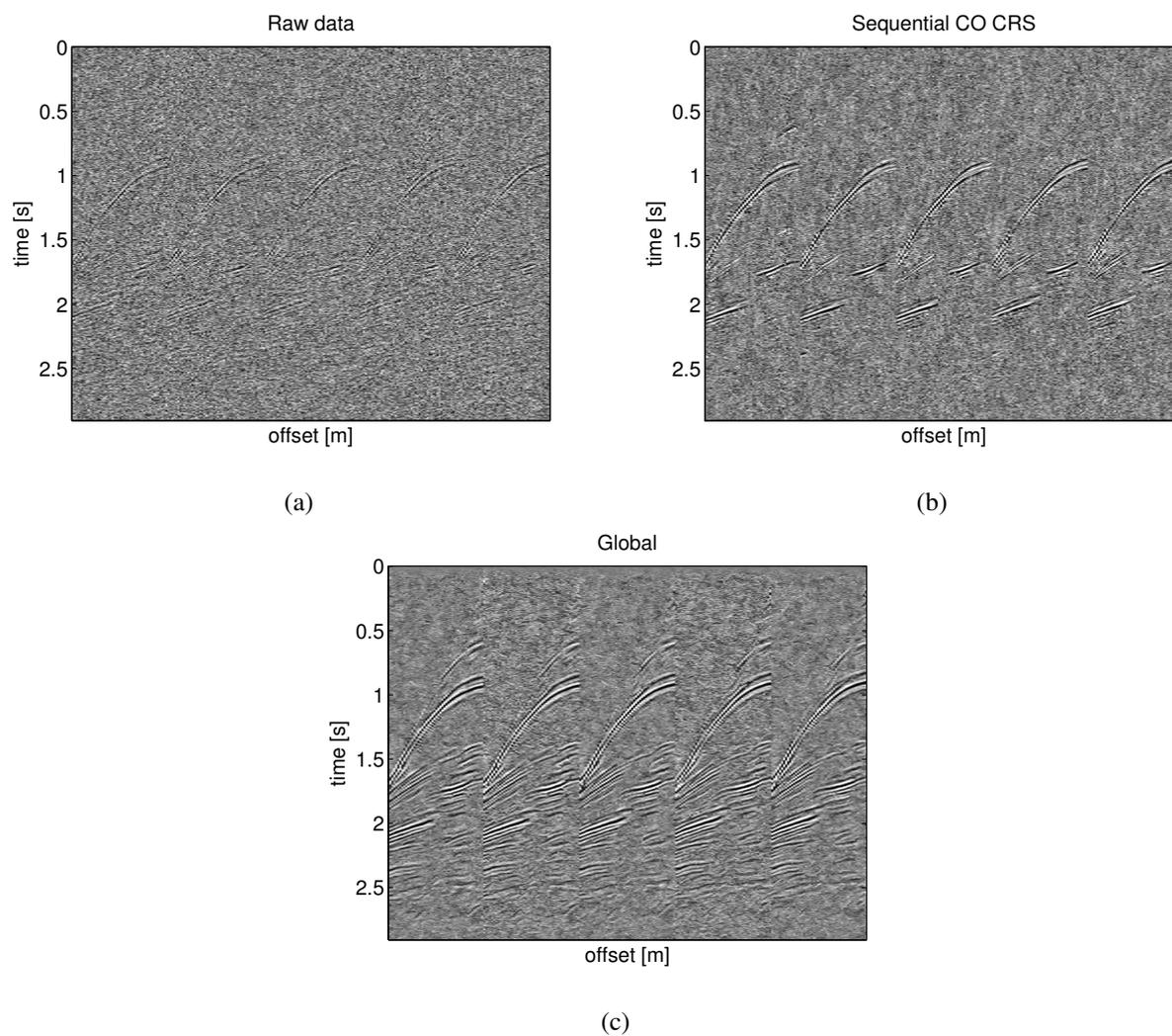


Figure 5.7: Raw CMP data (a); sequential enhanced CMP data (b); global enhanced CMP data (c).

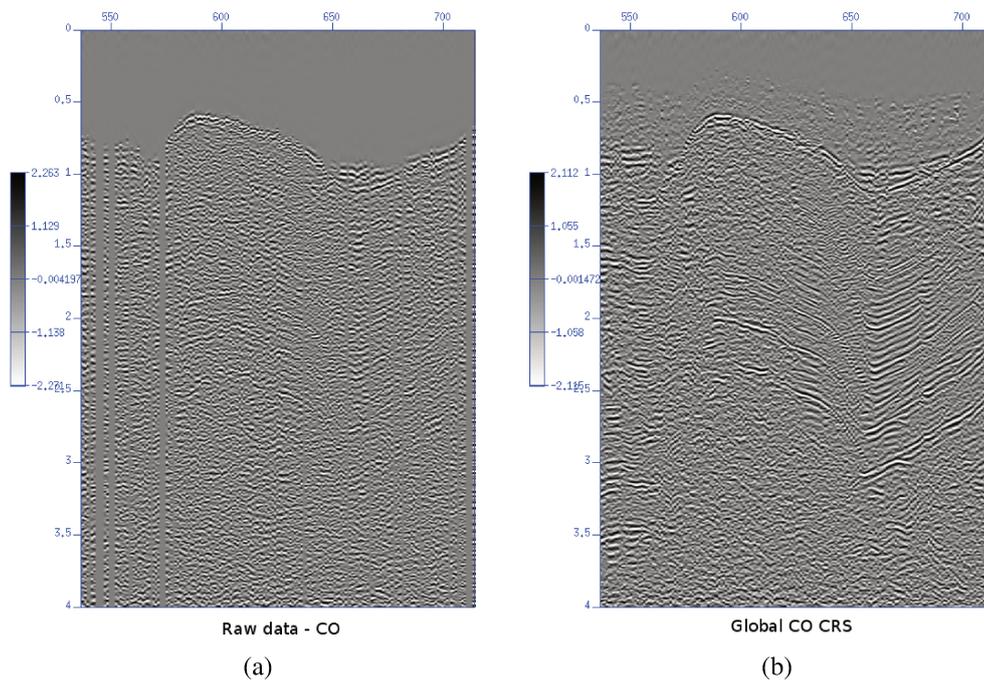


Figure 5.8: Field data: Raw (a) and global enhanced (b) CO gather with  $h = -2000$  m.

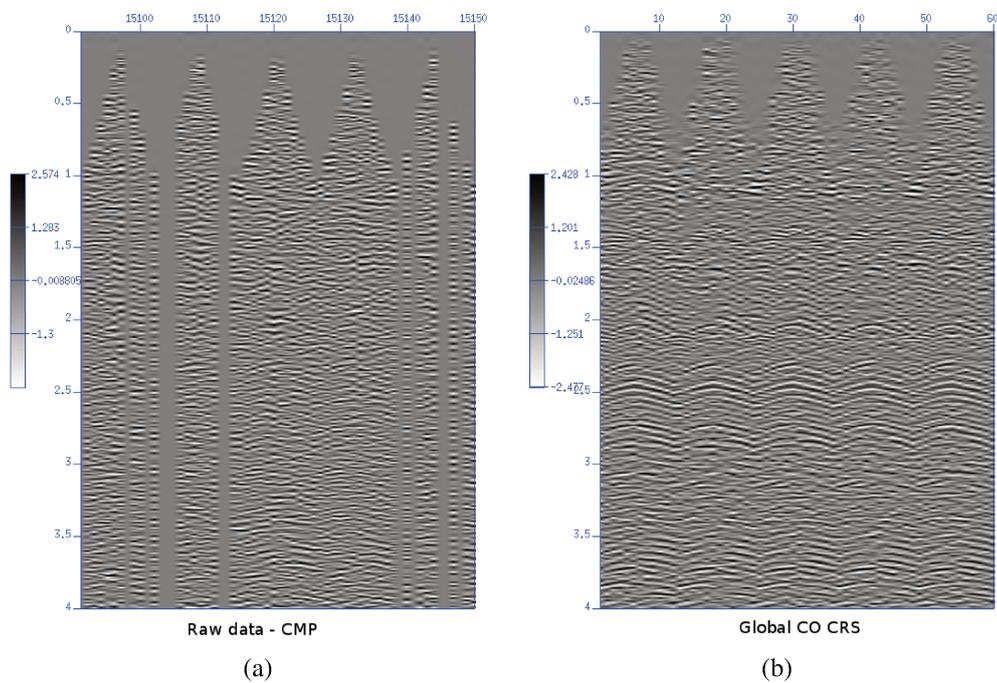


Figure 5.9: Field data: Five raw (a) and global enhanced (b) CMP gathers.

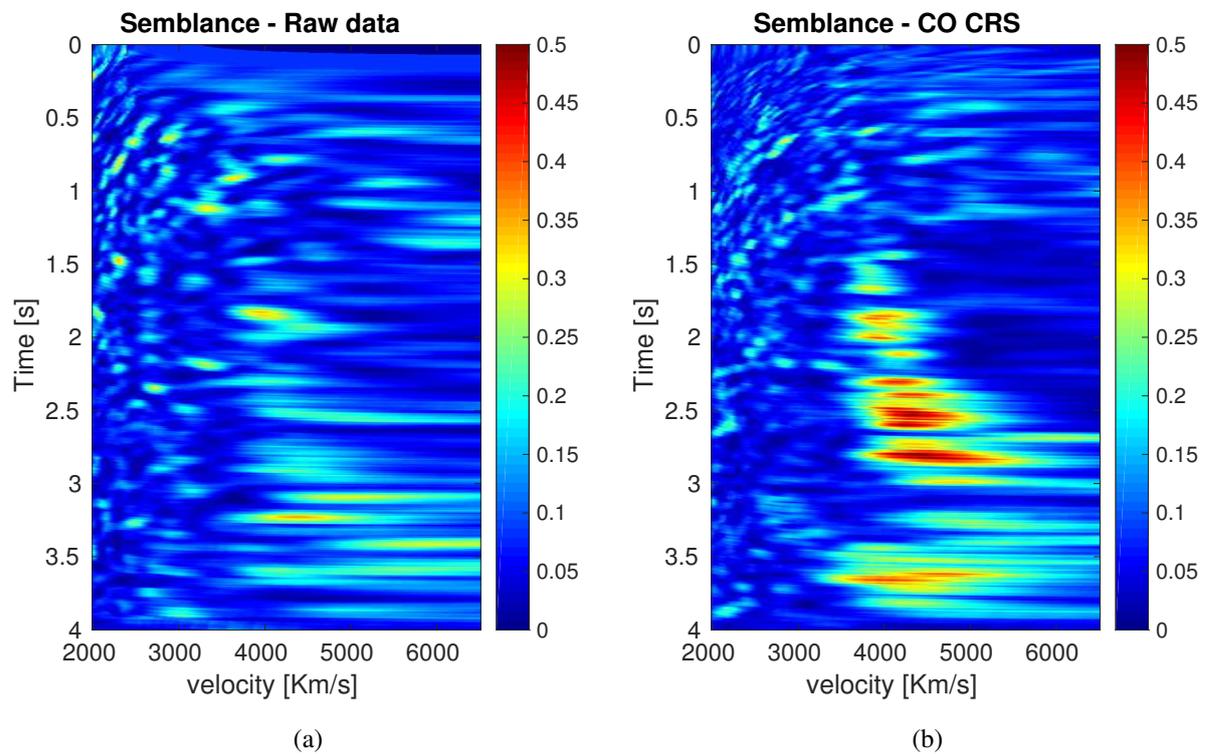


Figure 5.10: Field data: Coherence map computed from the raw data set (a) and from the CO-CRS enhanced version of the same data set (b).

## Chapter 6

# Slopes estimation and picking

Slopes estimation and picking are fundamental in stereotomography, since the input data for the inversion are obtained in these stages. In this chapter we briefly explain the manual and automatic current strategies for slopes estimation and data picking in stereotomography. We then compare the current stereotomography slopes estimation strategy with a new proposal, which employs the slopes estimation in the attributes obtained with the CO-CRS method. This comparison is performed in a simple synthetic seismic data, with a single dipping reflector. We also propose an automatic picking strategy to select the slopes in these CO-CRS attributes gathers, in the presence of several reflections. We finish the chapter with a synthetic seismic data example, where we obtain the macro-model of seismic acoustic velocities with the following work-flow: application of the CO-CRS method in the pre-stack seismic data for the slopes estimation; obtainment of the stereotomography pickings from the CO-CRS attributes using the automatic picking proposal; inversion of these pickings with stereotomography.

### 6.1 Local slant stack

The stereotomography tool used in slopes estimation and picking was proposed in the work of Billette [1998] and is strongly based in the Tau-p transform [Stoffa, 2012], also known as slant stack [Schultz and Claerbout, 1978, Gardner and Lu, 1991].

The slant stack is the Tau-p transform expressed as

$$R(\tau, p) = \int_{-\infty}^{+\infty} dx f(\tau + px, x), \quad (6.1)$$

for the continuous case or

$$R(\tau, p) = \sum_{i=0}^{N_x-1} dx_i f(\tau + px_i, x_i), \quad (6.2)$$

for the discrete case.

The slant stack is, therefore, a summation of the data. This summation maps the straight line in the trace gather  $(t, x)$  into a point in the panel  $(\tau, p)$ . On the other hand, for the stereotomography picking, we use the concept of local summation of traces in the neighbouring of a given central trace, instead of in all the traces. This is because, in stereotomography, we are interested in local values for the slopes. The method employed to do so, proposed in Billette [1998], is to weight the traces by a window function, centered at some reference trace.

As defined in Billette [1998], the local slant stack consists of a transformation from a trace gather (distance, time) to a stacked panel (slope, time), for a reference distance trace. This trace gather might be sorted in common-shot (CS), common-receiver (CR), common-midpoint (CMP) or common-offset (CO) gathers. Finally, the function  $f(\tau + px_i, x_i)$ , used in the stacking Tau-p operation, is the data sampled at the time instant  $\tau + px_i$  and position  $x_i$ , with respect to the central trace.

In the work of Billette [1998], it is also presented the semblance measure, which is a function that also maps the data from a (distance, time) trace gather into a coherence panel, as an alternative to the Tau-p in the stereotomography picking. The semblance coherence panel is computed using data windowed with a given traveltimes [Neidell and Taner, 1971], as opposed to the Tau-p panel, computed in data following a line given by the traveltimes. The semblance measure presents the advantage of a better performance in the presence of random noise, when compared to the Tau-p transform. In the next sections we explain how the picking operation is performed in stereotomography and we provide more information about the semblance.

### 6.1.1 Stereotomography data picking

The current strategy used for slopes estimation and picking, in stereotomography, might be performed in CS, CR, CMP, or CO seismic gathers. As we must determine the slopes at source and receiver positions, we must compute the Tau-p transform to estimate the slopes and pick these slopes in, at least, two different seismic gathers. Usually, the slopes at the source position ( $p_s$ ) are obtained in CR gathers and the slopes at the receiver position ( $p_r$ ) are obtained in CS gathers.

A very interesting approach for the stereotomography slopes estimation and picking, also discussed in Billette [1998], is to use the semblance measure, defined for stacking velocity estimation in Neidell and Taner [1971], instead of the Tau-p transform. The semblance function is a second-order energy measure computed for windows of  $2L + 1$  time samples, taken from traces at  $N_r$  receivers. If we adapt the semblance from stacking velocity analysis to the slopes estimation, each data window

used for semblance evaluation follows the equation defined by the  $(\tau, p)$  parameters and consists of a few samples before and after the window center. These samples might be interpolated if the Tau-p equation does not result in an exact time sample.

If we assume a seismic gather with a total of  $N_t$  time samples and  $N_x$  traces, we may follow a traveltime equation and form data windows from the sampled seismic data. We write this windowed data at the  $i$ -th time sample and  $j$ -th seismic trace as  $d(i, j)$ . For the semblance evaluation, we must design a data window centered in a reference trace, following the Tau-p equation. Finally, we center the window at a given sample  $i_t$ , corresponding to the  $\tau$  position, and write the semblance measure as

$$S_c = \frac{\sum_{i=i_t-L}^{i_t+L} \left| \sum_{j=1}^{N_r} d(i, j) \right|^2}{N_r \sum_{i=i_t-L}^{i_t+L} \sum_{j=1}^{N_r} |d(i, j)|^2}. \quad (6.3)$$

The time sample in the center of the time window gives the  $\tau$  value. This approach consists into testing several slopes values for each  $\tau$ . The semblance values for all the Tau-p possible values form the semblance spectrum.

The main idea behind the semblance function is to measure the coherence between seismic traces in a given neighbouring, by allowing that these traces be aligned in the tested window, if the Tau-p equation exactly fits a reflection.

Therefore, to form the picking set for one scatterer position, composed by the parameter values in equation (2.1), in page 22, we must choose source and receiver reference positions, given by  $s$  and  $r$ , and compute two slopes spectra. One semblance maximum, at the same  $\tau$  value for both spectra, gives the two local slopes. The value of  $t_{sr}$  is given by the  $\tau$  value. This operation must be repeated for all the scatterer positions, in order to generate the picked data. This is a very time consuming operation, which encouraged several attempts to generate an automatic picking tool [Billette, 1998, Billette et al., 2003].

### 6.1.2 Automatic picking

The automatic picking, proposed in Billette [1998], consists into finding, for a given  $\tau$ , slopes that jointly present coherence values within a defined interval, for all the seismic gathers under test. The authors suggest to perform this test in all the CS, CR, CMP and CO gathers. The justification is that a joint probability on the four semblance values in the desired interval increases the chances that the pick is pertinent.

Semblance computation in all the discussed seismic gathers does not highly affect the computa-

tional time, since we only need to compute the semblance measure in all the seismic gathers if we obtain a coherence value in the desired interval for the last seismic gather being tested. The authors also suggest to look at certain sampling of the shots and receivers, used as reference traces. We implemented an automatic picking tool similar to this one, in order to compare with the picked data obtained with the CO-CRS method.

## 6.2 Plane reflector with dip

In this section we analyze the quality of the local slopes estimation and inversion for three different approaches: local slant stack computed with semblance<sup>1</sup>, sequential and global CO-CRS. In order to perform this comparison we generated a very simple synthetic seismic data. We consider a model with a single dipping reflector with an angle of  $10^\circ$ . We further consider a homogeneous overburden with a velocity of 2 Km/s. The synthetic data were generated using 96 channels per shot in a split-spread geometry with offsets 2500 – 150 – 0 – 150 – 2500 m and 50 m between the receivers. The distance between the shots is 200 m, which results in the very low fold of only 12 traces per CMP.

In order to obtain a complete comparison between the slopes estimation methods, we tested this data in three situations, with different signal-to-noise ratios (SNR). We used the open-source software package *Seismic Unix* [Stockwell, 1999] to generate random noise with Gaussian probability distribution<sup>2</sup>, which we added to the synthetic data to obtain seismic data with SNR = 5, 2 and 1. For all the comparisons, we choose the common-offset located at –2500 m to be used as reference.

In Figures 6.1, 6.2, 6.3 and 6.4 we illustrate the semblance coherence spectra for the local slant stack slopes estimation in the four described seismic gathers, computed in the data with SNR = 5. The traces illustrated in red are the reference traces, with source and receiver coordinates resulting in the desired offset. It is possible to observe that even in this case, with the highest tested SNR, the trace gathers and the coherence spectra are highly affected by the noise. In section 6.2.1 we discuss with more details this slope estimation example and in section 6.2.2 we analyze the stereotomography inversion applied to the data picked with different slopes estimation methods.

### 6.2.1 Slopes quality analysis

In this section we compare the slopes obtained with the slant stack method, sequential and global CO-CRS, for three different SNR cases, in the single dipping reflector seismic data. On the other

<sup>1</sup>For simplicity, from this point on, we call this method of local slant stack.

<sup>2</sup>We employed the *suaddnoise* function, which adds either Gaussian or white (uniform) noise to each trace. The output trace is the input trace plus a scaled version of a random noise vector. The scale factor is the reciprocal of the sn (signal-to-noise ratio) parameter times the square root of the ratio of input (signal) power to the noise power.

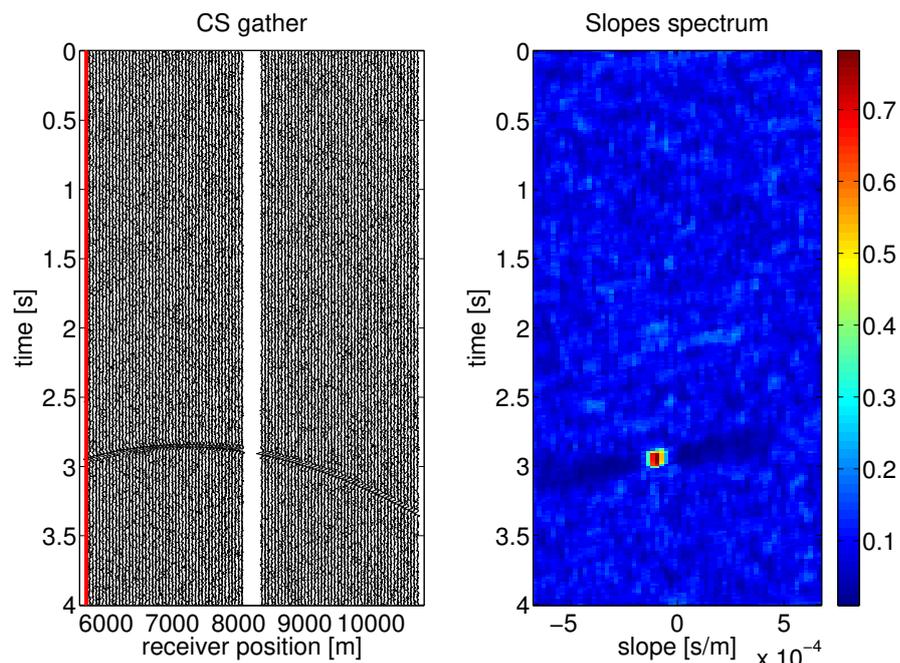


Figure 6.1: CS gather for  $\text{SNR} = 5$  and the respective semblance panel. The trace in red corresponds to the reference trace used for the semblance computation.

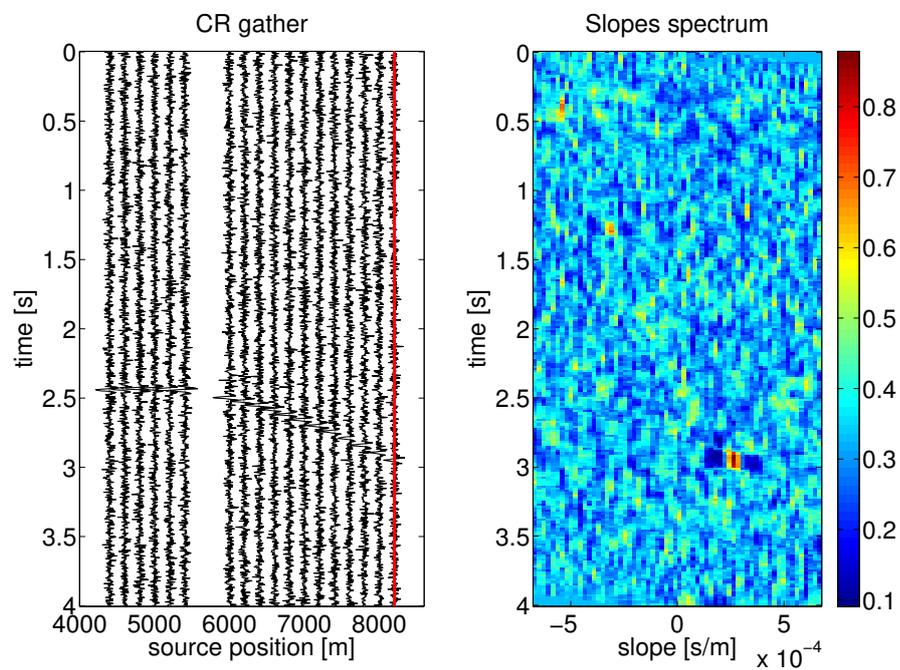


Figure 6.2: CR gather for  $\text{SNR} = 5$  and the respective semblance panel. The trace in red corresponds to the reference trace used for the semblance computation.

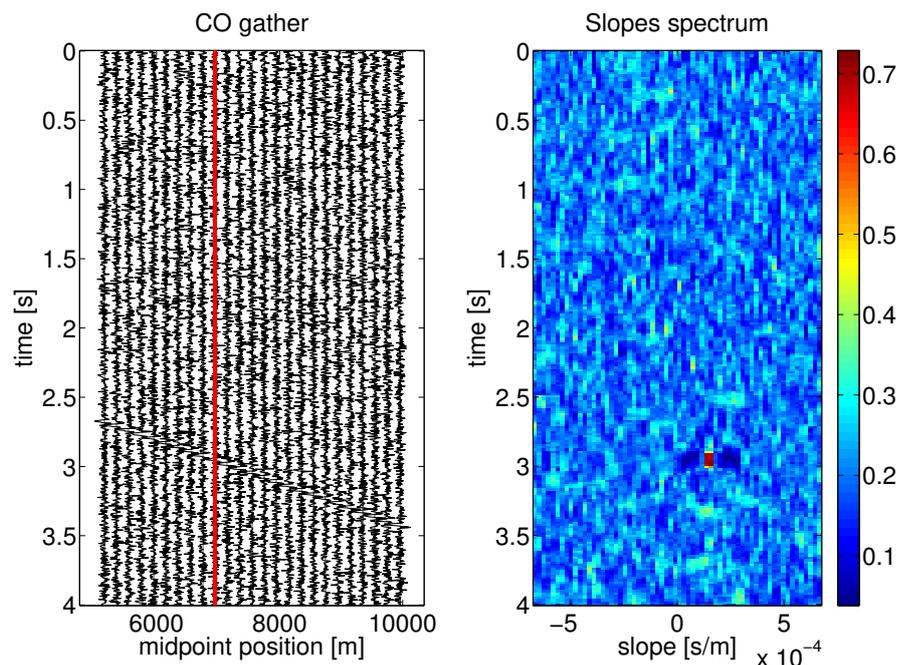


Figure 6.3: CO gather for  $\text{SNR} = 5$  and the respective semblance panel. The trace in red corresponds to the reference trace used for the semblance computation.

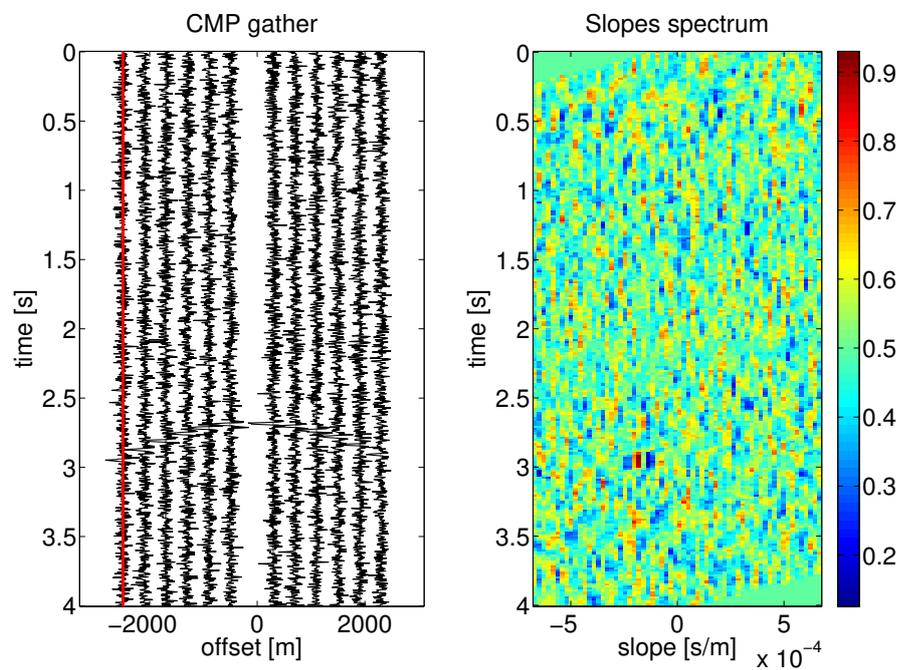


Figure 6.4: CMP gather for  $\text{SNR} = 5$  and the respective semblance panel. The trace in red corresponds to the reference trace used for the semblance computation.

hand, the data with different SNR may allow us to investigate the potential in the slope estimation methods. As we are only trying to analyze the quality in the obtained slopes, we use simpler ways of picking the data.

For the slant stack method, we used the automatic picking for selecting the slopes in all the CS, CR, CMP and CO gathers, with the coherence thresholds carefully adjusted to obtain only one pick at each pair of reference source and receiver traces. Note that this is not a practical strategy, since it is only possible because we know that each trace has only one reflection. For the CO-CRS case, we applied the sequential and global CO-CRS methods in the common-offset of  $-2500$  m. Regarding the picking in CO-CRS examples, we used the resulting coherence gathers as reference: we selected the slopes in the  $\theta_s$  and  $\theta_r$  attributes gathers, at the time samples where the coherence presented the maximum value, for each trace. For the global CO-CRS approach, we used the global coherence, while for the sequential CO-CRS, we used the coherence obtained in the second CO search, which seemed to present the highest SNR.

In Figures 6.5(a), 6.5(b) and 6.5(c) we show the source and receiver angles obtained with the slant stack method and also the absolute value of the specular angle, obtained with equation (4.12), for the three data with different SNR, for all the CMPs indexes, in the CO with offset equal to  $-2500$  m. It is possible to observe that only in the data with  $\text{SNR} = 5$  the angles are correctly estimated. In the data with  $\text{SNR} = 2$  the estimation starts to fail in some of the traces and in the data with  $\text{SNR} = 1$  it is safe to say that the estimation has failed in all the traces.

In order to obtain the stereotomography slopes, with the CO-CRS method, we needed to specify a way of performing the picking operation. As discussed for the NIP tomography performed with the ZO-CRS attributes [Mann and Duvencak, 2004, Klüver and Mann, 2005], the tomography picking can be performed in the CRS attributes with the use of coherence gathers. For the sequential CO-CRS we have coherence gathers in all the search stages and also in the data stacking. In Figures 6.6(a), 6.6(b), 6.6(c), 6.6(d) and 6.6(e) we illustrate these coherence gathers for the data with  $\text{SNR} = 5$ . By analyzing these spectra, and also the spectra obtained in different experiments, we decided to use the coherence gather obtained in the CS search, once that at this search stage it presents a reasonable SNR.

In Figures 6.7(a), 6.7(b) and 6.7(c) we show the coherence gathers obtained in the CS search with the data with different SNR values. The angle gathers where the picking was performed are shown in Figures 6.8(a), 6.8(b), 6.8(c), 6.9(a), 6.9(b) and 6.9(c).

We show the source and receiver angles and also the absolute value of the specular angle, obtained with the sequential CO-CRS method, in Figures 6.10(a), 6.10(b) and 6.10(c). In this test, it is possible to note that almost all the angles obtained from the data with  $\text{SNR} = 2$  are correctly estimated.

The coherence gathers used to obtain the slopes in the tests with the global CO-CRS are shown

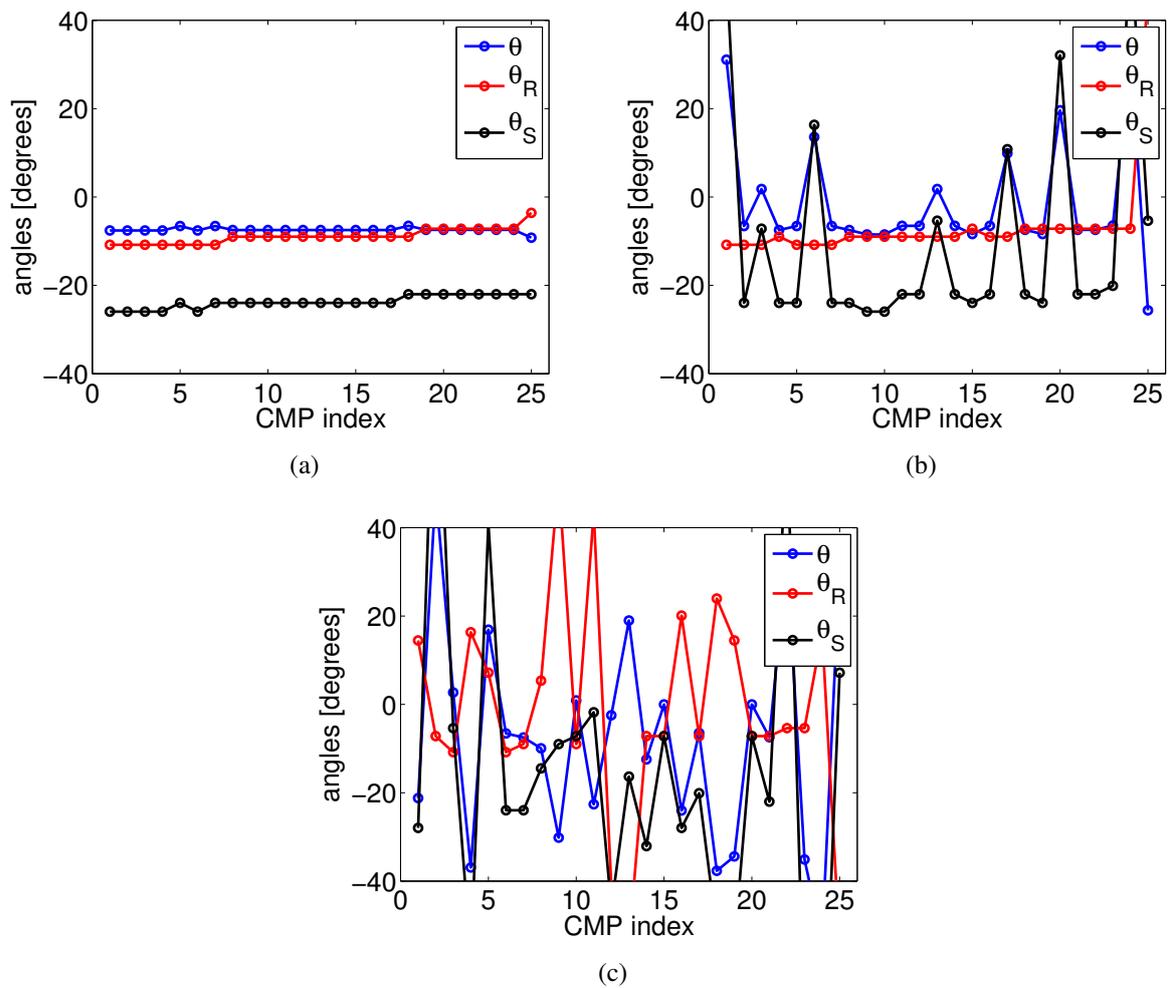


Figure 6.5: Values of  $\theta$ ,  $\theta_s$  and  $\theta_r$  obtained from local slant stack in data with SNR = 5 (a), 2 (b) and 1 (c).  $\theta$  is the specular angle and  $\theta_s$  and  $\theta_r$  are the angles from the scatterer to source and receiver positions, respectively.

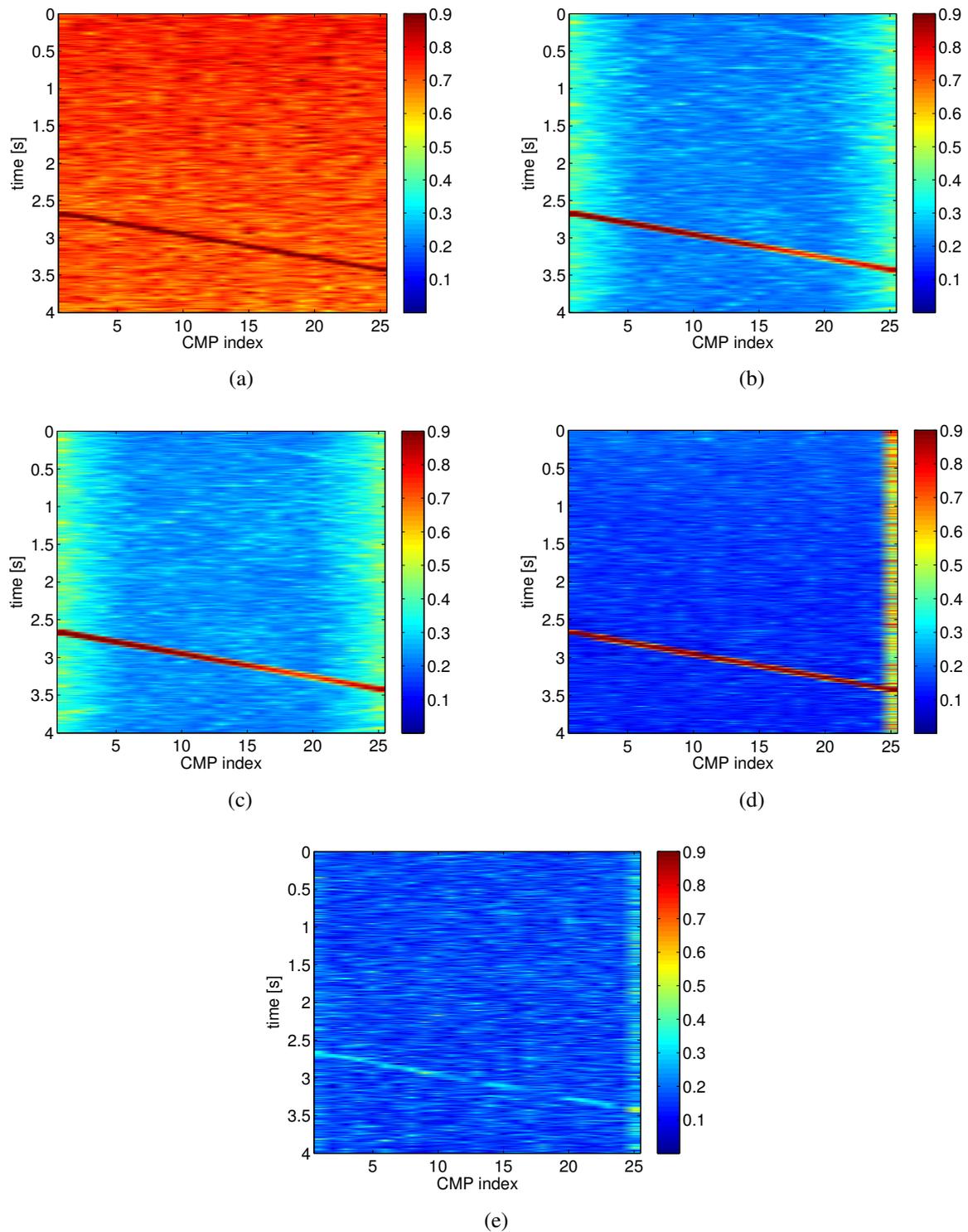


Figure 6.6: Coherence gathers obtained in the sequential CO-CRS search for the data with  $\text{SNR} = 5$ . These gathers come from: CMP search (a); first (b) and second (c) stages of CO search and CS search (d). Final coherence gather obtained with the estimated parameters used in the stacking (e).

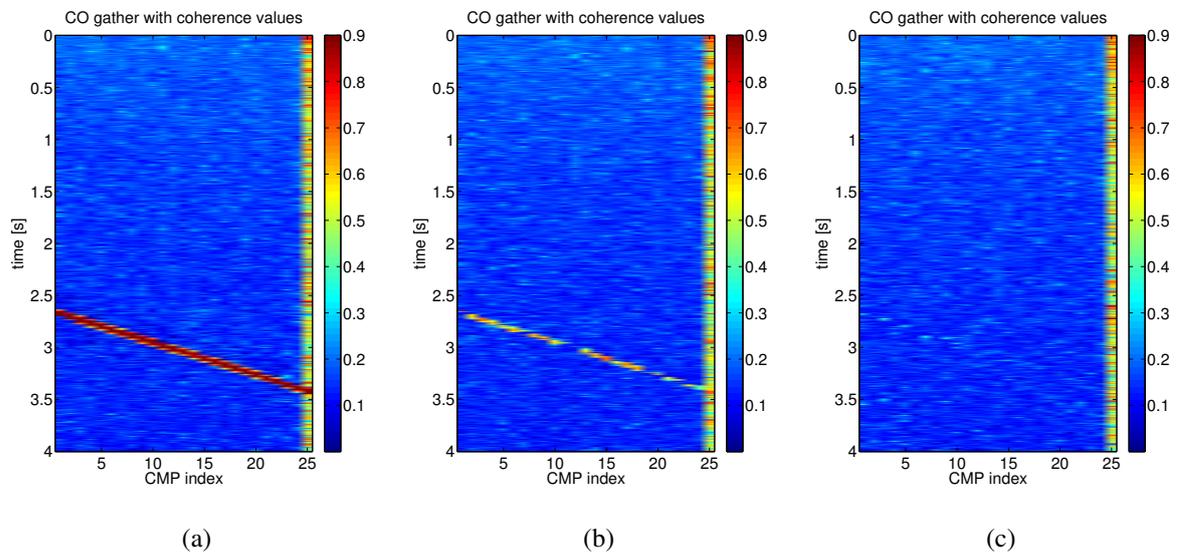


Figure 6.7: Coherence gathers obtained in CS search and used for the angles pick, in the sequential CO-CRS, for the data with SNR = 5 (a), 2 (b) and 1 (c).

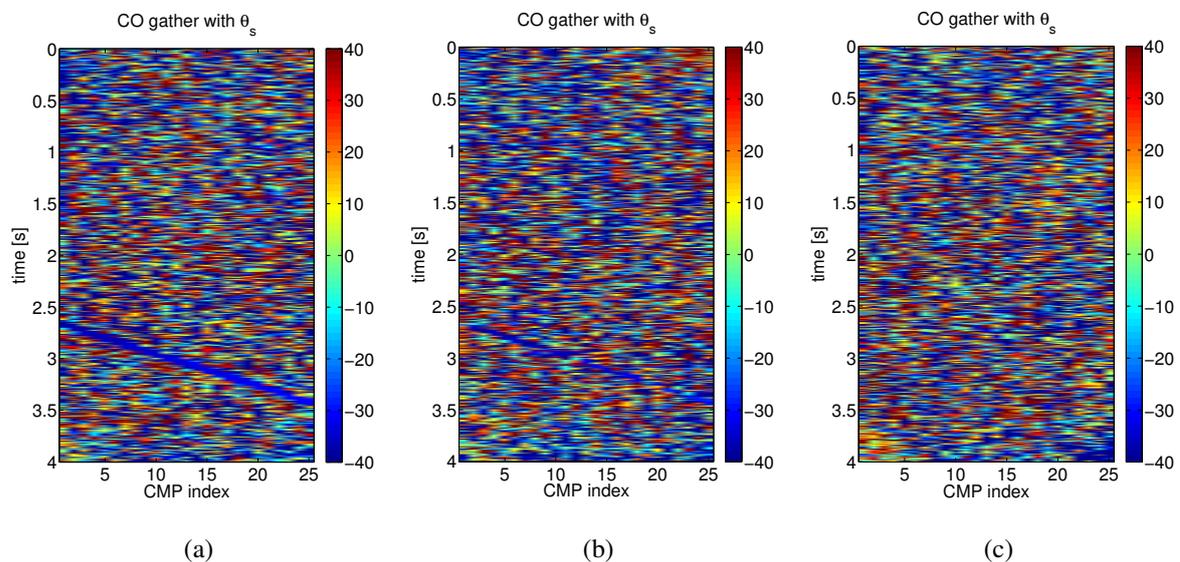


Figure 6.8:  $\theta_s$  obtained with the sequential CO-CRS for the data with SNR = 5 (a), 2 (b) and 1 (c).

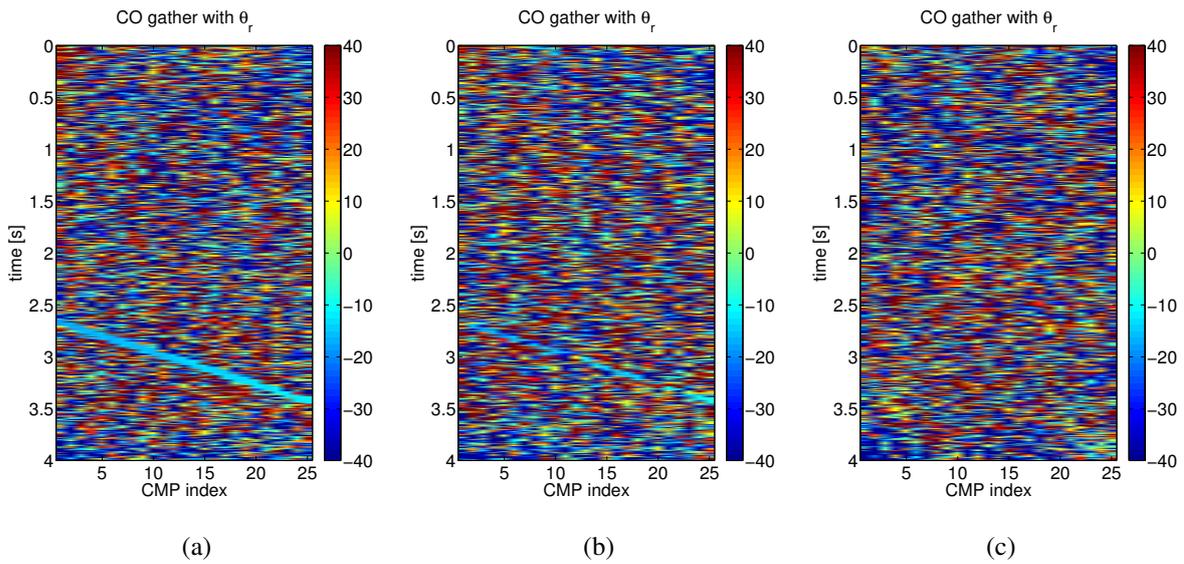


Figure 6.9:  $\theta_r$  obtained with the sequential CO-CRS for the data with SNR = 5 (a), 2 (b) and 1 (c).

in Figures 6.11(a), 6.11(b) and 6.11(c), where it is possible to see that the coherence values near the reflection are much higher and well-defined, in comparison to the ones obtained from the sequential CO-CRS. This is due to the better parameter estimation in the global CO-CRS method. We illustrate the angle gathers where the picking was performed in Figures 6.12(a), 6.12(b), 6.12(c), 6.13(a), 6.13(b) and 6.13(c). It is also possible to see a better definition of the angles, specially in the case with SNR = 1.

Finally, we show the source and receiver angles and also the absolute value of the specular angle, obtained with the global CO-CRS method, in Figures 6.14(a), 6.14(b) and 6.14(c). The parameter estimation in the global CO-CRS gathers presented correct values in all the tested data, being the only method in this example to present this performance in low SNR data.

## 6.2.2 Inversion comparison

We performed the stereotomography inversion in the seismic data from previous section, in order to evaluate the performance of the different tested slopes estimation methods. This inversion was performed aiming at obtaining the model with constant velocity of 2 Km/s. We used the 25 picked data from the common-offset with offset value of  $-2500$  m. The inversion strategy that we used in this example was divided in initialization, localization and single-step joint inversion of the model parameters and we used the Tikhonov regularization. There are several heuristics in the literature to choose a proper value for  $\lambda_r$ . One of the most known is the L-curve method [Hansen, 2005], which attempts to balance the misfit and regularization terms. We did not apply these sort of methods in our

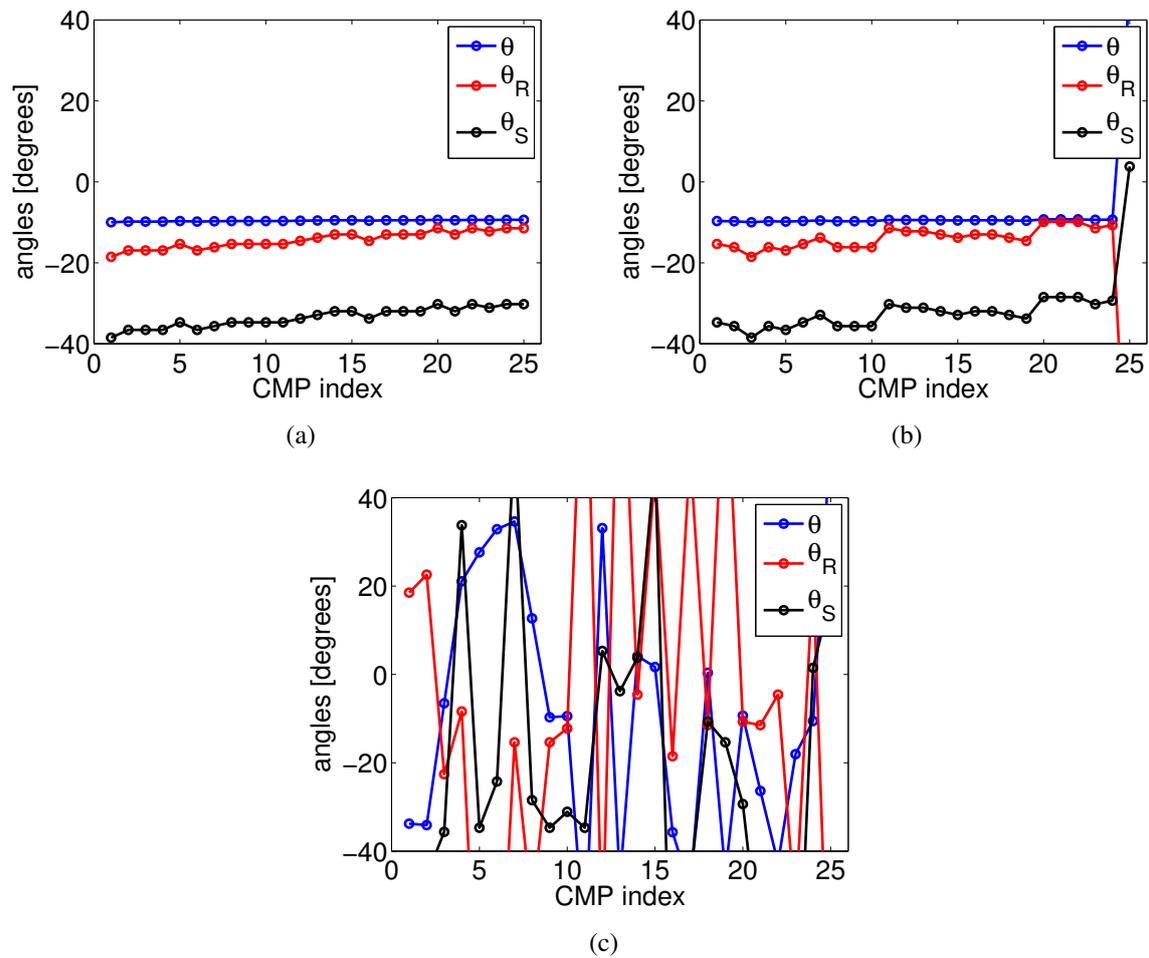


Figure 6.10: Values of  $\theta$ ,  $\theta_s$  and  $\theta_r$  obtained from the sequential CO-CRS in data with SNR = 5 (a), 2 (b) and 1 (c).  $\theta$  is the specular angle and  $\theta_s$  and  $\theta_r$  are the angles from the scatterer to source and receiver positions, respectively.

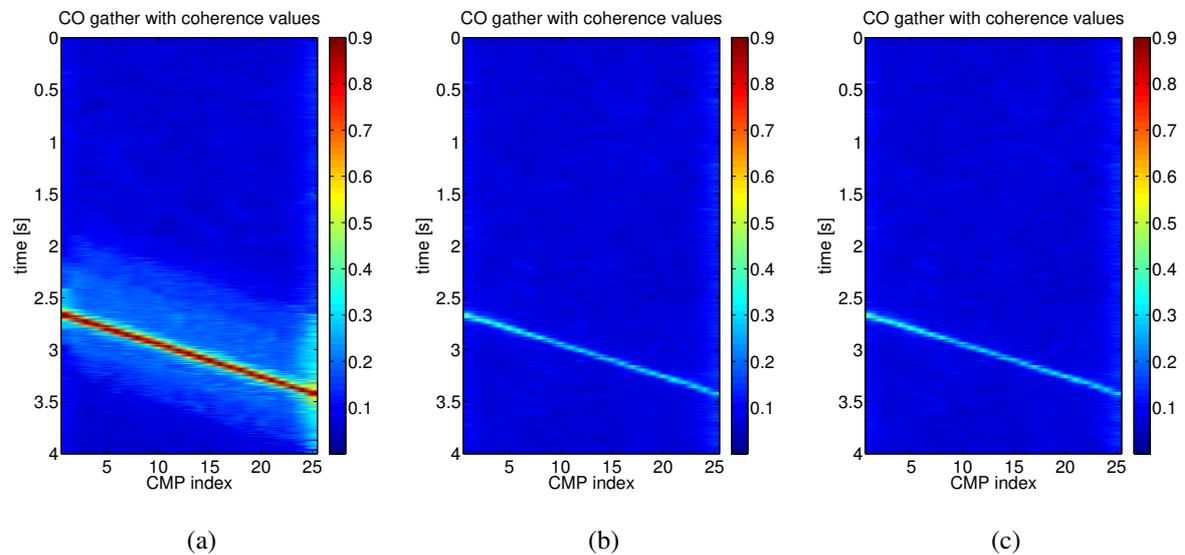


Figure 6.11: Coherence gathers obtained with the global CO-CRS, for the data with SNR = 5 (a), 2 (b) and 1 (c).

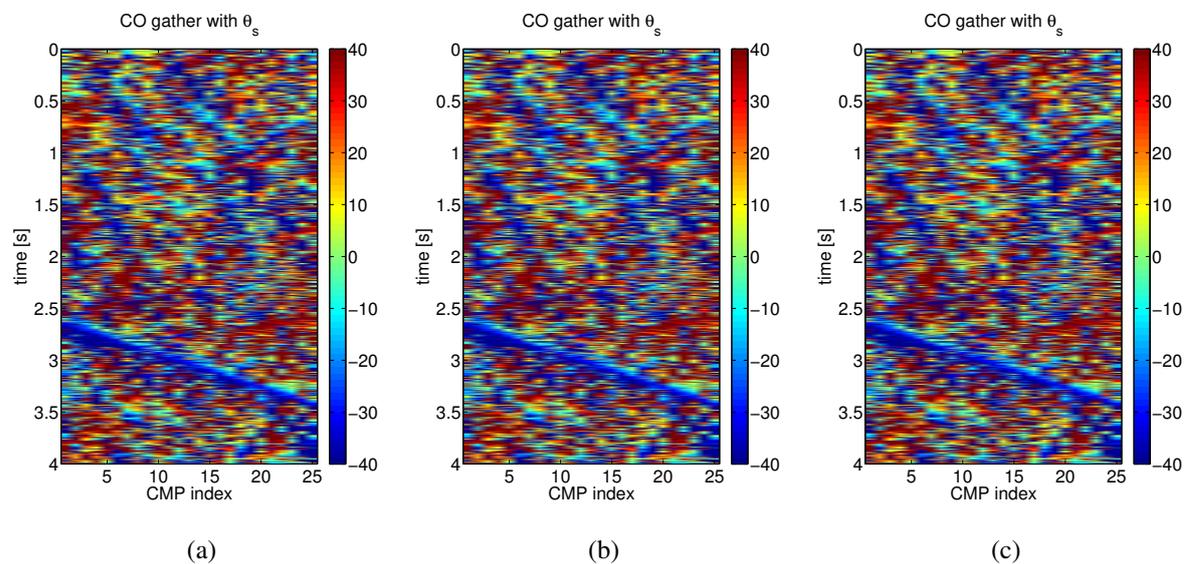


Figure 6.12:  $\theta_s$  for the global CO-CRS with SNR = 5 (a), 2 (b) and 1 (c).

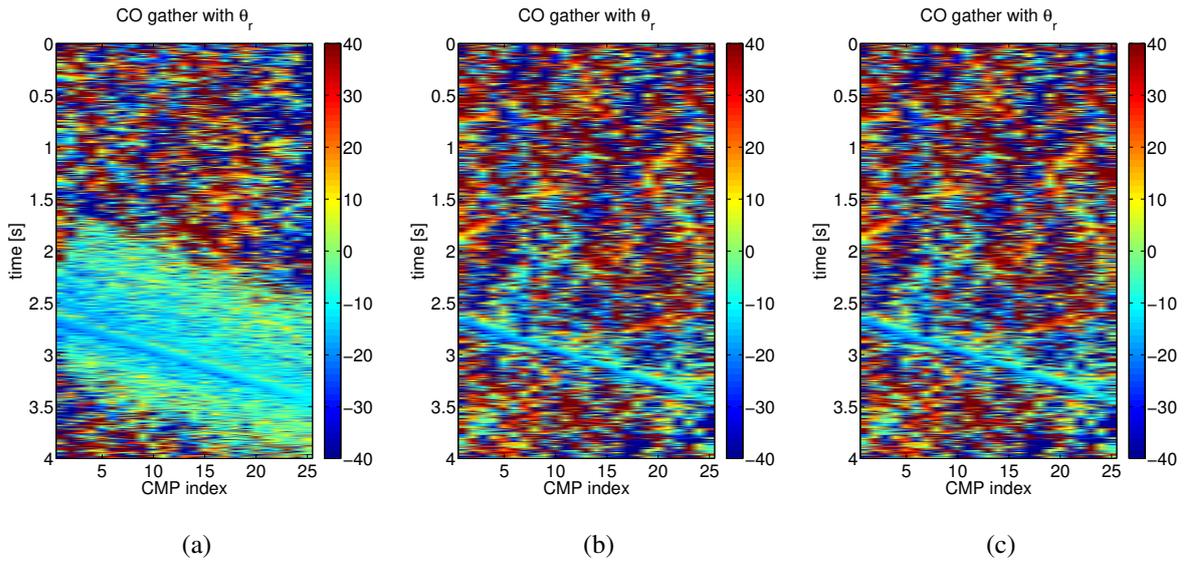


Figure 6.13:  $\theta_r$  for the global CO-CRS with SNR = 5 (a), 2 (b) and 1 (c).

experiments, but this is an interesting perspective for future studies. In our tests, we choose simply to set the parameter  $\lambda_r$  by trial and error, with the most suitable value, according to the test being performed.

The velocity model coordinates vary from 0 to 4 Km in depth and from 0 to 11.2 Km horizontally. The velocity model was sampled with a grid of 100 m, resulting in a velocity model matrix with dimensions  $41 \times 113$ . For generating the B-splines coefficients matrix, we downsampled this matrix by a factor of 5, in both directions, resulting in a B-splines coefficients matrix with dimensions  $9 \times 23$ , in a grid with 500 m of distance between consecutive points. For all the tests we used the constant velocity model with the value of 1.5 Km/s as initialization.

We first show the results for the picked data with the slopes estimated by the slant stack method. For this data, we performed the stereotomography inversion in all the three different SNR data. In Figures 6.15(a), 6.15(b) and 6.15(c) we show how the initialization was performed in these data, with the equivalent model approach. In order to better visualize the data, we performed a zoom in the regions close to the reflection. We can observe that the quality in these picked data implies in poor initialization in the cases of SNR = 2 and SNR = 1. In Figures 6.16(a), 6.16(b) and 6.16(c) we show the localization results and in Figures 6.16(d), 6.16(e) and 6.16(f) we show the results obtained in the joint inversion. We can observe, in these inversion stages, that the equivalent model plot appeared to be correct only in the case with SNR = 5. In order to analyze the estimated velocity, we plot a 2D velocity slice, extracted in the depth of 1.5 Km, in Figures 6.17(a), 6.17(b) and 6.17(c). The velocities are close to the actual value of 2 Km/s in all the distances only in the data with SNR = 5. Although in the other two data sets the velocities are wrong, it is possible to observe that they tend

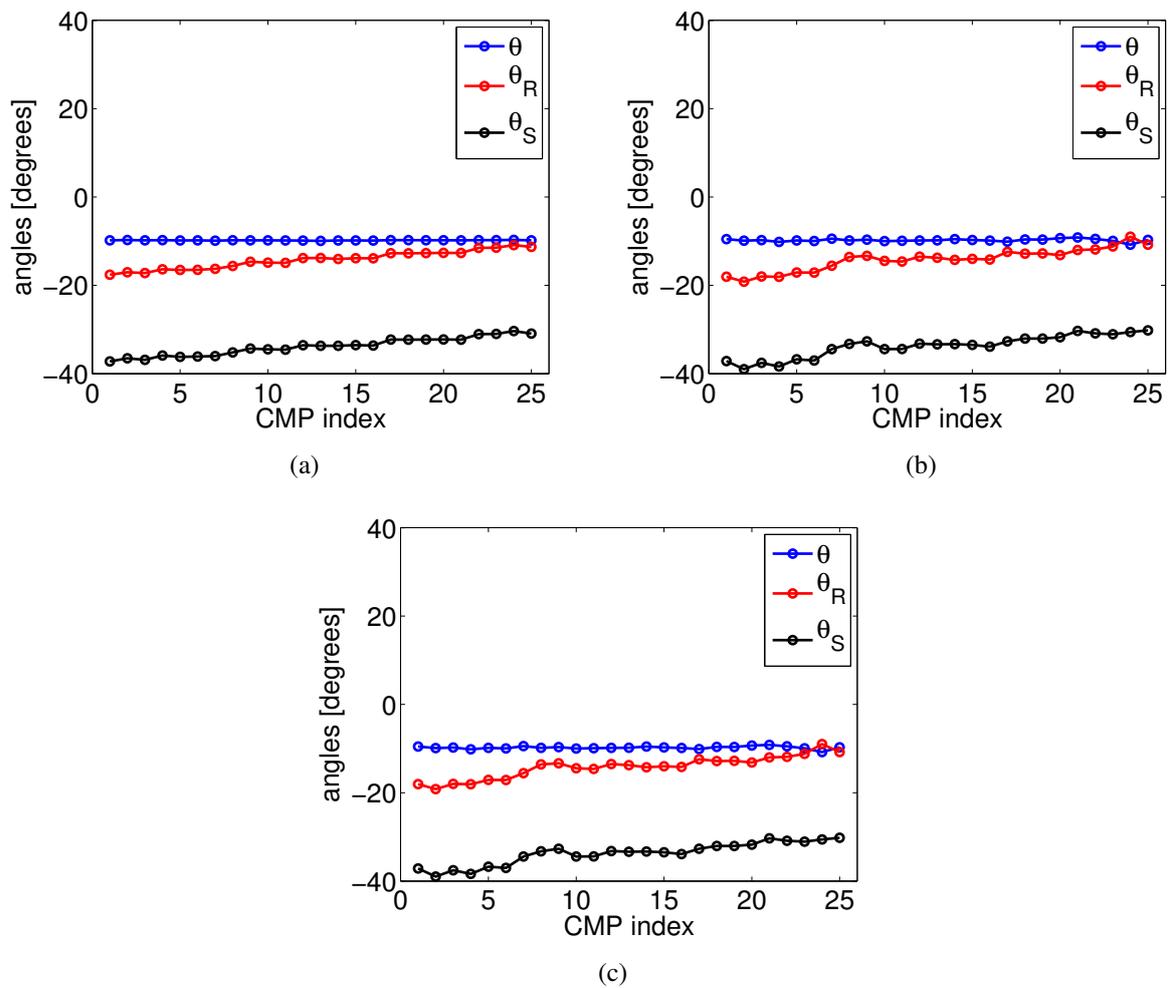


Figure 6.14: Values of  $\theta$ ,  $\theta_s$  and  $\theta_r$  obtained from the global CO-CRS in data with SNR = 5 (a), 2 (b) and 1 (c).  $\theta$  is the specular angle and  $\theta_s$  and  $\theta_r$  are the angles from the scatterer to source and receiver positions, respectively.

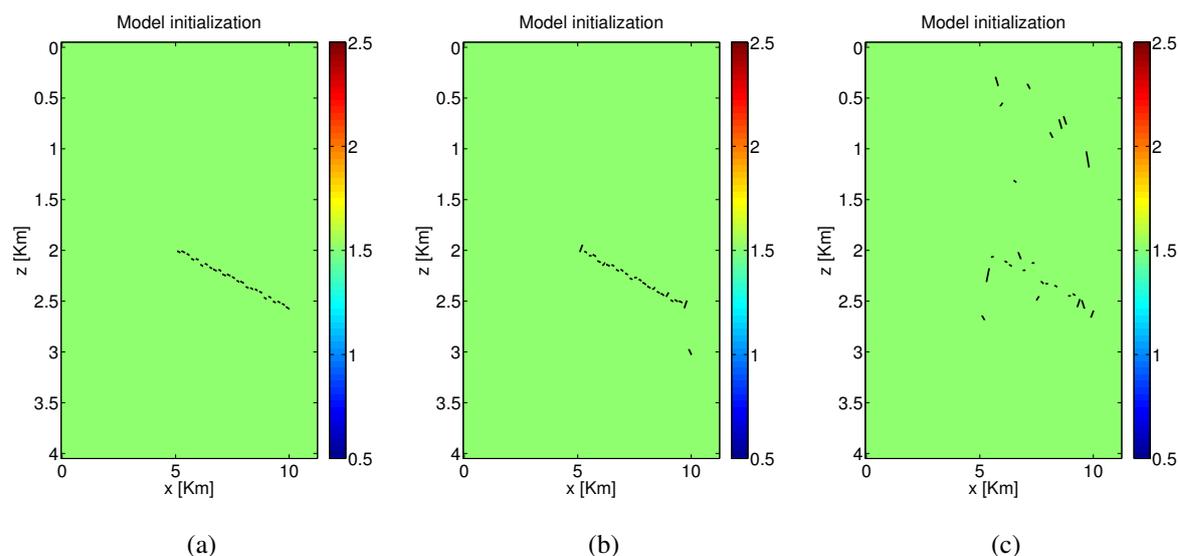


Figure 6.15: Equivalent model initialization obtained with picks from the local slant stack in data with SNR = 5 (a), 2 (b) and 1 (c).

to be close to 2 Km/s. This is mostly due to the strong regularization adopted in these examples. In Figures 6.18(a), 6.18(b) and 6.18(c) we show the cost function for the three different data. The red vertical lines divide the localization and joint inversion stages. It is possible to observe that in the cases of low SNR, stereotomography has converged to high cost function values, failing in the attempt of minimizing the misfit. This is mostly due to the impossibility of obtaining a good initialization in these cases.

For the picked data with the slopes estimated with sequential and global CO-CRS, we decided to analyze the stereotomography inversion only in the cases of SNR = 5 and SNR = 1. First, we see the results obtained for the slopes from the sequential CO-CRS. In Figures 6.19(a) and 6.19(b) we show the initialization, with the equivalent model approach. In these examples, the equivalent model dip bars, for the data with SNR = 1, present strongly incorrect values, which are difficult to analyze because they are out of the zoomed region. This is the reason why we only show, in Figures 6.20(a) and 6.20(b), the dip bars and equivalent model obtained in the localization and joint inversion for the data with SNR = 5. The 2D velocity slices, extracted in the depth of 1.5 Km, are shown in Figures 6.21(a), and 6.21(b). We can see that, in the data with SNR = 1, the velocities are quite far from the correct value of 2 Km/s. In Figures 6.22(a), and 6.22(b) we show the cost functions for the two data sets.

Now we show the results obtained with the slopes from the global CO-CRS method, for the data with SNR = 5 and SNR = 1. In Figures 6.23(a) and 6.23(b) we show the initialization, with the equivalent model approach. In both cases, the dip bars present better orientations, when compared to

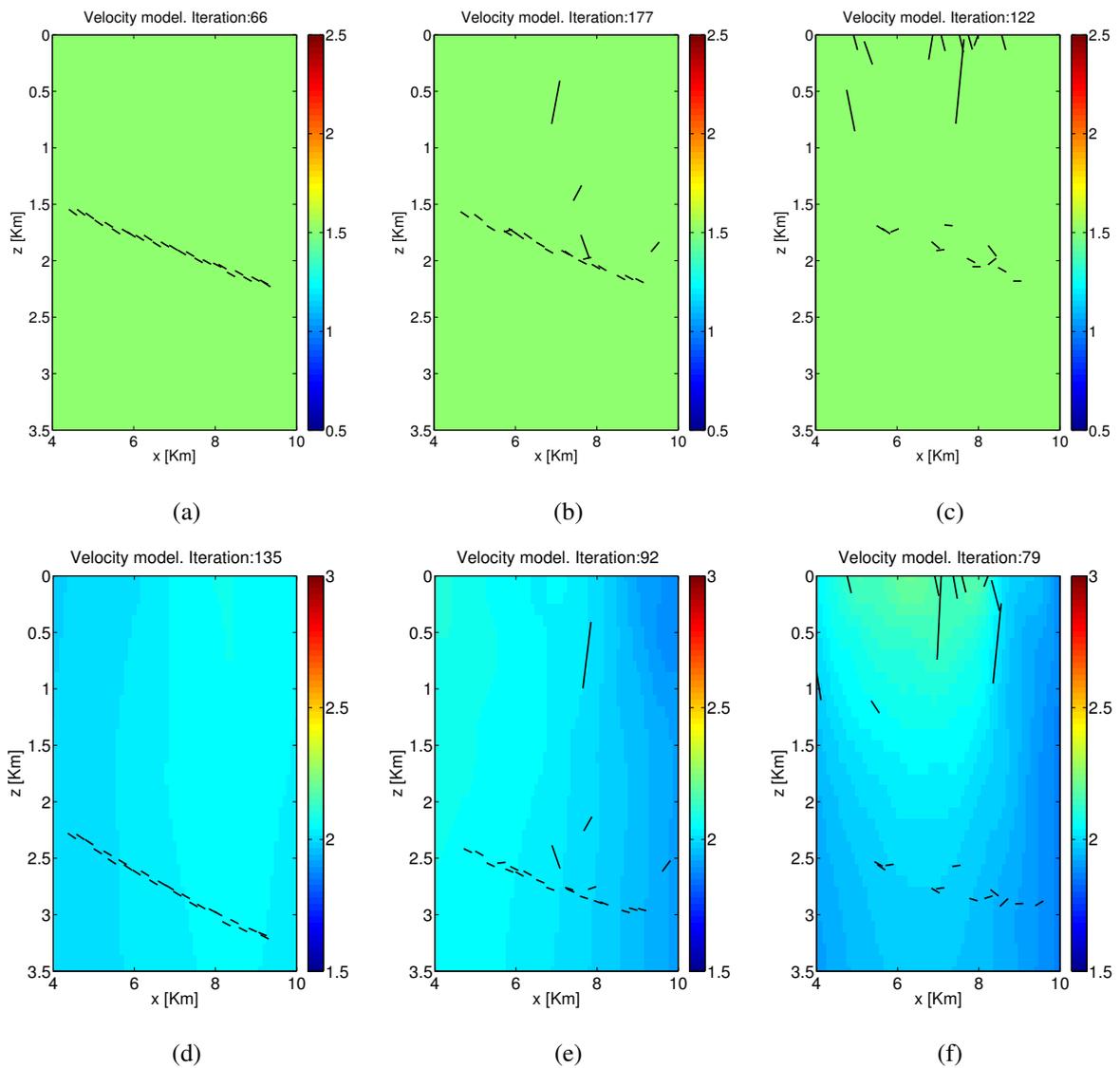


Figure 6.16: Localization and joint inversion results obtained with picks from local slant stack in data with SNR = 5 (a) and (d), 2 (b) and (e) and 1 (c) and (f).

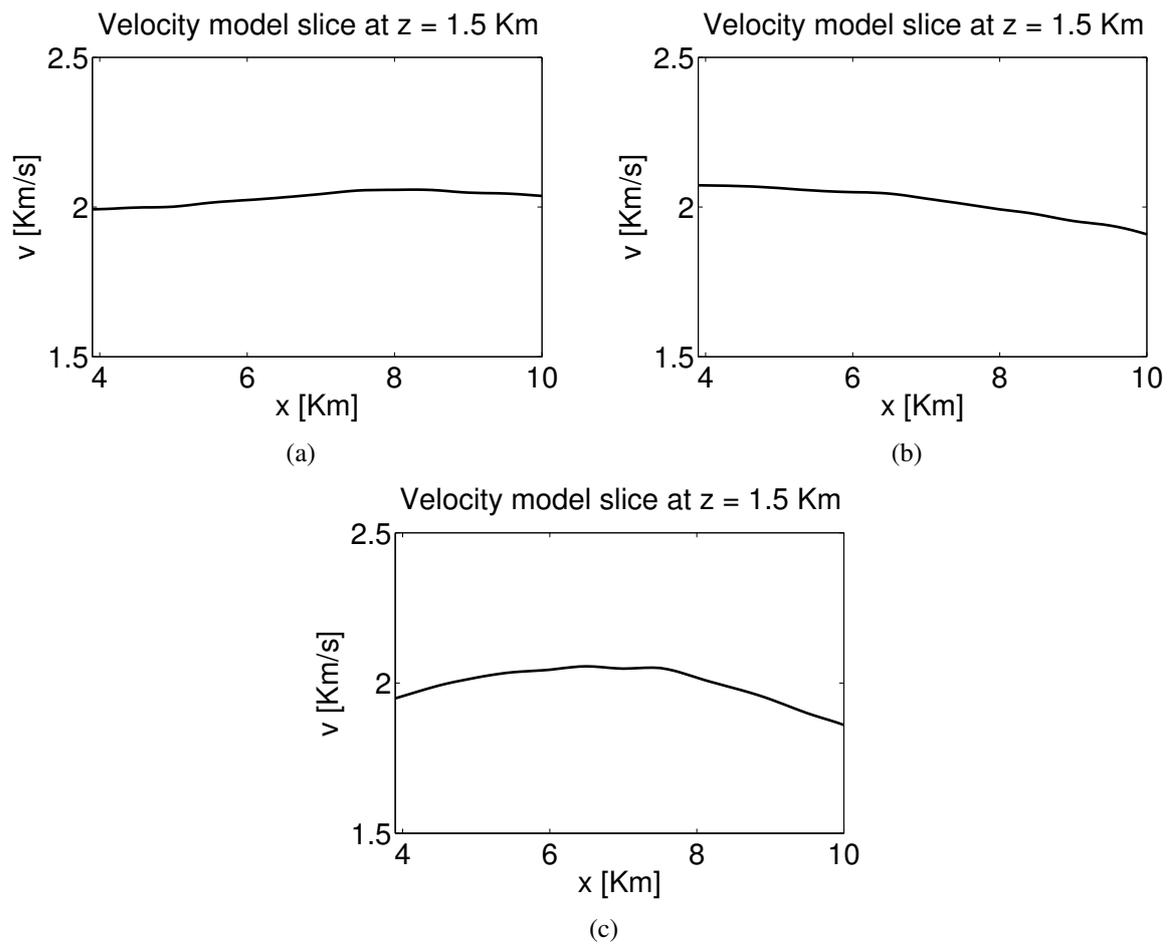


Figure 6.17: Velocity model slice for the picks obtained with local slant stack in data with SNR = 5 (a), 2 (b) and 1 (c).

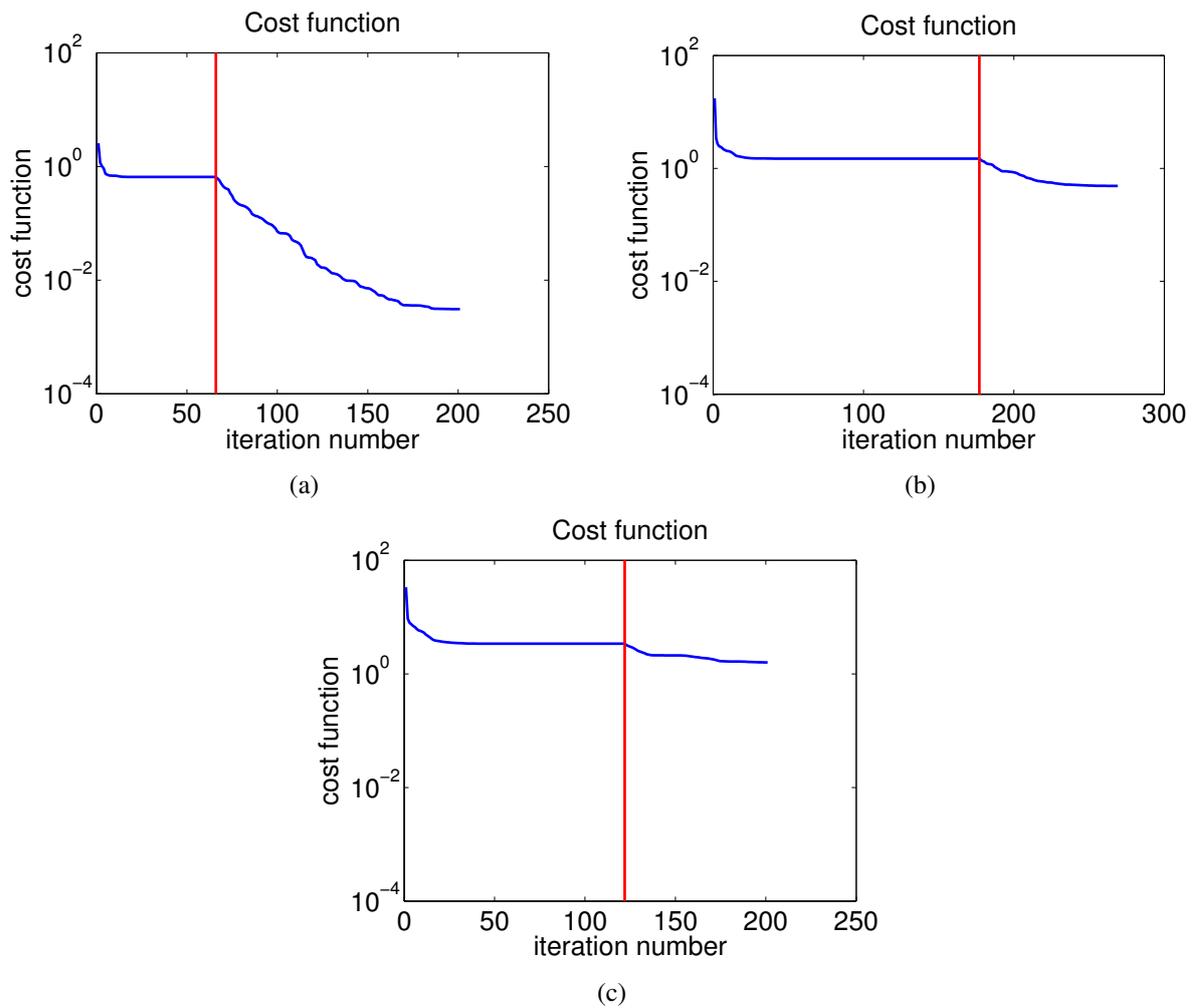


Figure 6.18: Cost functions for the picks obtained with local slant stack in data with SNR = 5 (a), 2 (b) and 1 (c).

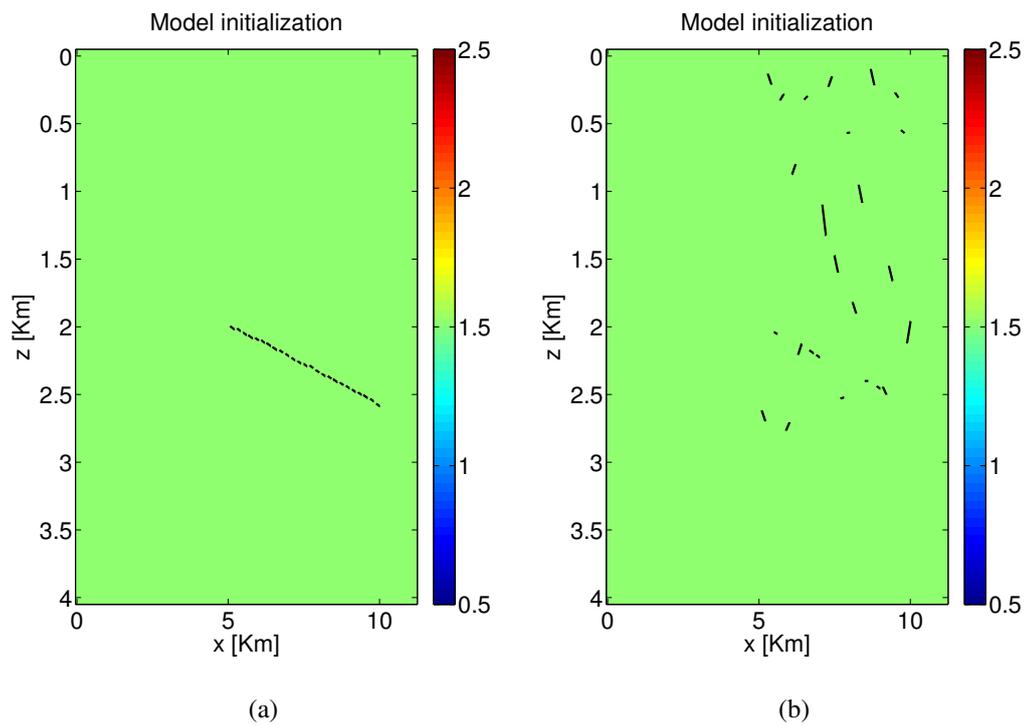


Figure 6.19: Equivalent model for the picks obtained with sequential CO-CRS in data with  $\text{SNR} = 5$  (a) and 1 (b).

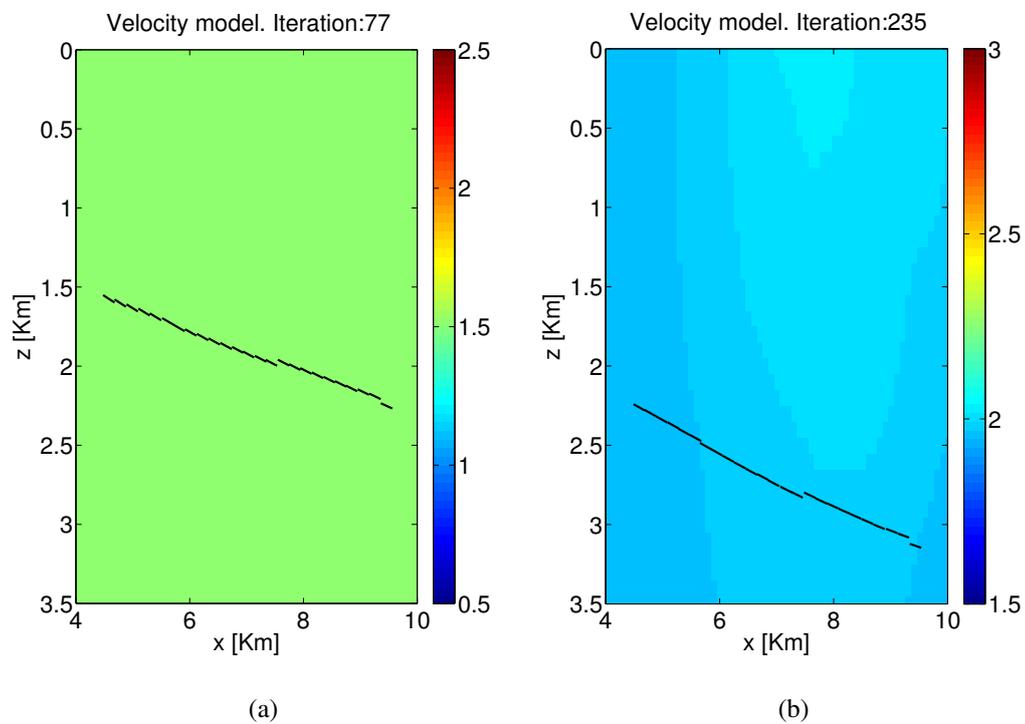


Figure 6.20: Localization (a) and joint inversion (b) results for the picks obtained with sequential CO-CRS in data with  $\text{SNR} = 5$ .

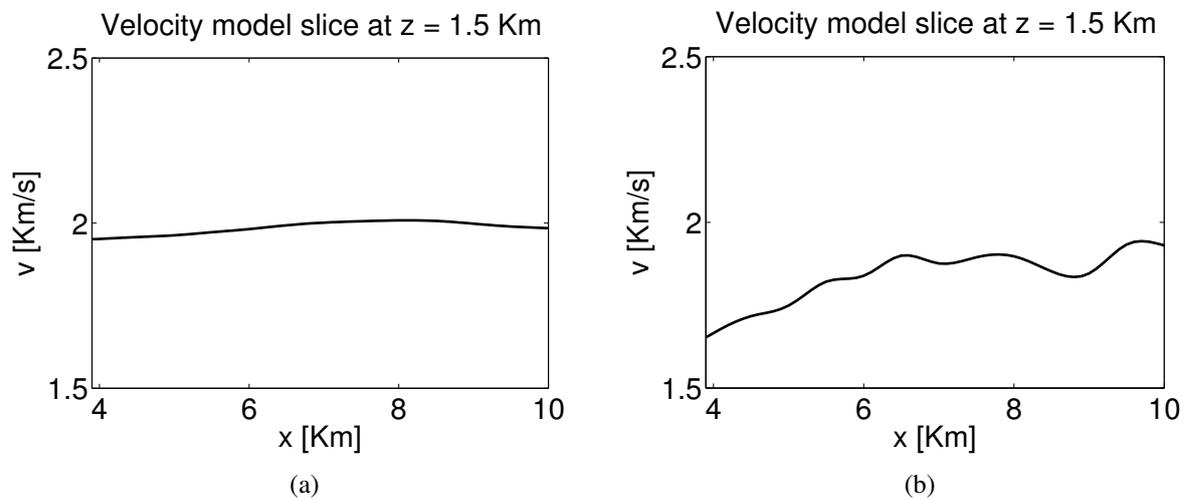


Figure 6.21: Velocity model slice for the picks obtained with sequential CO-CRS in data with SNR = 5 (a) and 1 (b).

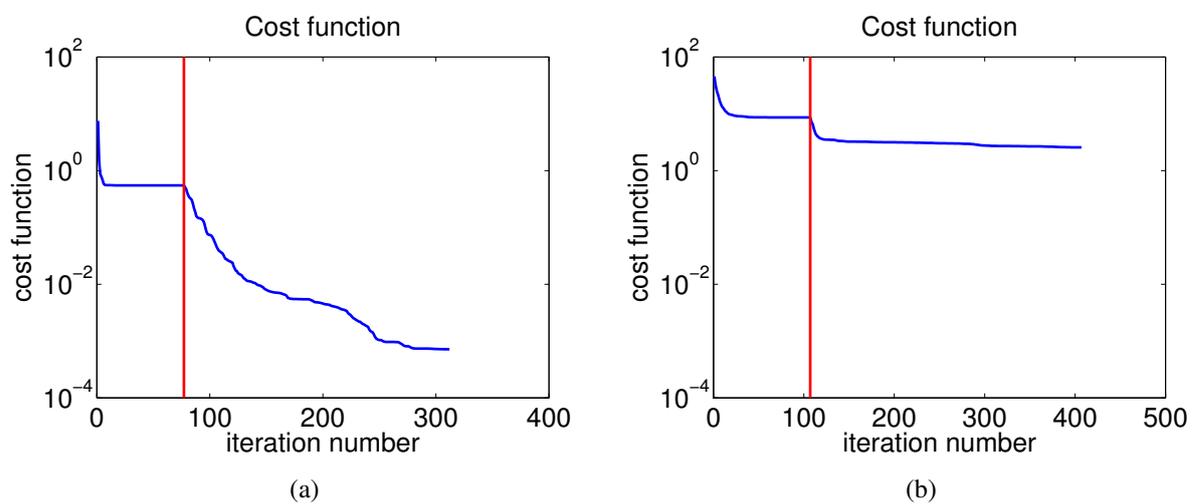


Figure 6.22: Cost functions for the picks obtained with sequential CO-CRS in data with SNR = 5 (a) and 1 (b).

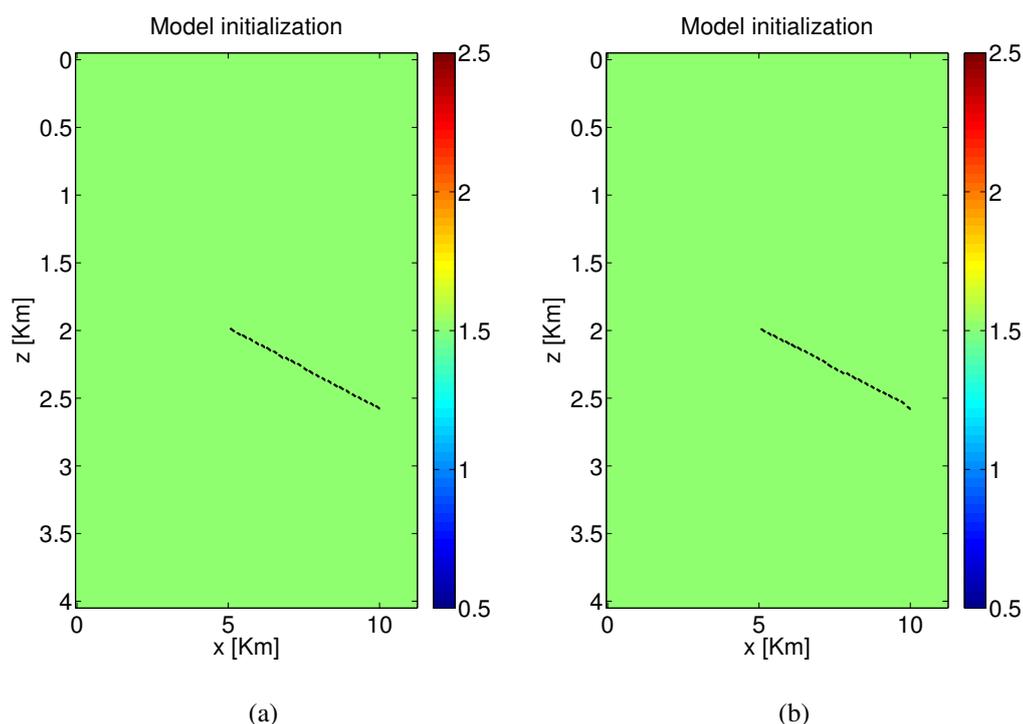


Figure 6.23: Equivalent model for the picks obtained with global CO-CRS in data with SNR = 5 (a) and 1 (b).

the other picking methods. The localization and joint inversion results are shown in Figures 6.24(a), 6.24(b), 6.24(c) and 6.24(d). We can observe that the global CO-CRS is the only slopes estimation method where the dip bars are correctly displaced for the data with SNR = 1. The 2D velocity slices, extracted in the depth of 1.5 Km, are shown in Figures 6.25(a), and 6.25(b). In both cases the velocities are well distributed with the distance, but in the data with SNR = 5 they are closer to the correct value. On the other hand, this is the slopes estimation method where the velocities are closer to the correct value in all the different SNR data sets, including the case of SNR = 1. In Figures 6.26(a), and 6.26(b) we show the cost functions for the two data sets.

### 6.3 Slopes picking with CO-CRS

In this section we discuss how to perform the picking operation, to obtain the stereotomography slopes, in the CO-CRS attributes gather, for more complex seismic data. We begin by proposing a method to perform automatic picking with the use of coherence gathers obtained by CO-CRS. We, then, illustrate this automatic picking strategy and the resulting stereotomography inversion in a synthetic seismic data, that presents a few reflections.

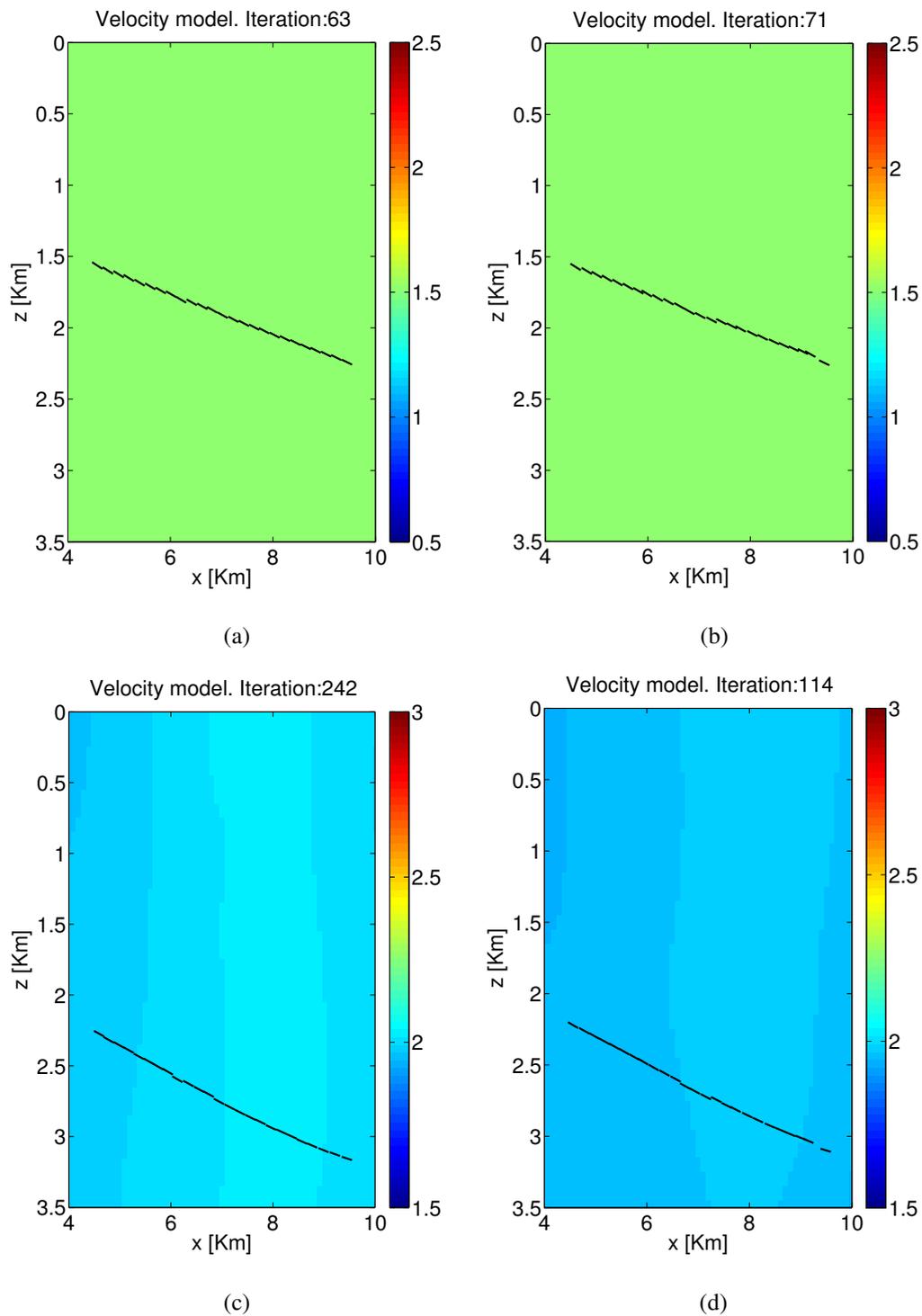


Figure 6.24: Localization and joint inversion results for the picks obtained with global CO-CRS in data with SNR = 5 (a) and 1 (b).

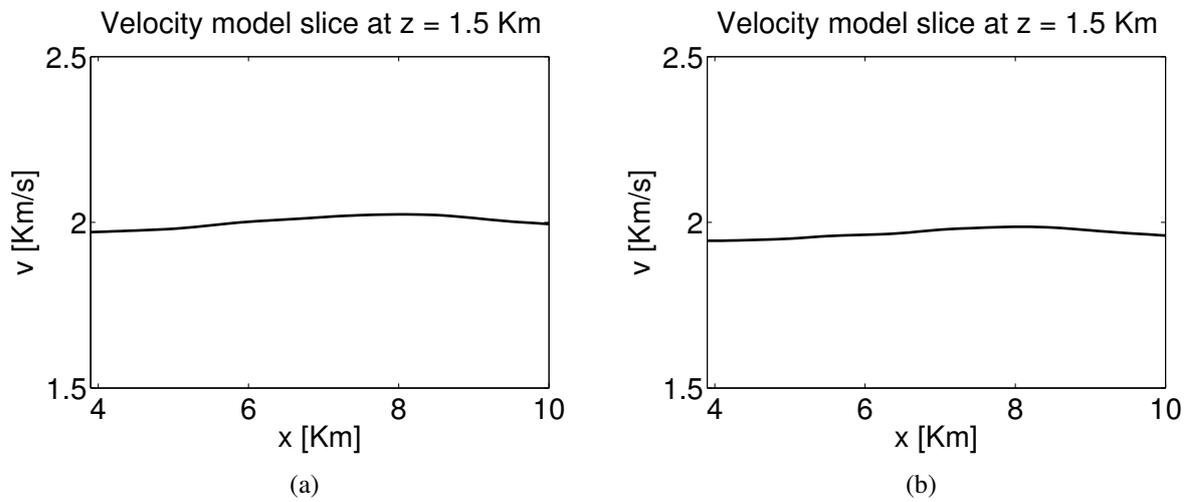


Figure 6.25: Velocity model slice for the picks obtained with global CO-CRS in data with SNR = 5 (a) and 1 (b).

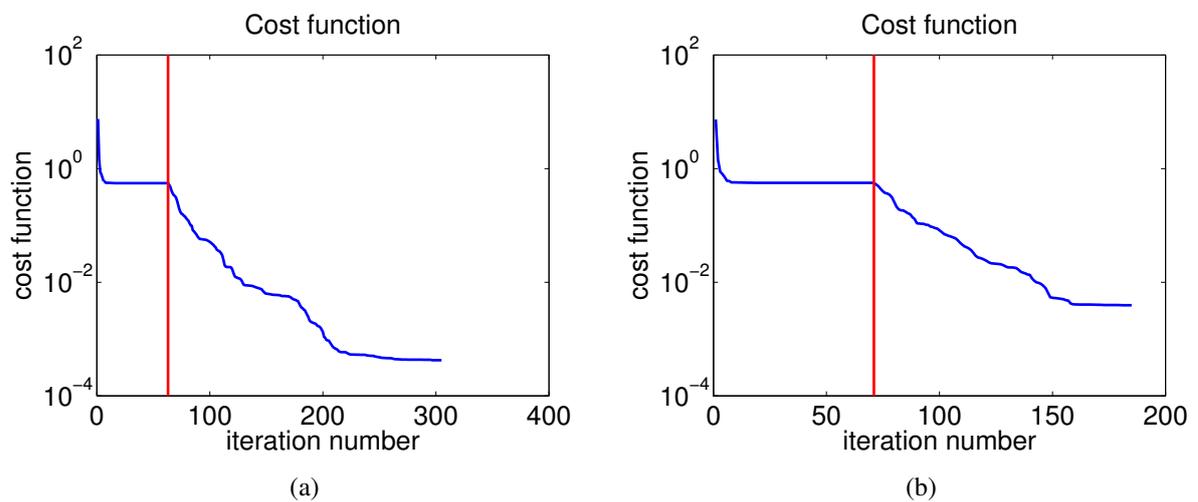


Figure 6.26: Cost functions for the picks obtained with global CO-CRS in data with SNR = 5 (a) and 1 (b).

### 6.3.1 Strategy for automatic picking with CO-CRS

Automatic picking strategies are important tools to extract the angle values from the CO-CRS attributes gathers, since the manual picking can become a very time consuming operation. In the works of Mann and Duvenceck [2004] and Klüver and Mann [2005] an automatic picking strategy is presented to extract the ZO-CRS parameters, which are used as input in the NIP tomography.

We follow this idea and propose, here, an automatic strategy to extract the slopes used in stereotomography, from the CO-CRS attributes gathers. As explained in chapter 5, we obtain, as output from the CO-CRS method, some attributes and coherence gathers. The authors of Mann and Duvenceck [2004] note that the coherence gives a direct measure of the reliability of the kinematic wave-field attributes. Therefore, using the coherence values to extract the information necessary to the picking, from the attributes gathers, is a very interesting way to automate the picking process. However, only using coherence as a reliability criterion for picking can be misleading. This is mostly because, in the presence of noise, some coherence values might present high values, resulting in incorrect picks. Therefore, some additional criteria must be considered.

We propose the following automatic picking algorithm. For each trace:

- Scan each time sample from this trace and form simple windows of size  $2L + 1$  and  $2K + 1$ , with  $L$  time samples and  $K$  traces around the scanned time sample.
- Check if the coherence (semblance) for this reference time sample is higher than a given threshold.
- Check if a user-defined percentage of all samples inside the window:
  - have coherence values higher than a (different) given threshold;
  - have a dip difference below a given threshold with respect to the central sample.
- Continue scanning the time samples on the selected trace until a user-defined maximum number of picks on this trace is reached.

We did not use windows aligned with the CO-CRS parameters nor performed filtering operations in the coherence gathers, as it is suggested for the ZO-CRS case [Mann and Duvenceck, 2004, Klüver and Mann, 2005]. For the numerical experiment shown in section 6.3.2, and also for other performed tests, the proposed algorithm presented a satisfactory performance.

### 6.3.2 Numerical example: synthetic data

In this numerical example, we investigate the performance of the proposed automatic picking strategy, discussed in section 6.3.1, in a synthetic seismic data.

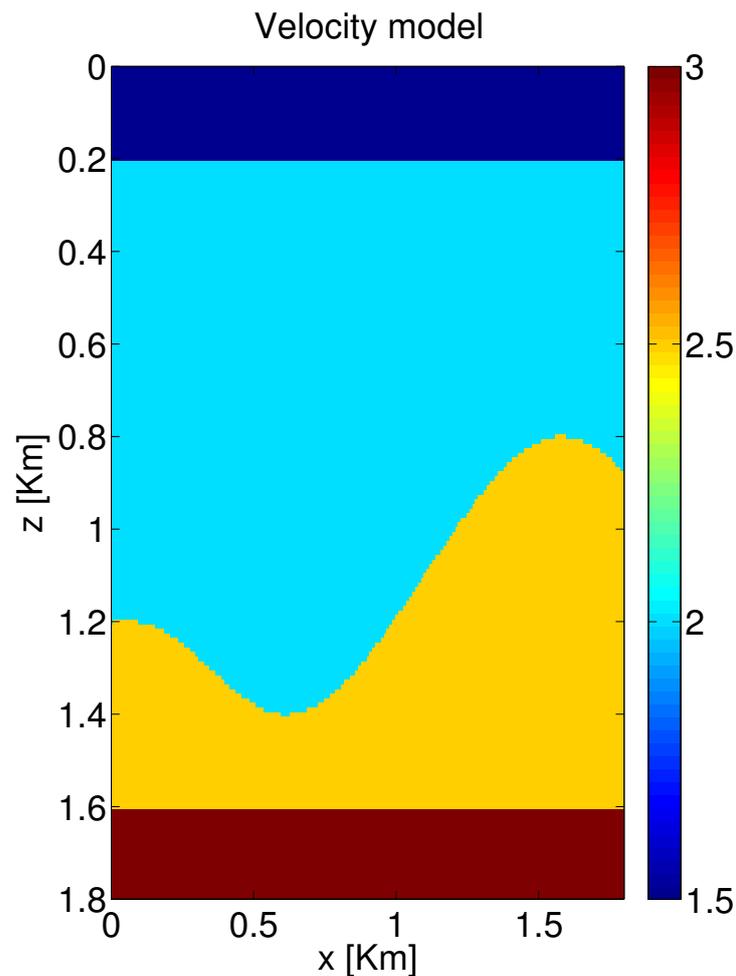


Figure 6.27: Velocity model used to generate the synthetic data.

We generated the data using the open-source ray tracing tool SEIS-88, developed by the Seismic waves in complex 3-D structures (SW3D) consortium and integrated in the open-source seismic processing package GêBR [Biloti, 2011, 2012].

The data contains 179 shots with 180 receivers, displaced at intervals of 10 m. The distance between the source and the first receiver is 10 m. The first shot was performed at the offset of 10 m and the last one at 1790 m. Each shot was increased of 10 m in relation to the last one. The data has a sample period of 2 ms and the total time of recording was 2 s. The velocity model used to generate the data contains four layers, with  $V_p$  velocities of 1.5 Km/s, 2.0 Km/s, 2.5 Km/s and 3.0 Km/s. We illustrate this velocity model in Figure 6.27.

We used Seismic Unix to add random noise to the data, resulting in a data with  $\text{SNR} = 20$ . In order to pick the slopes and generate the stereotomography input data, we applied the global CO-CRS

method in the CO gather with offset coordinate of 70 m. We used the DE algorithm with the crossover and mutation parameters configured with the values of  $CR = 0.7$  and  $F = 0.2$ . The population size was of  $N_P = 60$  individuals and the total number of iterations was  $G = 400$ . We used the following apertures for offset and midpoint:  $\Delta h^{\max} = 100$  m and  $\Delta x_m^{\max} = 100$  m. In Figure 6.28(a) we show the CO gather data with  $SNR = 20$ , used in this example. Figure 6.28(b) shows the same CO gather enhanced by the application of the global CO-CRS method. In Figures 6.28(c) and 6.28(d) we show the angle gathers estimated in this operation.

We performed our proposed automatic picking algorithm, obtaining, in the coherence gather, the picking positions that indicate a reflection. With these positions, we were able to select the stereotomography parameter values in the angle gathers. In Figure 6.29(a) we show the coherence gather superposed by black markers indicating the picked positions. In Figure 6.29(b) we show the constant gradient  $v(z) = 1.5 + z$  Km/s velocity model, used as initialization<sup>3</sup> to the stereotomography, superposed by the dip bars obtained by the equivalent model evaluation in the picked data. We obtained the total of 660 pickings for this offset.

The stereotomography inversion was employed to recover the model varying from 0 to 1.8 Km, both in distance and in depth. The constant gradient velocity model used as initialization was sampled in both coordinates with intervals of 10 m, resulting in a matrix with dimensions  $181 \times 181$ . In the initialization and localization inversion stages, we undersampled this model to 320 m in both dimensions, resulting in a matrix with dimensions  $6 \times 6$ . In the multi-step approach adopted in the joint inversion, we increased this sampling distance by two, alternately for distance and depth. We performed five multi-step inversions, in the sampling grids with depth  $\times$  distance coordinates of  $320 \text{ m} \times 320 \text{ m}$ ,  $320 \text{ m} \times 160 \text{ m}$ ,  $160 \text{ m} \times 160 \text{ m}$ ,  $160 \text{ m} \times 80 \text{ m}$  and  $80 \text{ m} \times 80 \text{ m}$ . In Figures 6.30(a), 6.30(b), 6.30(c), 6.30(d), 6.30(e) and 6.30(f) we illustrate the velocity model obtained in the localization and in each one of the multi-step stages, superposed by the respective equivalent model dip bars. These stages took 55, 43, 462, 19, 4 and 9 iterations, resulting in a total of 592 iterations. We used the Tikhonov regularization, with the value of  $\lambda_r = 0.1$ , obtained by trial and error. We used  $\sigma_x = \sigma_z = 0.01$  Km,  $\sigma_{px} = \sigma_{pz} = 0.01$  s/Km and  $\sigma_t = 0.001$  s, as suggested in Billette et al. [2003].

It is possible to observe that the dip bars model parameters are strongly updated in the localization and in the two first multi-step inversions, being slightly changed in the last three multi-step stages. On the other hand, the velocity model is strongly updated in the three first multi-step inversions. In Figures 6.31(a), 6.31(b), 6.31(c), 6.31(d) and 6.31(e) we show the true velocity model reconstructed with B-splines coefficients and using the same sampling as the one used in the five multi-step inversion stages. This is an interesting plot, since it might be used as the best possible velocity model

<sup>3</sup>We have tried to use several velocity models in the stereotomography initialization, such as homogeneous velocity models or even different constant gradient velocity models. The constant gradient velocity model chosen is the one which provided the best results for the velocities update in all the depths.

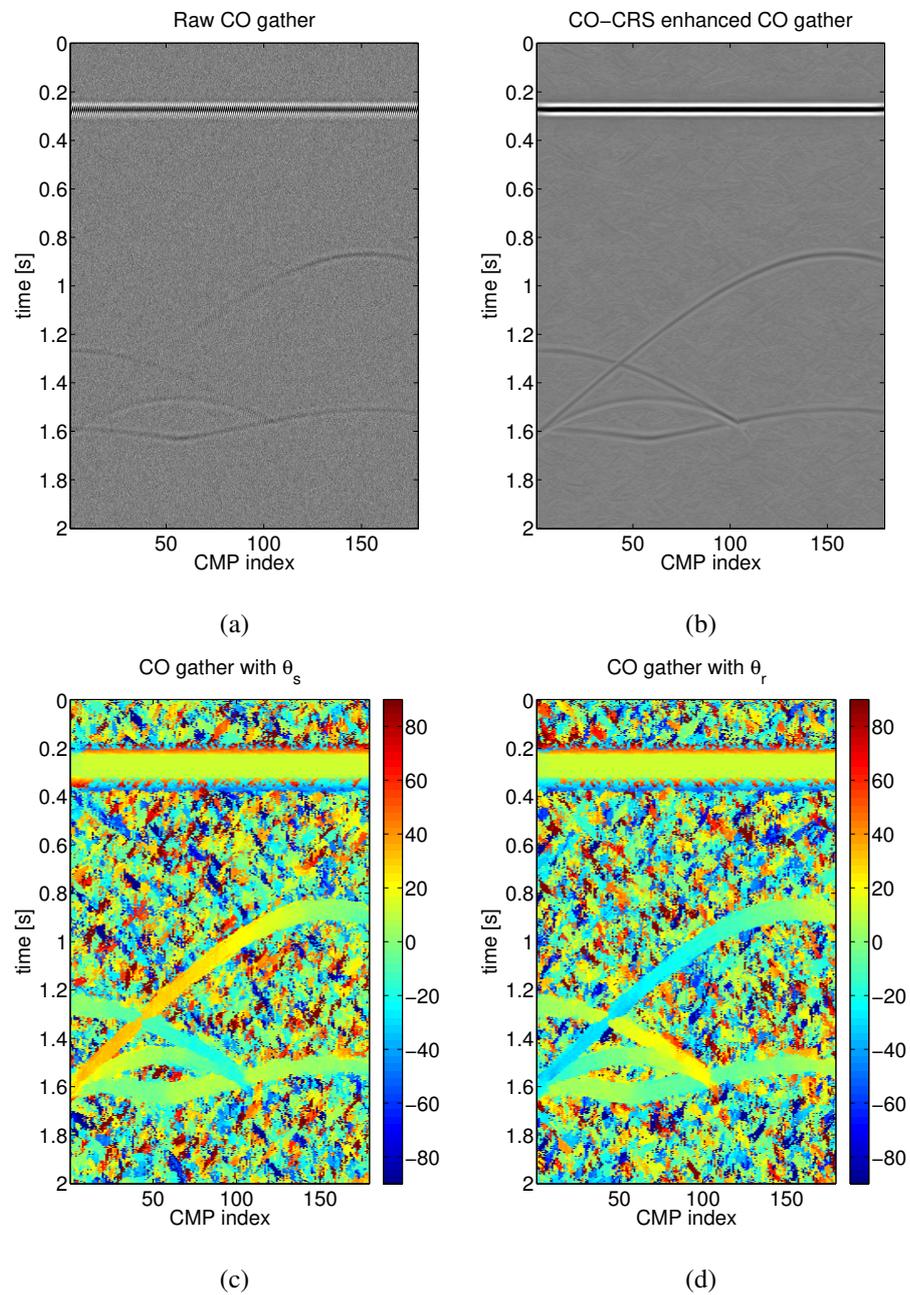


Figure 6.28: Raw CO gather data with  $\text{SNR} = 20$  (a) and the same data enhanced with CO-CRS (b).  $\theta_s$  (c) and  $\theta_r$  (d) estimated attributes.

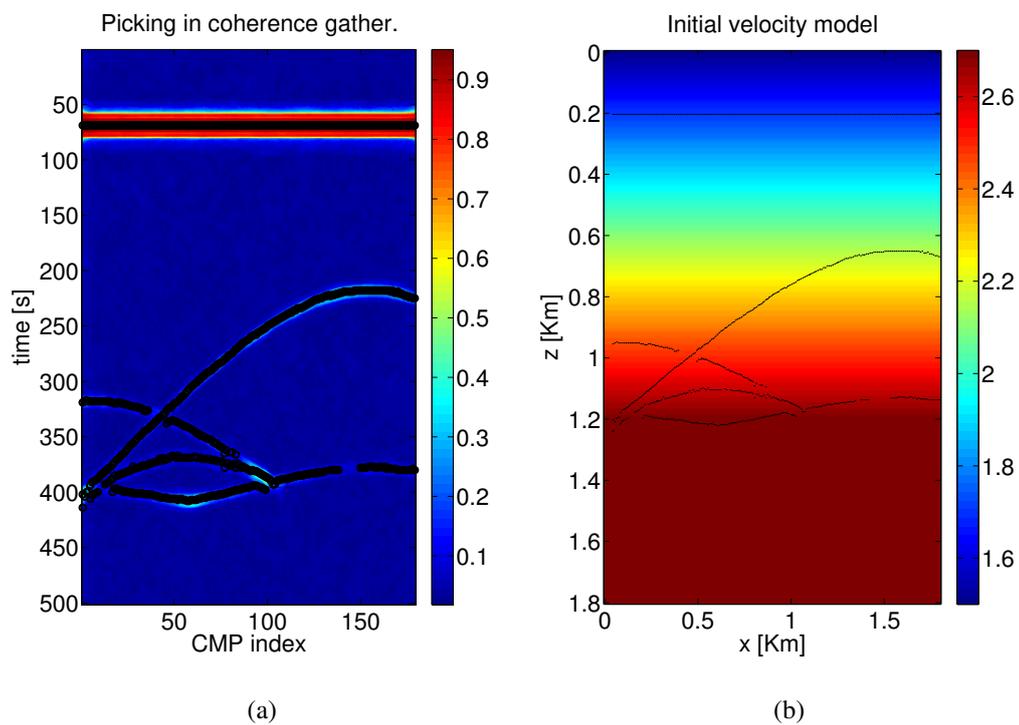


Figure 6.29: Coherence gather superposed by black markers indicating the picking positions (a). Constant gradient  $v(z) = 1.5 + z$  Km/s velocity model, used as initialization to the stereotomography, superposed by the dip bars obtained by the equivalent model evaluation in the picked data (b).

B-splines representation, with the employed grid. If we use this for comparison purposes, we can see that stereotomography performed quite well in the first two multi-step inversion stages. In Figure 6.32 we show the cost function for this example. The vertical lines correspond to the divisions in the inversion stages.

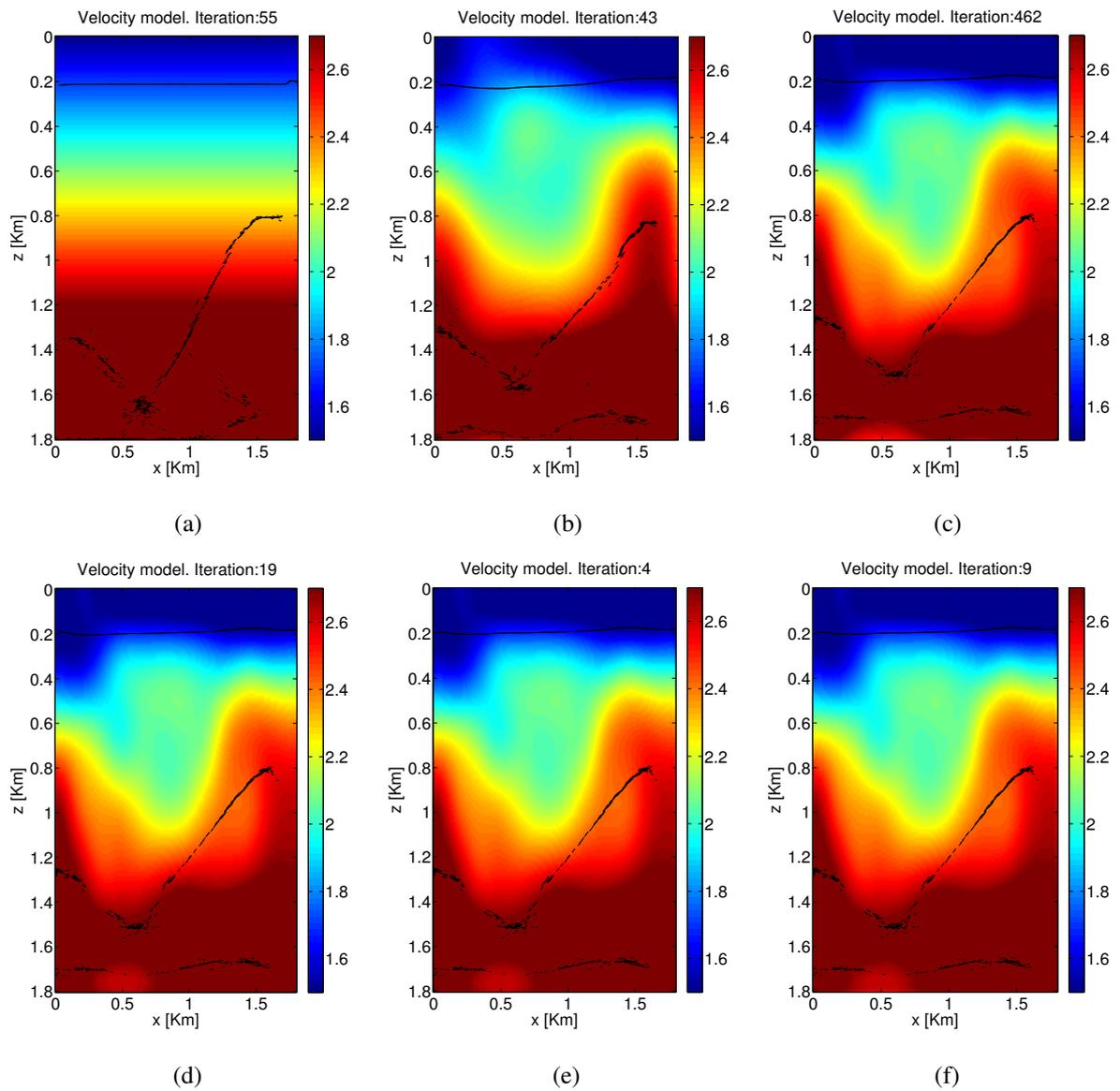


Figure 6.30: Velocity model obtained in the localization (a) and in each one of the multi-step stages (b)-(f), superposed by the respective equivalent model dip bars.

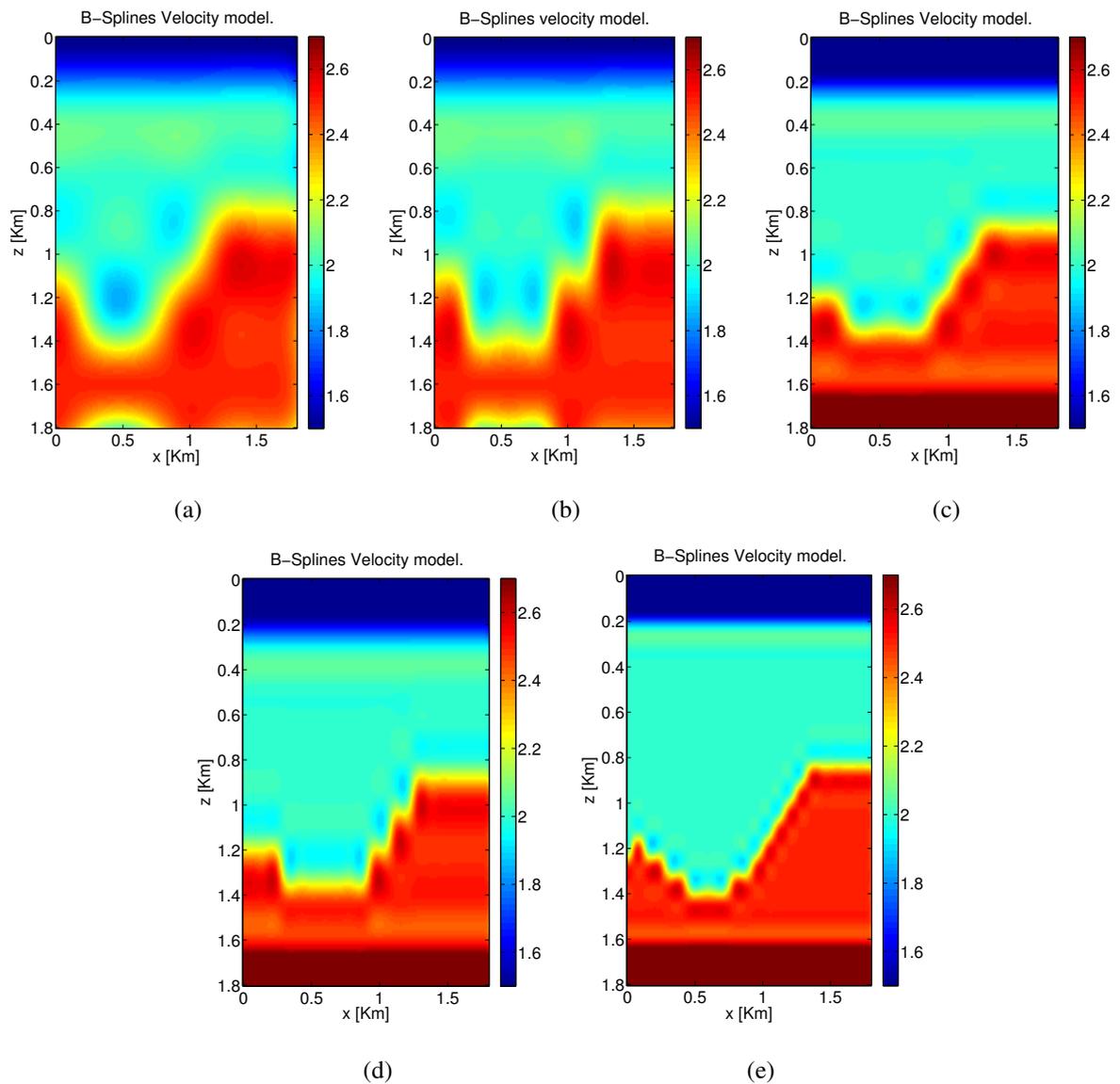


Figure 6.31: True velocity model reconstructed with B-splines coefficients using the same sampling as the one used in the five multi-step inversion stages (a)-(e).

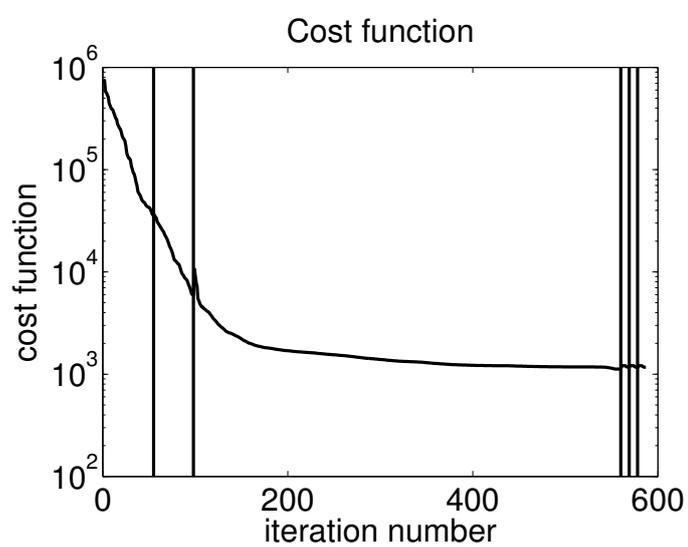


Figure 6.32: Cost function for the synthetic data example. The vertical lines correspond to the divisions in the inversion stages.

# Chapter 7

## Conclusions

In this thesis we discussed the problem of slopes estimation for a better initialization of 2D stereotomography. This is an important topic, since stereotomography is highly affected by its initialization, as it employs local optimization methods to solve the inverse problem.

We proposed to use the attributes obtained as outputs of the common-offset common-reflection-surface (CO-CRS) method as input data for stereotomography. We also proposed to use the global optimization method known as differential evolution (DE) to estimate the CO-CRS attributes. The method currently used to estimate the CO-CRS attributes is divided in a sequence of steps, where at each step some approximations are used in the estimation. By analyzing our tests, we may conclude that the most significant differences between the global and sequential CO-CRS parameter estimation strategies are:

- The global CO-CRS strategy presents better estimates in data sets highly corrupted by noise. The possible limitation for the sequential CO-CRS is that the method is strongly dependent of each search step. If the estimation fails in some of these steps, the following ones might present poor performances.
- The computational cost is higher for the global CO-CRS. However, the computational cost in the sequential CO-CRS can also be quite expensive. If we choose a bi-parametric search in the first sequential step, the sequential CO-CRS will present approximately the same computational cost as the global CO-CRS, with, most likely, a worse performance.

The link between the CO-CRS attributes and the stereotomography initialization requires a picking strategy. We completed this work-flow by proposing an automatic picking strategy to extract the slopes used in stereotomography, from the CO-CRS attributes gathers. We adopted a simplification of some existing methods, which are used to obtain the ZO-CRS parameters for the NIP tomography. Nevertheless, we still see possible improvements to be adopted in our automatic picking proposal,

---

specially regarding the formation of aligned windows and application of filtering methods in the coherence gathers.

We illustrated, with synthetic examples, the benefits of using the global CO-CRS slopes as initialization, for seismic data sets highly corrupted by noise. We now list the most significant aspects of these tests:

- In our tests, the stereotomography inversion with the slopes obtained from the slant stack method presented good performance only in seismic data with reasonably good signal-to-noise ratios (SNR). In data sets more corrupted by noise, the slopes obtained from the sequential and global CO-CRS presented better inversion results.
- The method more robust to noisy data was the global CO-CRS, which was the only method capable of presenting satisfactory results for all the data sets in our tests.

For the application of this complete work-flow in more complex data, we see the potential of exploring different aspects, in order to obtain satisfactory results. As future prospects for the continuation of these work, we highlight:

- The stereotomography initialization is still a great issue and the combination of slopes picked with the CO-CRS method from different CO gathers may increase the quality of the results.
- Local optimization is a very sensitive approach to solve the inversion problem. In most of our tests we obtained velocity models that were probably a local minimum solution. We believe that testing different regularization functions or even better velocities initialization could mitigate this problem.
- Regarding the stereotomography input, the proposed work-flow of slopes computation and picking should also be investigated in more complex synthetic and field seismic data.

## Bibliography

- M. Alérini, G. Lambaré, R. Baina, P. Podvin, and S. Le Bégat. 2D PP/PS-stereotomography: Application to a real 2D-OBC dataset. *Geophysical Prospecting*, 56(2):213–227, 2008.
- T. Barros, R. Ferrari, R. Krummenauer, R. Lopes, and M. Tygel. The impact of the parameter estimation strategy in the CRS method. In *SBGf 13th International Congress of the Brazilian Geophysical Society*, 2013.
- T. Barros, R. Ferrari, R. Krummenauer, and R. Lopes. Differential evolution-based optimization procedure for automatic estimation of the common-reflection surface traveltime parameters. *Geophysics*, 80(6):WD189–WD200, 2015.
- T. Barros, R. Krummenauer, R. Lopes, and H. Chauris. Pre-stack data recovery through common offset CRS stack with differential evolution. In *78th EAGE Conference and Exhibition 2016*, 2016.
- M. Baykulov and D. Gajewski. Prestack seismic data enhancement with partial common-reflection-surface (CRS) stack. *Geophysics*, 74(3):V49–V58, 2009.
- S. Bergler. The common-reflection-surface stack for common offset-theory and application. *Master's thesis, Universität Karlsruhe (Germany)*, 2001.
- D.P. Bertsekas. Projected newton methods for optimization problems with simple constraints. *SIAM Journal on control and Optimization*, 20(2):221–246, 1982.
- F. Billette. *Velocity macro model estimation in seismic reflection by Stereotomography*. PhD thesis, Paris 7 Denis Diderot University, 1998. English version.
- F. Billette and G. Lambaré. Velocity macro-model estimation from seismic reflection data by stereotomography. *Geophysical Journal International*, 135(2):671–690, 1998.
- F. Billette, S. Le Bégat, P. Podvin, and G. Lambaré. Practical aspects and applications of 2D stereotomography. *Geophysics*, 68(3):1008–1021, 2003.

- R. Biloti. GêBR: a free seismic processing interface. In *SBGf 12th International Congress of the Brazilian Geophysical Society*, 2011.
- R. Biloti. GêBR: a free seismic processing interface. In *74th EAGE Conference and Exhibition-Workshops*, 2012.
- Ernesto G Birgin, Jose Mario Martínez, Marcos Raydan, et al. Spectral projected gradient methods: review and perspectives. *J. Stat. Softw*, 60(3):1–21, 2014.
- T.N. Bishop, K.P. Bube, R.T. Cutler, R.T. Langan, P.L. Love, J.R. Resnick, R.T. Shuey, D.A. Spindler, and H.W. Wyld. Tomographic determination of velocity and depth in laterally varying media. *Geophysics*, 50(6):903–923, 1985.
- L. Boccato, R. Krummenauer, R. Attux, and A. Lopes. Application of natural computing algorithms to maximum likelihood estimation of direction of arrival. *Signal Processing*, 92(5):1338 – 1352, 2012. ISSN 0165-1684. doi: <http://dx.doi.org/10.1016/j.sigpro.2011.12.004>.
- A. Bourgeois, M. Bouget, P. Lailly, M. Poulet, P. Ricarte, and R. Versteeg. Marmousi, model and data. In *52nd. EAGE meeting, Proceedings of the Workshop on Practical Aspects of Seismic Data Inversion, Copenhagen.*, 1991.
- C. Bunks, F.M. Saleck, S. Zaleski, and G. Chavent. Multiscale seismic waveform inversion. *Geophysics*, 60(5):1457–1473, 1995.
- J.C. Butcher. *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*. Wiley-Interscience, 1987.
- J.C. Butcher. Runge-Kutta methods. *Scholarpedia*, 2(9):3147, 2007.
- R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.
- V. Červený. *Seismic ray theory*. Cambridge university press, 2005.
- V. Červený and R. Ravindra. *Theory of seismic head waves*. University of Toronto Press, 1971.
- H. Chauris, M. Noble, G. Lambaré, and P. Podvin. Migration velocity analysis from locally coherent events in 2-D laterally heterogeneous media, part I: Theoretical aspects. *Geophysics*, 67(4):1202–1212, 2002.

- S. Chiu and R.R. Stewart. Tomographic determination of three-dimensional seismic velocity structure using well logs, vertical seismic profiles, and surface seismic data. *Geophysics*, 52(8):1085–1098, 1987.
- J.C. Costa, F.J. da Silva, E.N. Gomes, J. Schleicher, L.A. Melo, and D. Amazonas. Regularization in slope tomography. *Geophysics*, 73(5):VE39–VE47, 2008.
- S. Das and P.N. Suganthan. Differential Evolution: A Survey of the State-of-the-Art. *Evolutionary Computation, IEEE Transactions on*, 15(1):4–31, 2011. ISSN 1089-778X. doi: 10.1109/TEVC.2010.2059031.
- D. Datta and M.K. Sen. Estimating a starting model for full-waveform inversion using a global optimization method. *Geophysics*, 2016.
- L.N. de Castro. Fundamentals of natural computing: an overview. *Physics of Life Reviews*, 4(1):1 – 36, 2007. ISSN 1571-0645. doi: <http://dx.doi.org/10.1016/j.plrev.2006.10.002>.
- S. Dümmling, K. Meier, D. Gajewski, and C. Hübscher. Comparison of prestack stereotomography and NIP wave tomography for velocity model building: Instances from the messinian evaporites. *Geophysics*, 73(5):VE291–VE302, 2008.
- E. Duveneck. Velocity model estimation with data-derived wavefront attributes. *Geophysics*, 69(1): 265–274, 2004.
- H.W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.
- V. Farra and R. Madariaga. Seismic waveform modeling in heterogeneous media by ray perturbation theory. *Journal of Geophysical Research: Solid Earth*, 92(B3):2697–2712, 1987.
- V. Farra and R. Madariaga. Non-linear reflection tomography. *Geophysical Journal International*, 95(1):135–147, 1988.
- S. Fomel, S. Luo, and H. Zhao. Fast sweeping method for the factored eikonal equation. *Journal of Computational Physics*, 228(17):6440–6455, 2009.
- G. Garabito, P.L. Stoffa, L. Lucena, and J. Cruz. Part I-CRS stack: Global optimization of the 2D CRS-attributes. *Journal of Applied Geophysics*, 85(0):92 – 101, 2012.
- G. Garabito, P. Stoffa, L. Lucena, and W. Söllner. Global optimization of the common-offset CRS-attributes: Synthetic and field data application. In *SBGf 13th International Congress of the Brazilian Geophysical Society*, 2013.

- G. Gardner and L. Lu. *Slant-stack processing*. Society of Exploration Geophysicists, 1991.
- A. Gholami and H.R. Siahkoohi. Regularization of linear and non-linear geophysical ill-posed problems with joint sparsity constraints. *Geophysical Journal International*, 180(2):871–882, 2010.
- A. Gosselet and S. Le Bégat. Combining borehole and surface seismic data for velocity field estimation through slope tomography. *Geophysical Journal International*, 176(3):897–908, 2009.
- Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*, volume 4. Siam, 2005.
- G. Höcht, E. De Bazelaire, P. Majer, and P. Hubral. Seismics and optics: hyperbolae and curvatures. *Journal of Applied Geophysics*, 42(3):261–281, 1999.
- G. Hoecht, P. Ricarte, S. Bergler, and E. Landa. Operator-oriented CRS interpolation. *Geophysical Prospecting*, 57(6):957–979, 2009.
- R. Jäger, J. Mann, G. Höcht, and P. Hubral. Common-reflection-surface stack: image and attributes. *Geophysics*, 66:97–109, 2001.
- T. Klüver and J. Mann. Smoothing and automated picking of kinematic wavefield attributes. In *9th International Congress of the Brazilian Geophysical Society*, 2005.
- G. Lambaré. Stereotomography. *Geophysics*, 73(5):VE25–VE34, 2008.
- G. Lambaré, P.S. Lucio, and A. Hanyga. Two-dimensional multivalued traveltime and amplitude maps by uniform sampling of a ray field. *Geophysical Journal International*, 125(2):584–598, 1996.
- G. Lambaré, M. Alerini, R. Baina, and P. Podvin. Stereotomography: a semi-automatic approach for velocity macromodel estimation. *Geophysical Prospecting*, 52(6):671–681, 2004.
- S. Le Bégat, H. Chauris, V. Devaux, S. Nguyen, and M. Noble. Velocity model estimation for depth imaging: Comparison of three tomography methods on a 2d real data set. *Geophysical prospecting*, 52(5):427–438, 2004.
- C.C. Lopez. *Speed up and regularization techniques for seismic full waveform inversion*. PhD thesis, Earth Sciences. Université Nice Sophia Antipolis, 2014. English version.
- J. Mann and E. Duvencok. Event-consistent smoothing in generalized high-density velocity analysis. In *SEG Technical Program Expanded Abstracts 2004*, pages 2176–2179. Society of Exploration Geophysicists, 2004.

- J. Mann, R. Jäger, T. Müller, G. Höcht, and P. Hubral. Common-reflection-surface stack - a real data example. *Journal of Applied Geophysics*, 42(3-4):301–318, 1999.
- G.F. Margrave. Numerical methods of exploration seismology with algorithms in Matlab. *The University of Calgary Publication*, 2001.
- W.H. Mayne. Common reflection point horizontal data stacking techniques. *Geophysics*, 27:927–938, December 1962.
- L. Métivier and R. Brossier. The SEISCOPE optimization toolbox: A large-scale nonlinear optimization library based on reverse communication. *Geophysics*, 2016.
- L. Métivier, F. Bretaudeau, R. Brossier, S. Operto, and J. Virieux. Full waveform inversion and the truncated newton method: Quantitative imaging of complex subsurface structures. *Geophysical Prospecting*, 62(6):1353–1375, 2014.
- N.-A. Müller, M. Brucher, and F. Clement. 3D common-offset CRS stack: Simplified formulation and application to Foothills data. In *SEG Technical Program Expanded Abstracts 2013*, pages 4262–4266. Society of Exploration Geophysicists, 2013.
- N. Neidell and M. Taner. Semblance and other coherency measures for multichannel data. *Geophysics*, 36(3):482–497, July 1971.
- S. Nguyen, R. Baina, M. Alerini, G. Lambaré, V. Devaux, and M. Noble. Stereotomography assisted by migration of attributes. *Geophysical Prospecting*, 56(5):613–625, 2008.
- J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.
- J. Nocedal and S. Wright. *Numerical optimization*. Springer, 2006.
- R. Ottolini. Signal/noise separation in dip space. *SEP report*, 37:143–149, 1983.
- R.E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503, 2006.
- E. Polak and G. Ribiere. Note sur la convergence de méthodes de directions conjuguées. *Revue française d'informatique et de recherche opérationnelle. Série rouge*, 3(16):35–43, 1969.
- W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in Fortran 77 and Fortran 90: Source Code for Recipes and Example Programs*. Cambridge University Press Cambridge, 1997.

- V. Prieux, G. Lambaré, S. Operto, and J. Virieux. Building starting models for full waveform inversion from wide-aperture data by stereotomography. *Geophysical Prospecting*, 61(s1):109–137, 2013.
- S. Rainer. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, Technical Report TR-95-012, ICSI, 1995.
- Marcos Raydan. On the barzilai and borwein choice of steplength for the gradient method. *IMA Journal of Numerical Analysis*, 13(3):321–326, 1993.
- A. Sajeva, M. Aleari, E. Stucchi, N. Bienati, and A. Mazzotti. Estimation of acoustic macro models using a genetic full-waveform inversion: Applications to the marmousi model. *Geophysics*, 2016.
- J. Schleicher, J.C. Costa, L.T. Santos, A. Novais, and M. Tygel. On the estimation of local slopes. *Geophysics*, 74(4):P25–P33, 2009.
- P.S. Schultz and J.F. Claerbout. Velocity estimation and downward continuation by wavefront synthesis. *Geophysics*, 43(4):691–714, 1978.
- B. Schwarz, C. Vanelle, S. Wißmath, A. Bauer, and D. Gajewski. Efficient common-reflection-surface-based prestack slope determination for stereotomography. In *SEG Technical Program Expanded Abstracts 2015*, pages 5248–5253. Society of Exploration Geophysicists, 2015.
- M. Sen and P. Stoffa. Global Optimization Methods in Geophysical Inversion: Advances in Exploration Geophysics. *Elsevier*, 4, 1995.
- C. Shin, S. Jang, and D.-J. Min. Improved amplitude preservation for prestack depth migration by inverse scattering theory. *Geophysical prospecting*, 49(5):592–606, 2001.
- J.W. Stockwell. The CWP/SU: Seismic Unix package. *Computers & Geosciences*, 25(4):415–419, 1999.
- P.L. Stoffa. *Tau-p: A plane wave approach to the analysis of seismic data*. Springer Science & Business Media, 2012.
- R. Storn and K. Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
- C.H. Sword. *Tomographic determination of interval velocities from reflection seismic data: The method of controlled directional reception*. PhD thesis, Stanford University, 1987.
- A. Tarantola. *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*. Elsevier, 1987.

- A. Tarantola. *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.
- F.B. Tavakoli, S. Operto, A. Ribodetti, and J. Virieux. Slope tomography based on eikonal solvers and the adjoint-state method. *Geophysical Journal International*, 209(3):1629–1647, 2017.
- M. Unser. Splines: A perfect fit for signal and image processing. *Signal Processing Magazine, IEEE*, 16(6):22–38, 1999.
- M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part I - Theory. *IEEE transactions on signal processing*, 41(2):821–833, 1993a.
- M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part II - Efficient design and applications. *IEEE transactions on signal processing*, 41(2):834–848, 1993b.
- J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009.
- Y. Zhang, S. Bergler, and P. Hubral. Common-reflection-surface (CRS) stack for common offset. *Geophysical Prospecting*, 49(6):709–718, 2001.
- H. Zhao. A fast sweeping method for eikonal equations. *Mathematics of computation*, 74(250):603–627, 2005.
- C. Zhu, R.H. Byrd, P. Lu, and J. Nocedal. LBFGS-B: Fortran subroutines for large-scale bound constrained optimization. *Report NAM-11, EECS Department, Northwestern University*, 1994.

# Appendix A

## B-Splines

One of the key aspects in stereotomography is the velocity model representation. In seismic tomography, the usual approach to represent the velocity model is to use representations such as:

$$v(x, z) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_z-1} w_{ij} B_i(x) B_j(z), \quad (\text{A.1})$$

where  $B_i(x)$  and  $B_j(z)$  represent a basis for the expansion of the velocity using, for instance, B-splines [Tarantola, 2005],  $w_{ij}$  are the coefficients of this expansion and  $N_x$  and  $N_z$  are, respectively, the number of rows and columns in the coefficients matrix. This approach with the B-splines is also common in stereotomography. We provide, now, more information about how we addressed the B-splines velocity model representation.

### A.1 Definition

Splines are piecewise polynomials with pieces that are smoothly connected together. The joining points of the polynomials are called knots. For a spline of degree  $n$ , each segment is a polynomial of degree  $n$ , so we need  $(n + 1)$  coefficients to describe each piece [Unser et al., 1993a,b]. The splines can be characterized in terms of a B-spline (basis spline) expansion:

$$s(x) = \sum_k c(k) \beta^n(x - k), \quad k = 0, \dots, N - 1. \quad (\text{A.2})$$

In the B-splines expansion,  $\beta^n$  is a central B-spline of degree  $n$  and  $c(k)$  are the B-splines coefficients. In this work, we used B-splines of degree  $n = 3$ , defined by:

$$\beta^3(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{|x|^3}{2}, & 0 \leq |x| < 1 \\ \frac{(2-|x|)^3}{6}, & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases} . \quad (\text{A.3})$$

We also used the first and second order derivatives of  $\beta^3$ , defined, respectively, by:

$$\frac{d\beta^3(x)}{dx} = \begin{cases} \frac{x(3|x|-4)}{2}, & 0 \leq |x| < 1 \\ \frac{-x(2-|x|)^2}{2|x|}, & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases} \quad (\text{A.4})$$

and

$$\frac{d^2\beta^3(x)}{dx^2} = \begin{cases} \frac{3x^2}{|x|-2}, & 0 \leq |x| < 1 \\ (2 - |x|), & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases} . \quad (\text{A.5})$$

## A.2 B-splines coefficients

Traditionally, the B-spline interpolation problem has been approached using a matrix framework and setting up a system of equations, which is then solved using standard numerical techniques. A different method for obtaining the B-splines coefficients using simpler digital filtering techniques was proposed by Unser [1999].

As described in Unser [1999], the B-splines coefficients can be obtained via digital filtering, using a filtering operation. In other words, given the digital sampled input signal  $\{s(k)\}_{k=0}^{N-1}$ , the B-splines are obtained by two consecutive filtering operations:

$$c^+(k) = s(k) + z_1 c^+(k-1), \quad k = 1, \dots, N-1, \quad (\text{A.6})$$

$$c^-(k) = z_1(c^-(k+1) - c^+(k)), \quad k = N-2, \dots, 0. \quad (\text{A.7})$$

with  $z_1 = -2 + \sqrt{3}$ . The first filter, with output given by  $c^+(k)$ , is causal, running from left to right, while the second filter, with output given by  $c^-(k)$ , is anti-causal, running from right to left. We illustrate the filters diagram in Figure A.1.

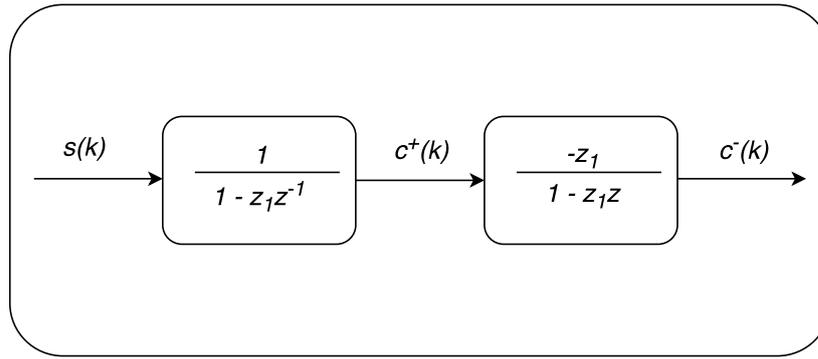


Figure A.1: Filters for B-Spline coefficients computation.

The starting values for the two filters,  $c^+(0)$  and  $c^-(N - 1)$ , are given by

$$c^+(0) = 6 \sum_{k=0}^{N-1} s(k) z_1^k, \quad (\text{A.8})$$

and

$$c^-(N - 1) = \frac{z_1}{z_1^2 - 1} (c^+(N - 1) + z_1 c^+(N - 2)). \quad (\text{A.9})$$

Because of the B-spline compact support, to interpolate signals with cubic B-splines we need to use only 4 neighbouring B-splines. The digital interpolated signal,  $s(m)$ , interpolated with B-splines coefficients, is given by:

$$s(m) = \sum_{i=k_1}^{k_1+K-1} c(i) \beta^3 \left( \frac{m}{r} - i \right), \quad (\text{A.10})$$

where each interpolated point is created at the position  $m = x/d_g$ , with  $d_g$  being the sample rate in space;  $r$  is the interpolation factor;  $K = n + 1$  is the B-spline support factor (for cubic B-splines,  $n = 3$  and  $K = 4$ ); and  $k_1 = \lfloor m/r \rfloor$ .

### A.3 2D B-splines

The two-dimension (2D) cubic B-splines interpolation is defined as:

$$s(x, z) = \sum_k \sum_l c(k, l) \beta^3(x - k) \beta^3(z - l), \quad (\text{A.11})$$

with  $k = 0, \dots, N - 1$ ,  $l = 0, \dots, M - 1$  and  $x, z \in \mathbb{R}$ ,  $k, l \in \mathbb{Z}$ .

It is shown in Unser [1999] that the 2D B-splines coefficients can be obtained by two consecutive 1D filtering operations, in the  $x$  and  $z$  dimensions, using the same 1D filter, with the input sampled

signal  $s(k, l)$ , for  $k = 0, \dots, N - 1$  and  $l = 0, \dots, M - 1$ . In that way, the sampled 2D B-spline interpolation might be given by the following expression:

$$s(m_1, m_2) = \sum_{i=k_1}^{k_1+K-1} \sum_{j=l_1}^{l_1+K-1} c(i, j) \beta^3\left(\frac{m_1}{r} - i\right) \beta^3\left(\frac{m_2}{r} - j\right). \quad (\text{A.12})$$

Now, each interpolated point is created at the position  $m_1 = x/d_g$  and  $m_2 = z/d_g$ , with  $k_1 = \lfloor m_1/r \rfloor$  and  $l_1 = \lfloor m_2/r \rfloor$ .

## Appendix B

### Gradient method

The functional  $J(\mathbf{m}) : \mathbb{R}^N \rightarrow \mathbb{R}$ , described as the stereotomography cost function in chapter 2, is known as the error surface for the stereotomography minimization problem, where  $N$  is the number of elements in the model parameter vector,  $\mathbf{m}$ . The minimization problem consists in finding the vector  $\mathbf{m}^* \in \mathbb{R}^N$  which minimizes the cost function  $J(\mathbf{m})$ :

$$\mathbf{m}^* = \arg \min_{\mathbf{m}} J(\mathbf{m}), \quad (\text{B.1})$$

with  $\mathbf{m} \in \mathbb{R}^N$ .

This problem can be solved iteratively, with the use of the initial condition  $\mathbf{m}_0$ , by an optimization algorithm of the form:

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \mathbf{f}_k, \quad (\text{B.2})$$

where the variable  $k \geq 0$  defines the iteration number,  $\mathbf{m}_k \in \mathbb{R}^N$  is the parameter vector,  $\alpha_k \in \mathbb{R}^+$  is a scalar that defines the step size and  $\mathbf{f}_k \in \mathbb{R}^N$  is a vector that defines the direction of adjustment.

One of the optimization algorithms most employed in minimization problems is the gradient method [Nocedal and Wright, 2006]. The gradient method (also known as steepest-descent) is a first order method, once it uses only the first order derivatives of the cost function  $J(\mathbf{m})$ . In this method, the direction of adjustment is given by:

$$\mathbf{f} = -\nabla J(\mathbf{m}). \quad (\text{B.3})$$

The adjustment, in the gradient method, is, then, given by:

$$\mathbf{m}_{k+1} = \mathbf{m}_k - \alpha_k \nabla J(\mathbf{m}_k). \quad (\text{B.4})$$

Finding the step size at each iteration,  $\alpha_k$ , is a different problem, largely addressed in the literature [Nocedal and Wright, 2006].

Interesting alternatives to the gradient method would be the methods which try to improve the steepest-descent gradient, such as the conjugate gradient method [Nocedal and Wright, 2006]. The conjugate gradient algorithm is an iterative linear solver for symmetric positive definite systems. This method can be interpreted as a minimization algorithm for quadratic functions [Métivier and Brossier, 2016]. Also, there are other alternatives which use the second-order information, present in the Hessian matrix, such as the Newton method [Nocedal and Wright, 2006], or even methods which try to approximate the computation of this matrix, such as the BFGS method [Nocedal and Wright, 2006].

In the next sections we briefly introduce some of the methods that we investigated in our work, which are related to the step size determination and to the use of the nonlinear conjugate gradient, which is an extension of the conjugate gradient to nonlinear functions.

## B.1 Step size determination

The step size value plays a crucial role the optimization techniques based on the gradient, once it determines how much the solution moves on the gradient direction [Nocedal and Wright, 2006]. Too small values may cause a slow convergence for the algorithm, or even worse, may imply result in failure to converge. On the other hand, too large values may cause the algorithm to oscillate in the solution neighbouring, failing to converge.

The search for the step size value usually involves, at each iteration of the optimization technique, the solution of a unidimensional minimization of the type:

$$\alpha_k = \arg \min_{\alpha} J(\mathbf{m}_k + \alpha \mathbf{f}_k). \quad (\text{B.5})$$

Hence, at each  $k$ -th iteration, we apply an algorithm to find a value of  $\alpha_k > 0$  which minimizes equation (B.5). The line search methods are the most employed methods to estimate the step size. Some of the most known line search algorithms are the back-tracking, the Fibonacci and the golden ratio search methods [Nocedal and Wright, 2006]. A different approach for finding the step size is to use the spectral projected gradient methods [Raydan, 1993, Birgin et al., 2014], which can achieve faster convergence for steepest-descent methods.

## B.2 Nonlinear conjugate gradient

The nonlinear conjugate gradient method is an extension of the conjugate gradient algorithm to the minimization of general nonlinear functions. In the nonlinear conjugate gradient the model update is still performed with equation (B.2) and the step size is computed as in equation (B.5). But we have now a coefficient,  $\beta_k \in \mathbb{R}$ , in the computation of the direction:

$$\begin{cases} \mathbf{f}_0 &= -\nabla J(\mathbf{m}_0), \\ \mathbf{f}_k &= -\nabla J(\mathbf{m}_k) + \beta_k \mathbf{f}_{k-1}, \quad k \geq 1. \end{cases} \quad (\text{B.6})$$

The coefficient  $\beta_k$  tries to approximate second-order information and is computed with first-order information, given by the gradient. We choose to compute  $\beta_k$  with the Polak-Ribiere method [Polak and Ribiere, 1969], which is a popular method in the literature due to its good results. The coefficient  $\beta_k$ , computed with the Polak-Ribiere method, is given by

$$\beta_k = \frac{\nabla J(\mathbf{m}_k)^T (\nabla J(\mathbf{m}_k) - \nabla J(\mathbf{m}_{k-1}))}{\nabla J(\mathbf{m}_{k-1})^T \nabla J(\mathbf{m}_{k-1})}. \quad (\text{B.7})$$

## Appendix C

# Broyden-Fletcher-Goldfarb-Shanno method (BFGS)

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [Nocedal and Wright, 2006], is a *quasi*-Newton method, which tries to estimate iteratively the inverse of the hessian matrix,  $\mathbf{H}$ .

To compute the BFGS algorithm, assume that the cost function  $J(\mathbf{m})$  has continuous partial derivatives until second order. Define  $\mathbf{g}_k = -\nabla J(\mathbf{m}_k)^T$  and  $\mathbf{g}_{k+1} = -\nabla J(\mathbf{m}_{k+1})^T$ . If the hessian,  $\nabla^2 J(\mathbf{m})$ , is constant, then:

$$\mathbf{q}_k \equiv \mathbf{g}_{k+1} - \mathbf{g}_k = \nabla^2 J(\mathbf{m})\mathbf{p}_k, \quad (\text{C.1})$$

$$\mathbf{p}_k = \alpha_k \mathbf{f}_k. \quad (\text{C.2})$$

In this context, we have the direction of adjustment given by  $\mathbf{f}_k = \mathbf{H}_k \mathbf{g}_k$ .

The BFGS algorithm determines the matrix  $\mathbf{H}_k$ , iteratively, by:

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} \left[ 1 + \frac{\mathbf{q}_k^T \mathbf{H}_k \mathbf{q}_k}{\mathbf{p}_k^T \mathbf{q}_k} \right] - \frac{\mathbf{H}_k \mathbf{q}_k \mathbf{p}_k^T + \mathbf{p}_k \mathbf{q}_k^T \mathbf{H}_k}{\mathbf{p}_k^T \mathbf{q}_k}, \quad (\text{C.3})$$

with  $k = 0, \dots, N - 1$ . The startup is  $\mathbf{H}_0 = \mathbf{I}_N$ , where  $\mathbf{I}_N$  is the identity matrix with dimension  $N$ .

### C.1 Limited memory BFGS

Unfortunately, the BFGS method presents large limitations concerning its implementation for a large number of model parameters. For large values of  $N$ , the BFGS starts to occupy much computer memory, which significantly slows its execution. To overcome this issue, the work of Nocedal [1980] proposes the *limited memory* BFGS (*l*-BFGS). Later, it was proposed the limited memory BFGS *bounded* (*l*-BFGS-*b*) [Zhu et al., 1994, Byrd et al., 1995], for bounded problems.

This algorithm was implemented for generical optimization problems in the FORTRAN programming language and made available by its authors [Zhu et al., 1994]. This implementation of l-BFGS-b is highly used for different sort of optimization problems and we integrated this library in our stereotomography implementation.

The l-BFGS-b is employed in stereotomography in the sense of minimizing a nonlinear function of  $N$  variables<sup>1</sup>,

$$\min J(\mathbf{m}) \quad (\text{C.4})$$

subject to

$$\mathbf{l} \leq \mathbf{m} \leq \mathbf{u}, \quad (\text{C.5})$$

with the vector  $\mathbf{l}$  and  $\mathbf{u}$  representing the lower and upper bounds of the variables. The user must provide the gradient to the algorithm, but no knowledge about the Hessian is necessary. In order to approximate the Hessian matrix the authors use limited memory BFGS matrices.

The l-BFGS-b estimates an approximate inverse Hessian operator  $\mathbf{Q}_k$ . The resulting descent direction is computed as

$$\mathbf{f}_k = -\mathbf{Q}_k \nabla J(\mathbf{m}_k). \quad (\text{C.6})$$

The algorithm is also appropriate to solve unbounded problems, making it unnecessary to provide the bounds to all the variables. The authors also use the gradient projection method to determine a set of active constraints at each iteration. This is performed before the line search to determine the step size  $\alpha$ . The line search is done with the Wolfe condition [Nocedal and Wright, 2006], which presents two testing conditions.

At each iteration of the line search, before testing the two Wolfe conditions, the updated model is projected into the feasible domain  $\Omega$  through the operator  $\text{Proj}(\mathbf{m}, \mathbf{l}, \mathbf{u})$  defined, for the  $i$ -th element of the vectors  $\mathbf{m}$ ,  $\mathbf{l}$ , and  $\mathbf{u}$ , as

$$\text{Proj}(m_i, l_i, u_i) = \begin{cases} m_i, & \text{if } l_i \leq m_i \leq u_i, \\ l_i, & \text{if } m_i < l_i, \\ u_i, & \text{if } m_i > u_i, \end{cases} \quad (\text{C.7})$$

This procedure ensures that the estimated model always remains in the feasible domain  $\Omega$  [Métivier and Brossier, 2016].

The two Wolfe conditions are the sufficient decrease condition

$$J(\mathbf{m}_k + \alpha \mathbf{f}_k) \leq J(\mathbf{m}_k) + c_1 \alpha \nabla J(\mathbf{m}_k)^T \mathbf{f}_k \quad (\text{C.8})$$

---

<sup>1</sup>In order to ease the notation we consider  $N$  instead of the  $N + M$  variables defined in chapter 2.

and the curvature condition

$$J(\mathbf{m}_k + \alpha \mathbf{f}_k)^T \mathbf{f}_k \geq c_2 \nabla J(\mathbf{m}_k)^T \mathbf{f}_k, \quad (\text{C.9})$$

with the coefficients  $c_1$  and  $c_2$  being in the interval

$$0 < c_1 < c_2 \leq 1. \quad (\text{C.10})$$

In practice we may use  $c_1 = 10^{-4}$  and  $c_2 = 0.9$ , as suggested in the works of Nocedal and Wright [2006] and Métivier and Brossier [2016].

For more information about the l-BFGS-b we suggest the reader to see the works of Zhu et al. [1994] and Byrd et al. [1995].

# Appendix D

## Differential evolution (DE)

Differential Evolution (DE) [Storn and Price, 1997] is a parallel direct search method for continuous space variables which utilizes  $N_P$   $D$ -dimensional parameter vectors  $\mathbf{x}_{i,G}$ ,  $i = 1, \dots, N_P$  as population, on each generation  $G$ . Each parameter vector constitutes a candidate solution of the optimization problem. The DE algorithm is divided in three stages: *mutation*, *crossover* and *selection*.

The mutation operation generates of a new vector for each individual by the following expression

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F(\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}). \quad (\text{D.1})$$

The indexes  $r_1, r_2, r_3 \in \{1, \dots, N_P\}$  are mutually distinct, chosen randomly and different from the index  $i$ .  $F$  is a real and constant factor in the range of  $[0, 2]$ , which controls the length of the step given in the direction defined by  $\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}$ . The crossover operation is employed with the goal of enhancing the diversity of the mutated parameter vectors. Let  $\mathbf{x}_{i,G}$  be the vector under analysis and  $\mathbf{v}_{i,G+1}$  the mutated vector obtained by equation (D.1). The crossover resultant vector  $\mathbf{u}_{i,G+1}$  is obtained by

$$\mathbf{u}_{ji,G+1} = \begin{cases} \mathbf{v}_{ji,G+1}, & \text{if } r_j \leq CR \quad \text{or} \quad j = l_i \\ \mathbf{x}_{ji,G+1}, & \text{if } r_j > CR \quad \text{and} \quad j \neq l_i \end{cases}, \quad (\text{D.2})$$

where  $j = 1, \dots, D$ ,  $r_j \sim U(0, 1)$ ,  $CR \in [0, 1]$  is the crossover constant factor defined by the user and  $l_i$  is a random index  $\in \{1, \dots, D\}$ , which ensures that  $\mathbf{u}_{i,G+1}$  receives at least one component from  $\mathbf{v}_{i,G+1}$ . After the stages of mutation and crossover, the selection of the vectors to be preserved in the next generation is made by the use of a greedy criterion. The vector  $\mathbf{u}_{i,G+1}$  is compared to the vector  $\mathbf{x}_{i,G}$ . If vector  $\mathbf{u}_{i,G+1}$  yields a larger cost function value than  $\mathbf{x}_{i,G}$ , then  $\mathbf{x}_{i,G+1}$  is set to  $\mathbf{u}_{i,G+1}$ ; otherwise, the old value  $\mathbf{x}_{i,G}$  is retained.

The differential evolution strategy employed in this work is known in the literature by the notation: *DE/rand/1/bin*. In that notation, *rand* specifies the random choice of the population vector in the

**Algorithm 1** Differential Evolution

---

```

Function  $\mathbf{x} = \text{DE}(N_P, CR, F, \text{range}, f_{cost})$ 
 $[\mathbf{x}, f_{\mathbf{x}}] \leftarrow \text{initialize}(N_P, \text{range}, f_{cost})$ 
while stopping criterion is not reached do
  for  $i = 1$  until  $N_P$  do
     $\mathbf{v}_{i,G+1} \leftarrow \text{mutation}(\mathbf{x}_{i,G}, F)$ 
     $\mathbf{u}_{i,G+1} \leftarrow \text{crossover}(\mathbf{x}_{i,G}, \mathbf{v}_{i,G+1}, CR)$ 
  end for
   $f_{\mathbf{u}} \leftarrow \text{evaluate}(\mathbf{u}, f_{cost})$ 
  for  $i = 1$  until  $N_P$  do
    if  $f_{\mathbf{u}}(i) > f_{\mathbf{x}}(i)$  then
       $\mathbf{x}_{i,G+1} \leftarrow \mathbf{u}_{i,G+1}$ 
    else
       $\mathbf{x}_{i,G+1} \leftarrow \mathbf{x}_{i,G}$ 
    end if
  end for
end while

```

---

mutation;  $l$  is the number of difference vectors used in the crossover operation and *bin* denotes the crossover scheme employed, related to independent binomial experiments. In Algorithm 1 we show the pseudocode of DE [Boccatto et al., 2012]. The parameters illustrated in Algorithm 1 are: the number of individuals in a population ( $N_P$ ); the crossover rate ( $CR$ ); the mutation scale factor ( $F$ ); the search space limits (range) and the cost function that we are evaluating to solve the problem ( $f_{cost}$ ).

In Figure D.1.1 we present a graphic illustration of the *DE/rand/1/bin* algorithm. Assume that, at a given generation we have an individual given by the vector  $\mathbf{x}_1$ , with 3 parameters to be estimated. The DE algorithm is performed as follows: First, two other individuals from the same generation,  $\mathbf{x}_2$  and  $\mathbf{x}_3$ , are randomly chosen. The mutation operation is performed by adding individual  $\mathbf{x}_1$  with the difference between individuals  $\mathbf{x}_2$  and  $\mathbf{x}_3$  weighted by the control parameter  $F$ , yielding the individual  $\mathbf{v}$ . For the crossover operation, other individual from the same generation,  $\mathbf{x}_4$ , is also randomly chosen. The crossover is the operation of replacing a position of the individual  $\mathbf{v}$  by the value of that same position of the individual  $\mathbf{x}_4$ , with a probability given by the control parameter  $CR$ , resulting in a new individual  $\mathbf{u}$ . The selection criterion is a greedy one, which selects the individual yielding the higher value of the cost function being evaluated (we illustrate the maximization case). These operations must be applied to all the individuals at every generation.

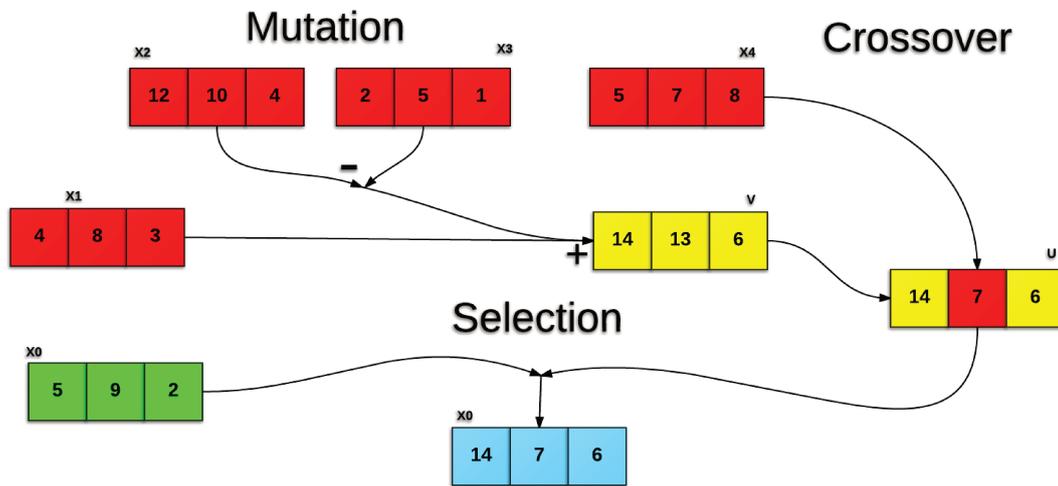


Figure D.1: DE algorithm.