AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Towards Reducing Structural Interpretation Uncertainties Using Seismic Data

THÈSE

présentée et soutenue publiquement le 25 avril 2019
pour l'obtention du grade de

Docteur de l'Université de Lorraine

Spécialité Géosciences

par

Modeste Irakarama

Composition du jury:

Alison MALCOLM	Maître de conf., Mem. University of Newfoundland	Rapporteur
Thomas BOHLEN	Professeur, Karlsruhe Institute of Technology	Rapporteur
Alexandrine GESRET	Maître de conf., MINES ParisTech	Examinateur
Isabelle LECOMTE	Maître de conf., University of Bergen	Examinateur
Paul SAVA	Professeur, Colorado School of Mines	Examinateur
Guillaume CAUMON	Professeur, Université de Lorraine	Directeur de thèse
Paul CUPILLARD	Maître de conf., Université de Lorraine	Co-dir. de thèse, invité
Pierre THORE	Chercheur Sénior, TOTAL	Invité

UMR 7359 - GeoRessources
Université de Lorraine - CNRS - CREGU
ENSG - Campus Brabois - TSA 70605
54518 Vandœuvre-lès-Nancy cedex - FRANCE

Contents

Acknowledgements **1**

Résumé **5**

Abstract **7**

Résumé long **9**

Extended Abstract **15**

Bibliography **21**

General Introduction **22**
Introduction . . . 22
Three optimization problems . . . 27
Contributions . . . 28
Seismic imaging . . . 28
Structural modeling . . . 28
Appraising structural models using seismic data . . . 28
Other related papers . . . 29
Acknowledgements . . . 29

Bibliography **30**

I Notes on Seismic Imaging **32**

Introductory Comments **33**

I.1 Seismic Imaging **34**
I.1.1 Introduction . . . 34
I.1.2 Linearized waveform inversion . . . 35
I.1.3 Nonlinear waveform inversion . . . 38
I.1.4 Discussion . . . 39
I.1.5 Appendix . . . 42

I.2 Reverse time migration as a preconditioner for waveform inversion **45**
I.2.1 Summary . . . 45
I.2.2 Introduction . . . 46
I.2.3 Theory . . . 47
I.2.3.1 Heuristics . . . 47
I.2.3.2 Underlying assumptions . . . 49
I.2.4 Numerical examples . . . 51
I.2.5 Conclusion . . . 52

I.2.6 Acknowledgements . . . 52

I.3 Extended Kirchhoff Imaging for Migration Velocity Analysis 53
I.3.1 Summary . . . 53
I.3.2 Introduction . . . 54
I.3.3 Imaging with single arrivals . . . 54
I.3.3.1 Data-domain extensions . . . 54
I.3.3.2 Image-domain extensions . . . 55
I.3.4 Imaging with multiple arrivals . . . 56
I.3.5 Velocity model building via ray tracing . . . 58
I.3.6 Velocity model building via the adjoint-state method . . . 60
I.3.7 Discussion . . . 62
I.3.8 Conclusion . . . 63
I.3.9 Acknowledgements . . . 63
I.3.10 Appendix . . . 64

Bibliography 65

II Notes on Structural Modeling 68

Introductory Comments 69

II.1 Finite difference implicit structural modeling of geological structures 72
II.1.1 Summary . . . 72
II.1.2 Introduction . . . 73
II.1.3 Domain and problem discretization . . . 75
II.1.3.1 Domain discretization . . . 75
II.1.3.2 Problem discretization . . . 77
II.1.3.3 Solving the discrete problem . . . 78
II.1.4 Interpolation in 1D: problem formulation . . . 78
II.1.4.1 Interpolation in 1D . . . 78
II.1.4.2 Problem formulation . . . 78
II.1.4.3 Towards interpolation in higher dimensions . . . 79
II.1.5 Interpolation in 2D . . . 80
II.1.5.1 Directional second derivatives . . . 80
II.1.5.2 Mixed derivatives . . . 81
II.1.6 Interpolation in 3D . . . 82
II.1.7 Handling discontinuities . . . 84
II.1.8 Limitations and discussion . . . 85
II.1.8.1 Large thickness variations . . . 85
II.1.8.2 Discontinuities . . . 86
II.1.8.3 Performance . . . 86
II.1.8.4 Beyond 3D . . . 86
II.1.9 Conclusion . . . 87
II.1.10 Acknowledgements . . . 87
II.1.11 Appendix . . . 88

II.2 Finite element implicit structural modeling of geological structures 89
II.2.1 Summary . . . 89
II.2.2 Introduction . . . 90
II.2.3 Finite elements for boundary value problems . . . 91
II.2.4 Finite elements for structural interpolation . . . 92

II.2.5 Numerical experiments . . . 94
II.2.6 Discussion & Perspectives . . . 94
II.2.7 Conclusion . . . 95
II.2.8 Acknowledgements . . . 95

II.3 Discussion 96

Bibliography 99

III Reducing Structural Uncertainties using Seismic Data 103

Introductory Comments 104

III.1 Structural Interpretation Uncertainties 106
III.1.1 Acknowledgements . . . 110

III.2 Appraising Structural Models using Seismic Data — Theory 111
III.2.1 Summary . . . 111
III.2.2 Introduction . . . 112
III.2.3 The structural interpretation appraisal problem . . . 113
III.2.4 Mathematical formulation . . . 116
III.2.5 Macro-layered velocity models . . . 117
III.2.5.1 Block macro-layered velocity model . . . 118
III.2.5.2 Wire macro-layered velocity model . . . 119
III.2.5.3 Reflectivity macro-layered velocity model . . . 120
III.2.5.4 Density macro-layered velocity models . . . 120
III.2.6 Data-space misfit functions . . . 121
III.2.6.1 Example for vertical seismic profiling data . . . 122
III.2.6.2 Example for surface seismic data . . . 123
III.2.7 Application & Discussion . . . 128
III.2.8 Conclusion . . . 130
III.2.9 Acknowledgements . . . 131

III.3 Appraising Structural Models using Seismic Data — Practice 132
III.3.1 Summary . . . 132
III.3.2 Introduction . . . 133
III.3.3 Appraising models with the same number of layers . . . 133
III.3.4 Appraising models with different number of layers . . . 142
III.3.5 Conclusion & Discussion . . . 147
III.3.6 Acknowledgements . . . 148
III.3.7 Appendix . . . 149

Bibliography 152

General Conclusion 156

IV Seismic Imaging for GeoModel Analysis - SIGMA 159

IV.1 Introducing the SIGMA library 160
IV.1.1 Introduction & Motivation . . . 160
IV.1.2 Hardware requirements . . . 160
IV.1.3 Getting started . . . 161

IV.1.4 The RSF file format . . . 161

IV.2 Tutorial I: Seismic Modeling & Imaging 164
IV.2.1 In this tutorial . . . 164
IV.2.2 Seismic modeling . . . 164
IV.2.2.1 Preparing for wave simulation . . . 164
IV.2.2.2 A seismic survey . . . 166
IV.2.3 Seismic imaging . . . 167
IV.2.3.1 Common image gathers . . . 169
IV.2.3.2 Stacking common image gathers . . . 171
IV.2.4 Acknowledgements . . . 173

IV.3 Tutorial II: Structural Interpolation 174
IV.3.1 In this tutorial . . . 174
IV.3.2 Structural interpolation . . . 174
IV.3.2.1 Interpolation in 2D . . . 174
IV.3.2.2 Interpolation in 3D . . . 178
IV.3.3 Acknowledgements . . . 181

Bibliography 182

Acknowledgements

Firstly, I would sincerely like to thank my PhD jury composed of Alison Malcolm, Thomas Bohlen, Alexandrine Gesret, and Isabelle Lecomte for accepting the invitation to evaluate my work. Isabelle Lecomte also reviewed chapter III.2 when it was submitted to Geophysics and improved the chapter considerably. Secondly, I would like to thank my supervisors Guillaume Caumon, Paul Cupillard and Paul Sava [1] . I have known Guillaume Caumon and Paul Cupillard since I was a master's student at the École Nationale Supérieure de Géologie – ENSG – in Nancy; they taught me way more than just science and I continue to learn a lot just by observing and discussing with them. I hope one day I'll be as good a mentor to my students as Guillaume Caumon and Paul Cupillard have been to me. Guillaume Caumon and Paul Cupillard have also helped me with a lot of personal issues that I'm truly greatful for. When I started my PhD, Guillaume Caumon and Paul Cupillard invited Paul Sava to join the project. Thank you Paul Sava for accepting the invitation. My knowledge of seismic imaging has greatly benefited from discussion with Paul Sava and his students, especially Daniel Rocha and Esteban Diaz. I also learn more than just science by discussing with Paul Sava.

Before I came to Nancy, I was in Toulouse where I also had remarkable mentors at the Université Toulouse III – Paul Sabatier – whom I would like to thank. Francis Odonne taught me structural geology; he somehow managed to get me very interested in his course without any equations. Christine Destrigneville and Muriel Lubes taught me thermodynamics; this was the first time I really appreciated the use of mathematics to solve "real life" practical problems. My fascination of applying mathematics to practical problems was further nurtured by Raphaël Garcia who taught me geophysics and initiated me to potential theory.

My list of remarkable teachers would not be complete without mentioning Pauline Collon, who taught me C++ programming. I think the most important skill I learned from the ENSG is probably C++ programming. Thank you Pauline! Pauline, thank you also for my final year internship at CEREMA. There is also François Bonneau, who I have known since I was a student at the ENSG. François has always been available to discuss both scientific and non-scientific matters; I feel like he has also supervised me in some sense during my stay in Nancy. When I was still a student at the ENSG, I did a summer internship at the Laboratoire de Géologie de Lyon (thanks to Paul Cupillard) where I was supervised by Benoit Tauzin, who taught me a lot about inverse problems. Over the course of my PhD, I had the opportunity to supervise three master students: Jean Langanay, Thomas Beraud, and Morgan Thierry-Coudon; I learned as much from them as they learned from me. Thomas Beraud made significant contributions to chapter I.3 and Morgan Thierry-Coudon made significant contributions to chapter II.2. Other colleagues who made significant contributions to this manuscript include Gautier Laurent (chapter II.1), Julien Renaudeau (chapter II.1. I also got the chance to collaborate with Julien on one of his papers), Mustapha Zakari (chapter II.2), and Jonathan Edwards (chapter III.2). Melchior Schuh-Senlis also made important contributions by proofreading part of this manuscript and proofchecking my algebra in chapters I.2 and I.3. There are many more colleagues who have contributed to this manuscript, some of whom are mentioned in the ac-

[1] I was not able to list Paul Sava as one of the supervisors on the cover page because of administrative constraints

knowledgements section of relevant chapters throughout the manuscript.

I'm very grateful to my PhD advisory committee composed of Ludovic Métivier and Pierre Thore. I got stuck a number of times during my PhD and discussing with Ludovic Métivier and Pierre Thore was often all that I needed to make some progress. Equally important, Ludovic Métivier and Pierre Thore pointed out many flaws in my reasoning, thus enabling me to propose less flawed arguments.

I would like to thank the following members of the Association Scientifique pour la Géologie et ses Applications – ASGA – : Armelle Faynot, Christophe Antoine, Fatima Chtioui, Maryse Villeneuve, Nicole Galloy and Sophie Romain, for making my life in Nancy much easier (and consequently happier). I also acknowledge the RING-GOCAD Consortium, managed by ASGA, for funding my PhD. I enjoyed my stay in Nancy greatly due to my thoughtful colleagues in the RING-team (in the following list I omit those already mentioned above) : Antoine Mazuyer, Arnaud Botella, Benjamin Chauvin, Corentin Gouache, Gabriel Godefroy, Margaux Raguenel, Marion Parquer, Nicolas Clausolles, Nicolas Mastio, Paul Baville, Pierre Anquez (whith whom I got an opportunity to collaborate on a paper), and Yves Frantz. Thanks guys: I'm really lucky, thankful, and happy to have been part of your team!

Last but not least, I would like to thank my family: my parents, two sisters, one brother, and my newly born niece, for their support; they have **ALWAYS** been supportive and I would not have finished my master's degree at the ENSG had it not been for their support. Speaking of everlasting support, I would also like to thank my lifelong friends, who are now practically family, Ralph P. and Michel N.

"Restons factuels les mecs!" — Guillaume Caumon

"Rien n'est simple!" — Alfred B. (a.k.a my dad)

I would like to dedicate this thesis to my mother who was convinced I had what it took from the beginning (before I started university). Thanks mum!

Résumé

Les modèles géologiques sont couramment utilisés pour estimer les ressources souterraines, pour faire des simulations numériques, et pour évaluer les risques naturels; il est donc important que les modèles géologiques représentent la géométrie des objets géologiques de façon précise. La première étape pour construire un modèle géologique consiste souvent à interpréter des surfaces structurales, telles que les failles et horizons, à partir d'une image sismique; les objets géologiques identifiés sont ensuite utilisés pour construire le modèle géologique par des méthodes d'interpolation. Les modèles géologiques construits de cette façon héritent donc les incertitudes d'interprétation car une image sismique peut souvent supporter plusieurs interprétations structurales. Dans ce manuscrit, j'étudie le problème de réduire les incertitudes d'interprtation à l'aide des données sismiques. Particulièrement, j'étudie le problème de déterminer, à l'aide des données sismiques, quels modèles sont plus probables que d'autres dans un ensemble des modèles géologiques cohérents. Ce problème sera connu par la suite comme *le problème d'évaluation des modèles géologiques par données sismiques.*

La première partie du manuscrit est consacrée à l'imagerie sismique. J'y propose de se servir de la "reverse time migration" (RTM) pour préconditionner l'inversion des formes d'ondes. Les expériences numériques montrent que le préconditionneur proposé accélère l'inversion des formes d'ondes linéarisée ("least squares reverse time migration") et l'inversion des formes d'ondes non-linéaire ("full waveform invesion") d'au moins un ordre de grandeur. Je justifie la performance numérique positive du préconditionneur proposé en montrant algébriquement qu'un filtre passe-bas d'une image RTM peut approximer les éléments de la diagonale de la matrice hessienne de la fonction-objectif sous des hypothèses appropriées. Cependant, je ne suis toujours pas en mesure de proposer une interprétation physique du filtre passe-bas et de la manière dont il relie l'image RTM aux éléments de la diagonale de la matrice hessienne. Ensuite, je propose un opérateur généralisé du "Kirchhoff extended imaging" pour la construction de modèles de vitesses; l'opérateur est généralisé dans le sens où il décrit simultanément plusieurs extensions dans le "data-domain" (ex. extension en source, extension en distance source-récepteur, extension en angle de réflexion) et des extensions dans l'"image-domain" (ex. extension en temps, extension en décalage spatial). Il existe au moins deux avantages de l'opérateur généralisé proposé: premièrement, il permet une implémentation unifiée pour plusieurs extensions (c'est-à-dire une seule implémentation valable pour plusieurs extensions); deuxièmement, lopérateur aboutit à un algorithme du gradient unifié de la "migration velocity analysis" (MVA). J'illustre la sensibilité de l'opérateur généralisé proposé aux vitesses imprécises en utilisant cet opérateur dans une expérience MVA basée sur des rayons sismiques.

La deuxième partie du manuscrit est consacrée à la modélisation structurale, en particulier à l'interpolation structurale implicite. J'y propose une nouvelle méthode sous le nom de "Finite Difference Structural Implicit Modeling" (FDSIM). Il existe au moins deux avantages de la FDSIM: premièrement, il est relativement facile d'implémenter cette méthode sur ordinateur et de l'optimiser car elle est basée sur les différences finies sur des grilles régulières; deuxièmement, comme la FDSIM gère les discontinuités par rastérisation, cette méthode s'est montrée capable de gérer facilement des réseaux de failles très complexes. Le principal inconvénient de

la méthode est qu'elle peut nécessiter une résolution très fine en fonction de la complexité du réseau de failles, conduisant parfois à des limites de mémoire. Je propose également de nouveaux opérateurs de régularisation; la particularité de ces opérateurs est qu'ils n'ont pas besoin dêtre implémentés sur les noeuds aux bords, une propriété très convenable pour la modélisation structurale implicite où les conditions aux limites sont généralement inconnues. Je propose ensuite une autre méthode sous le nom de "Finite Element Structural Implicit Modeling" (FESIM). La FESIM est basée sur une implémentation éléments finis des opérateurs de régularisation dernièrement proposés. Je montre que la méthode des éléments finis classique connue pour résoudre les problèmes aux limites doit être légèrement modifiée pour la modélisation implicite où les conditions aux limites sont généralement inconnues.

La troisième partie du manuscrit est consacrée à l'évaluation des modèles structuraux à l'aide des données sismiques. J'y introduis et formalise le problème d'évaluation des interprétations structurales à l'aide des données sismiques. Je propose de résoudre ce problème par génération des données sismiques synthétiques pour chaque interprétation structurale dans un premier temps, ensuite d'utiliser ces données synthétiques pour calculer la fonction-objectif pour chaque interprétation; cela permet de classer les différentes interprétations structurales. La difficulté majeure d'évaluer les modèles structuraux à l'aide des données sismiques consiste à proposer des fonctions-objectifs adéquates. Je propose un ensemble de conditions qui doivent être satisfaites par la fonction-objectif pour une évaluation réussie des modèles structuraux à l'aide des données sismiques. Je montre que, puisqu'il n'est pas possible de satisfaire ces conditions en utilisant les données de type "vertical seismic profile" (VSP), il n'est pas possible d'évaluer les interprétations structurales à l'aide des données VSP dans le cas le plus général. Ces conditions imposées à la fonction-objectif peuvent en principe être satisfaites en utilisant les données sismiques de surface ("surface seismic data"). Cependant, en pratique il reste tout de même difficile de proposer et de calculer des fonctions-objectifs qui satisfassent ces conditions. Je termine le manuscrit en illustrant les difficultés rencontrées en pratique lorsque nous cherchons à évaluer les interprétations structurales à l'aide des donnés sismiques de surface. Je propose une fonction-objectif générale faite de deux composants principaux: (1) un opérateur de résidus qui calcule les résidus des données (c'est-à-dire la différence entre données synthétiques et données observées), et (2) un opérateur de projection qui projette les résidus de données depuis l'espace de données vers l'espace physique (c'est-à-dire le sous-sol). Cette fonction-objectif est donc localisée dans l'espace car elle génère des valeurs en fonction de l'espace. Cependant, je ne suis toujours pas en mesure de proposer une implémentation pratique de cette fonction-objectif qui satisfasse les conditions imposées pour une évaluation réussie des interprétations structurales; cela reste un sujet de recherche.

Abstract

Subsurface structural models are routinely used for resource estimation, numerical simulations, and risk management; it is therefore important that subsurface models represent the geometry of geological objects accurately. The first step in building a subsurface model is usually to interpret structural features, such as faults and horizons, from a seismic image; the identified structural features are then used to build a subsurface model using interpolation methods. Subsurface models built this way therefore inherit interpretation uncertainties since a single seismic image often supports multiple structural interpretations. In this manuscript, I study the problem of reducing interpretation uncertainties using seismic data. In particular, I study the problem of using seismic data to determine which structural models are more likely than others in an ensemble of geologically plausible structural models. I refer to this problem as *appraising structural models using seismic data*.

The first Part of the manuscript is devoted to seismic imaging. I first propose to use reverse-time migration (RTM) as a preconditioner for waveform inversion. Numerical experiments show that the proposed preconditioner accelerates both linearized waveform inversion (least squares reverse time migration) and nonlinear waveform inversion (full waveform inversion) by at least an order of magnitude. I justify the positive numerical performance of the proposed preconditioner by showing algebraically that a low pass filter of the RTM image can approximate the diagonal elements of the Hessian matrix of the objective function under appropriate assumptions. However, I am still unable to propose a physical meaning of the low pass filtering and how it relates the RTM image to the elements of the diagonal of the Hessian matrix. Then, I propose a generalized extended Kirchhoff imaging operator for velocity modeling; the operator is generalized in the sense that it describes multiple data-domain extensions (e.g. shot, offset, and angle extensions) and image-domain extensions (e.g. time-lag and space-lag extensions) simultaneously. The advantages of the proposed generalized extended operator are twofold: firstly, it allows a unified implementation for multiple extensions (i.e. a single implementation that is valid for multiple extensions); secondly, the operator leads to a unified gradient-based migration velocity analysis (MVA) scheme. I confirm the ability of the proposed generalized extended operator to capture image distortion caused by inaccurate velocity by applying it to a ray-based MVA experiment.

The second Part of the manuscript is devoted to structural modeling, particularly implicit structural interpolation. I introduce Finite Difference Structural Implicit Modeling (FDSIM). The advantages of FDSIM are twofold: firstly, it is relatively easy to implement and to optimize since it is based on finite differences on regular grids; secondly, because it handles discontinuities by rasterization, FDSIM has shown to easily handle very complex fault networks. The main disadvantage of the method is that it may require a very fine resolution depending on the complexity of the fault network, sometimes leading to memory limits. I also propose new regularization operators; the particularity of these operators is that they do not need to be implemented on boundary nodes, a property which is very appealing in implicit modeling where boundary conditions are usually unknown. I then introduce Finite Element Structural Implicit Modeling (FESIM). FESIM is based on a finite element implementation of the newly proposed

regularization operators. I show that the conventional finite element familiar for solving boundary value problems has to be slightly modified for implicit modeling where boundary conditions are usually unknown.

The third Part of the manuscript is devoted to appraising structural models/interpretations using seismic data. I introduce and formalize the problem of appraising structural interpretations using seismic data. I propose to solve the problem by generating synthetic data for each structural interpretation and then to compute misfit values for each interpretation; this allows us to rank the different structural interpretations. The main challenge of appraising structural models using seismic data is to propose appropriate data misfit functions. I derive a set of conditions that have to be satisfied by the data misfit function for a successful appraisal of structural models. I argue that since it is not possible to satisfy these conditions using vertical seismic profile (VSP) data, it is not possible to appraise structural interpretations using VSP data in the most general case. The conditions imposed on the data misfit function can in principle be satisfied for surface seismic data. In practice, however, it remains a challenge to propose and compute data misfit functions that satisfy those conditions. I conclude the manuscript by highlighting practical issues of appraising structural interpretations using surface seismic data. I propose a general data misfit function that is made of two main components: (1) a residual operator that computes data residuals, and (2) a projection operator that projects the data residuals from the data-space into the image-domain. This misfit function is therefore localized in space, as it outputs data misfit values in the image-domain. However, I am still unable to propose a practical implementation of this misfit function that satisfies the conditions imposed for a successful appraisal of structural interpretations; this is a subject for further research.

Résumé long

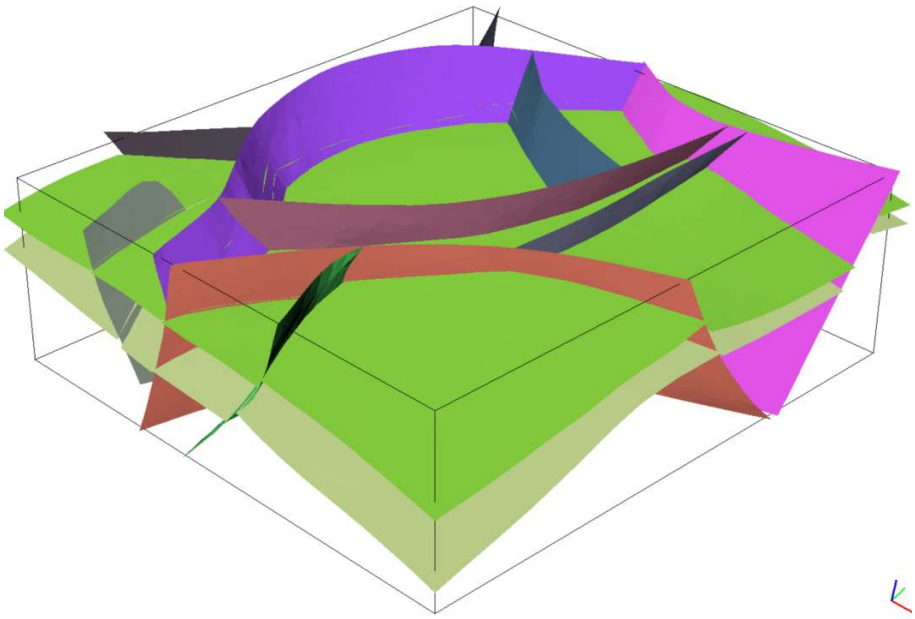

Figure 1: Exemple d'un modèle structural 3D du sous-sol. Le modèle est composé de deux surfaces d'horizon et dix surfaces de faille.

La modélisation structurale est la pratique de construire des modèles géologiques du sous-sol. Un exemple d'un modèle géologique du sous-sol en 3D est illustré dans la Figure 1. Les modèles géologiques servent dans plusieurs applications stratégiques comme dans l'évaluation des risques naturels, ou encore dans l'estimation des ressources naturelles souterraines. Il est donc important que les modéles géologiques représentent la géométrie des structures géologiques souterraines de façon précise. Dans ce manuscrit, nous allons nous intéresser surtout à des domaines stratigraphiques dans lesquels les modèles structuraux sont souvent construits à partir des données interprétées sur des images sismiques; en particulier, nous allons nous intéresser à la *modélisation structurale implicite* (voir par exemple Caumon et al., 2013) telle qu'illustrée dans la Figure 2. Malheureusement, les modèles structuraux construits de la manière décrite précédemment peuvent être sujets à des *incertitudes d'interprétation structurale* (voir par exemple Bond et al., 2007) telles qu'illustrées dans la Figure 3. Dans ce manuscrit, nous allons étudier le problème de reduire ce type d'incertitudes à l'aide des données sismiques. En particulier, nous allons étudier le problème de se servir des données sismiques pour determiner quels modèles sont plus probables que d'autres dans un ensemble des modèles structuraux géologiquement cohérents; ce problème sera connu sous le nom du *problème d'évaluation des modèles structuraux par données sismiques* par la suite.

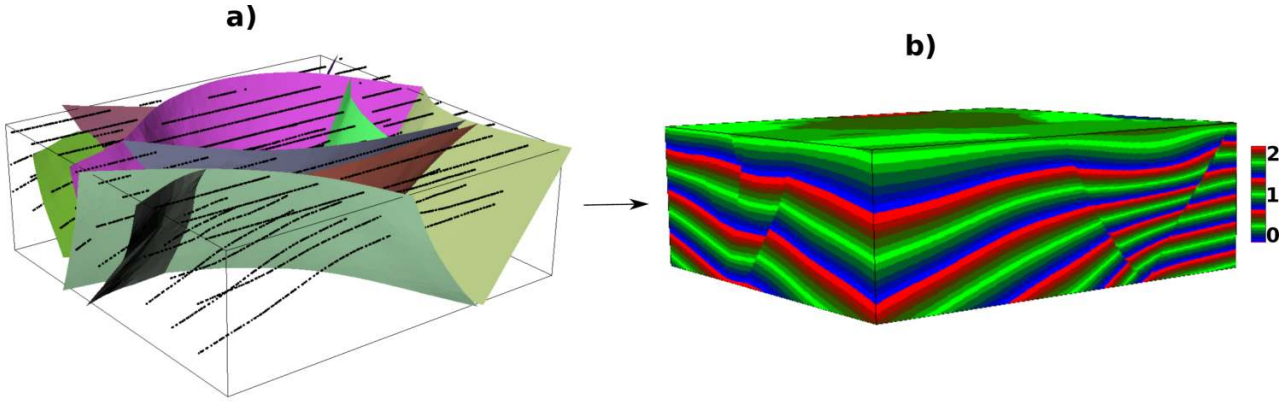

Figure 2: Le concept de la modélisation structurale implicite. **a**) Données initiales (fournies par Total); les lignes noires ont été interprétées à partir d'une image sismique par un processus similaire à celui illustré dans la Figure 3b. **b**) Les données initiales sont ensuite interpolées pour obtenir un champ scalaire qui représente le temps stratigraphique. Le champs scalaire servira enfin pour construire le modèle dans la Figure 1 par extraction des isovaleurs: les surfaces d'horizons sont des isovaleurs du champs scalaire. Voir Caumon et al. (2009), par exemple, pour la construction des surfaces de failles.

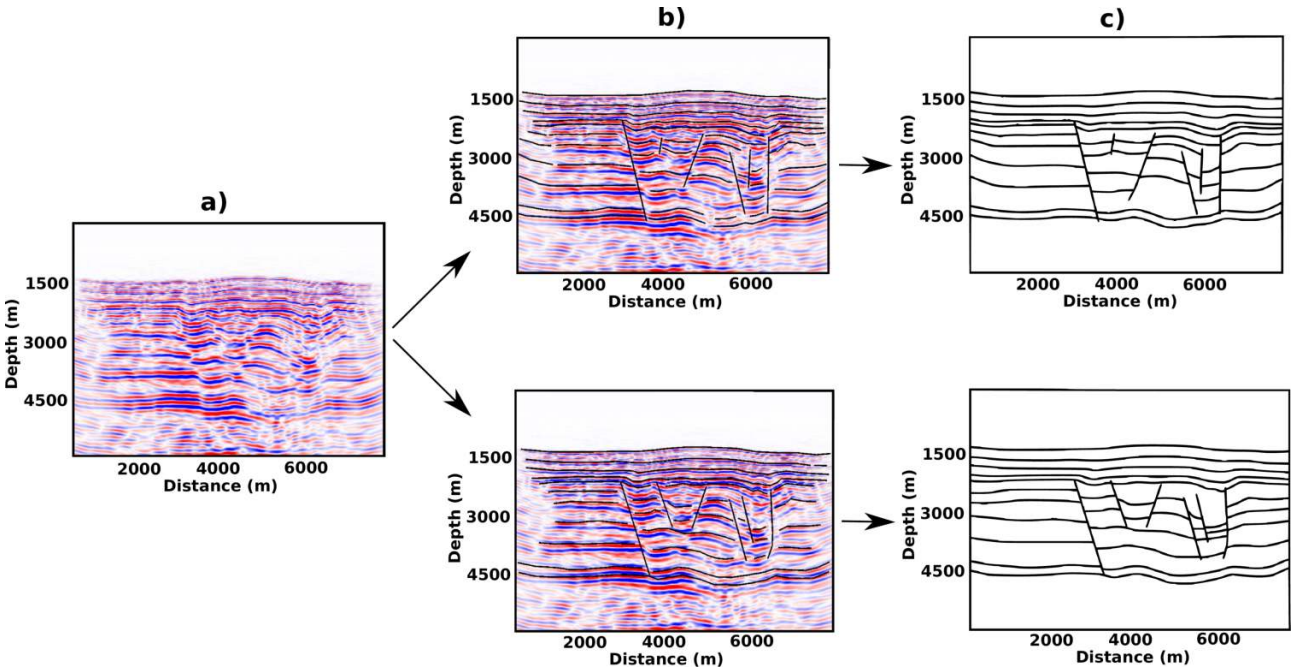

Figure 3: Illustration des incertitudes d'interprétation structurale. **a**) Image sismique 2D du sous-sol. **b**) Deux interprétations structurales par deux géologues différents. **c**) Les modèles géologiques 2D du sous-sol issus des interprétations structurales.

Dans la première partie du manuscrit je propose deux contributions dans le domaine de l'imagerie sismique. Premièrement, dans le chapitre I.2, je propose d'utiliser la "reverse-time migration" (RTM) (voir par exemple Baysal et al., 1983, pour plus d'informations sur la RTM) en tant que préconditionneur pour accélérer l'inversion des formes d'ondes (voir par exemple Virieux and Operto, 2009, pour plus d'informations sur l'inversion des formes d'ondes). L'idée de base repose sur l'observation que l'image RTM donne une bonne estimation de l'illumination sismique dans le sous-sol pour une géométrie d'acquisition donnée. Cette méthode a le mérite d'être simple à implémenter tout en restant numériquement efficace comme l'illustre la Figure 4. Deuxièmement, dans le chapitre I.3, je propose une formule généralisée pour faire du "Kirchhoff extended imaging" (voir par exemple Symes, 2008, pour plus d'informations sur le concept

d'"extended imaging").

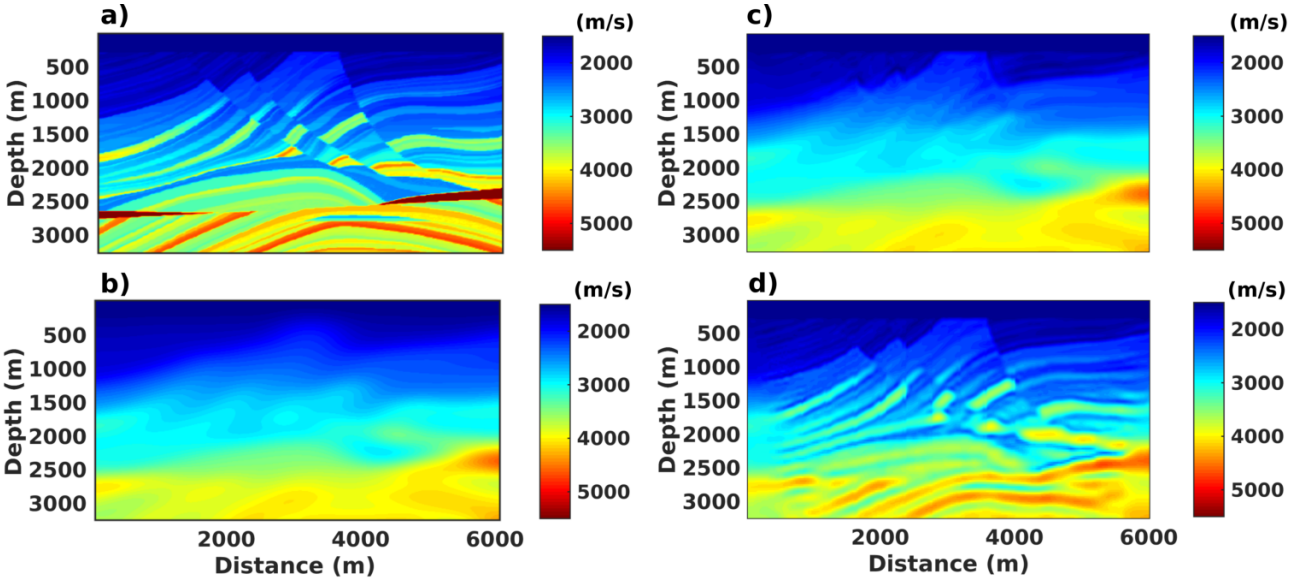

Figure 4: Accélération de l'inversion des formes d'ondes par la RTM. **a**) Modèle de vitesses de reference. **b**) Modèle de vitesses initial. **c**) Modèle de vitesses obtenu après inversion sans préconditionneur. **d**) Modèle de vitesses obtenu après inversion en utilisant la RTM comme préconditionneur.

L'idée de base repose sur l'observation que la formule du "Kirchhoff imaging" (voir par exemple Etgen et al., 2009, pour plus d'informations sur le "Kirchhoff imaging") standard

$$I(\mathbf{x}) = \int_{\mathbf{s},\mathbf{r}} W(\mathbf{s},\mathbf{r},\mathbf{x}) \frac{\partial}{\partial t} D\big[\mathbf{s},\mathbf{r},T(\mathbf{s},\mathbf{r},\mathbf{x})\big] = \int_{\mathbf{s},\mathbf{r}} F(\mathbf{s},\mathbf{r},\mathbf{x}) \tag{1}$$

peut être *étendue* en écrivant

$$I(\mathbf{x},\beta) = \int_{\mathbf{s},\mathbf{r}} \delta\big[\beta - \zeta(\mathbf{s},\mathbf{r},\mathbf{x})\big] F(\mathbf{s},\mathbf{r},\mathbf{x}), \tag{2}$$

où β est le *paramètre d'extension*. Dans l'équation 1, $I(\mathbf{x})$ désigne l'image sismique calculée, $\mathbf{s}$ la position de la source, $\mathbf{r}$ la position du récepteur, $W(\mathbf{s},\mathbf{r},\mathbf{x})$ une fonction de pondération, $T(\mathbf{s},\mathbf{r},\mathbf{x})$ le temps de trajet d'un rayon sismique issu de la source $\mathbf{s}$ puis réfléchi sur le point $\mathbf{x}$ et enfin enregistré au récepteur $\mathbf{r}$, et $D(\mathbf{s},\mathbf{r},t)$ les données sismiques. La fonction $\zeta(\mathbf{s},\mathbf{r},\mathbf{x})$ dans l'équation 2 est un paramètre libre permettant à l'utilsateur de définir la nature du paramètre d'extension β; par exemple, si on définit $\zeta(\mathbf{s},\mathbf{r},\mathbf{x}) = |\mathbf{s}-\mathbf{r}|$, l'image construite par l'équation 2 est par définition une "surface-offset extended image". L'équation 2 a le mérite d'aboutir à une formulation unifiée de la "migration velocity analysis" (MVA) (voir par exemple Woodward et al., 2008, pour plus d'informations sur la MVA), c.-à-d. l'équation aboutit potentiellement à une implémentation unique valable pour plusieurs types de MVA selon la définition de $\zeta(\mathbf{s},\mathbf{r},\mathbf{x})$ (voir la section I.3.6 pour plus de détails).

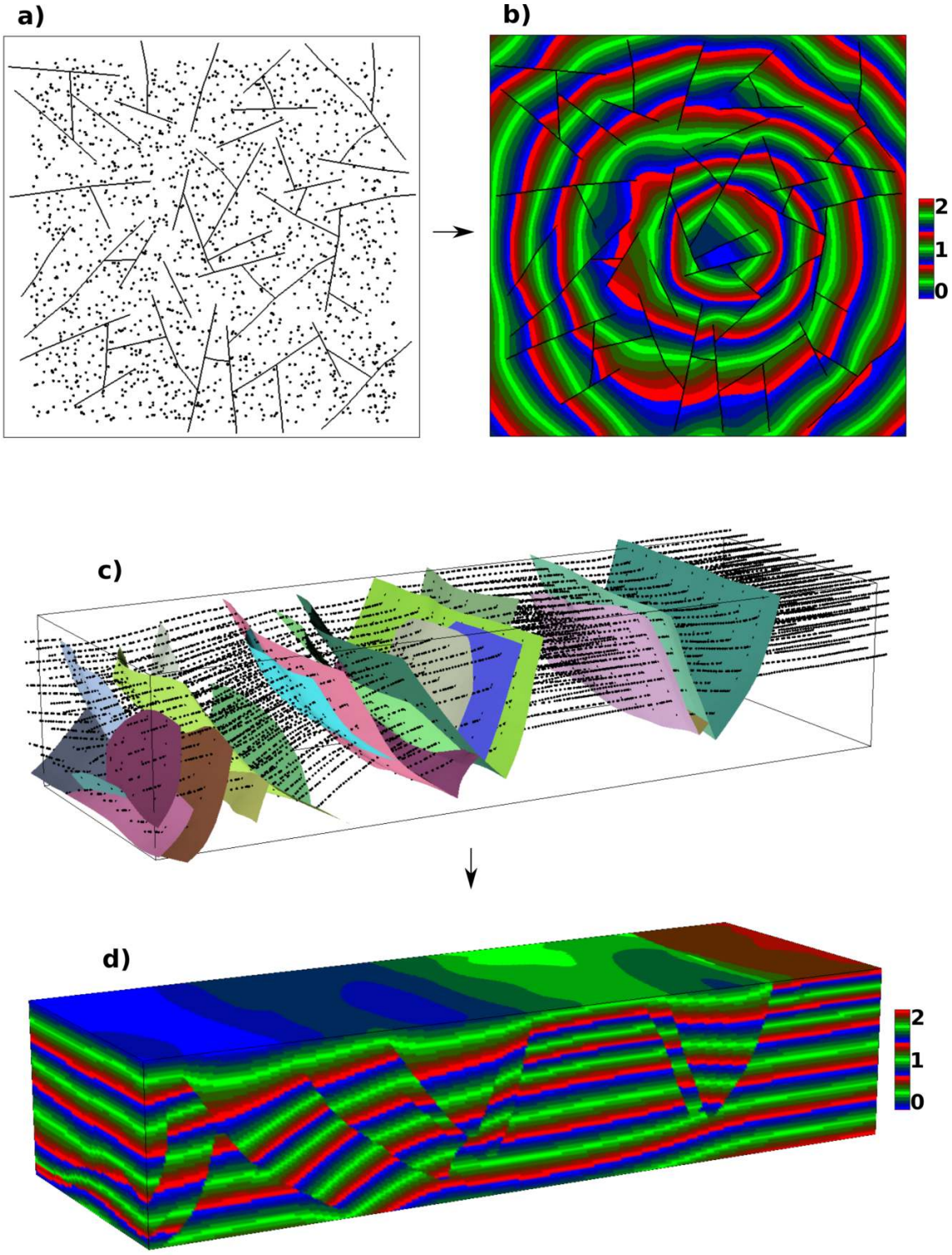

Figure 5: La modélisation structurale implicite par différences-finies. **a**) Données 2D initiales (fournies par ExxonMobil). **b**) Interpolation structurale 2D par différences-finies. **c**) Données 3D initiales (fournies par l'IFPEN). **d**) Interpolation structurale 3D par différences-finies.

Dans la deuxième partie du manuscrit je propose deux contributions dans le domaine de la modélisation structurale implicite. Premièrement, dans le chapitre II.1, je propose une nouvelle méthode pour la modélisation implicite par différences-finies. Cette méthode a le mérite de gérer facilement les réseaux complèxes de failles, comme l'illustre la Figure 5 , car la méthode gère les discontinuités par rastérisation; en plus, la méthode est simple à implémenter et optimiser. Dans ce travail, je propose aussi de nouveaux opérateurs de régularisation, par exemple l'opérateur

$$\mathscr{R}(\phi) = \begin{cases} \partial_x^2 \phi \\ \partial_z^2 \phi \\ \frac{1}{2}(\partial_x^2 + \partial_z^2 - 2\partial_x \partial_z)\phi \\ \frac{1}{2}(\partial_x^2 + \partial_z^2 + 2\partial_x \partial_z)\phi \end{cases}$$

en 2D (voir chapitre II.1 pour la 3D), où $\phi(\mathbf{x})$ est la fonction implicite qui nous intéresse. La particularité de ces opérateurs est qu'ils n'ont pas besoin d'être implémentés sur les noeuds aux bords, une propriété très convenable pour la modélisation structurale implicite où les conditions aux limites sont généralement inconnues. Deuxièmement, dans le chapitre II.2, je propose une autre méthode pour la modélisation implicite par éléments-finis. Cette méthode est basée sur l'implémentation éléments-finis de nouveaux opérateurs proposés dans le chapitre II.1. Je montre que la méthode des éléments finis classique connue pour résoudre les problèmes aux limites doit être légèrement modifiée pour la modélisation implicite où les conditions aux limites sont généralement inconnues.

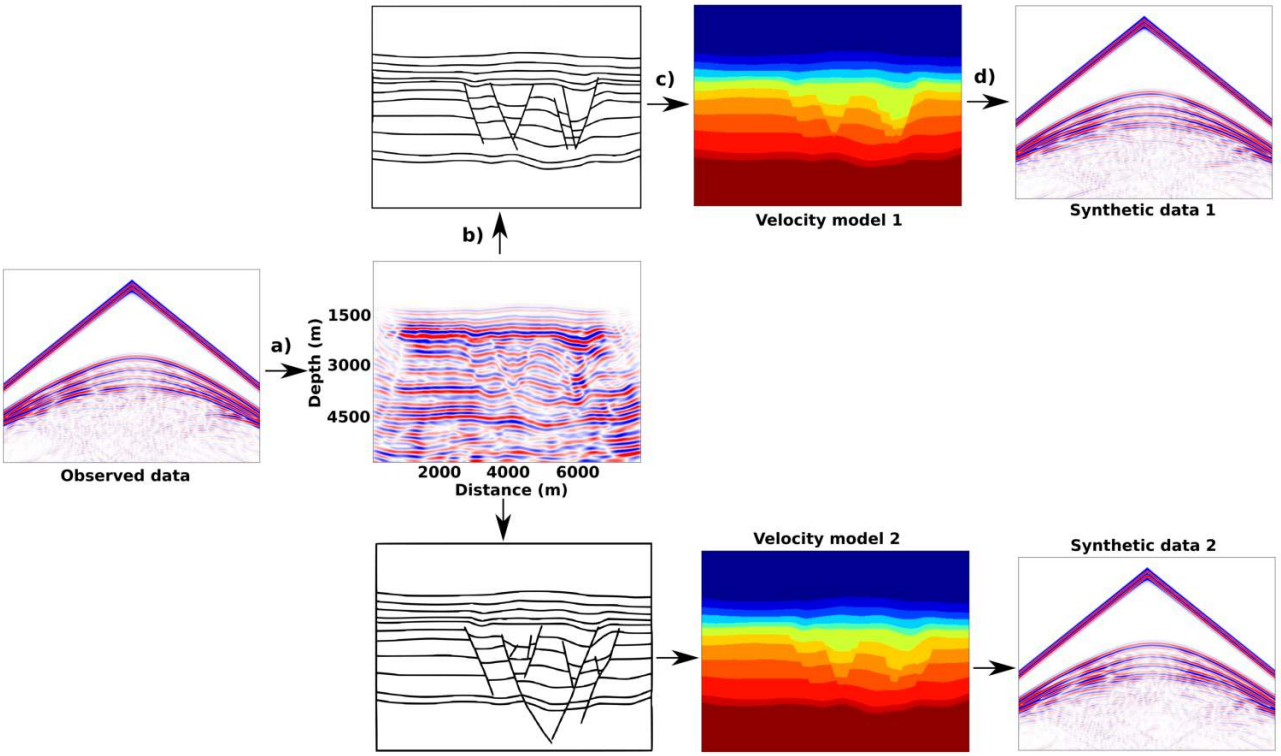

Figure 6: Stratégie globale pour l'évaluation des modèles structuraux par données sismiques. **a**) Données observées utilisées pour l'imagerie sismique. **b**) Modèles structuraux construits pour chaque interprétation structurale. **c**) Modèles de vitesses contruits pour chaque modèle structural. **d**) Données synthétiques calculées pour chaque modèle de vitesses, permettant ainsi de calculer la fonction-objectif pour chaque interprétation structurale.

La troisième partie du manuscrit est consacrée à l'évaluation des modèles structuraux à l'aide des données sismiques. Je propose deux contributions dans ce domaine. Premièrement, dans le chapitre III.2, je développe une théorie quantitative qui formalise le problème d'évaluation des modèles structuraux par données sismiques. La stratégie globale proposée pour résoudre ce problème est illustrée dans la Figure 6: l'idée de base est de générer des données synthétiques pour chaque interprétation structurale, puis comparer ces données aux données observées par calcul d'une fonction-objectif adéquate. Je propose un ensemble de conditions qui doivent être satisfaites par la fonction-objectif pour une évaluation réussie, puis j'argumente qu'une fonction-objectif susceptible de satisfaire ces conditions est de forme

$$\Phi_i(\Omega) = \sum_{\mathbf{x}\in\Omega} \mathbf{E}_i(\mathbf{x}) = \sum_{\mathbf{x}\in\Omega} \mathbf{P}\big[\mathbf{R}(d_i, d_o)\big], \tag{3}$$

où $\mathbf{E}(\mathbf{x})$ désigne la carte d'erreurs d'interprétation, $\mathbf{R}$ un *opérateur de résidus* qui calcule la différence entre les données observées d_o et les données synthétiques d_i calculées dans le i^{eme} modèle structural, $\mathbf{P}$ un *opérateur de projection* qui projette les résidus des données depuis l'espace des données vers l'espace physique (le sous-sol); Ω est la region que nous aimerions évaluer dans le sous-sol. Deuxièmement, dans le chapitre III.3, j'étudie les aspects pratiques du problème et montre que cela reste difficile à proposer une implémentation de l'équation 3 qui satisfait les conditions imposées pour une évaluation réussie des modèles structuraux. Malgré mon échec à démontrer une implémentation pratique qui satisfasse les conditoins requises, je conclus que l'implémentation la plus prometteuse de la l'équation 3 est de forme

$$\begin{cases} a_i(\mathbf{s},\mathbf{r},t) = & \mathbf{R}\big[d_i(\mathbf{s},\mathbf{r},t), d_o(\mathbf{s},\mathbf{r},t)\big] & = d_i(\mathbf{s},\mathbf{r},t) - d_o(\mathbf{s},\mathbf{r},t) \\ \mathbf{E}_i(\mathbf{x}) = & \mathbf{P}\big[\mathbf{R}(d_i, d_o)\big] & = \underset{\Delta m(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},t} \mathbf{R}_2\big[a(\mathbf{s},\mathbf{r},t,\Delta m), a_i(\mathbf{s},\mathbf{r},t)\big] \\ & & = \underset{\Delta m(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},\mathbf{x},t} \delta(\mathbf{x}-\mathbf{r})\mathbf{R}_2\big[a(\mathbf{s},\mathbf{x},t,\Delta m), a_i(\mathbf{s},\mathbf{x},t)\big], \\ & & \text{subject to } \mathbf{L}(m+\Delta m)a(\mathbf{x},t,\Delta m;\mathbf{s}) = \delta(\mathbf{x}-\mathbf{s})w(t) \\ \Phi_i(\Omega) = & \sum_{\mathbf{x}\in\Omega} \big|\mathbf{E}_i(\mathbf{x})\big| \end{cases}.$$

Dans cette équation, $\mathbf{s}$ désigne la position de l'ondelette source $w(t)$, $\mathbf{r}$ la position des récepteurs, m le modèle initial de vitesses lisse, $\mathbf{L}$ l'équation d'onde, et $\mathbf{R}_2$ un opérateur de résidus de type "phase-shift" (voir Ma and Hale, 2013; Warner and Guasch, 2016; Yang, 2015, pour quelques exemples des opérateurs de résidus de type "phase-shift").

Enfin, dans la quatrième partie du manuscrit, je présente brièvement SIGMA, un code développé pendant ma thèse. Tous les résultats présentés dans ce manuscrit sont reproductibles et ont été produits par SIGMA.

Extended Abstract

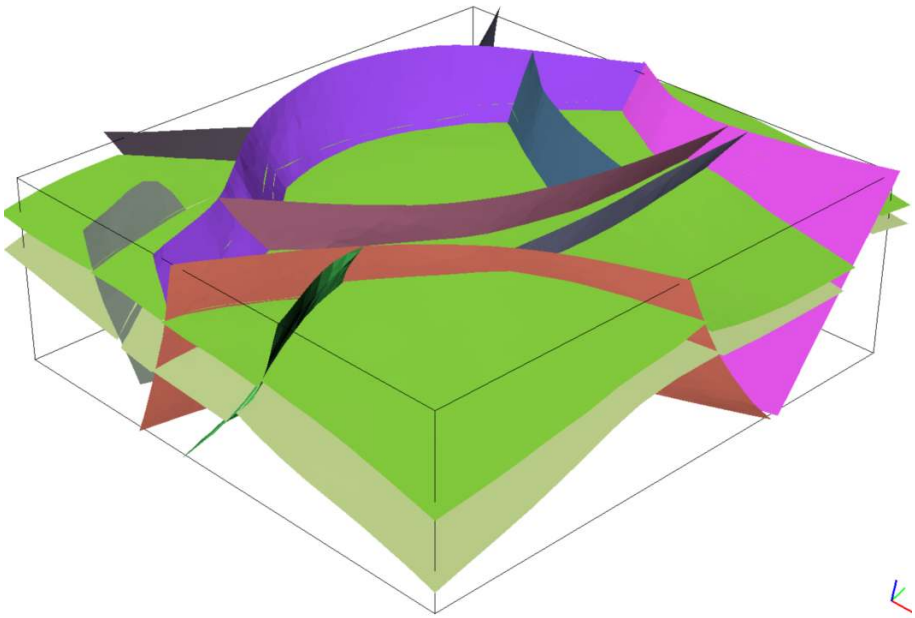

Figure 7: Example of a 3D subsurface structural model. The model is made of two horizon surfaces and ten fault surfaces.

Structural modeling is the practice of building subsurface geological models. An example of a 3D subsurface geological model is illustrated in Figure 7. Geological models are useful in a number of strategic applications such as the evaluation of natural risks and the estimation of subsurface natural resources. It is therefore important that geological models represent the geometry of subsurface geological structures accurately. In this manuscript, we will focus mainly on stratigraphic domains where structural models are often built from data interpreted from seismic images; in particular, we are going to focus on *structural implicit modeling* (see for example Caumon et al., 2013) as illustrated in Figure 8. Unfortunately, structural models built as just described can be subjected to *structural interpretation uncertainties* (see for example Bond et al., 2007) as illustrated in Figure 9. In this manuscript, we are going to study the problem of reducing these types of uncertainties using seismic data. In particular, we are going to study the problem of using seismic data to determine which models are more likely than others from a set of geologically consistent structural models; this problem will be referred to as *the problem of appraising structural models using seismic data* in what follows.

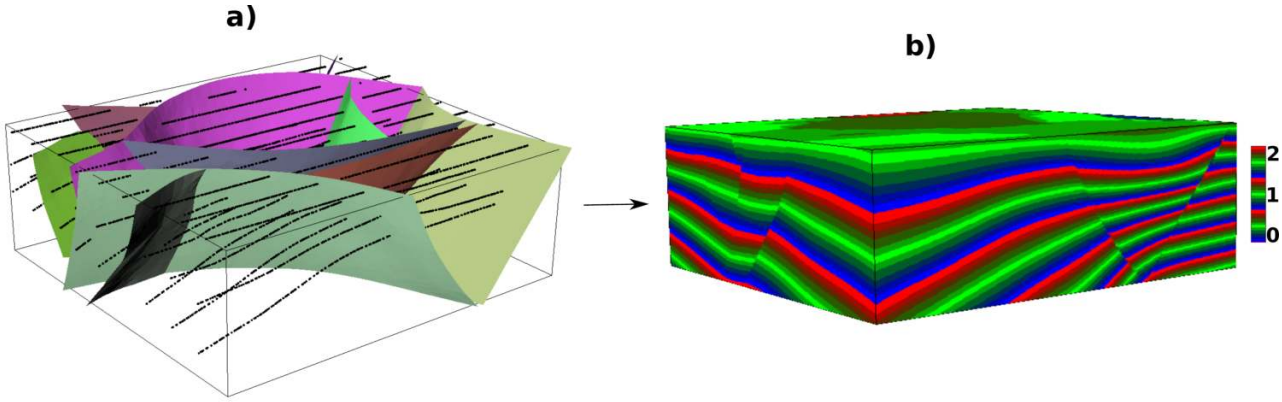

Figure 8: The concept of structural implicit modeling. **a**) Input data (courtesy of Total); the black lines have been interpreted from a seismic image by a process similar to that illustrated in Figure 9. **b**) The input data are then interpolated to obtain a scalar field which represents the stratigraphic time. The scalar field is then finally used to build the model in Figure 7 by extracting isovalues: horizon surfaces are isovalues of the scalar field. See Caumon et al. (2009), for example, on how to build fault surfaces.

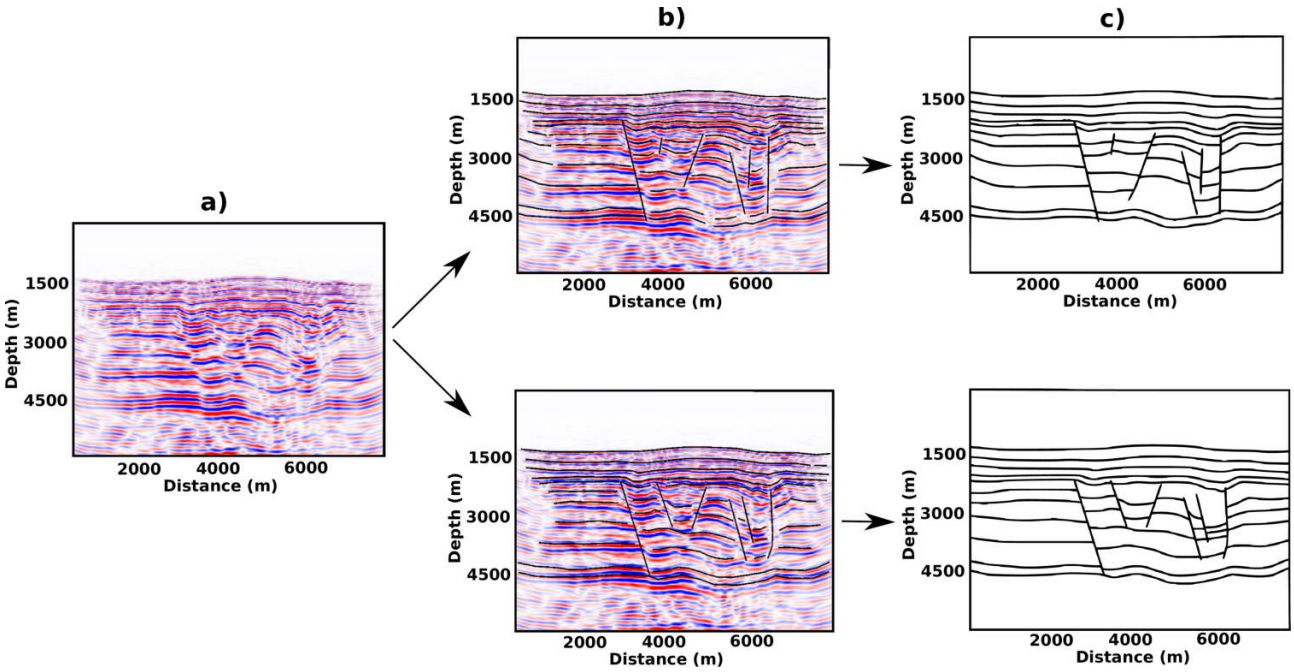

Figure 9: Illustration of structural interpretation uncertainties. **a**) A 2D seismic image of the subsurface. **b**) Two structural interpretations from two different geologists. **c**) Subsurface geological models resulting from the structural interpretations.

In the first Part of the manuscript I propose two contributions in the field of seismic imaging. Firstly, in chapter I.2, I propose to use reverse time migration (RTM) (see for example Baysal et al., 1983, for more information on RTM) as a preconditioner to speed up waveform inversion (see for example Virieux and Operto, 2009, for more information on waveform inversion). The underlying idea is based on the observation that the RTM image gives a good estimation of the seismic illumination in the subsurface for a given acquisition geometry. This method has the advantage of being simple to implement while still remaining numerically efficient as illustrated in Figure 10. Secondly, in chapter I.3, I propose a generalized formula for Kirchhoff extending imaging (see for example Symes, 2008, for more information on the concept of extended imaging).

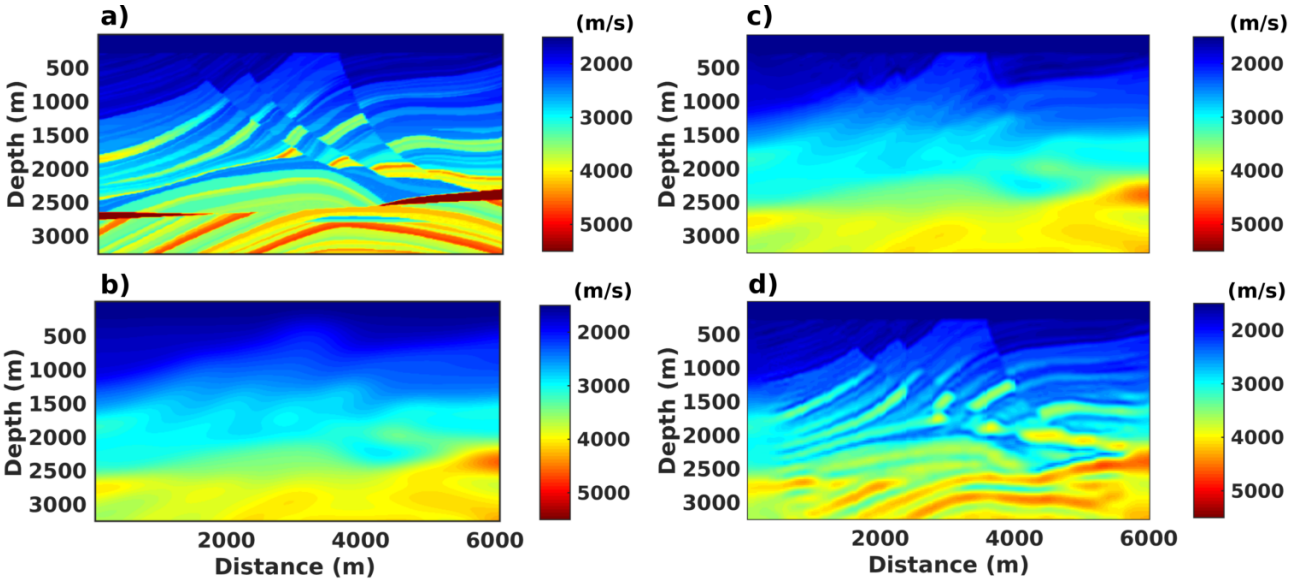

Figure 10: Speeding up waveform inversion with RTM. **a**) Reference velocity model. **b**) Initial velocity model. **c**) Velocity model obtained after inversion without preconditioning. **d**) Velocity model obtained after inversion using RTM as a precontioner.

The underlying idea is based on the observation that the standard Kirchhoff imaging formula (see for example Etgen et al., 2009, for more information on Kirchhoff imaging)

$$I(\mathbf{x}) = \int_{\mathbf{s},\mathbf{r}} W(\mathbf{s},\mathbf{r},\mathbf{x}) \frac{\partial}{\partial t} D\big[\mathbf{s},\mathbf{r},T(\mathbf{s},\mathbf{r},\mathbf{x})\big] = \int_{\mathbf{s},\mathbf{r}} F(\mathbf{s},\mathbf{r},\mathbf{x}) \tag{4}$$

can be *extended* by writing

$$I(\mathbf{x},\beta) = \int_{\mathbf{s},\mathbf{r}} \delta\big[\beta - \zeta(\mathbf{s},\mathbf{r},\mathbf{x})\big] F(\mathbf{s},\mathbf{r},\mathbf{x}), \tag{5}$$

where β is the *extension parameter*. In Equation 4, $I(\mathbf{x})$ designates the output seismic image, $\mathbf{s}$ the source location , $\mathbf{r}$ the receiver location, $W(\mathbf{s},\mathbf{r},\mathbf{x})$ a weighting function , $T(\mathbf{s},\mathbf{r},\mathbf{x})$ the travel time of a seismic ray from the source $\mathbf{s}$ then reflected off the point $\mathbf{x}$ and finally recorded at the receiver $\mathbf{r}$, and $D(\mathbf{s},\mathbf{r},t)$ the input seismic data. The function $\zeta(\mathbf{s},\mathbf{r},\mathbf{x})$ in Equation 5 is a free parameter that allows the user to define the nature of the extension parameter β; for example, if one defines $\zeta(\mathbf{s},\mathbf{r},\mathbf{x}) = |\mathbf{s}-\mathbf{r}|$, the image volume output by Equation 5 is by definition a surface-offset extended image. Equation 5 has the advantage of leading to a unified formulation of gradient-based migration velocity analysis (MVA) (see for example Woodward et al., 2008, for more information on MVA), i.e. the equation potentially leads to a unique implementation that is valid for multiple types of MVA depending on the definition of $\zeta(\mathbf{s},\mathbf{r},\mathbf{x})$ (see section I.3.6 for more details).

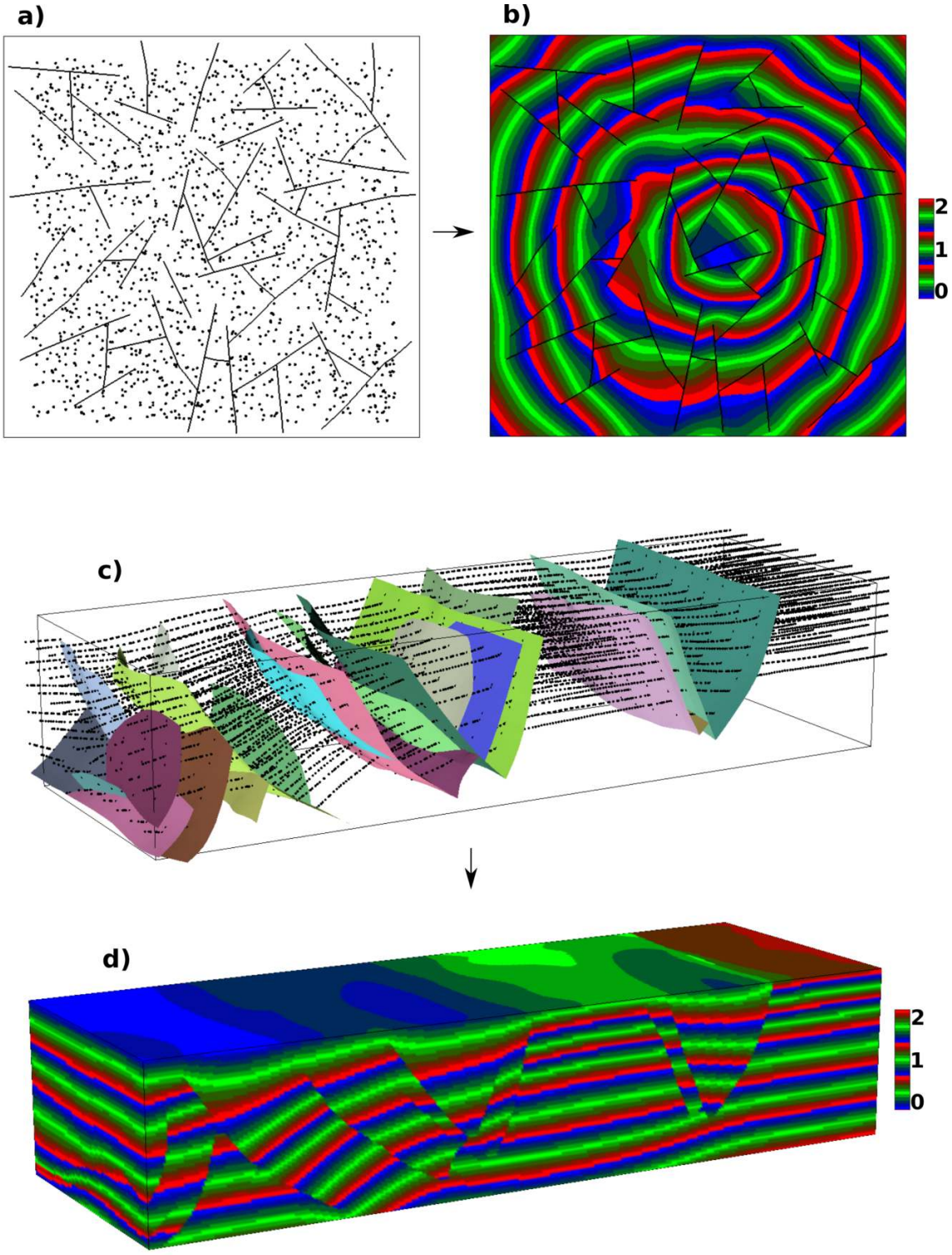

Figure 11: Finite difference structural implicit modeling. **a**) 2D input data (courtesy of ExxonMobil). **b**) 2D finite difference structural interpolation. **c**) 3D input data (courtesy of IFPEN). **d**) 3D finite difference structural interpolation.

In the second Part of the manuscript I propose two contributions in the field of structural implicit modeling. Firstly, in chapter II.1, I propose a new method for implicit modeling based on finite differences. This method has the advantage of readily handling complex fault networks, as illustrated in Figure 11, since the method handles discontinuities by rasterization; furthermore, the method is easy to implement and optimize. In this work, I also propose new regularization operators, for example the operator

$$\mathscr{R}(\phi) = \begin{cases} \partial_x^2 \phi \\ \partial_z^2 \phi \\ \frac{1}{2}(\partial_x^2 + \partial_z^2 - 2\partial_x \partial_z)\phi \\ \frac{1}{2}(\partial_x^2 + \partial_z^2 + 2\partial_x \partial_z)\phi \end{cases}$$

in 2D (see chapter II.1 for 3D), where $\phi(\mathbf{x})$ is the implicit function of interest. The particularity of these operators is that they do not need to be implemented on boundary nodes, a property which is very appealing in implicit modeling where boundary conditions are usually unknown. Secondly, in chapter II.2, I propose another method for implicit modeling based on finite elements. This method is merely a finite element implementation of the new regularization operators proposed in chapter II.1. I show that the conventional finite element familiar for solving boundary value problems has to be slightly modified for implicit modeling where boundary conditions are usually unknown.

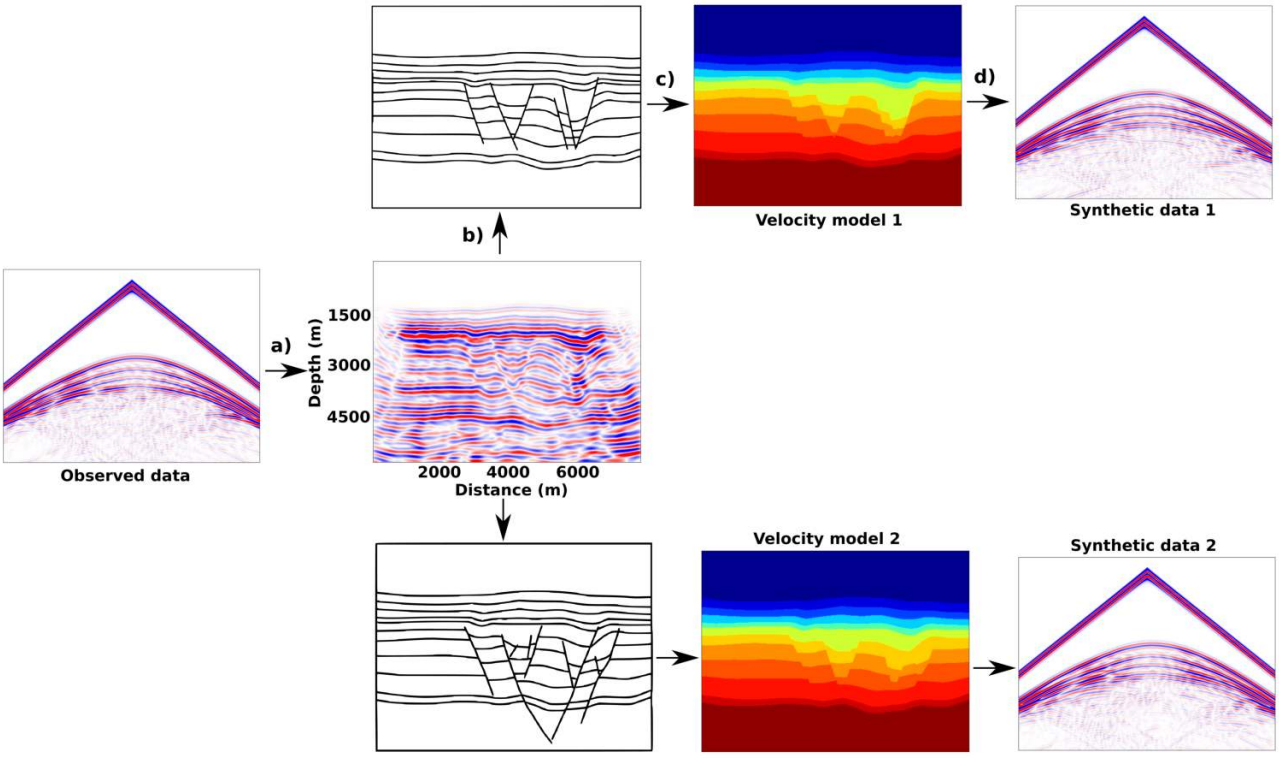

Figure 12: Global workflow for appraising structural models using seismic data. **a**) Observed data is used for seismic imaging. **b**) Structural models are built for each interpretation of the seismic image. **c**) Velocity models are built for each structural model. **d**) Synthetic data are generated for each velocity model, allowing to compute a data misfit value for each interpretation.

The third Part of the manuscript is devoted to appraising structural models/interpretations using seismic data. I propose two contributions in this field. Firstly, in chapter III.2, I develop a quantitative theory that formalizes the problem of appraising structural models using seismic data. The global workflow proposed to tackle this problem is illustrated in Figure 12: the main idea is to generate synthetic data for each structural interpretation, and then to compare these data to observed data by computing an appropriate data misfit function. I derive a set of conditions that have to be satisfied by the data misfit function for a successful appraisal of structural models, then I argue that a misfit function that is likely to satisfy those conditions has the form

$$\Phi_i(\Omega) = \sum_{\mathbf{x}\in\Omega} \mathbf{E}_i(\mathbf{x}) = \sum_{\mathbf{x}\in\Omega} \mathbf{P}\big[\mathbf{R}(d_i, d_o)\big], \tag{6}$$

where $\mathbf{E}(\mathbf{x})$ designates the interpretation error map, $\mathbf{R}$ a *residual operator* that computes the difference between observed d_o and the synthetic data d_i computed in the i^{th} structural model, $\mathbf{P}$ a *projection operator* that projects data residuals from the data-space into the image-domain; Ω is the region we would like to evaluate in the subsurface. Secondly, in chapter III.3, I study the practical aspects of the problem and show that it remains a challenge to propose an implementation of Equation 6 that satisfies the conditions imposed for a successful appraisal of structural models. Although I fail to demonstrate a practical implementation that satisfies the required conditions, I conclude that the most promising implementation has the form

$$\begin{cases} a_i(\mathbf{s},\mathbf{r},t) = \mathbf{R}\big[d_i(\mathbf{s},\mathbf{r},t), d_o(\mathbf{s},\mathbf{r},t)\big] & = d_i(\mathbf{s},\mathbf{r},t) - d_o(\mathbf{s},\mathbf{r},t) \\ \mathbf{E}_i(\mathbf{x}) = \mathbf{P}\big[\mathbf{R}(d_i, d_o)\big] & = \underset{\Delta m(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},t} \mathbf{R}_2\big[a(\mathbf{s},\mathbf{r},t,\Delta m), a_i(\mathbf{s},\mathbf{r},t)\big] \\ & = \underset{\Delta m(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},\mathbf{x},t} \delta(\mathbf{x}-\mathbf{r})\mathbf{R}_2\big[a(\mathbf{s},\mathbf{x},t,\Delta m), a_i(\mathbf{s},\mathbf{x},t)\big], \\ & \text{subject to } \mathbf{L}(m+\Delta m)a(\mathbf{x},t,\Delta m;\mathbf{s}) = \delta(\mathbf{x}-\mathbf{s})w(t) \\ \Phi_i(\Omega) = \sum_{\mathbf{x}\in\Omega} \mathbf{E}_i(\mathbf{x}) & \end{cases}.$$

In this equation, $\mathbf{s}$ designates the location of the source wavelet $w(t)$, $\mathbf{r}$ the location of receivers, m the initial smooth imaging velocity model, $\mathbf{L}$ the wave equation, and $\mathbf{R}_2$ a phase-shit residual operator (see Ma and Hale, 2013; Warner and Guasch, 2016; Yang, 2015, for some examples of phase-shift residual operators).

Finally, in the fourth Part of the manuscript, I briefly present SIGMA, a software developed during my PhD. All the results presented in this manuscript are reproducible and were obtained using SIGMA.

Bibliography

E. Baysal, D. Kosloff, and J. Sherwood. Reverse time migration. *GEOPHYSICS*, 48:1514–1524, 1983.

C. Bond, Z. Shipton, and S. Jones. What do you think this is? Conceptual uncertainty in geoscience interpretation. *GSA Today*, 17:4–10, 2007.

G. Caumon, P. Collon, C. L. C. de Veslud, S. Viseur, and J. Sausse. Surface-Based 3D Modeling of Geological Structures. *Mathematical Geosciences*, 41:927–945, 2009.

G. Caumon, G. Gray, C. Antoine, and M.-O. Titeux. Three-dimensional implicit stratigraphic model building from remote sensing data on tetrahedral meshes: theory and application to regional model of La Popa Basin, NE Mexico. *IEEE Transactions on Geoscience and Remote Sensing*, 51(3), 2013.

J. Etgen, S. Gray, and Y. Zhang. An overview of depth migration in exploration geophysics. *GEOPHYSICS*, 74(6), 2009.

Y. Ma and D. Hale. Wave-equation reflection traveltime inversion with dynamic warping and full-waveform inversion. *GEOPHYSICS*, 78:R223–R233, 2013.

W. Symes. Migration velocity analysis and waveform inversion. *Geophysical Prospecting*, 56: 765–790, 2008.

J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *GEOPHYSICS*, 74(6), 2009.

M. Warner and L. Guasch. Adaptive waveform inversion: Theory. *Geophysics*, 81:R429–R445, 2016.

M. Woodward, D. Nichols, O. Zdraveva, P. Whitfield, and T. Johns. A decade of tomography. *GEOPHYSICS*, 73(5), 2008.

H. Yang. Recent advances in imaging crustal fault zones: a review. *Earthquake Science*, 28: 151–162, 2015.

General Introduction

Introduction

Structural modeling is the practice of building subsurface structural models. A structural model is a set geologically consistent structural surfaces such as faults and horizons (e.g. Caumon et al., 2009); an example of a structural model is shown in Figure 13. Structural modeling has been around at least since the early nineties and it is now a well established field. The reader interested in learning more on structural modeling is referred to Mallet (2002); Caumon et al. (2004); Chilès et al. (2004); Calcagno et al. (2008); Caumon et al. (2009, 2013); Jackson et al. (2013); Mallet (2014); Wellmann and Caumon (2018).

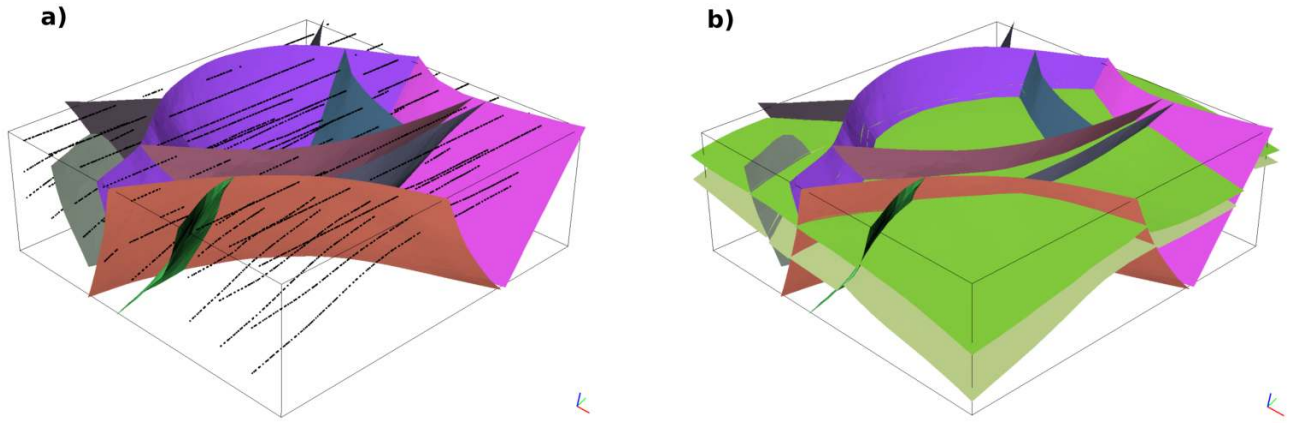

Figure 13: Example of a 3D structural model (data courtesy of Total). **a**) Input data consisting of 10 fault surfaces and horizon points picked along 3 horizons. **b**) Structural model built from the input data.

Subsurface structural models are routinely used for resource estimation, numerical simulations, and risk management; it is therefore important that structural models represent the geometry of subsurface geological objects accurately. In this manuscript, we will only discuss structural models that are built from input data interpreted from seismic images, as shown in Figure 14. Seismic images are obtained by seismic imaging. Seismic imaging is the practice of using of elastic waves propagated in the subsurface to image geological structures of the subsurface. A typical seismic imaging workflow is illustrated in Figure 15. Seismic waves are first propagated in the subsurface (e.g. Figure 15a) and then recorded at the surface of the Earth as seismic data (e.g. Figure 15b) after scattering off heterogeneities in the subsurface. Recorded data are then used to estimate a *migration velocity model* (e.g. Figure 15c) which is used to *migrate* the observed data to obtain a seismic image of the subsurface (e.g. Figure 15d). The reader interested in learning more on seismic imaging is referred to Baysal et al. (1983); Claerbout (1985); Miller et al. (1987); Bleistein et al. (2001); Xu et al. (2001); Sava and Fomel (2003); Biondi (2006); Plessix (2006); Etgen et al. (2009).

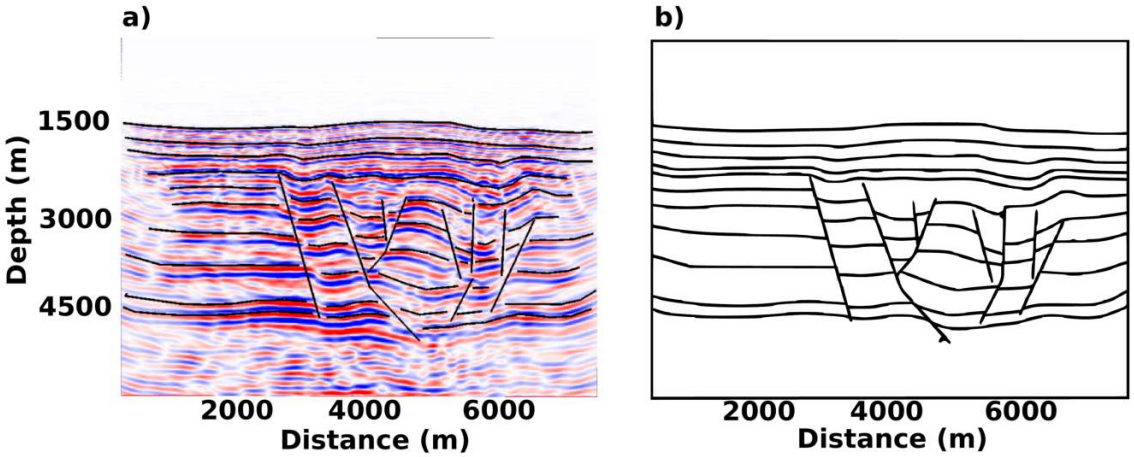

Figure 14: Structural modeling from seismic interpretation. **a**) Input data picked on the seismic image. **b**) A 2D structural model built from the input data.

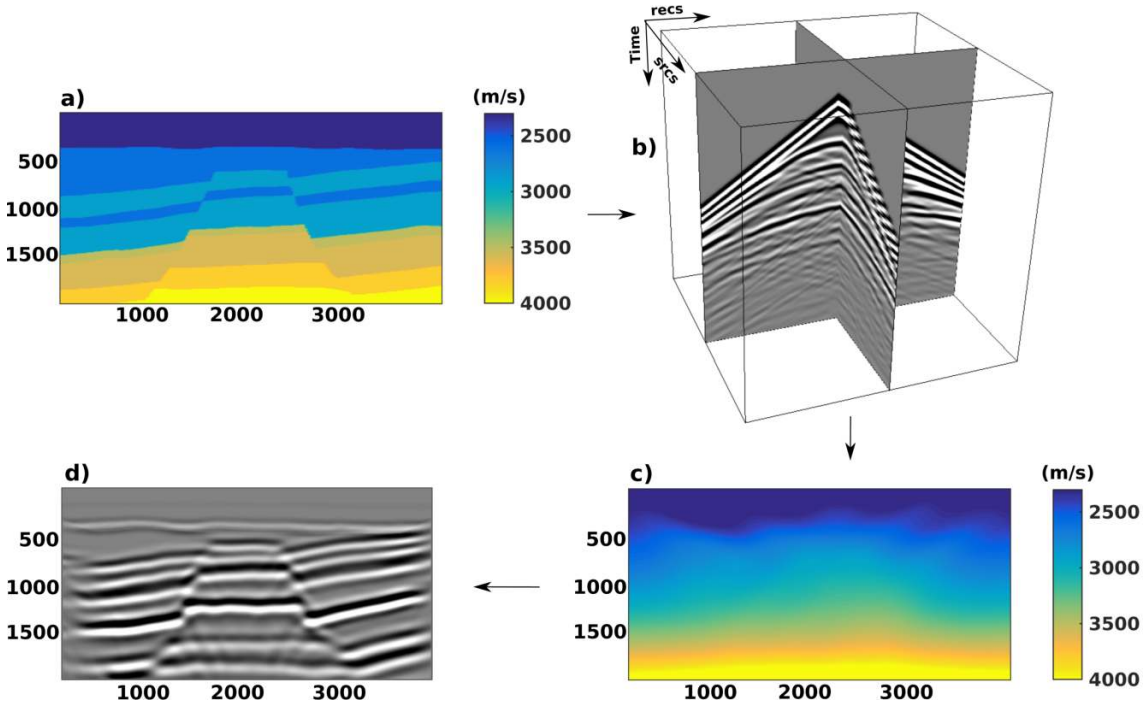

Figure 15: Typical seismic imaging workflow. **a**) Reference velocity model, unknown in practice. **b**) "Observed" data generated from the reference model. **c**) Imaging (migration) velocity model estimated from the observed data. **d**) Seismic image of the reference model obtained by migrating the observed data using the imaging velocity model.

Structural models built from seismic interpretations may be subjected to (structural) uncertainties inherited from seismic imaging. We speak of structural uncertainties when the position and/or the presence of structural surfaces is uncertain in the subsurface (e.g. Thore et al., 2002; Wellmann et al., 2010). We can distinguish two types of structural uncertainties inherited from seismic imaging

1. Structural uncertainties due to the non-uniqueness of the imaging velocity. The problem of estimating the imaging velocity from seismic data is ill-posed. Its solution usually involves some user-dependent parameters which can be tuned to give different equally acceptable imaging velocity models (e.g. Woodward et al., 2008; Messud et al., 2017; Beraud et al., 2018). These velocity models can lead to different structural models. For example, consider the velocity modeling experiment of Beraud et al. (2018) in Figure 16 carried out using the data set in Figure 15b. Figure 16a shows the initial velocity model used for velocity modeling. Figures 16b, 16c, and 16d show the final velocity model obtained by ray-based migration velocity analysis (e.g. Stork, 1992; Woodward et al., 2008) with a "low" regularization weight, a "moderate" regularization weight, and a "high" regularization weight, respectively. These 3 velocity models are considered "accurate" enough for imaging because they all flatten common-image-gathers as shown in Figures 17b, 17c, and 17d. However, the 3 velocity models lead to different structural models as illustrated in Figure 18.

2. Interpretation uncertainties due to a limited resolution of seismic images. In complex geological settings where the seismic image has a limited resolution, it is usually possible to propose multiple structural interpretations of the same seismic image (e.g. Bond et al., 2007; Bond, 2015). For example, Figure 19 shows an alternative structural interpretation of the seismic image in Figure 14a by a different interpreter.

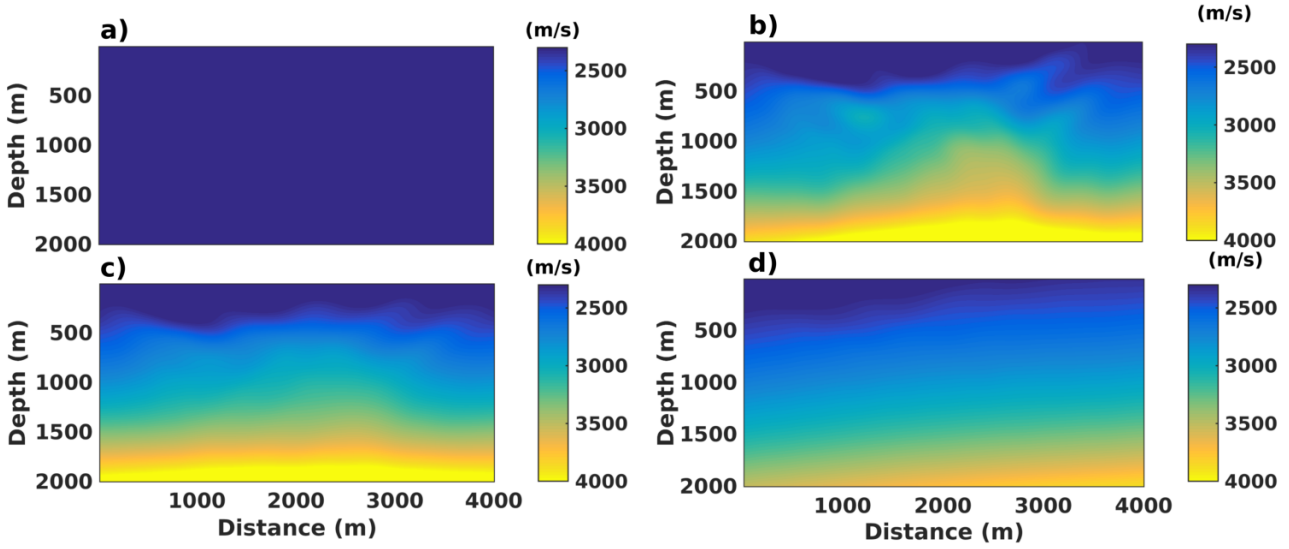

Figure 16: Velocity modeling by ray-based migration velocity analysis. **a**) Initial velocity model. **b**) Final velocity model obtained using a low regularization weight. **c**) Final velocity model obtained using a moderate regularization weight. **d**) Final velocity model obtained using a high regularization weight.

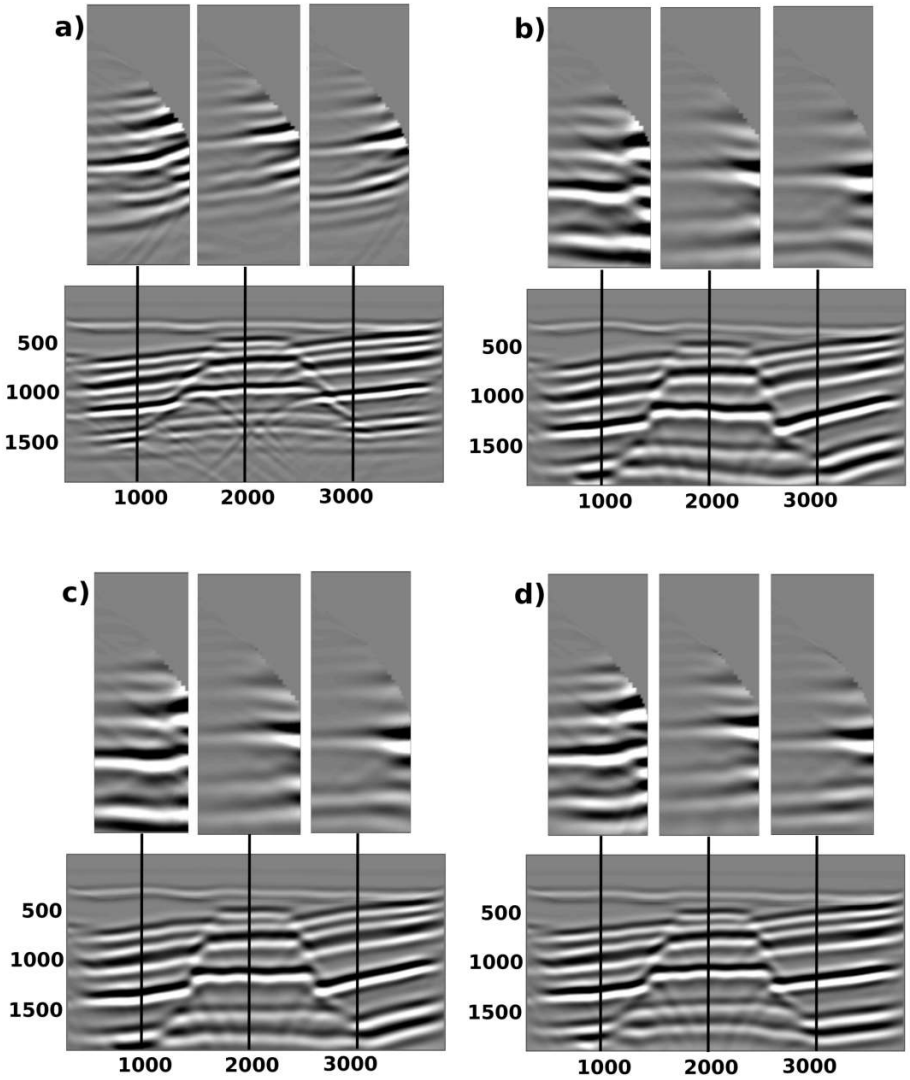

Figure 17: **a**) Seismic image and common-image-gathers obtained using the initial velocity model in Figure 16a. **b**) Seismic image and common-image-gathers obtained using the velocity model in Figure 16b. **c**) Seismic image and common-image-gathers obtained using the velocity model in Figure 16c. **d**) Seismic image and common-image-gathers obtained using the velocity model in Figure 16d.

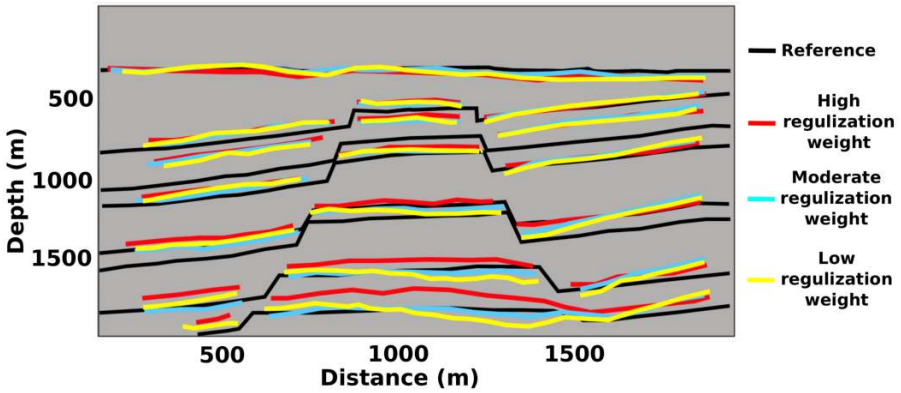

Figure 18: Structural uncertainties due to the non-uniqueness of the imaging velocity model. **Red**) Horizon picking on the seismic image in Figure 17d. **Blue**) Horizon picking on the seismic image in Figure 17c. **Yellow**) Horizon picking on the seismic image in Figure 17b.

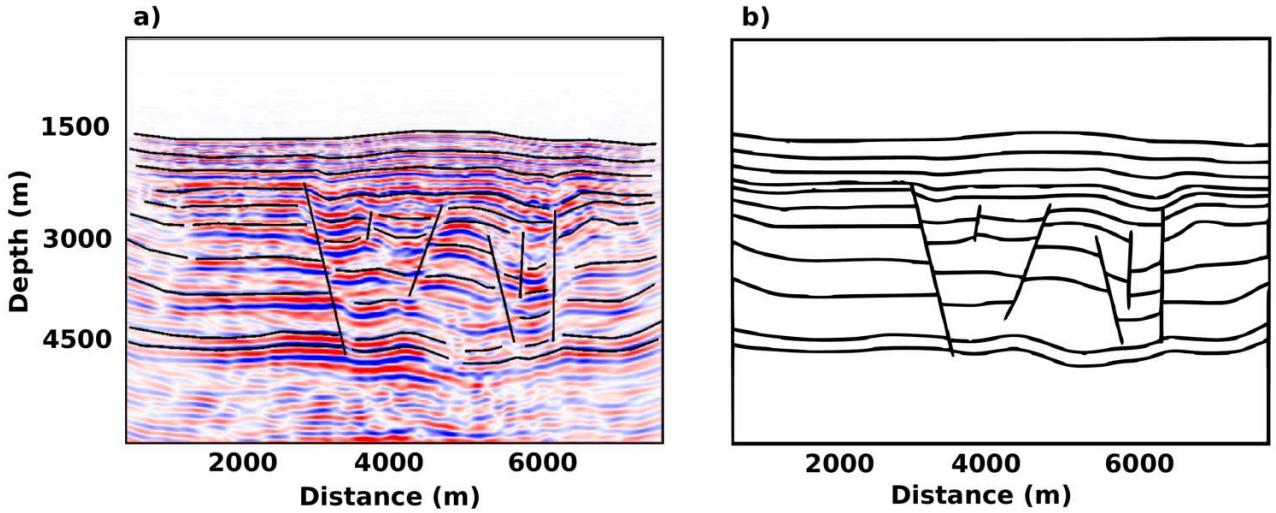

Figure 19: A different structural interpretation of the seismic image in Figure 14a. **a**) Input data picked on the seismic image. **b**) A 2D structural model built from the input data.

In this manuscript, we will assume that we have a unique imaging velocity model so that we can only focus on interpretation uncertainties. In particular, given a set of geologically consistent structural models from a single seismic image, we will study the problem of determining which structural models are more likely than others using seismic data. For example, can we use seismic data to determine which structural model is more accurate between Figure 14b and Figure 19b? This problem will be referred to as *appraising structural models/interpretations using seismic data* hereafter. Our strategy for appraising structural interpretations using seismic data is illustrated in Figure 20. First, we assume that the seismic data used for seismic imaging are available. The seismic image is given to different interpreters who will propose different structural interpretations. The structural interpretations are then used to create subsurface structural models. The structural models are in turn used to create velocity models. Finally, the velocity models are used to generate synthetic data for each structural interpretation. The synthetic data are then compared with observed data in order to determine which interpretations are more likely. The main challenge of the proposed workflow is to find an appropriate way to compare observed data with synthetic data; that is, to compute appropriate data misfit values. This problem will be investigated in detail in chapters III.2 and III.3, where we show that appropriate data misfit functions should be localized in space (i.e. data misfit function as a function of space). The localization is typically achieved by projecting data residuals from the data-space into the image-domain; the projection operators are borrowed from the field of seismic imaging.

Appraising structural models using the workflow proposed above is therefore an integrative problem which involves seismic imaging and structural modeling. It is therefore reasonable to give a short review of seismic imaging and structural modeling before diving into the problem of appraising structural models using seismic data. As we were reviewing the seismic imaging and structural modeling literature, it occurred to us that we could propose alternative solutions, and sometimes simpler ones, to existing problems. Consequently, instead of giving a traditional literature review, we will present alternative solutions to some existing problems in seismic imaging and in structural modeling. As a result, this manuscript has 3 main parts:

1. Part I: Notes on Seismic Imaging.
2. Part II: Notes on Structural Modeling.
3. Part III: Reducing Structural Uncertainties using Seismic Data.

Part I presents seismic imaging methods that are considered to be useful for Part III; it's therefore recommended to read Part I before reading Part III. Part II, on the other hand, is independent and can safely be skipped as it is not essential to Part III.

There is a fourth Part, the last Part of the manuscript, where I present SIGMA, a software that my students (students I supervised for a master's project) and I developed over the course of my thesis. All the results presented in this thesis are reproducible and were obtained using SIGMA.

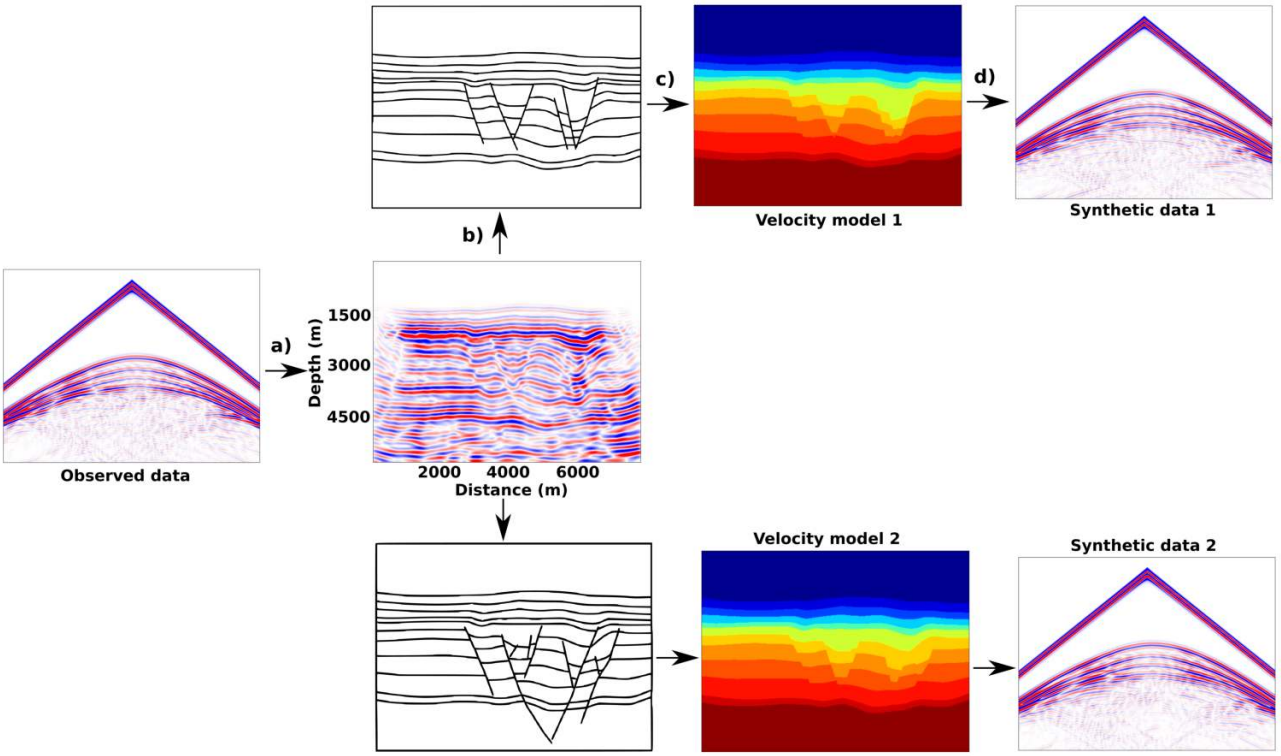

Figure 20: Workflow for appraising structural models using seismic data. **a**) Observed data is used for seismic imaging. **b**) Structural models are built for each interpretation of the seismic image. **c**) Velocity models are built for each structural model. **d**) Synthetic data are generated for each velocity model, allowing to compute a data misfit value for each interpretation.

Three optimization problems

Parts I, II, and III of the manuscript begin with an introductory text which presents seismic imaging, implicit structural modeling, and appraising structural interpretations using seismic data as optimization problems:

- The optimization approach to seismic imaging is a more modern and general (i.e. abstract) way of presenting seismic imaging.
- Presenting the implicit structural modeling problem as an optimization problem has the advantage of highlighting the challenges of interpolation for structural modeling compared to the standard scattered data interpolation problem (e.g. Anjyo et al., 2014); in particular, interpolation for structural modeling has additional optimization constraints.
- The problem of appraising structural interpretations using seismic data can also be seen from an optimization point of view; the advantage of doing so is to highlight the challenges of the problem, which appear as optimization constraints.

Contributions

Seismic imaging

In chapter I.2, I propose to use reverse time migration to precondition waveform inversion. I start the chapter with a short review of similar existing preconditioning techniques in order to highlight the difference with the proposed technique.

In chapter I.3, we propose a generalized Kirchhoff extended imaging formula. While Kirchhoff extended imaging is not new, our generalized extension formula allowed us to derive a generalized differential semblance ray-based migration velocity analysis formula. The generalized extension formula also has a practical advantage: it allows a unified implementation (i.e. the same code) of multiple extended imaging operators (shot extension, offset extension, angle extension, ...) depending on input parameters.

Structural modeling

In chapter II.1, we propose an implicit structural modeling (i.e. structural interpolation) method based on finite differences. This technique has the advantage of being simple to implement and being very robust in handling discontinuities. We also derive new regularization operators in this chapter. This chapter resulted in the following publications:

- [Journal paper]: Irakarama M., Laurent G., Renaudeau J., Caumon G. (2019) Finite difference implicit structural modeling of geological structures. In prep.
- [Conference paper]: Irakarama M., Laurent G., Renaudeau J., Caumon G. (2018) Finite difference implicit modeling of geological structures. In:. **80th EAGE Conference & Exhibition**, Copenhagen, Denmark.

In chapter II.2, we propose an implicit structural modeling (i.e. structural interpolation) method based on a finite element implementation of the new regularization operators proposed in chapter II.1.

Appraising structural models using seismic data

In chapter III.2, we introduce and formalize the problem of appraising structural interpretations/models using seismic data. This chapter mainly focuses on theoretical aspects of the problem and resulted in the following publications:

- [Journal paper]: Irakarama M., Cupillard P., Caumon G., Sava P., Edwards J. (2019) Appraising structural interpretations using seismic data – theoretical elements. **Geophysics**, 84:N29–N40. Nominated for Geophysics Bright Spots.
- [Conference paper]: Irakarama M., Cupillard P., Caumon G, Sava P. (2017) Appraising structural models using seismic data: problem and challenges. In: **87th SEG Annual Meeting**, Houston, Texas.
- [Conference paper]: Irakarama M., Cupillard P., Caumon G, Sava P. (2017) Appraising structural interpretations using seismic data misfit functionals. In: **79th EAGE Conference & Exhibition**, Paris, France.

In chapter III.3, we highlight the practical challenges of appraising structural models using seismic data. We also consider appraising structural models by waveform inversion.

Other related papers

The following publications also benefited from this PhD project:

- [Journal paper]: Anquez P., Pellerin J., Irakarama M., Cupillard P., Lévy B., Caumon G. (2019) Automatic correction and simplification of geological maps and cross-sections for numerical simulations. **Comptes Rendus Geosciences**, 351:48–58.
- [Conference paper]: Renaudeau J., Irakarama M., Laurent G., Maerten F., Caumon G. (2018) Implicit modeling of geological structures: a Cartesian grid method handling discontinuities with ghost points. In: **41st Conference on Boundary Elements and Mesh Reduction Methods**, New Forest, UK.

Acknowledgements

I would like to thank Christelle Butault and Paul Angrand for the seismic interpretations presented in this general introduction, and Thomas Beraud for carrying out the velocity modeling experiment.

Bibliography

K. Anjyo, J. Lewis, and F. Pighin. Scattered data interpolation for computer graphics. In *ACM SIGGRAPH 2014 Courses*, Vancouver, Canada, 2014.

E. Baysal, D. Kosloff, and J. Sherwood. Reverse time migration. *GEOPHYSICS*, 48:1514–1524, 1983.

T. Beraud, M. Irakarama, and P. Cupillard. Ray tomography: implementation, structural uncertainties and geological constraints. In *2018 RING meeting*, Université de Lorraine-ENSG, Nancy, France, September 2018. ASGA.

B. Biondi. *3D Seismic Imaging*. Investigations in GEOPHYSICS. Society of Exploration Geophysicists, 2006.

N. Bleistein, J. W. S. Jr., and J. K. Cohen. *Mathematics of Multidimensional Seismic Imaging, Migration, and Inversion*. Interdisciplinary Applied Mathematics. Springer New York, 2001.

C. Bond. Uncertainty in structural interpretation: Lessons to be learnt. *Journal of Structural Geology*, 74:185–200, 2015.

C. Bond, Z. Shipton, and S. Jones. What do you think this is? Conceptual uncertainty in geoscience interpretation. *GSA Today*, 17:4–10, 2007.

P. Calcagno, J. Chilès, G. Courrioux, and A. Guillen. Geological modelling from the field data and geological knowledge Part I. Modeling method coupling 3D potential-field interpolation and geological rules. *Physics of the Earth and Planetary Interiors*, 17:147–157, 2008.

G. Caumon, F. cois Lepage, C. H. Sword, and J.-L. Mallet. Building and Editing a Sealed Geological Model. *Mathematical Geology*, 36:405–424, 2004.

G. Caumon, P. Collon, C. L. C. de Veslud, S. Viseur, and J. Sausse. Surface-Based 3D Modeling of Geological Structures. *Mathematical Geosciences*, 41:927–945, 2009.

G. Caumon, G. Gray, C. Antoine, and M.-O. Titeux. Three-dimensional implicit stratigraphic model building from remote sensing data on tetrahedral meshes: theory and application to regional model of La Popa Basin, NE Mexico. *IEEE Transactions on Geoscience and Remote Sensing*, 51(3), 2013.

J.-P. Chilès, C. Aug, A. Guillen, and T. Less. Modeling the geometry of geological units and its uncertainty in 3d form structural data: the potential-field method. In *Orebody Modeling and Strategic Mine Planning*, volume Spectrum 14, Perth, WA, 2004.

J. Claerbout. *Imaging the Earth's interior*. Blackwell Science Inc, 1985.

J. Etgen, S. Gray, and Y. Zhang. An overview of depth migration in exploration geophysics. *GEOPHYSICS*, 74(6), 2009.

M. D. Jackson, G. J. Hampson, J. H. Saunders, A. El-Sheikh, G. H. Graham, and B. Y. G. Massart. Surface-based reservoir modelling for flow simulation. *Geological Society, London Special Publications*, 387:271–292, 2013.

J.-L. Mallet. *Geomodeling*. Applied Geostatistics Series. Oxford University Press, 2002.

J.-L. Mallet. *Elements of mathematical sedimentary geology: The geochron model*. EAGE, 2014.

J. Messud, M. Reinier, H. Prigent, P. Guillaume, T. Coléou, and S. Masclet. Extracting seismic uncertainties from tomographic velocity inversion and their use in reservoir analysis. *The Leading Edge*, 36:127–132, 2017.

D. Miller, M. Oristaglio, and G. Beylkin. A new slant on seismic imaging: migration and integral geometry. *GEOPHYSICS*, 52:943–964, 1987.

R. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167:495–503, 2006.

P. Sava and S. Fomel. Angle-domain common-image gathers by wavefield continuation methods. *Geophysics*, 63:1065–1074, 2003.

C. Stork. Reflection tomography in the postmigrated domain. *GEOPHYSICS*, 57:680–692, 1992.

P. Thore, A. Shtuka, M. Lecour, T. Ait-Ettajer, and R. Cognot. Structural uncertainties: Determination, management, and applications. *GEOPHYSICS*, 67:840–852, 2002.

F. Wellmann and G. Caumon. 3D Structural geological models: Concepts, methods, and uncertainties. *Advances in Geophysics*, 59:1–121, 2018.

J. Wellmann, F. G.Horowitz, E. Schill, and K. Regenauer-Lieb. Towards incorporating uncertainty of structural data in 3D geological inversion. *Tectonophysics*, 490:141–151, 2010.

M. Woodward, D. Nichols, O. Zdraveva, P. Whitfield, and T. Johns. A decade of tomography. *GEOPHYSICS*, 73(5), 2008.

S. Xu, H. Chauris, G. Lambaré, and M. Noble. Common-angle migration: A strategy for imaging complex media. *GEOPHYSICS*, 66(6), 2001.

Part I

Notes on Seismic Imaging

Introductory Comments

Structural models are usually built from interpretations of seismic images. This first Part of the thesis is therefore devoted to seismic imaging. The problem of seismic imaging is that of transforming seismic traces, such as those illustrated in Figure 21a, into an image of the subsurface, such as that shown in Figure 21b, where geological structures can be identified.

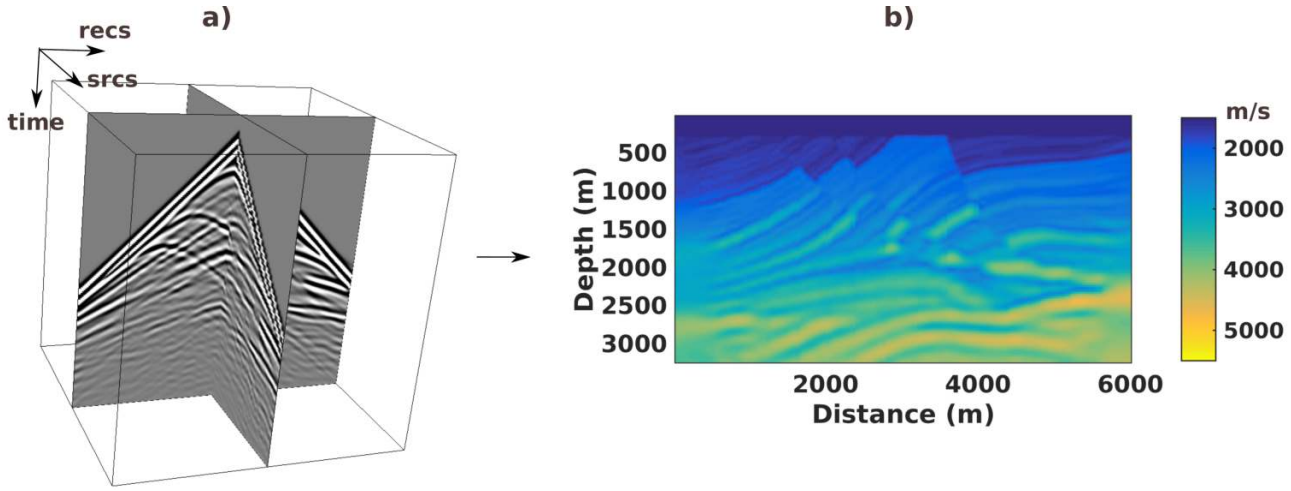

Figure 21: Waveform inversion of synthetic data computed in a portion of the Marmousi model (Figure 22a). **a**) Input seismic traces. **b**) Subsurface image obtained after waveform inversion. The initial velocity model is shown in Figure 22b.

Traditionally, seismic imaging has been developed from concepts based on geophysical intuition (Claerbout, 1971; Schneider, 1978; Claerbout, 1985). Arguably, a more modern way to present the seismic imaging problem is to formulate it as an optimization problem (Tarantola, 1984a,b; Plessix, 2006; Fichtner et al., 2006; Sava, 2014). I follow the optimization approach to seismic imaging in this thesis as I find it to more accessible to readers, like myself, with a limited grasp of the enormous seismic imaging literature. Plessix (2006) provides a good summary for the optimization approach to seismic imaging. In this thesis, the generic (L2-norm based) seismic imaging problem is formulated as

$$\left\{\begin{array}{rcl} m(\mathbf{x}) & = & \underset{m(\mathbf{x})}{\text{argmin}} \int_{\mathbf{s},\mathbf{r},t} \big[D_{obs}(\mathbf{r},t,\mathbf{s}) - D_{syn}(\mathbf{r},t,\mathbf{s},m)\big]^2 \\ & = & \underset{m(\mathbf{x})}{\text{argmin}} \int_{\mathbf{s},\mathbf{r},t,\mathbf{x}} \delta(\mathbf{x}-\mathbf{r}) \big[D_{obs}(\mathbf{x},t,\mathbf{s}) - D_{syn}(\mathbf{x},t,\mathbf{s},m)\big]^2 \ , \\ & & \text{subject to } D_{syn}(\mathbf{x},t,\mathbf{s},m) = \mathbf{F}\big[m(\mathbf{x}),\mathbf{s}\big] \end{array}\right. \tag{7}$$

where $D_{obs}(\mathbf{r},t,\mathbf{s})$ denotes the observed data from a source fired at $\mathbf{s}$ and recorded at receiver $\mathbf{r}$, $D_{syn}(\mathbf{r},t,\mathbf{s},m)$ denotes the corresponding synthetic data, $\mathbf{F}$ is the forward modeling operator, and $m(\mathbf{x})$ is the subsurface Earth model. When the operator $\mathbf{F}$ is linear in $m(\mathbf{x})$, problem 7 leads to what is known as migration and least-squares migration (Tarantola, 1984a; Zhang et al., 2015); when the operator $\mathbf{F}$ is nonlinear in $m(\mathbf{x})$, problem 7 leads to what is known as (full) waveform inversion (Tarantola, 1984b; Pratt et al., 1998).

Chapter I.1

Seismic Imaging

I.1.1 Introduction

Seismic imaging is a vast subject; in this chapter, I only give a summary of what I consider to be relevant to this thesis (in particular Part III). I present seismic imaging as an optimization problem (Plessix, 2006; Fichtner et al., 2006; Virieux and Operto, 2009). The reader who wishes to get more insights and a more intuition-based description of seismic imaging is referred to Claerbout (1985); Bleistein et al. (2001); Biondi (2006). Seismic imaging methods can be grouped into two main categories (Etgen et al., 2009):

1. In ray-based methods (Miller et al., 1987; Nemeth et al., 1999; Thierry et al., 1999; Xu and Lambaré, 2004), the forward modeling operator $\mathbf{F}$ in equation 7 is derived from ray-theory and/or the eikonal equation.
2. In wave-based methods (Kosloff and Baysal, 1983; Baysal et al., 1983; McMechan, 1983; Mulder and Plessix, 2004; Zhang and Sun, 2009; Zhang et al., 2015), the forward modeling operator $\mathbf{F}$ in equation 7 is derived from the wave equation.

In this chapter, we consider seismic imaging under the acoustic approximation; that is, the forward modeling operator $\mathbf{F}$ in equation 7 is derived from the acoustic wave equation

$$\mathbf{L}(m)P(\mathbf{x},t;\mathbf{s}) = \left[\frac{1}{m^2(\mathbf{x})}\frac{\partial^2}{\partial t^2} - \Delta\right]P(\mathbf{x},t;\mathbf{s}) = w(t)\delta(\mathbf{x}-\mathbf{s}), \tag{8}$$

where $P(\mathbf{x},t;\mathbf{s})$ denotes the synthetic wavefield for a source fired at $\mathbf{s}$, and $w(t)$ is the source wavelet. Equation 8 is nonlinear in the subsurface model $m(\mathbf{x})$; it can be linearized by decomposing the subsurface model into a smooth varying term $m_0(\mathbf{x})$ and an oscillatory term $m_1(\mathbf{x})$ (Tarantola, 1984a): $m(\mathbf{x}) = m_0(\mathbf{x}) + m_1(\mathbf{x})$. Assuming that the synthetic wavefield can also be decomposed in a similar manner: $P(\mathbf{x},t;\mathbf{s}) = P_0(\mathbf{x},t;\mathbf{s}) + P_1(\mathbf{x},t;\mathbf{s})$, it follows from the Taylor expansion

$$\frac{1}{m^2(\mathbf{x})} = \frac{1}{[m_0(\mathbf{x}) + m_1(\mathbf{x})]^2} \approx \frac{1}{m_0^2(\mathbf{x})} - 2\frac{m_1(\mathbf{x})}{m_0^3(\mathbf{x})}$$

that

$$\begin{aligned}\mathbf{L}(m_0)P_0(\mathbf{x},t;\mathbf{s}) + \mathbf{L}(m_0)P_1(\mathbf{x},t;\mathbf{s}) &= w(t)\delta(\mathbf{x}-\mathbf{s}) + 2\frac{m_1(\mathbf{x})}{m_0^3(\mathbf{x})}\frac{\partial^2}{\partial t^2}P(\mathbf{x},t;\mathbf{s}) \\ &\approx w(t)\delta(\mathbf{x}-\mathbf{s}) + 2\frac{m_1(\mathbf{x})}{m_0^3(\mathbf{x})}\frac{\partial^2}{\partial t^2}P_0(\mathbf{x},t;\mathbf{s}),\end{aligned}$$

leading to the linear (linear in m_1) Born modeling wave equation

$$\begin{cases} \mathbf{L}(m_0)P_0(\mathbf{x},t;\mathbf{s}) = w(t)\delta(\mathbf{x}-\mathbf{s}) \\ \mathbf{L}(m_0)P_1(\mathbf{x},t;\mathbf{s}) = 2\frac{m_1(\mathbf{x})}{m_0^3(\mathbf{x})}\frac{\partial^2}{\partial t^2}P_0(\mathbf{x},t;\mathbf{s}) \end{cases}. \tag{9}$$

In equation 9, the smooth varying part of the model $m_0(\mathbf{x})$ is assumed to be known and the oscillatory part of the model $m_1(\mathbf{x})$ is the unknown subsurface model of interest; $P_1(\mathbf{x}, t; \mathbf{s})$ represents the synthetic wavefield.

In the next two sections, I will consider the case where the forward modeling operator $\mathbf{F}$ in equation 7 is the solution of equation 9 and equation 8 to estimate the subsurface model from the data set in Figure 21a. This data set was generated using finite differences (Virieux, 1984) in the portion of the Marmousi model (Versteeg, 1994) shown in Figure 22a. The model is discretized on a grid of size 200×215 with $dx = 30\,m$ and $dz = 15\,m$, and data are computed with a Ricker source wavelet with a dominant frequency of $15\,Hz$. The data set consists of 60 shot-gathers recorded at 200 receivers; both sources and receivers are evenly distributed along the free surface.

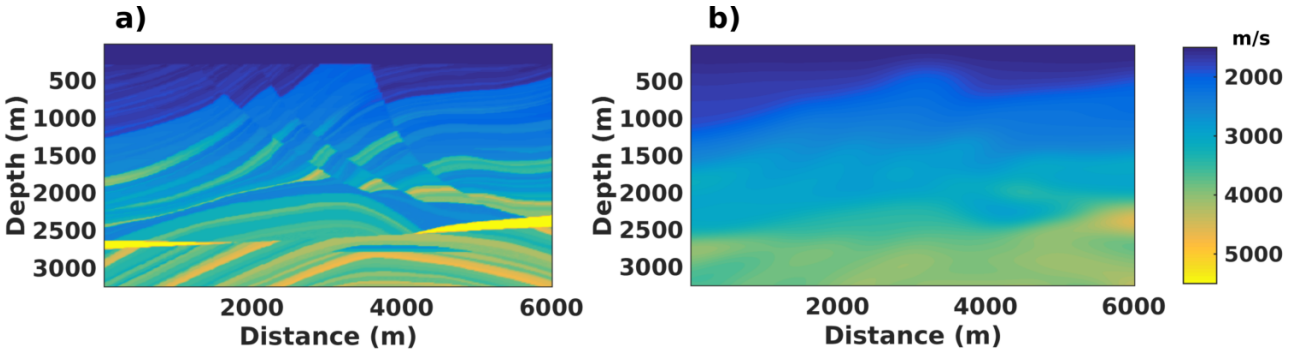

Figure 22: Portion of the Marmousi velocity model (Versteeg, 1994) used for numerical experiments. **a**) The reference (true) velocity model. **b**) The migration velocity model obtained by smoothing the reference model.

I.1.2 Linearized waveform inversion

Linearized seismic imaging techniques are obtained from equation 7 by equating the forward modeling operator $\mathbf{F}$ to the solution of the Born equation 9. In particular, the seismic imaging problem can be stated as

$$\begin{cases} m_1(\mathbf{x}) = \underset{m_1(\mathbf{x})}{\operatorname{argmin}}\, \Phi(m_1) = \underset{m_1(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},t,\mathbf{x}} \delta(\mathbf{x}-\mathbf{r}) \big[D_o(\mathbf{x},t,\mathbf{s}) - P_1(\mathbf{x},t,\mathbf{s},m_1)\big]^2 \\ \text{subject to } \mathbf{L}(m_0) P_1(\mathbf{x},t,\mathbf{s},m_1) = 2\frac{m_1(\mathbf{x})}{m_0^3(\mathbf{x})}\frac{\partial^2}{\partial t^2} P_0(\mathbf{x},t;\mathbf{s}) \end{cases}, \tag{10}$$

where $m_0(\mathbf{x})$ and $P_0(\mathbf{x}, t; \mathbf{s})$ are as defined in section I.1.1. In what follows, I omit parameters to simplify the notation; I also drop the subscript 1 in m_1 and P_1. The problem 10 can be solved by iterative methods like, for example, the conjugate gradient method (Shewchuk, 1994; Nocedal and Wright, 2006). Such iterative methods require an explicit computation of the gradient of the objective function $\Phi(m)$ with respect to the model m, which can be obtained using the adjoint-state method (Plessix, 2006; Fichtner et al., 2006). Equation 10 leads to the Lagrangian functional

$$\begin{aligned} \mathscr{L} &= \Big\langle 1, \delta(\mathbf{x}-\mathbf{r})\big[D_o - P\big]^2 \Big\rangle_{\mathbf{s},\mathbf{r},t,\mathbf{x}} \\ &+ \Big\langle \lambda, \big[\mathbf{L}(m_0)P - 2\frac{m}{m_0^3}\frac{\partial^2}{\partial t^2}P_0\big] \Big\rangle_{\mathbf{s},t,\mathbf{x}}, \end{aligned} \tag{11}$$

where λ is the Lagrange multiplier and we have used the notation $\langle f, g \rangle_{\mathbf{x},t,..} = \int_{\mathbf{x},t,..} fg$. The gradient of the Lagrangian with respect to m is given by

$$\begin{aligned}\frac{\partial}{\partial m}\mathscr{L} &= \left\langle -2\delta(\mathbf{x}-\mathbf{r})\big[D_o - P\big], \frac{\partial}{\partial m}P \right\rangle_{\mathbf{s},\mathbf{r},t,\mathbf{x}} \\ &+ \left\langle \lambda, \mathbf{L}(m_0)\frac{\partial}{\partial m}P \right\rangle_{\mathbf{s},t,\mathbf{x}} \\ &+ \left\langle \lambda, -\frac{2}{m_0^3}\frac{\partial^2}{\partial t^2}P_0 \right\rangle_{\mathbf{s},t,\mathbf{x}}. \end{aligned} \tag{12}$$

Using the adjoint of the acoustic modeling operator $\mathbf{L}^*$, that is

$$\left\langle \lambda, \mathbf{L}(m_0)\frac{\partial}{\partial m}P \right\rangle_{\mathbf{s},t,\mathbf{x}} = \left\langle \mathbf{L}^*(m_0)\lambda, \frac{\partial}{\partial m}P \right\rangle_{\mathbf{s},t,\mathbf{x}}, \tag{13}$$

equation 12 leads to

$$\begin{aligned}\frac{\partial}{\partial m}\mathscr{L} &= \left\langle \frac{\partial}{\partial m}P, \mathbf{L}^*(m_0)\lambda - \int_{\mathbf{r}} 2\delta(\mathbf{x}-\mathbf{r})\big[D_o - P\big] \right\rangle_{\mathbf{s},t,\mathbf{x}} \\ &- \left\langle \lambda, \frac{2}{m_0^3}\frac{\partial^2}{\partial t^2}P_0 \right\rangle_{\mathbf{s},t,\mathbf{x}}. \end{aligned} \tag{14}$$

In practice, applying the adjoint operator $\mathbf{L}^*$ translates to solving the acoustic wave equation 8 backwards in time (e.g. Fichtner et al., 2006). It follows from equation 14 that

$$\begin{cases} \mathbf{L}^*(m_0)\lambda = 2\int_{\mathbf{r}} \delta(\mathbf{x}-\mathbf{r})\big[D_o - P\big] \\ \frac{\partial}{\partial m}\mathscr{L} = -\big\langle \lambda, \frac{2}{m_0^3}\frac{\partial^2}{\partial t^2}P_0 \big\rangle_{\mathbf{s},t,\mathbf{x}} = \big\langle \frac{\partial}{\partial t}\lambda, \frac{2}{m_0^3}\frac{\partial}{\partial t}P_0 \big\rangle_{\mathbf{s},t,\mathbf{x}} \end{cases}. \tag{15}$$

The derivative of the objective function $\Phi(m)$ with respect to m is given by

$$\frac{\partial}{\partial m}\Phi(m) = \frac{\partial}{\partial m}\mathscr{L}, \tag{16}$$

which essentially follows by noting from equation 11 that $\Phi(m) = \mathscr{L}$ when P satisfies the Born equation 9. The reader is referred to Plessix (2006) and the appendix of Virieux and Operto (2009) for more details on this argument. Now that an expression of the derivative of the objective function with respect to the model is available (eq. 16), the problem in equation 10 can be solved iteratively using an algorithm of the form

$$\begin{cases} \mathbf{L}(m_0)P^n = 2\frac{m^n}{m_0^3}\frac{\partial^2}{\partial t^2}P_0 \\ \mathbf{L}^*(m_0)\lambda^n = 2\int_{\mathbf{r}} \delta(\mathbf{x}-\mathbf{r})\big[D_o - P^n\big] \\ \Delta m^n = \big\langle \frac{\partial}{\partial t}\lambda^n, \frac{2}{m_0^3}\frac{\partial}{\partial t}P_0 \big\rangle_{\mathbf{s},t} \\ m^{n+1} = m^n + \Delta m^n \end{cases}, \tag{17}$$

Observation 1. The term $\frac{\partial}{\partial m}\mathscr{L}$ in equation 15 was computed for the entire domain; if it were computed at a fixed point $\mathbf{x}$ the integral over $\mathbf{x}$ would be absent. This is the reason why there is no integral over $\mathbf{x}$ in the discrete (pointwise) gradient Δm^n in equation 17.

Figure 23a shows the subsurface image obtained from the data in Figure 21a after 10 conjugate gradient iterations of the algorithm in equation 17, using the model in Figure 22b as m_0.

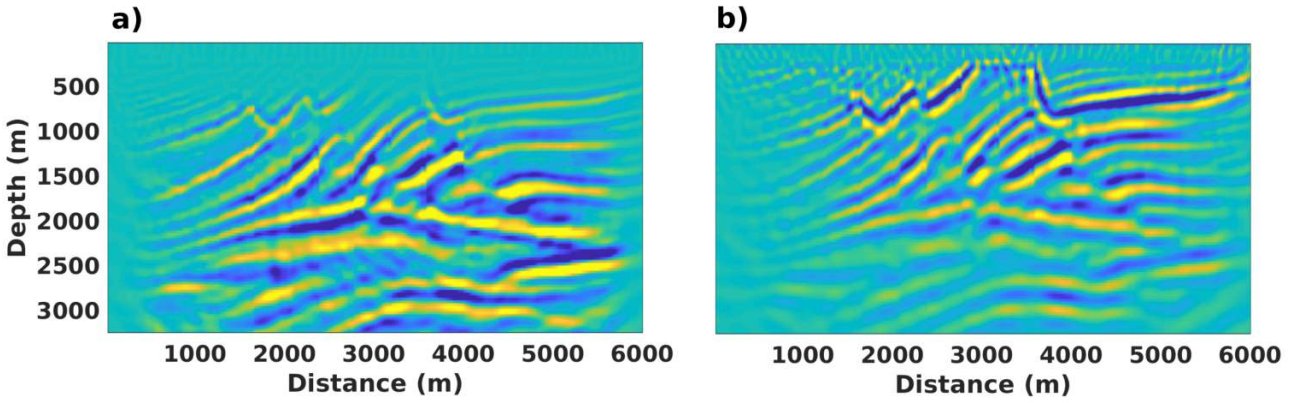

Figure 23: Linearized seismic imaging. **a**) Subsurface image obtained by linearized inversion after 10 iterations (least-squares reverse time migration). **b**) Subsurface image obtained by a single iteration of linearized inversion (reverse time migration).

Migration

Traditionally, seismic imaging seems to have been developed from concepts based on geophysical intuition (Claerbout, 1971; Schneider, 1978; Gazdag, 1980; Baysal et al., 1983; Claerbout, 1985) under the name of migration. It was later shown that migration is in fact the first iteration of the algorithm in equation 17 (Tarantola, 1984a). In particular, at the first iteration of the algorithm in equation 17, we have $m^0 = P^0 = 0$, leading to the familiar reverse time migration formula (e.g. Zhang and Sun, 2009)

$$\begin{cases} \mathbf{L}^*(m_0)\lambda = 2\int_{\mathbf{r}} \delta(\mathbf{x}-\mathbf{r})D_o \\ m = \left\langle \frac{\partial}{\partial t}\lambda^n, \frac{2}{m_0^3}\frac{\partial}{\partial t}P_0 \right\rangle_{\mathbf{s},t} = -\left\langle \lambda^n, \frac{2}{m_0^3}\frac{\partial^2}{\partial t^2}P_0 \right\rangle_{\mathbf{s},t} \end{cases}. \tag{18}$$

An alternative approach to present migration is as the adjoint of any linearized forward seismic modeling operator (e.g. Claerbout, 2004). Consider for example the Born modeling formula appearing in equation 10, it is shown in the appendix (section I.1.5) that its solution can be written as

$$P(\mathbf{r},t;\mathbf{s}) = -2\int_{\mathbf{x},t'} G(\mathbf{x},t-t';\mathbf{r})\ddot{P}_0(\mathbf{x},t';\mathbf{s})\frac{m(\mathbf{x})}{m_0^3(\mathbf{x})}, \tag{19}$$

where the Green's function $G(\mathbf{x},t;\mathbf{r})$ is the solution to

$$\mathbf{L}(m_0)G(\mathbf{x},t;\mathbf{r}) = \delta(\mathbf{x}-\mathbf{r})\delta(t).$$

The adjoint of the Born modeling equation 19 leads to another familiar reverse time migration formula (e.g. Symes, 2008)

$$m(\mathbf{x}) = -2\int_{\mathbf{r},\mathbf{s},t,t'} G(\mathbf{x},t-t';\mathbf{r})\ddot{P}_0(\mathbf{x},t';\mathbf{s})\frac{1}{m_0^3(\mathbf{x})}P(\mathbf{r},t;\mathbf{s}), \tag{20}$$

where, in this case, $P(\mathbf{r},t;\mathbf{s})$ is replaced by $D_o(\mathbf{r},t;\mathbf{s})$ in practice. The equivalence between equation 18 and equation 20 is shown in the appendix (section I.1.5). Figure 23b shows the subsurface image obtained by equation 20 (or alternatively equation 18).

I.1.3 Nonlinear waveform inversion

Nonlinear seismic imaging is obtained from equation 7 by equating the forward modeling operator $\mathbf{F}$ to the solution of the acoustic equation 8. In particular, the seismic imaging problem can be stated as

$$\begin{cases} m(\mathbf{x}) = \underset{m(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},t,\mathbf{x}} \delta(\mathbf{x}-\mathbf{r})\big[D_o(\mathbf{x},t,\mathbf{s}) - P(\mathbf{x},t,\mathbf{s},m)\big]^2 \\ \text{subject to } \mathbf{L}(m)P(\mathbf{x},t,\mathbf{s},m) = w(t)\delta(\mathbf{x}-\mathbf{s}) \end{cases}, \tag{21}$$

In what follows, I omit parameters to simplify the notation. The procedure to solve problem 21 is very similar to that described in the previous section for solving the linearized problem 10. We first define the Lagrangian

$$\begin{aligned} \mathscr{L} &= \Big\langle 1, \delta(\mathbf{x}-\mathbf{r})\big[D_o - P\big]^2 \Big\rangle_{\mathbf{s},\mathbf{r},t,\mathbf{x}} \\ &+ \Big\langle \lambda, \big[\mathbf{L}(m)P - w(t)\delta(\mathbf{x}-\mathbf{s})\big] \Big\rangle_{\mathbf{s},t,\mathbf{x}} ; \end{aligned}$$

then we compute its derivative with respect the model $m(\mathbf{x})$

$$\begin{aligned} \frac{\partial}{\partial m}\mathscr{L} &= \Big\langle -2\delta(\mathbf{x}-\mathbf{r})\big[D_o - P\big], \frac{\partial}{\partial m}P \Big\rangle_{\mathbf{s},\mathbf{r},t,\mathbf{x}} \\ &+ \Big\langle \lambda, \Big[\frac{\partial}{\partial m}\mathbf{L}(m)\Big]P \Big\rangle_{\mathbf{s},t,\mathbf{x}} \\ &+ \Big\langle \lambda, \mathbf{L}(m)\frac{\partial}{\partial m}P \Big\rangle_{\mathbf{s},t,\mathbf{x}} . \end{aligned} \tag{22}$$

Noting that

$$\frac{\partial}{\partial m}\mathbf{L}(m) = -\frac{2}{m^3}\frac{\partial^2}{\partial t^2},$$

and using the adjoint as defined in equation 13, equation 22 leads to

$$\begin{aligned} \frac{\partial}{\partial m}\mathscr{L} &= \Big\langle \frac{\partial}{\partial m}P, \mathbf{L}^*(m)\lambda - 2\int_{\mathbf{r}} \delta(\mathbf{x}-\mathbf{r})\big[D_o - P\big] \Big\rangle_{\mathbf{s},t,\mathbf{x}} \\ &- \Big\langle \lambda, \frac{2}{m^3}\frac{\partial^2}{\partial t^2}P \Big\rangle_{\mathbf{s},t,\mathbf{x}} . \end{aligned} \tag{23}$$

It follows from equation 23 that

$$\begin{cases} \mathbf{L}^*(m)\lambda = 2\int_{\mathbf{r}} \delta(\mathbf{x}-\mathbf{r})\big[D_o - P\big] \\ \frac{\partial}{\partial m}\mathscr{L} = -\big\langle \lambda, \frac{2}{2m^3}\frac{\partial^2}{\partial t^2}P\big\rangle_{\mathbf{s},t,\mathbf{x}} = \big\langle \frac{\partial}{\partial t}\lambda, \frac{2}{m^3}\frac{\partial}{\partial t}P\big\rangle_{\mathbf{s},t,\mathbf{x}} \end{cases},$$

which leads to an iterative solution to problem 21 of the form

$$\begin{cases} \mathbf{L}(m)P^n = w(t)\delta(\mathbf{x}-\mathbf{s}) \\ \mathbf{L}^*(m)\lambda^n = 2\int_{\mathbf{r}} \delta(\mathbf{x}-\mathbf{r})\big[D_o - P^n\big] \\ \Delta m^n = \big\langle \frac{\partial}{\partial t}\lambda^n, \frac{2}{m^3}\frac{\partial}{\partial t}P^n\big\rangle_{\mathbf{s},t} \\ m^{n+1} = m^n + \Delta m^n \end{cases}. \tag{24}$$

Figure 24a shows the subsurface image obtained after a multi-scale inversion (Bunks et al., 1995) using equation 24: we start with 10 conjugate gradient iterations up to $6\,Hz$, then 10 iterations up to $8\,Hz, 10\,Hz$, and finally up to $12\,Hz$. In practice, both linearized and nonlinear inversion require an efficient preconditioner. For example, Figure 24b shows the subsurface image that would have been obtained if no preconditioner was used in Figure 24a. Preconditioning is discussed in chapter I.2.

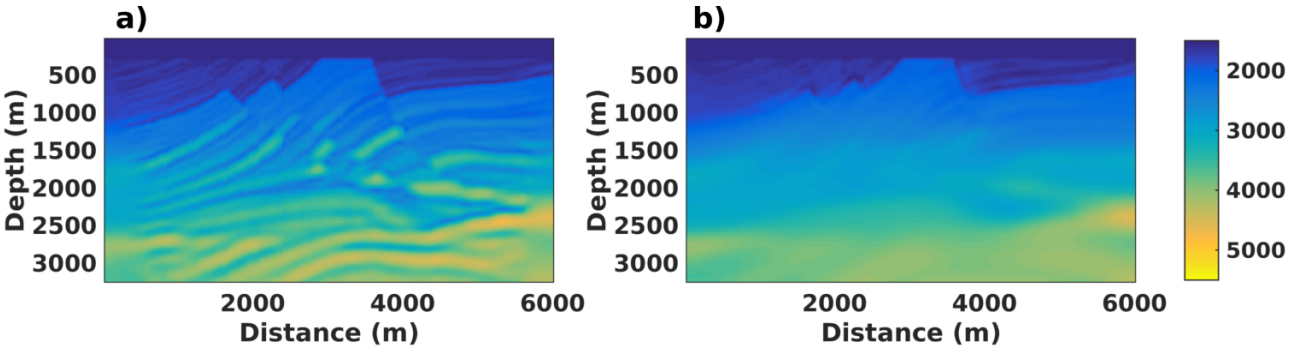

Figure 24: Nonlinear seismic imaging (waveform inversion). **a**) Subsurface image obtained using conjugate gradients with a preconditioner. **b**) Subsurface image obtained using conjugate gradients without a preconditioner.

I.1.4 Discussion

In section I.1.2, we presented a nonlinear conjugate gradient solution to linearized inversion; this was mainly to highlight the similarity with nonlinear inversion. Linearized inversion can also be performed using linear conjugate gradients. The requirements for solving a problem with linear conjugate gradients are:

1. The forward modeling operator should be linear with respect to the model $m(\mathbf{x})$, which is the case for the Born modeling equation 19.

2. The adjoint of the forward modeling operator should be available, which is equation 20 in our example.

In practice, the linearized inversion scheme in equation 17 where all shot gathers are inverted simultaneously did not work very well. The image in Figure 23a was obtained by inverting each shot gather independently and then stacking the resulting partial images. Figure 25 compares images obtained by shotwise inversion against images obtained by simultaneous inversion of all the shot gathers.

The discussion in section I.1.2 suggests that a seismic image has the same units as velocity (m_1 in equation 9). It is customary however to plot seismic images without units. This is because seismic images usually undergo a series of post-processing steps for better visual effects. Figure 26 shows an application of some common post-processing steps to the reverse-time migration image in Figure 23b.

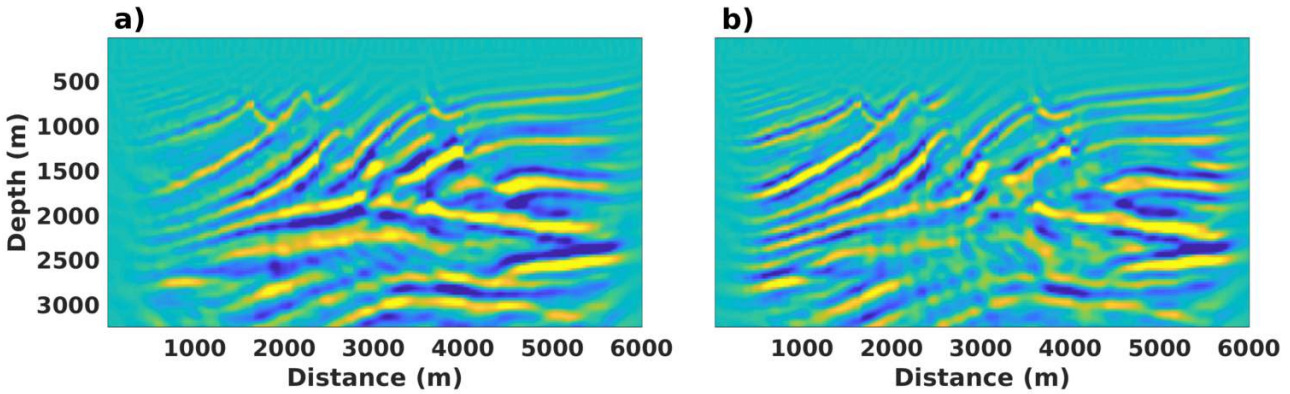

Figure 25: Linearized seismic imaging, 10 iterations. **a**) Subsurface image obtained by shotwise inversion followed by stacking. **b**) Subsurface image obtained by simultaneously inversion of all shot gathers.

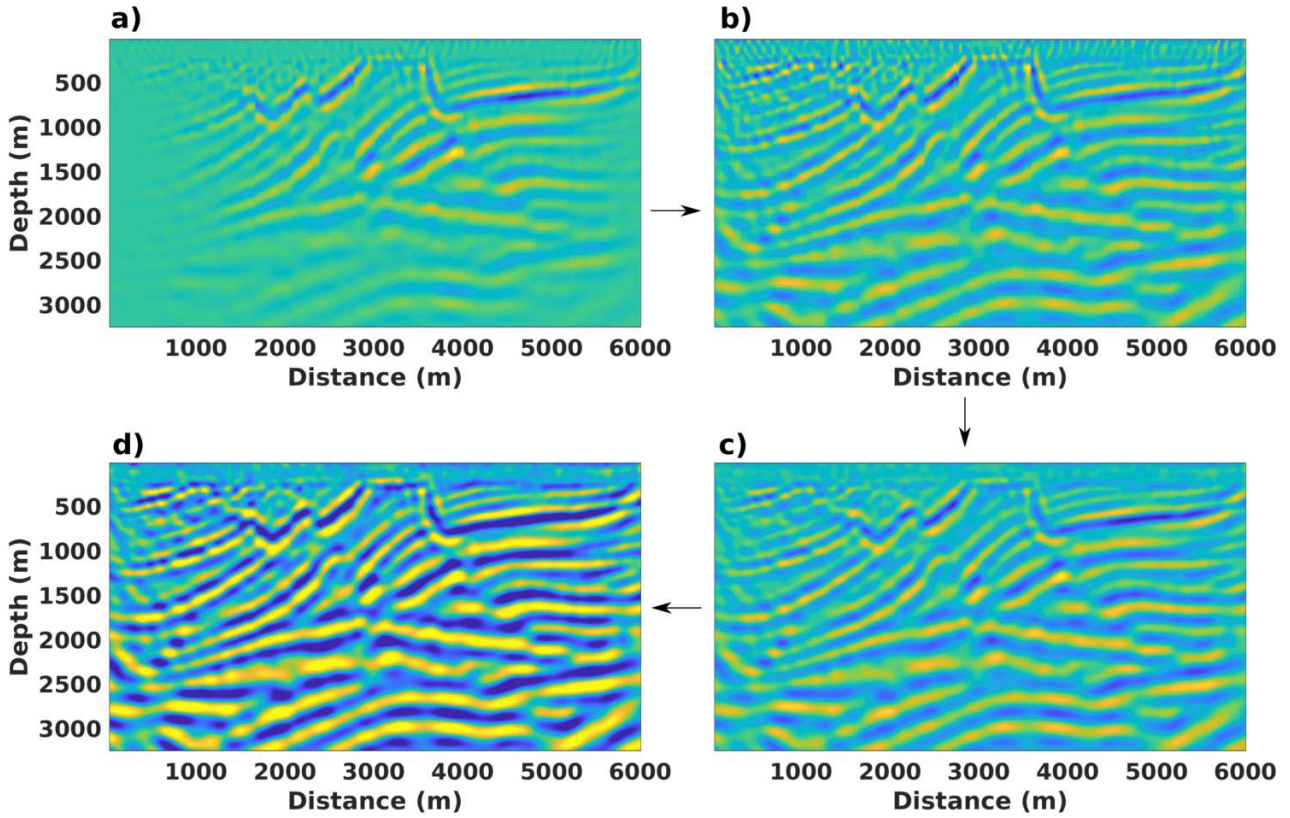

Figure 26: Some common post-processing steps in seismic imaging. **a**) Reverse-time migration image (Figure 23b before clipping). **b**) Automatic gain control (here, normalization by the smoothed absolute value of the initial image). **c**) Smoothing. **d**) Clipping.

In chapter I.3, it is shown that replacing $P_0(\mathbf{x}, t'; \mathbf{s})$ with $G(\mathbf{x}, t'; \mathbf{s})$ in equations 19 and 20 (which corresponds to replacing the source wavelet $w(t)$ in equation 9 by $\delta(t)$), and then replacing the Green's functions $G(\mathbf{x}, t - t'; \mathbf{r})$ and $G(\mathbf{x}, t'; \mathbf{s})$ by their ray-based counterpart, equations 19 and 20 become Kirchhoff modeling and Kirchhoff migration operators, respectively. Figure 27 shows the subsurface images obtained by ray-based least-squares migration and ray-based migration, after post-processing as described in Figure 26.

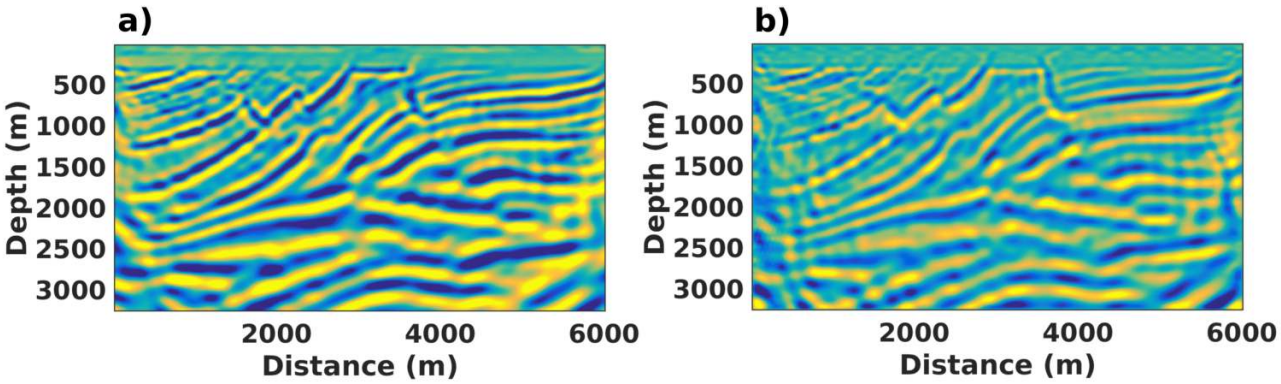

Figure 27: Ray-based seismic imaging. **a**) Subsurface image obtained by linearized inversion after 10 iterations (least-squares Kirchhoff migration). **b**) Subsurface image obtained by a single iteration of linearized inversion (Kirchhoff migration). These images have been post-processed as described in Figure 26.

I.1.5 Appendix

Derivation of the Born modeling equation 19

In this paragraph, we demonstrate the Born modeling equation 19. We start from the Born modeling formula in equation 10, that is

$$\mathbf{L}(m_0)P(\mathbf{x},t;\mathbf{s}) = 2\frac{m(\mathbf{x})}{m_0^3(\mathbf{x})}\frac{\partial^2}{\partial t^2}P_0(\mathbf{x},t;\mathbf{s}). \tag{25}$$

For simplicity, we define

$$g(\mathbf{x},t;\mathbf{s}) = 2\frac{m(\mathbf{x})}{m_0^3(\mathbf{x})}\ddot{P}_0(\mathbf{x},t;\mathbf{s}),$$

allowing us to express equation 25 as

$$\begin{aligned}\mathbf{L}(m_0)P(\mathbf{x},t;\mathbf{s}) &= g(\mathbf{x},t;\mathbf{s}) &(26)\\ &= \int_{\mathbf{x}',t'}\delta(\mathbf{x}-\mathbf{x}')\delta(t-t')g(\mathbf{x}',t';\mathbf{s})\\ &= \int_{\mathbf{x}',t'}\mathbf{L}(m_0)G(\mathbf{x},t;\mathbf{x}',t')g(\mathbf{x}',t';\mathbf{s}). &(27)\end{aligned}$$

In equation 27, we have defined

$$\mathbf{L}(m_0)G(\mathbf{x},t;\mathbf{x}',t') = \delta(\mathbf{x}-\mathbf{x}')\delta(t-t'). \tag{28}$$

It follows from equation 27 that

$$\mathbf{L}(m_0)\big[P(\mathbf{x},t;\mathbf{s}) - \int_{\mathbf{x}',t'}G(\mathbf{x},t;\mathbf{x}',t')g(\mathbf{x}',t';\mathbf{s})\big] = 0,$$

leading to

$$P(\mathbf{x},t;\mathbf{s}) = \int_{\mathbf{x}',t'}G(\mathbf{x},t;\mathbf{x}',t')g(\mathbf{x}',t';\mathbf{s}). \tag{29}$$

Equation 29 represents the solution to equation 25 in terms of the Green's function $G(\mathbf{x},t;\mathbf{x}',t')$ (i.e., the solution to equation 28). The modeled seismic traces are obtained by sampling $P(\mathbf{x},t;\mathbf{s})$ at receivers $\mathbf{r}$; that is,

$$
\begin{aligned}
P(\mathbf{r},t;\mathbf{s}) &= \int_{\mathbf{x}} \delta(\mathbf{x}-\mathbf{r}) P(\mathbf{x},t;\mathbf{s}) \\
&= \int_{\mathbf{x}} \delta(\mathbf{x}-\mathbf{r}) \int_{\mathbf{x}',t'} G(\mathbf{x},t;\mathbf{x}',t') g(\mathbf{x}',t';\mathbf{s}) \\
&= \int_{\mathbf{x}',t'} G(\mathbf{r},t;\mathbf{x}',t') g(\mathbf{x}',t';\mathbf{s}) \\
&= \int_{\mathbf{x}',t'} G(\mathbf{r},t-t';\mathbf{x}') g(\mathbf{x}',t';\mathbf{s}) \qquad (30) \\
&= \int_{\mathbf{x}',t'} G(\mathbf{x}',t-t';\mathbf{r}) g(\mathbf{x}',t';\mathbf{s}) \qquad (31) \\
&= \int_{\mathbf{x},t'} 2\frac{m(\mathbf{x})}{m_0^3(\mathbf{x})} G(\mathbf{x},t-t';\mathbf{r}) \frac{\partial^2}{\partial t'^2} P_0(\mathbf{x},t';\mathbf{s}) \\
&= -\int_{\mathbf{x},t'} 2\frac{m(\mathbf{x})}{m_0^3(\mathbf{x})} \frac{\partial}{\partial t'} G(\mathbf{x},t-t';\mathbf{r}) \frac{\partial}{\partial t'} P_0(\mathbf{x},t';\mathbf{s}) \qquad (32) \\
&= -\int_{\mathbf{x},t'} 2\frac{m(\mathbf{x})}{m_0^3(\mathbf{x})} \frac{\partial}{\partial t'} \tau \frac{\partial}{\partial \tau} G(\mathbf{x},\tau;\mathbf{r}) \frac{\partial}{\partial t'} P_0(\mathbf{x},t';\mathbf{s}), \text{ with } \tau = t-t' \qquad (33) \\
&= \int_{\mathbf{x},t'} 2\frac{m(\mathbf{x})}{m_0^3(\mathbf{x})} \frac{\partial}{\partial \tau} G(\mathbf{x},\tau;\mathbf{r}) \frac{\partial}{\partial t'} P_0(\mathbf{x},t';\mathbf{s}), \text{ with } \tau = t-t' \qquad (34) \\
&= 2\int_{\mathbf{x},t'} \dot{G}(\mathbf{x},t-t';\mathbf{r}) \dot{P}_0(\mathbf{x},t';\mathbf{s}) \frac{m(\mathbf{x})}{m_0^3(\mathbf{x})}, \qquad (35)
\end{aligned}
$$

which demonstrates equation 19.

In equation 30, we have used

$$G(\mathbf{x},t;\mathbf{x}',t') = G(\mathbf{x},t-t';\mathbf{x}',0) = G(\mathbf{x},t-t';\mathbf{x}'). \qquad (36)$$

This follows from the definition of $G(\mathbf{x},t;\mathbf{x}',t')$ in equation 28; in particular,

$$
\begin{aligned}
\delta(\mathbf{x}-\mathbf{x}')\delta(t) &= \delta(\mathbf{x}-\mathbf{x}')\delta(t-0) = \mathbf{L}(m_0)G(\mathbf{x},t;\mathbf{x}',0) \\
&= \delta(\mathbf{x}-\mathbf{x}')\delta(t+t'-t') = \mathbf{L}(m_0)G(\mathbf{x},t;\mathbf{x}',0) \\
&\quad \text{using } l = t+t' \rightarrow t = l-t', \text{ it follows that} \\
&= \delta(\mathbf{x}-\mathbf{x}')\delta(l-t') = \mathbf{L}(m_0)G(\mathbf{x},l-t';\mathbf{x}',0) \\
\mathbf{L}(m_0)G(\mathbf{x},l;\mathbf{x}',t') &= \delta(\mathbf{x}-\mathbf{x}')\delta(l-t') = \mathbf{L}(m_0)G(\mathbf{x},l-t';\mathbf{x}',0)
\end{aligned}
$$

In equation 31, we have used

$$G(\mathbf{x},t;\mathbf{x}',t') = G(\mathbf{x}',t;\mathbf{x},t'). \qquad (37)$$

This is follows from the definition of $G(\mathbf{x},t;\mathbf{x}',t')$ in equation 28; in particular,

$$
\begin{aligned}
\mathbf{L}(m_0)G(\mathbf{x},t;\mathbf{x}',t') &= \delta(\mathbf{x}-\mathbf{x}')\delta(t-t') \\
&= \delta(\mathbf{x}'-\mathbf{x})\delta(t-t') \\
&= \mathbf{L}(m_0)G(\mathbf{x}',t;\mathbf{x},t')
\end{aligned}
$$

The equivalence between equation 18 and equation 20

In this paragraph, we demonstrate the equivalence between equation 18 and equation 20. First, let us define

$$g(\mathbf{r},\mathbf{x},t,\mathbf{s}) = 2\int_{\mathbf{r}} \delta(\mathbf{x}-\mathbf{r}) D_o(\mathbf{x},t,\mathbf{s}),$$

so that equation 18 can simplify to

$$\left\{ \begin{array}{l} \mathbf{L}^*(m_0)\lambda = g(\mathbf{r},\mathbf{x},t,\mathbf{s}) \\ m = -\left\langle \lambda, \frac{2}{m_0^3}\frac{\partial^2}{\partial t^2} P_0 \right\rangle_{\mathbf{s},t} \end{array} \right. . \tag{38}$$

By looking at equation 26 and its solution, equation 29, it follows that the solution to the first equation in 38 is

$$\begin{aligned} \lambda &= \int_{\mathbf{x}',t'} G(\mathbf{x},t;\mathbf{x}',t') g(\mathbf{r},\mathbf{x}',t';\mathbf{s}) \\ &= 2\int_{\mathbf{x}',t'} G(\mathbf{x},t;\mathbf{x}',t') \int_{\mathbf{r}} \delta(\mathbf{x}'-\mathbf{r}) D_o(\mathbf{x}',t',\mathbf{s}) \\ &= 2\int_{\mathbf{x}',t',\mathbf{r}} G(\mathbf{x},t;\mathbf{x}',t') \delta(\mathbf{x}'-\mathbf{r}) D_o(\mathbf{x}',t',\mathbf{s}) \\ &= 2\int_{t',\mathbf{r}} G(\mathbf{x},t;\mathbf{r},t') D_o(\mathbf{r},t',\mathbf{s}) \\ &= 2\int_{t',\mathbf{r}} G(\mathbf{x},t';\mathbf{r},t) D_o(\mathbf{r},t',\mathbf{s}) \qquad (39) \\ &= 2\int_{t',\mathbf{r}} G(\mathbf{x},t'-t;\mathbf{r}) D_o(\mathbf{r},t',\mathbf{s}), \qquad (40) \end{aligned}$$

where we have used equation 37 in equation 39, and equation 36 in equation 40. Inserting the expression of λ, equation 40, in the second equation in 38 leads to

$$\begin{aligned} m(\mathbf{x}) &= -2\int_{\mathbf{s},t} \frac{2}{m_0^3(\mathbf{x})}\frac{\partial^2}{\partial t^2} P_0(\mathbf{x},t;\mathbf{s}) \int_{\mathbf{r},t'} G(\mathbf{x},t'-t;\mathbf{r}) D_o(\mathbf{r},t',\mathbf{s}) \\ &= -2\int_{\mathbf{r},\mathbf{s},t,t'} G(\mathbf{x},t'-t;\mathbf{r}) \frac{\partial^2}{\partial t^2} P_0(\mathbf{x},t;\mathbf{s}) \frac{2}{m_0^3(\mathbf{x})} D_o(\mathbf{r},t',\mathbf{s}), \end{aligned}$$

which demonstrates the equivalence between equation 18 and equation 20.

Chapter I.2

Reverse time migration as a preconditioner for waveform inversion

I.2.1 Summary

Waveform inversion has gained recognition as a powerful technique for imaging the subsurface quantitatively by fitting synthetic data to observed data. On the one hand, linearized waveform inversion estimates the subsurface reflectivity model and, on the other hand, nonlinear waveform inversion estimates the velocity model in the subsurface. Waveform inversion problems are usually solved using iterative methods due to their large sizes. The high computational cost of waveform inversion is a challenge that has motivated geoscientists to develop preconditioning techniques to accelerate these iterative methods. In this chapter, I propose to use the reverse time migration (RTM) image as a preconditioner. The premise of the proposed method is that because the RTM image is obtained by a zero-lag cross-correlation of the source and receiver wavefields, the RTM image is a measure of both the subsurface reflectivity and illumination for a given acquisition geometry. I argue that the low frequency background signal traditionally regarded as noise in conventional RTM imaging is an approximation of the diagonal elements of the Hessian matrix of the objective function and can therefore be exploited to accelerate waveform inversion.

I.2.2 Introduction

Linearized waveform inversion (Tarantola, 1984a; Zhang et al., 2015; Rocha et al., 2018) and nonlinear waveform inversion (Tarantola, 1984b; Pratt et al., 1998) allow to generate optimal subsurface images by fitting synthetic data to observed data. Linearized inversion estimates the subsurface reflectivity, while nonlinear inversion seeks to estimate the subsurface velocity. Both linear and nonlinear waveform inversion problems are usually solved using iterative methods. Popular solvers for waveform inversion problems include gradient methods and quasi-Newton methods (Virieux and Operto, 2009). Gradient solvers generally show a low convergence rate, requiring preconditioning in order to make them competitive with quasi-Newton methods. The objective of this chapter is to propose a simple and efficient preconditioner for gradient methods.

A popular technique for preconditioning gradient based waveform inversion is to design a preconditioner that approximates the diagonal of the Hessian matrix of the L2-norm objective function (e.g. Plessix and Mulder, 2004)

$$\mathbf{H}(\mathbf{x},\mathbf{y}) \;=\; \int_{\mathbf{s},\mathbf{r},t} \left[\int_{\tau} \dot{G}_{\mathbf{r}}(t-\tau,\mathbf{x}) \dot{G}_{\mathbf{s}}(\tau,\mathbf{x}) \int_{\tau} \dot{G}_{\mathbf{r}}(t-\tau,\mathbf{y}) \dot{G}_{\mathbf{s}}(\tau,\mathbf{y}) \right],$$

where $\dot{G}_{\mathbf{s}}$ and $\dot{G}_{\mathbf{r}}$ represent time derivatives of the source and receiver wavefields; $\mathbf{x}$ and $\mathbf{y}$ are two points in space. Shin et al. (2001) proposed to make the assumption that the receiver wavefields' contribution to the Hessian is negligible, resulting into a diagonal of the Hessian matrix obtained by a product of source wavefields; Guitton (2007) further proposed to smooth the product of the source wavefields for more stability. Plessix and Mulder (2004) show that the product of source wavefields approximation is valid under infinite and uniform receiver coverage; they therefore proposed to approximate the receiver wavefields by analytical formulae, derived for a constant velocity model, for a finite receiver coverage acquisition. Rickett (2003) and Guitton (2004) use a first migrated image to generate synthetic data, which are migrated to produce a second image; they note that the two images are related by an approximation of the Hessian matrix, thus allowing them to estimate the Hessian. Hou and Symes (2016) use ray theory approximation to derive a pseudo inverse of the (extended) Born modeling operator; they show that the inverse of the Born modeling operator can be approximated by a weighted version of RTM. I reach a similar conclusion here. In particular, I argue that, under some crude assumptions, the diagonal elements of the pseudo Hessian matrix can be approximated by a low pass filter of the RTM image. This implies that the low frequency background signal in conventional RTM imaging, traditionally regarded as noise (Zhang and Sun, 2009), can be exploited to accelerate waveform inversion.

In section I.2.3.1, I present a heuristic motivation for the proposed method based on the fact that an RTM image is a good measure of the illumination distribution in the subsurface for a given acquisition geometry. In section I.2.3.2, I attempt to provide a more formal justification for the proposed method; the objective of this part is to reveal the underlying assumptions of the proposed method. In section I.2.4, I challenge the proposed method with numerical experiments on the Marmousi model (Versteeg, 1994); both linearized inversion and nonlinear inversion are tested.

I.2.3 Theory

I.2.3.1 Heuristics

The RTM image, under the Born approximation, can be expressed as (e.g. Symes, 2008, see also the last equation in section I.1.5)

$$\begin{aligned} I(\mathbf{x}) &= \frac{1}{m_0(\mathbf{x})^3} \int_{\mathbf{s},\mathbf{r},t,\tau} \dot{G}(t-\tau,\mathbf{x},\mathbf{r})\dot{G}(\tau,\mathbf{x},\mathbf{s})D_{obs}(\mathbf{s},\mathbf{r},t) \\ &= \sum_{\mathbf{s},\mathbf{r},t} \mathbf{B}(\mathbf{x},\mathbf{s},\mathbf{r},t)D_{obs}(\mathbf{s},\mathbf{r},t) \\ &= \mathbf{B}^*\big[D_{obs}(\mathbf{s},\mathbf{r},t)\big], \end{aligned} \tag{41}$$

where $\mathbf{B}[\cdot] : I(\mathbf{x}) \rightarrow D_{obs}(\mathbf{s},\mathbf{r},t)$ is the Born modeling operator (see for example section I.1.1), the square brackets indicate a linear relationship and the asterisk symbol indicates the adjoint operation; $\dot{G}(t,\mathbf{x},\mathbf{s})$ denotes the time derivative of the Green's function of the acoustic wave equation for a source positioned at $\mathbf{s}$, and $D_{obs}(\mathbf{s},\mathbf{r},t)$ are the input data recorded at $\mathbf{r}$; $m_0(\mathbf{x})$ is the smooth background velocity model, and $I(\mathbf{x})$ is the output RTM image. According to equation 41, if we ignore the data term $D_{obs}(\mathbf{s},\mathbf{r},t)$ for the sake of argument, the RTM image is a measure of the illumination distribution in the subsurface as a function of acquisition geometry and the background velocity model. The effect of the data term can be thought of as multiplying the illumination map by a highly oscillatory signal, whose effect can be removed by smoothing the absolute value of the output RTM image.

As an example, consider the synthetic model in Figure 28.a, discretized on a regular grid (dx=20m, dz=10m). For a survey with 39 sources, 200 receivers evenly distributed at the surface, and a Ricker source wavelet with a dominant frequency of 20Hz, the raw RTM image (using non muted data) is shown in Figure 28.d. Smoothing the absolute value of RTM image removes its oscillatory component, resulting in the illumination map in Figure 28.e; this illumination map can be used to normalize the raw RTM image (Figure 28.f). Numerical experiments suggest that this illumination map can be used to accelerate gradient methods for waveform inversion. Figure 29 compares the convergence rate for the linearized inversion (least squares RTM) problem using a conjugate gradient solver (e.g. Shewchuk, 1994) with and without the proposed preconditioner; the preconditioned version shows a convergence rate about an order of magnitude higher. Figure 30 compares the convergence rate for the nonlinear waveform inversion (full waveform inversion) problem using a nonlinear conjugate gradient solver with and without the proposed preconditioner; the preconditioned version shows a convergence rate about two orders of magnitude higher.

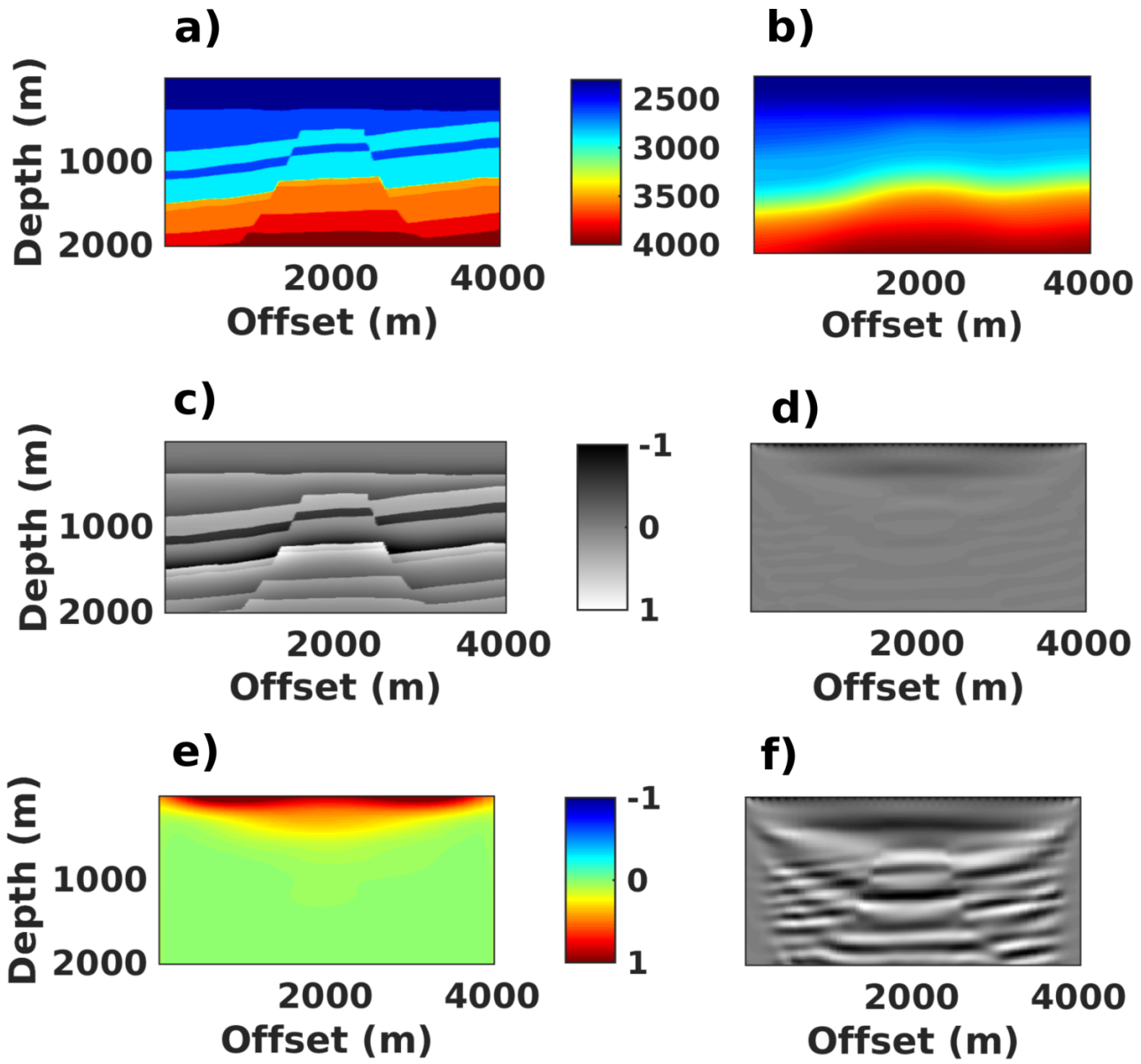

Figure 28: Synthetic model used for numerical experiments. **a**) Reference velocity model. **b**) Imaging model obtained by smoothing the reference model. **c**) Reference reflectivity obtained by taking the difference between the two previous models. **d**) Raw output of RTM. **e**) Illumination map estimated by smoothing the RTM image. **f**) The RTM image, normalized using the illumination map.

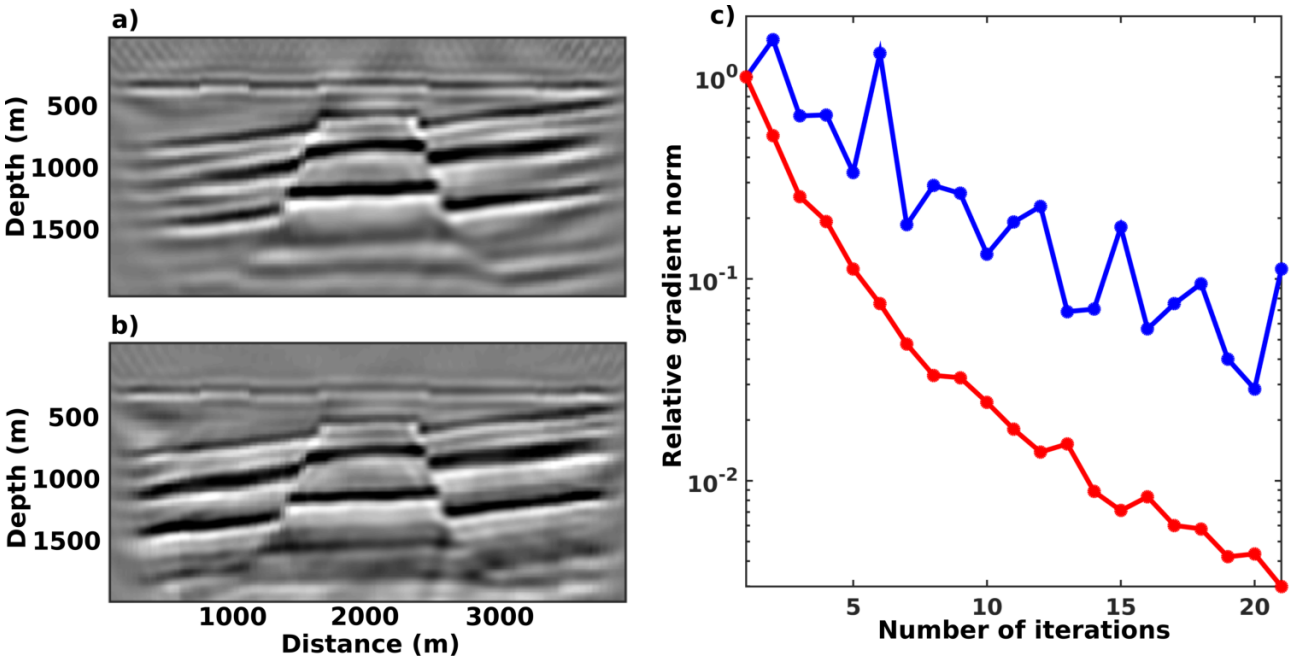

Figure 29: Convergence rate for linearized waveform inversion. **a**) Estimated reflectivity, non-preconditioned conjugate gradient. **b**) Estimated reflectivity, preconditioned conjugate gradient. **c**) Blue: non-preconditioned; red: preconditioned.

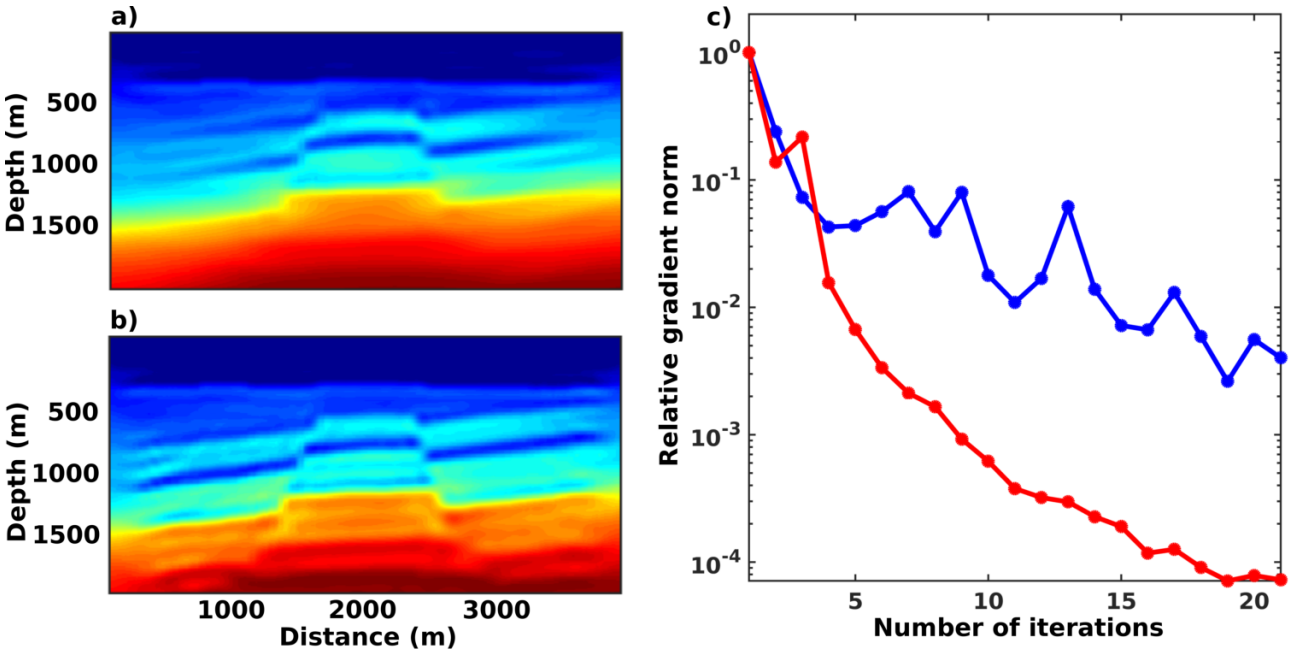

Figure 30: Convergence rate for nonlinear waveform inversion with data filtered (low pass) up to a 10Hz cutoff frequency. **a**) Estimated velocity, non-preconditioned nonlinear conjugate gradient. **b**) Estimated velocity, preconditioned nonlinear conjugate gradient. **c**) Blue: non-preconditioned; red: preconditioned.

I.2.3.2 Underlying assumptions

I now attempt to formally justify the proposition that, under suitable assumptions, the smoothed RTM image approximates the diagonal of the Hessian matrix. Assuming that the recorded data $D_{obs}(\mathbf{s}, \mathbf{r}, t)$ can be approximated by Born modeling, for some unknown reference reflectivity

image $I_{ref}(\mathbf{x})$, it follows from equation 41 that (Rickett, 2003; Guitton, 2004)

$$\begin{aligned} I(\mathbf{x}) &= \sum_{\mathbf{s},\mathbf{r},t} \mathbf{B}(\mathbf{x},\mathbf{s},\mathbf{r},t) D_{obs}(\mathbf{s},\mathbf{r},t) \\ &= \sum_{\mathbf{s},\mathbf{r},t} \mathbf{B}(\mathbf{x},\mathbf{s},\mathbf{r},t) \sum_{\mathbf{y}} \mathbf{B}(\mathbf{y},\mathbf{s},\mathbf{r},t) I_{ref}(\mathbf{y}) \\ &= \sum_{\mathbf{y}} \mathbf{H}(\mathbf{x},\mathbf{y}) I_{ref}(\mathbf{y}) \\ &= \mathbf{H}\big[I_{ref}(\mathbf{y})\big], \end{aligned} \tag{42}$$

where $\mathbf{H}$ is the approximate Hessian matrix. First, let us assume that the Hessian $\mathbf{H}$ can be approximated by its diagonal elements, denoted by $h(\mathbf{x})$; this reduces equation 42 to

$$I(\mathbf{x}) = h(\mathbf{x}) I_{ref}(\mathbf{x}). \tag{43}$$

According to equation 43, the RTM image is the product of the diagonal elements of the Hessian and the reference reflectivity. The idea will be to smooth out the oscillatory reference reflectivity. Let $\mathscr{N}(\mathbf{x})$ denote the set of points "near" $\mathbf{x}$, and let $f(\mathbf{x})$ denote a low-pass filter. Define the filtered version of equation 43 as

$$\bar{I}(\mathbf{x}) = \sum_{\mathbf{x}_i \in \mathscr{N}(\mathbf{x})} f(\mathbf{x}_i) I(\mathbf{x}_i) = \sum_{\mathbf{x}_i \in \mathscr{N}(\mathbf{x})} f(\mathbf{x}_i) h(\mathbf{x}_i) I_{ref}(\mathbf{x}_i). \tag{44}$$

If the diagonal elements $h(\mathbf{x})$ vary much slower than the (unknown) reference reflectivity $I_{ref}(\mathbf{x})$, $h(\mathbf{x})$ can be approximated by a quasi-constant function in $\mathscr{N}(\mathbf{x})$, leading to

$$I(\mathbf{x}) \approx \hat{h}(\mathbf{x}) \sum_{\mathbf{x}_i \in \mathscr{N}(\mathbf{x})} f(\mathbf{x}_i) I_{ref}(\mathbf{x}_i), \tag{45}$$

where $\hat{h}(\mathbf{x})$ is the average value of $h(\mathbf{x})$ in the neighborhood $\mathscr{N}(\mathbf{x})$. Finally, let us suppose that the reference reflectivity $I_{ref}(\mathbf{x})$ oscillates around some (unknown) average value, denoted by i_{ref}, and that the smoothed reference reflectivity $\bar{I}_{ref}(\mathbf{x})$ does not deviate too much from i_{ref}; it follows from 45 that

$$\begin{aligned} I(\mathbf{x}) &\approx \hat{h}(\mathbf{x}) \bar{I}_{ref}(\mathbf{x}), \\ &\approx i_{ref} \hat{h}(\mathbf{x}). \end{aligned} \tag{46}$$

The average value of $I_{ref}(\mathbf{x})$, i_{ref}, is likely to be zero in some cases; this is circumvented by first taking the absolute value of $I(\mathbf{x})$ before filtering. The unknown i_{ref} has no effect on image normalization since it is a constant. In summary, the smoothed RTM image approximates the Hessian under the following assumptions:

1. The Hessian $\mathbf{H}$ can be approximated by its diagonal elements, denoted by $h(\mathbf{x})$.
2. The diagonal elements $h(\mathbf{x})$ vary much slower than the (unknown) reference reflectivity $I_{ref}(\mathbf{x})$.
3. The reference reflectivity $I_{ref}(\mathbf{x})$ oscillates around some (unknown) average value, denoted by i_{ref}; furthermore, the smoothed reference reflectivity does not deviate too much from i_{ref}.

The first assumption is valid under a high frequency approximation (e.g. Miller et al., 1987); this may be a crude approximation in some cases but it has proven useful in practice (Plessix and Mulder, 2004). The third assumption is clearly reasonable under the Born approximation.

I.2.4 Numerical examples

I now test the proposed preconditioner on the portion of the Marmousi velocity model (Versteeg, 1994) shown in Figure 31a. The model is discretized on a grid of size 200×215 with $dx = 30m$ and $dz = 15m$. The model is used to generate synthetic data by finite difference time domain acoustic wave simulation (Virieux, 1984) with perfectly matched absorbing boundaries (Collino and Tsogka, 2001); the source is a Ricker wavelet with a dominant frequency of 15Hz. The input velocity model is also used to simulate the imaging velocity model shown in Figure 31b by smoothing. Figure 32 compares the results obtained by a multi-scale nonlinear waveform inversion (Bunks et al., 1995) without preconditioning (Figure 32a) and with preconditioning (Figure 32b). The preconditioned inversion in Figure 32b clearly converges much faster than the non-preconditioned version in Figure 32a.

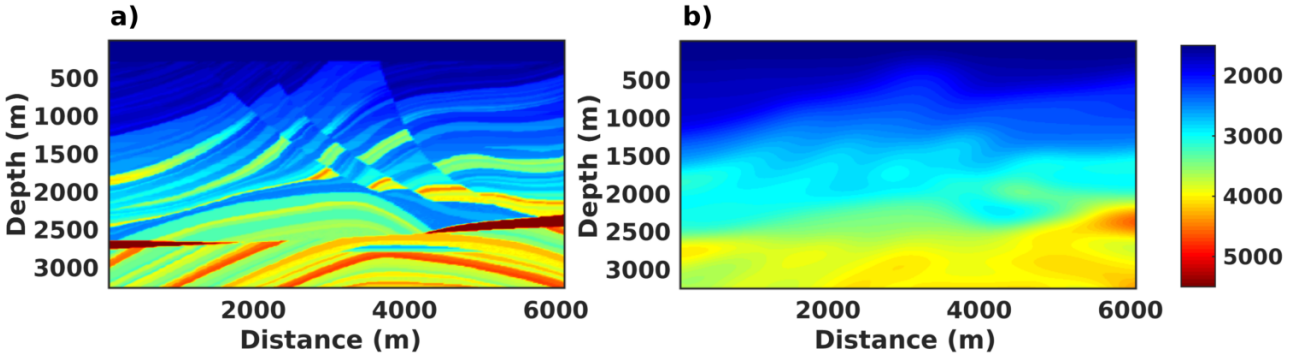

Figure 31: Marmousi velocity model (Versteeg, 1994). **a**) Reference velocity model. **b**) Imaging velocity model obtained by smoothing the reference model.

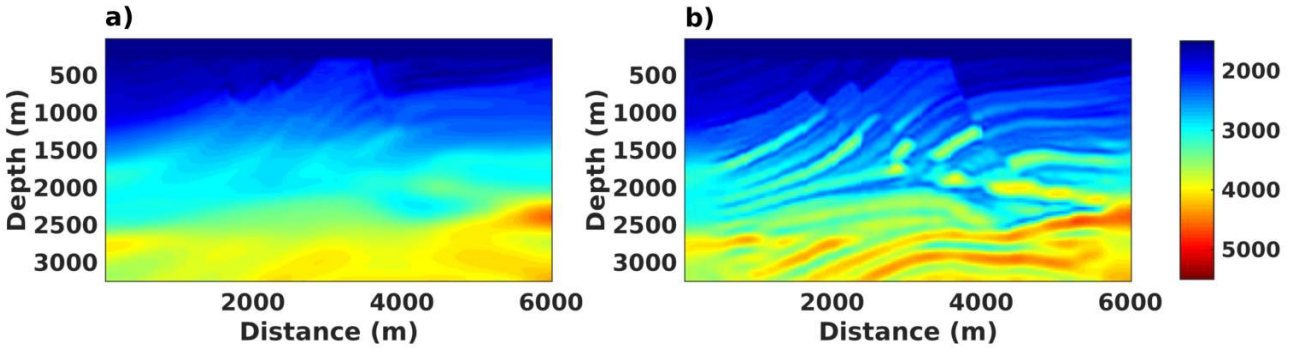

Figure 32: Nonlinear waveform inversion with data filtered (low pass) up to 6, 8, 10, and a 12 Hz cutoff frequency. The final velocity model inverted for one frequency is used as the initial velocity model for the next frequency. **a**) Estimated velocity, non-preconditioned conjugate gradient. **b**) Estimated velocity, preconditioned conjugate gradient.

The difference between the results obtained for linearized inversion without preconditioning (Figure 33a) and with preconditioning (Figure 33b) is less striking than that previously observed for nonlinear inversion (Figure 32). However, a look at the residual curves in Figure 33c reveals that the preconditioned linearized inversion converges about an order of magnitude faster than the non-preconditioned version.

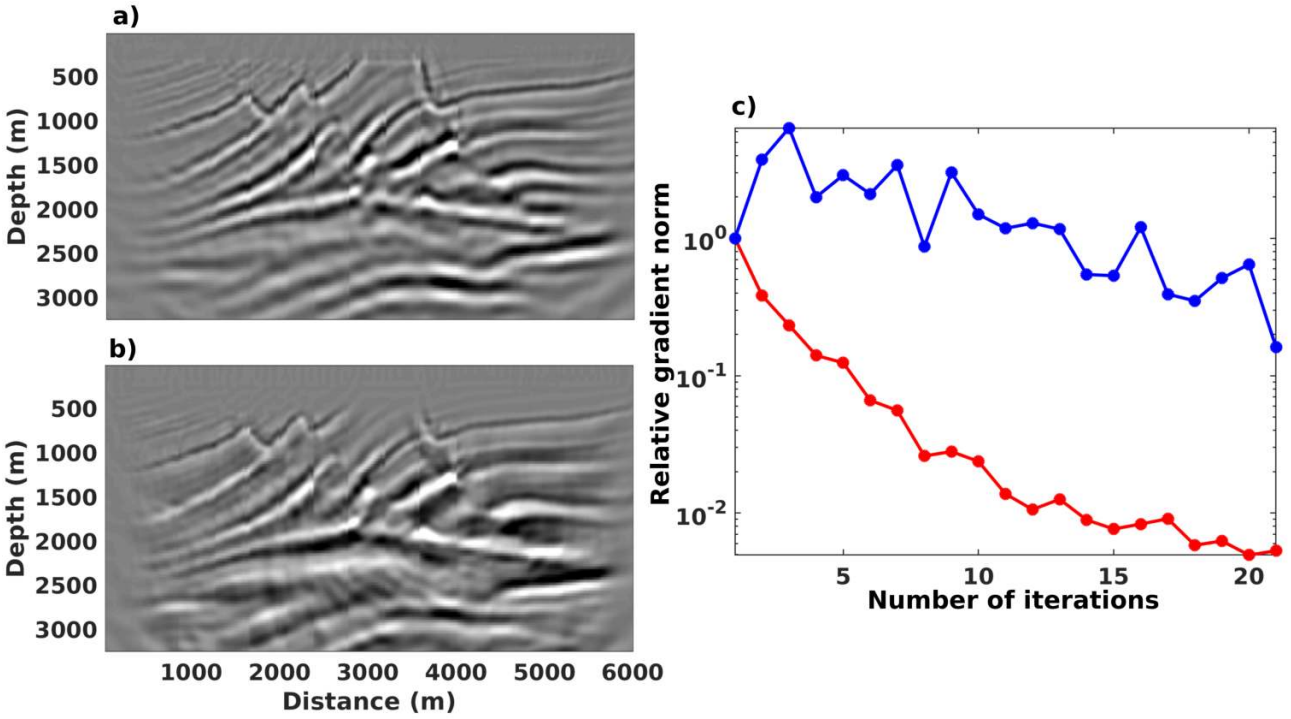

Figure 33: Linearized waveform inversion (least squares reverse time migration). **a**) Image obtained with non-preconditioned conjugate gradient. **b**) Image obtained with preconditioned conjugate gradient. **c**) Residual curves; blue: non-preconditioned, red: preconditioned.

I.2.5 Conclusion

Waveform inversion has proven to be a promising technology for quantitative seismic depth imaging. One of the challenges of waveform inversion is its high computational cost. A number of techniques have been proposed to accelerate the convergence rate of waveform inversion problems. In this paper, I propose another method to accelerate gradient based waveform inversion by using the low frequency background signal in reverse time migration (RTM) images as a preconditioner. Numerical experiments suggest that the proposed preconditioner accelarates notably both linearized and nonlinear waveform inversion. The success of the preconditioner is justified by showing that, under appropriate assumptions, the low frequency background RTM signal can approximate the diagonal elements of the Hessian matrix of the objective function.

I.2.6 Acknowledgements

A lot of what I know about seismic imaging comes from discussions with Daniel Rocha, Esteban Diaz, Ludovic Métivier, Paul Cupillard, and Paul Sava. My knowledge of seismic imaging has also greatly benefited from exchanges within the HIWAI project led by Yann Capdeville. I am thankful to Guillaume Caumon for having sent me to many EAGE and SEG seismic imaging conferences and workshops over the course of my PhD. I would also like to thank Melchior Schuh-senlis for many discussions about the algebraic manipulations in I.2.3.2.

Chapter I.3

Extended Kirchhoff Imaging for Migration Velocity Analysis

This chapter resulted from an ongoing technical report co-authored with Paul Sava and Thomas Beraud.

I.3.1 Summary

Extended images are a substantial component of velocity analysis. We investigate extensions of the Kirchhoff imaging operator that capture image distortions caused by inaccurate velocity. We propose a general formula that describes all data-domain extensions (e.g. offset and reflection angle extensions), as well as image-domain extensions (e.g. space lag and time lag extensions). Both single-arrival and multi-arrival imaging are discussed. We propose a method to handle multi-arrivals without shooting rays from the subsurface, thus reducing the computational cost. One practical advantage of our generalized extension formula is that it allows a unified implementation for all the extensions mentioned herein. Another advantage of the proposed generalized extension formula is that it leads to a unified differential semblance migration velocity analysis formulation for all the data-domain extensions mentioned herein.

I.3.2 Introduction

Kirchhoff extended images, also known as common image gathers, find applications in velocity analysis (Stork, 1992), imaging in complex media (Xu et al., 2001), and reservoir characterization (Fatti et al., 1994). In this chapter, extended images are grouped into two classes: data-domain extensions, and image-domain extensions. Data-domain extensions are those for which the extension parameter depends explicitly on data parameters, such as source-receiver geometry; these include shot extension, receiver extension, offset extension and reflection angle extension, and have been extensively used in ray based imaging and velocity analysis (Xu et al., 2001; Woodward et al., 2008). Image-domain extensions are those for which the extension parameters do not depend explicitly on data parameters; these include time-lag and space-lag extensions extensively used in wave-equation based imaging and velocity analysis (Sava and Fomel, 2006; Symes, 2008; Sava and Vasconcelos, 2010).

In this chapter, we propose a general Kirchhoff extension formula that describes all the extensions mentioned above. The generalized extension formula naturally leads to a unified implementation for constructing Kirchhoff extended images, and to a unified differential semblance (Symes and Carazzone, 1991; Mulder and ten Kroode, 2002) migration velocity analysis (MVA) formulation. First, we propose the general Kirchhoff extension formula for single-arrival imaging, then we propose a strategy to handle multi-arrivals without shooting rays from subsurface. We then apply the proposed extension formula within a typical ray-based MVA workflow (Stork, 1992; Woodward et al., 2008) to measure its suitability for velocity analysis. Finally, the proposed extension formula is used to formulate a differential semblance MVA that is simultaneously valid for multiple data-domain extensions.

I.3.3 Imaging with single arrivals

I.3.3.1 Data-domain extensions

The standard Kirchhoff imaging integral (e.g. Etgen et al., 2009) can be expressed as

$$\begin{aligned} I(\mathbf{x}) &= \int_{\mathbf{s},\mathbf{r},t} W(\mathbf{s},\mathbf{r},\mathbf{x})\delta\big[t - T(\mathbf{s},\mathbf{r},\mathbf{x})\big]\frac{\partial}{\partial t}D(\mathbf{s},\mathbf{r},t) \\ &= \int_{\mathbf{s},\mathbf{r}} W(\mathbf{s},\mathbf{r},\mathbf{x})\frac{\partial}{\partial t}D\big[\mathbf{s},\mathbf{r},T(\mathbf{s},\mathbf{r},\mathbf{x})\big] \\ &= \int_{\mathbf{s},\mathbf{r}} F_1(\mathbf{s},\mathbf{r},\mathbf{x}). \end{aligned} \tag{47}$$

In equation 47, $\mathbf{x}$ is an image point in the subsurface; $\mathbf{s}$ and $\mathbf{r}$ are source and receiver coordinates, assumed to be at the free surface; $T(\mathbf{s},\mathbf{r},\mathbf{x})$ is the traveltime of a ray shot from $\mathbf{s}$, scattered off $\mathbf{x}$, then recorded at $\mathbf{r}$; $D(\mathbf{s},\mathbf{r},t)$ are the input data; $W(\mathbf{s},\mathbf{r},\mathbf{x})$ are integration weights, whose physical meaning will become apparent when we discuss about image-domain extensions further below; $I(\mathbf{x})$ is the output image. In this chapter, we define *data-domain extensions* as those for which the extension parameter depends explicitly on data parameters $(\mathbf{s},\mathbf{r},\mathbf{x})$. Given this definition, all the data-domain extensions of equation 47 can be constructed using the generalized Radon transform

$$I(\mathbf{x},\beta) = \int_{\mathbf{s},\mathbf{r}} \delta\big[\beta - \zeta(\mathbf{s},\mathbf{r},\mathbf{x})\big]F_1(\mathbf{s},\mathbf{r},\mathbf{x}), \tag{48}$$

where $I(\mathbf{x},\beta)$ is the output extended image, and β is the extension parameter whose physical meaning depends on the user defined function $\zeta(\mathbf{s},\mathbf{r},\mathbf{x})$. β can be a scalar or a vector depending

on the definition of ζ. The advantage of equation 48 is that it allows multiple extensions to be constructed using the same kernel of code. For example, the shot extended image is obtained by defining

$$\zeta(\mathbf{s},\mathbf{r},\mathbf{x}) = \mathbf{s}; \tag{49}$$

the surface offset extended image is obtained by defining

$$\zeta(\mathbf{s},\mathbf{r},\mathbf{x}) = \frac{|\mathbf{s}-\mathbf{r}|}{2}; \tag{50}$$

and the reflection angle extended image is obtained by defining

$$\zeta(\mathbf{s},\mathbf{r},\mathbf{x}) = \text{acos}\left(\frac{\nabla T(\mathbf{x},\mathbf{s})\cdot\nabla T(\mathbf{x},\mathbf{r})}{\nabla T(\mathbf{x},\mathbf{s})\cdot\nabla T(\mathbf{x},\mathbf{s})}\right), \tag{51}$$

where $T(\mathbf{x},\mathbf{s})$ denotes first arrival traveltime maps from a source positioned at $\mathbf{s}$.

I.3.3.2 Image-domain extensions

The standard wavefield imaging integral (e.g. Symes, 2008) can be expressed as

$$I(\mathbf{x}) = \int_{\mathbf{s},\mathbf{r},t,t'} G(t',\mathbf{x},\mathbf{s})G(t-t',\mathbf{x},\mathbf{r})\frac{\partial}{\partial t}D(\mathbf{s},\mathbf{r},t), \tag{52}$$

where $G(t,\mathbf{x},\mathbf{s})$ denotes the Green's function of the acoustic wave equation with a source positioned at $\mathbf{s}$, and $D(\mathbf{s},\mathbf{r},t)$ are the input data. Following Sava and Vasconcelos (2010), we introduce the time-lag (scalar) parameter τ, and space-lag (vector) parameter $\boldsymbol{\lambda}$, and define an extension of equation 52 as

$$\begin{aligned} I(\mathbf{x},\boldsymbol{\lambda},\tau) &= \int_{\mathbf{s},\mathbf{r},t,t'} G\big[(t'-\tau),\mathbf{x}-\boldsymbol{\lambda},\mathbf{s}\big]\times \\ &\quad G\big[t-(t'+\tau),\mathbf{x}+\boldsymbol{\lambda},\mathbf{r}\big]\frac{\partial}{\partial t}D(\mathbf{s},\mathbf{r},t) \\ &= \int_{\mathbf{s},\mathbf{r},t,t'} G(t'-\tau,\mathbf{x}^-,\mathbf{s})\times \\ &\quad G(t-t'-\tau,\mathbf{x}^+,\mathbf{r})\frac{\partial}{\partial t}D(\mathbf{s},\mathbf{r},t). \end{aligned} \tag{53}$$

Using the ray approximation

$$G(t,\mathbf{x},\mathbf{s}) = A(\mathbf{x},\mathbf{s})\delta\big[t-T(\mathbf{x},\mathbf{s})\big], \tag{54}$$

where $A(\mathbf{x},\mathbf{s})$ and $T(\mathbf{x},\mathbf{s})$ represent the amplitude and traveltime of a ray from a source at $\mathbf{s}$, equation 53 leads to

$$\begin{aligned} I(\mathbf{x},\boldsymbol{\lambda},\tau) &= \int_{\mathbf{s},\mathbf{r},t,t'} A(\mathbf{x}^-,\mathbf{s})\delta\big[t'-\tau-T(\mathbf{x}^-,\mathbf{s})\big]\times \\ &\quad A(\mathbf{x}^+,\mathbf{r})\delta\big[t-t'-\tau-T(\mathbf{x}^+,\mathbf{r})\big]\frac{\partial}{\partial t}D(\mathbf{s},\mathbf{r},t) \\ &= \int_{\mathbf{s},\mathbf{r},t} A(\mathbf{x}^-,\mathbf{s})A(\mathbf{x}^+,\mathbf{r})\times \\ &\quad \delta\big[t-2\tau-T(\mathbf{x}^-,\mathbf{s})-T(\mathbf{x}^+,\mathbf{r})\big]\frac{\partial}{\partial t}D(\mathbf{s},\mathbf{r},t) \\ &= \int_{\mathbf{s},\mathbf{r}} W(\mathbf{s},\mathbf{r},\mathbf{x},\boldsymbol{\lambda})\frac{\partial}{\partial t}D\big[\mathbf{s},\mathbf{r},2\tau+T(\mathbf{s},\mathbf{r},\mathbf{x},\boldsymbol{\lambda})\big] \\ &= \int_{\mathbf{s},\mathbf{r}} F_2(\mathbf{s},\mathbf{r},\mathbf{x},\boldsymbol{\lambda},\tau). \end{aligned} \tag{55}$$

When $\boldsymbol{\lambda} = \mathbf{0}$ and $\tau = 0$, equation 55 reduces to equation 47. Based on equation 48 and equation 55, we propose the more general Kirchhoff extension formula

$$I(\mathbf{x}, \beta, \boldsymbol{\lambda}, \tau) = \int_{\mathbf{s},\mathbf{r}} \delta\big[\beta - \zeta(\mathbf{s}, \mathbf{r}, \mathbf{x}, \boldsymbol{\lambda})\big] F_2(\mathbf{s}, \mathbf{r}, \mathbf{x}, \boldsymbol{\lambda}, \tau). \tag{56}$$

I.3.4 Imaging with multiple arrivals

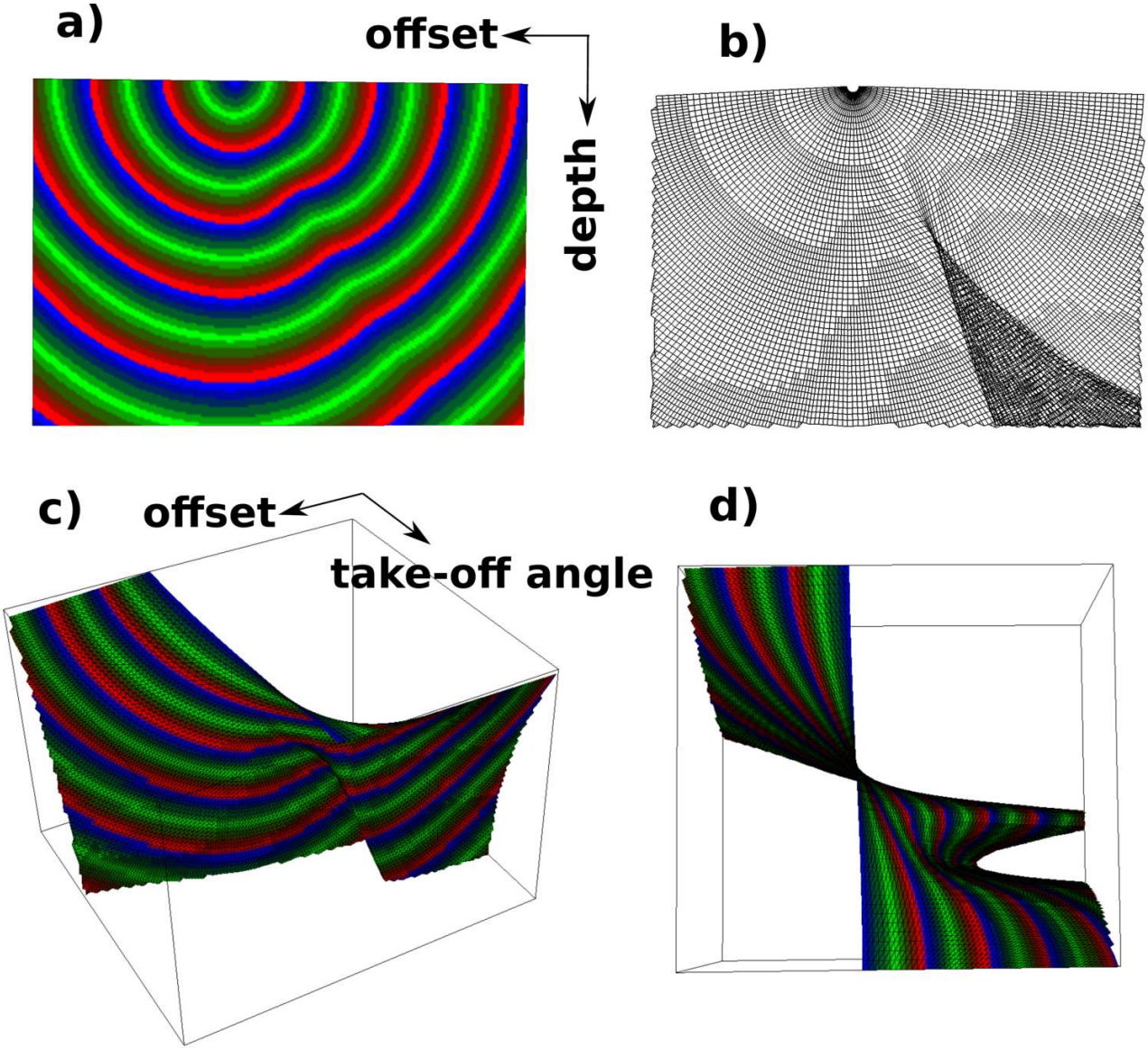

Figure 34: Traveltimes computed in the model of Figure 35b. **a**) First arrival traveltimes section computed by fast marching (Sethian and Popovici, 1999). **b**) Multi-valued traveltimes section computed by wavefront construction (Vinje et al., 1993); only ray cells are displayed. **c**), **d**) Single-valued traveltimes surfaces obtained by expressing the multi-valued traveltimes section **b**) in (offset,depth,take-off angle) coordinates.

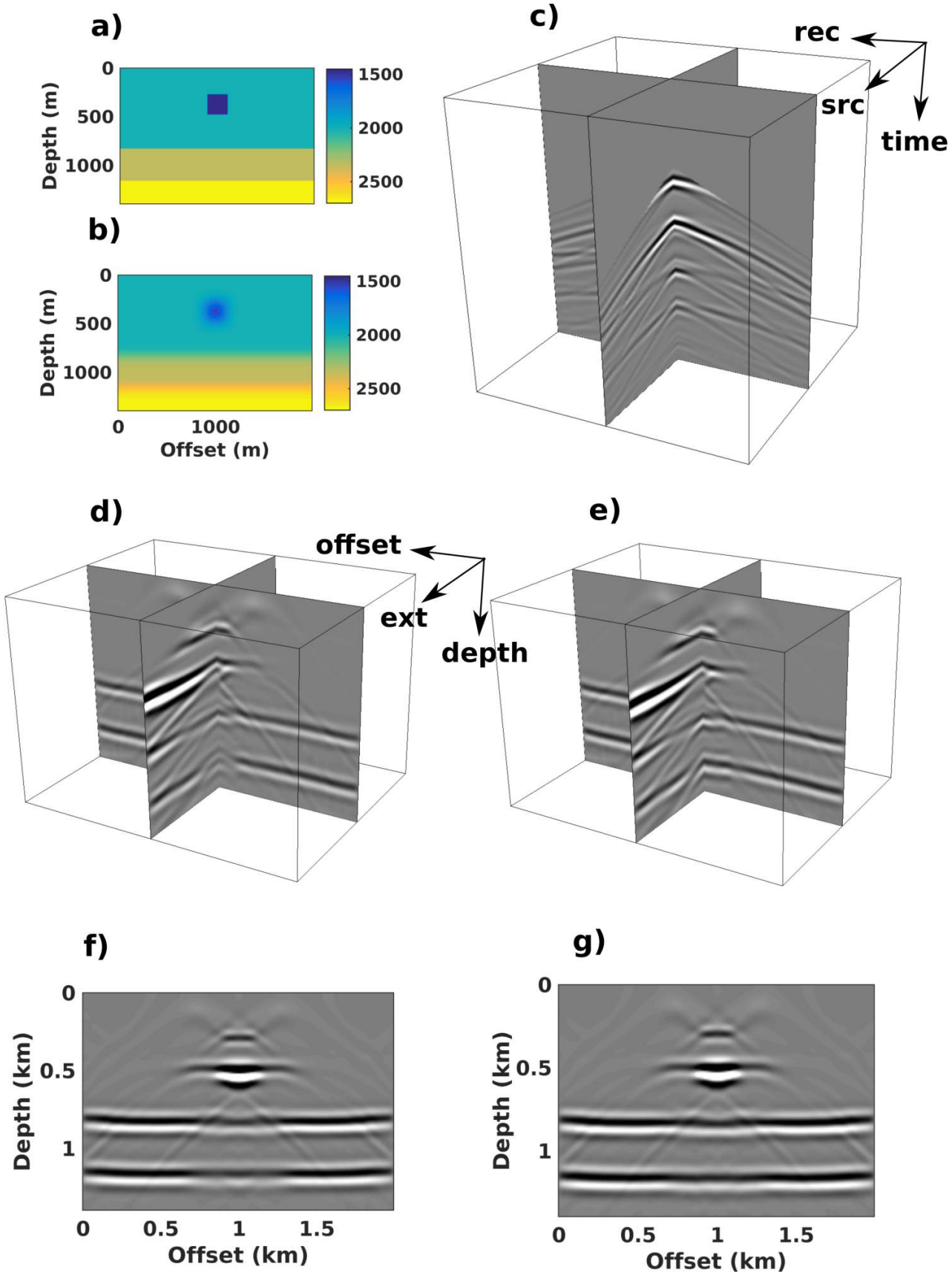

Figure 35: Illustrative example for offset extension (i.e. using equation 50). **a**) Reference velocity model. **b**) Migration model obtained by smoothing the reference model. **c**) Synthetic data computed in the reference model; sources and receivers cover the free surface. **d**) Offset extension using first arrivals. **e**) Offset extension using multiple arrivals. **f**) First arrival image. **g**) Multi-arrival image. An improved continuity of reflectors below the strong velocity anomaly is observed in the multi-arrival image.

Let $\mathbf{x} = (x, y, z)$ denote a point in 3D; the traveltime map $T(\mathbf{x}, \mathbf{s})$, from a shot positioned at $\mathbf{s}$, is a cube that can be multi-valued when the velocity field has strong spatial variations. Let us denote the two ray take-off angles (dip and strike) by $\boldsymbol{\phi} = (\phi_1, \phi_2)$, and introduce the 5D coordinates $\mathbf{x}^* = (\mathbf{x}, \boldsymbol{\phi})$. $\mathbf{x}^*$ will be referred to as phase-space coordinates (e.g. Cerveny, 2001; Bashkardin, 2014), and $\mathbf{x}$ as physical-space coordinates. The phase-space traveltime map $T(\mathbf{x}^*, \mathbf{s})$ is always single-valued; this concept is easier to illustrate in 2D as shown in Figure 34. Instead of projecting the single-valued phase-space traveltime map $T(\mathbf{x}^*, \mathbf{s})$ to the lower dimensional multi-valued physical-space traveltime map $T(\mathbf{x}, \mathbf{s})$, which is later re-interpolated during the imaging loop (e.g. Xu and Lambaré, 2004), we propose to interpolate ray information from $T(\mathbf{x}^*, \mathbf{s})$ directly during the imaging loop. That is, we propose the multi-valued version of equation 56 as

$$\begin{aligned} I(\mathbf{x}, \beta, \boldsymbol{\lambda}, \tau) \;\; = \;\; & \int_{\mathbf{s},\mathbf{r},\boldsymbol{\phi}} \delta\big[\beta - \zeta(\mathbf{s}, \mathbf{r}, \mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\phi})\big] W(\mathbf{s}, \mathbf{r}, \mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\phi}) \times \\ & \frac{\partial}{\partial t} D\big[\mathbf{s}, \mathbf{r}, 2\tau + T(\mathbf{s}, \mathbf{r}, \mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\phi})\big]. \end{aligned} \tag{57}$$

In practice, the traveltime map $T(\mathbf{x}^*, \mathbf{s})$ is discretized into a set of cells $\{\mathbf{C}_i(\mathbf{x}^*, \mathbf{s})\}$. We loop over the cells in $\{\mathbf{C}_i(\mathbf{x}^*, \mathbf{s})\}$ implicitly through grid points $\mathbf{x}$. We are allowed to do this because the order in which the cells are visited is irrelevant. A pseudo-code to implement equation 57 is given in algorithm 1.

```
for each source s do
    for each receiver r do
        for each image-point x do
            x⁻ = x − λ;
            x⁺ = x + λ;
            for C_s ∈ {C(x*_j, s) | x*_j = (x⁻, φ_j)} do
                for C_r ∈ {C(x*_k, r) | x*_k = (x⁺, φ_k)} do
                    β = ζ(C_s, C_r);
                    t = T(C_s) + T(C_r) + 2τ;
                    I(x, β, λ, τ) += A(C_s)A(C_r) ∂/∂t D(s, r, t);
                end
            end
        end
    end
end
```

Algorithm 1: A general pseudo-code to construct Kirchhoff extensions.

$\{\mathbf{C}(\mathbf{x}_i^*, \mathbf{s}) \,|\, \mathbf{x}^* = (\mathbf{x}, \boldsymbol{\phi}_i)\}$ is the set of cells that belong to a phase-space traveltime map $T(\mathbf{x}^*, \mathbf{s})$ from a shot positioned at $\mathbf{s}$ and whose projection in the physical-space contains the point $\mathbf{x}$. The notation $f(\mathbf{C})$ means that the value of $f(\mathbf{x}, \mathbf{s})$ is estimated by interpolation from the cell $\mathbf{C}(\mathbf{x}, \boldsymbol{\phi}, \mathbf{s})$. A 2D example comparing single-arrival with multi-arrival imaging is shown in Figure 35.

I.3.5 Velocity model building via ray tracing

We now test the ability of the proposed extended imaging operators (equations 56 and 57) to capture image distortion caused by inaccurate velocity by an application to ray-based MVA (Stork, 1992; Woodward et al., 2008). We use the simple velocity model in Figure 36a as the true model, in which we generate "observed" data by finite-difference acoustic modeling (Virieux, 1984) with perfectly matched absorbing boundaries (Collino and Tsogka, 2001). Sources and receivers cover the free surface. We then attempt to build a migration velocity model starting from the initial velocity model in Figure 36b.

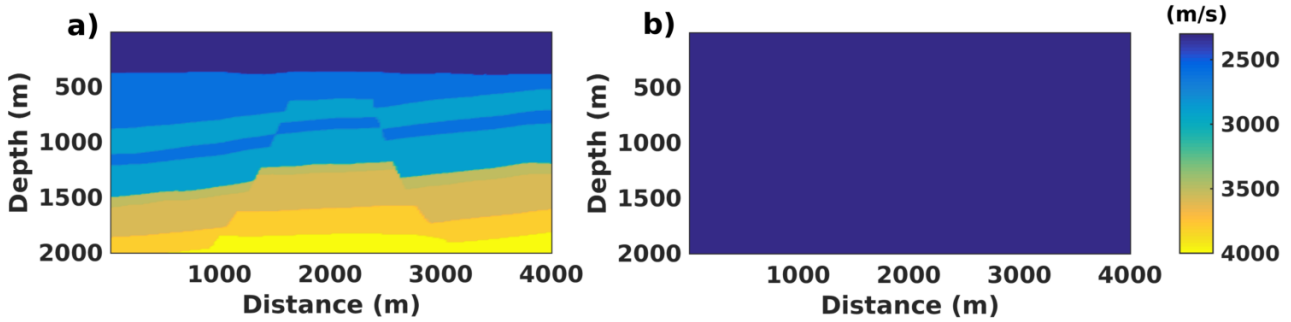

Figure 36: **a**) Reference velocity model. **b**) Initial velocity model for ray-based migration velocity analysis.

Figure 37a shows the output of equation 56 using the initial model (Figure 36b), with $\tau = 0$, $\boldsymbol{\lambda} = \mathbf{0}$, and ζ as defined in equation 50. For this simple example, residual moveout curves can be picked automatically by parabola fitting as shown in Figure 37b. Rays are then traced from each residual moveout curve towards the surface (Figure 37c), making it possible to build a ray tomographic system which can then be inverted to update the velocity model. Figure 37d shows the velocity model obtained after 15 iterations; the tomographic system is regularized using the smoothing operators proposed in chapter II.1 and in Irakarama et al. (2018).

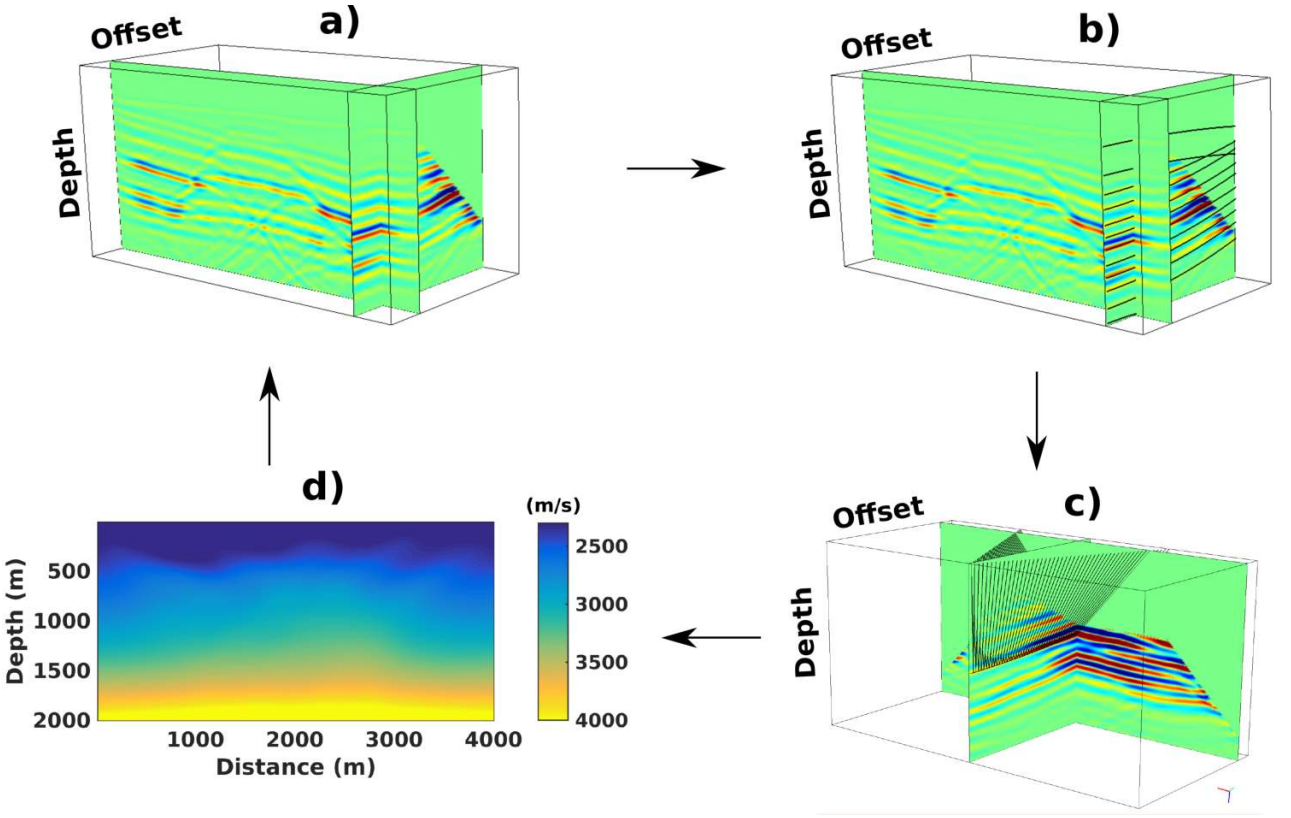

Figure 37: Typical ray-based migration velocity analysis workflow (Stork, 1992; Woodward et al., 2008). **a**) Offset extended image using the initial velocity model (Figure 36b). **b**) Residual moveout picking by parabola fitting. **c**) Ray-tracing from residual moveout curves towards the surface. **d**) Updated model after 15 iterations.

Figure 38 compares common image gathers computed in the initial model (Figure 36b) with common image gathers computed in the updated model (Figure 37d). We clearly note that the initial gathers (Figure 38a) are flattened (Figure 38b) by the updated velocity model. Figure

39 compares the migrated image obtained using the initial velocity model (Figure 39a) with the migrated image obtained using the updated velocity model (Figure 39b). The migrated image resulting from the updated model is more focused and has less artifacts. The analysis done in this section confirms that the proposed extended imaging operators can indeed be used for velocity modeling.

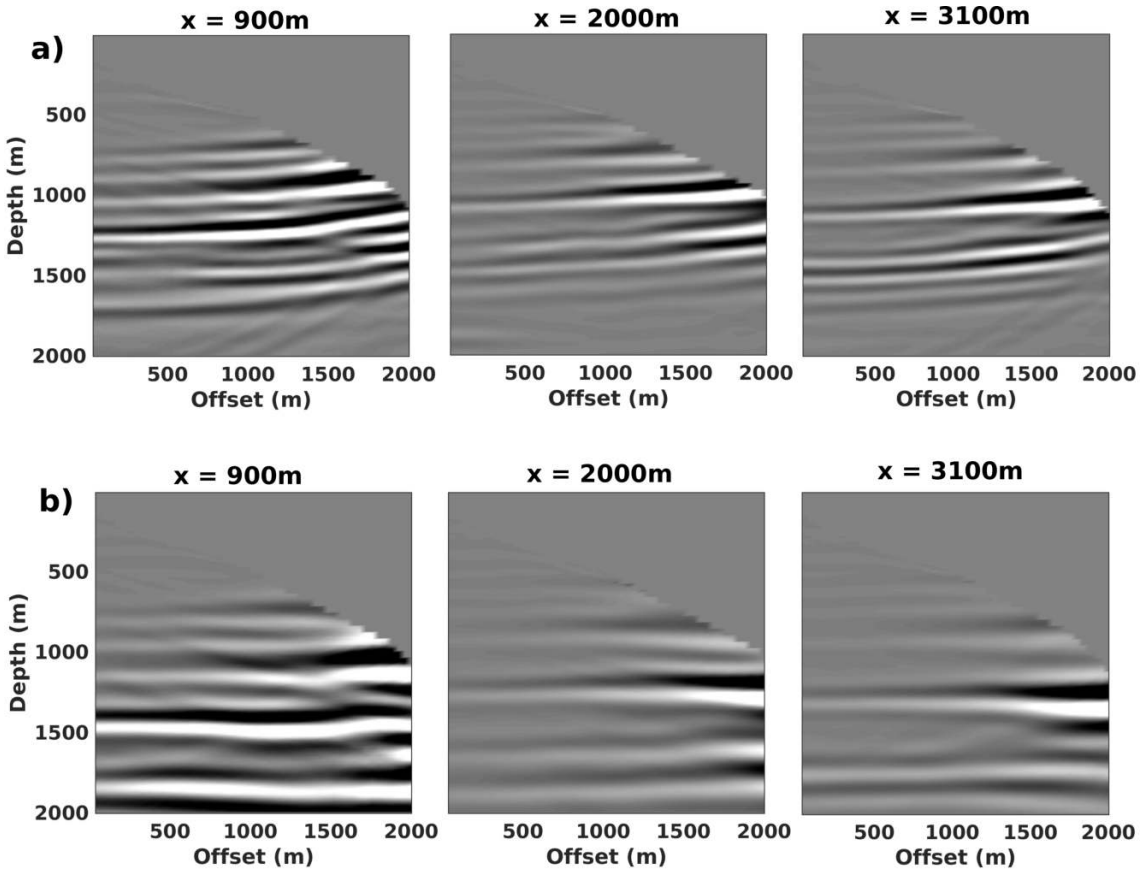

Figure 38: **a**) Common image gathers obtained using the initial velocity model (Figure 36b). **b**) Common image gathers obtained using the updated velocity model (Figure 37d).

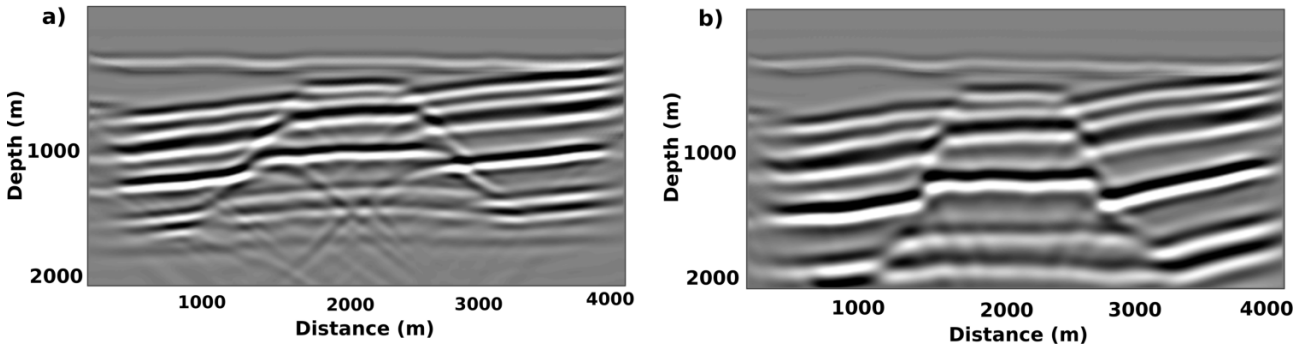

Figure 39: **a**) Migrated image obtained using the initial velocity model (Figure 36b). **b**) Migrated image obtained using the updated velocity model (Figure 37d).

I.3.6 Velocity model building via the adjoint-state method

In this section, we investigate using the proposed generalized extended imaging operators to develop a general gradient for gradient-based MVA. Velocity modeling can also be formulated

as an optimization problem (e.g. Mulder and ten Kroode, 2002); we consider such a formulation in this section using a differential semblance objective function similar to that of Plessix (2006). For simplicity, we only investigate velocity modeling using the single arrival data-domain extended imaging equation 48. The velocity modeling problem in this framework can be stated as

$$\left\{\begin{array}{l} m(\mathbf{x}) = \underset{m(\mathbf{x})}{\operatorname{argmin}}\, \Phi(m) = \underset{m(\mathbf{x})}{\operatorname{argmin}}\, \sum_{\mathbf{x},\beta} \big[I(\mathbf{x},\beta,m) - I(\mathbf{x},\beta+1,m)\big]^2 \\ \text{subject to} \\ I(\mathbf{x},\beta,m) = \int_{\mathbf{s},\mathbf{r}} \delta\big[\beta - \zeta(\mathbf{s},\mathbf{r},\mathbf{x},m)\big]\dot{D}\big[\mathbf{s},\mathbf{r},T(\mathbf{s},\mathbf{r},\mathbf{x})\big] \\ T(\mathbf{s},\mathbf{r},\mathbf{x},m) = T_s(\mathbf{x},m) + T_r(\mathbf{x},m) \\ \big[\nabla T_s(\mathbf{x},m)\big]^2 = \frac{1}{m(\mathbf{x})^2},\, T_s(\mathbf{s},m) \\ \big[\nabla T_r(\mathbf{x},m)\big]^2 = \frac{1}{m(\mathbf{x})^2},\, T_r(\mathbf{r},m) \end{array}\right. , \tag{58}$$

where the notation $\big[I(\mathbf{x},\beta,m) - I(\mathbf{x},\beta+1,m)\big]$ denotes a difference of two neihgbouring (partial) images along the β-axis. The problem in equation 58 can be solved using the adjoint-state method (e.g. Plessix, 2006). Other authors that have used the adjoint-state method and an eikonal solver for velocity modeling include Taillandier et al. (2009) who used the adjoint-state method to solve a first-arrival traveltime tomography problem, and Tavakoli et al. (2017) who used the adjoint-state method to solve a slope tomography problem. Here we consider using the adjoint-state method to solve a differential semblance ray-based MVA problem as done in Mulder and ten Kroode (2002). Our approach differs from that of Mulder and ten Kroode (2002) in two ways. First, we use an eikonal solver (Sethian and Popovici, 1999) to compute traveltimes instead of a ray-tracer (Vinje et al., 1993). Second, because we use the generalized data-domain extended imaging equation 48, our result is simultaneously valid for multiple data-domain extensions. In what follows, we omit some arguments in expressions in equation 58 to simplify notation. The Lagrangian of the problem in equation 58 is given by

$$\begin{aligned} \mathscr{L} &= \Big\langle 1, \big[I(\beta) - I(\beta+1)\big]^2 \Big\rangle_{\mathbf{x},\beta} \\ &+ \Big\langle \lambda_1, I(\beta) - \int_{\mathbf{s},\mathbf{r}} \delta\big[\beta - \zeta(\mathbf{s},\mathbf{r})\big]\dot{D}(\mathbf{s},\mathbf{r}) \Big\rangle_{\mathbf{x},\beta} \\ &+ \Big\langle \lambda_2, \big[\nabla T_s\big]^2 - \frac{1}{m^2} \Big\rangle_{\mathbf{x},\mathbf{s}} \\ &+ \Big\langle \lambda_3, \big[\nabla T_r\big]^2 - \frac{1}{m^2} \Big\rangle_{\mathbf{x},\mathbf{r}} . \end{aligned} \tag{59}$$

The derivative of the Lagrangian with respect to the model is given by (see the appendix, section I.3.10, for more details)

$$\begin{aligned} \frac{\partial}{\partial_m}\mathscr{L} &= \Big\langle 2\big[-I(\beta+1) + 2I(\beta) - I(\beta-1)\big], \frac{\partial}{\partial_m} I(\beta) \Big\rangle_{\mathbf{x},\beta} \\ &+ \Big\langle \lambda_1, \frac{\partial}{\partial_m} I(\beta) \Big\rangle_{\mathbf{x},\beta} - \Big\langle \lambda_1, \int_{\mathbf{s},\mathbf{r}} \delta\big[\beta - \zeta\big]\ddot{D}\frac{\partial}{\partial_m} T_s \Big\rangle_{\mathbf{x},\beta} - \Big\langle \lambda_1, \int_{\mathbf{s},\mathbf{r}} \delta\big[\beta - \zeta\big]\ddot{D}\frac{\partial}{\partial_m} T_r \Big\rangle_{\mathbf{x},\beta} \\ &+ \Big\langle \lambda_2, 2\nabla T_s \cdot \nabla\big[\frac{\partial}{\partial_m} T_s\big] \Big\rangle_{\mathbf{x},\mathbf{s}} + \Big\langle \lambda_2, \frac{2}{m^3} \Big\rangle_{\mathbf{x},\mathbf{s}} \\ &+ \Big\langle \lambda_3, 2\nabla T_r \cdot \nabla\big[\frac{\partial}{\partial_m} T_r\big] \Big\rangle_{\mathbf{x},\mathbf{r}} + \Big\langle \lambda_3, \frac{2}{m^3} \Big\rangle_{\mathbf{x},\mathbf{r}} . \end{aligned} \tag{60}$$

The divergence theorem implies that

$$\left\langle \bar{F}(\mathbf{x}), \nabla g(x) \right\rangle_{\mathbf{x}} = \left\langle \nabla \cdot \bar{F}(\mathbf{x}), g(x) \right\rangle_{\mathbf{x}}, \tag{61}$$

for some vector valued function $\bar{F}$ and a scalar valued function g (we have assumed that g or $\bar{F}$ is zero at boundaries). Using equation 61, equation 60 can be written as

$$\begin{aligned}
\frac{\partial}{\partial_m}\mathscr{L} &= \left\langle \lambda_1 + 2\big[-I(\beta+1) + 2I(\beta) - I(\beta-1)\big], \frac{\partial}{\partial_m} I(\beta) \right\rangle_{\mathbf{x},\beta} \\
&+ \left\langle 2\nabla\cdot\big[\lambda_2 \nabla T_s\big] - \int_{\mathbf{r},\beta} \lambda_1 \delta\big[\beta-\zeta\big]\ddot{D}, \frac{\partial}{\partial_m} T_s \right\rangle_{\mathbf{x},\mathbf{s}} + \left\langle \lambda_2, \frac{2}{m^3} \right\rangle_{\mathbf{x},\mathbf{s}} \\
&+ \left\langle 2\nabla\cdot\big[\lambda_3 \nabla T_r\big] - \int_{\mathbf{s},\beta} \lambda_1 \delta\big[\beta-\zeta\big]\ddot{D}, \frac{\partial}{\partial_m} T_r \right\rangle_{\mathbf{x},\mathbf{r}} + \left\langle \lambda_3, \frac{2}{m^3} \right\rangle_{\mathbf{x},\mathbf{r}},
\end{aligned}$$

leading to

$$\begin{cases}
\lambda_1 = 2\big[I(\beta+1) - 2I(\beta) + I(\beta-1)\big] \\
2\nabla\cdot\big[\lambda_2\nabla T_s\big] = \int_{\mathbf{r},\beta} \lambda_1 \delta\big[\beta-\zeta\big]\ddot{D} \\
2\nabla\cdot\big[\lambda_3\nabla T_r\big] = \int_{\mathbf{s},\beta} \lambda_1 \delta\big[\beta-\zeta\big]\ddot{D} \\
\frac{\partial}{\partial m}\mathscr{L} = \big\langle \lambda_2, \frac{2}{m^3}\big\rangle_{\mathbf{x},\mathbf{s}} + \big\langle \lambda_3, \frac{2}{m^3}\big\rangle_{\mathbf{x},\mathbf{r}}
\end{cases}. \tag{62}$$

The gradient in equation 62 allows to solve the problem in equation 58 using an iterative algorithm of the form

$$\begin{cases}
\big[\nabla T_s^n\big]^2 = \frac{1}{m_n^2} & (a) \\
\big[\nabla T_r^n\big]^2 = \frac{1}{m_n^2} & (b) \\
I^n(\beta) = \int_{\mathbf{s},\mathbf{r}} \delta\big[\beta-\zeta\big]\dot{D}(\mathbf{s},\mathbf{r},T_s^n + T_r^n) & (c) \\
\lambda_1^n = 2\big[I^n(\beta+1) - 2I^n(\beta) + I^n(\beta-1)\big] & (d) \\
2\nabla\cdot\big[\lambda_2^n\nabla T_s^n\big] = \int_{\mathbf{r},\beta} \lambda_1^n \delta\big[\beta-\zeta\big]\ddot{D} & (e) \\
2\nabla\cdot\big[\lambda_3^n\nabla T_r^n\big] = \int_{\mathbf{s},\beta} \lambda_1^n \delta\big[\beta-\zeta\big]\ddot{D} & (f) \\
\Delta m_n = \big\langle \lambda_2^n, \frac{2}{m_n^3}\big\rangle_{\mathbf{s}} + \big\langle \lambda_3^n, \frac{2}{m_n^3}\big\rangle_{\mathbf{r}} & (g) \\
m_{n+1} = m_n + \Delta m_n & (h)
\end{cases}. \tag{63}$$

Different definitions of ζ in equation 63 give different types of MVA; for example, equation 49 leads to shot-based MVA, equation 50 leads to (surface) offset-based MVA, and equation 51 leads to (reflection) angle-based MVA.

I.3.7 Discussion

The multivalued imaging equation 57 is included merely for completeness. In practice, the single valued version of the same equation is much easier to implement and gives comparable results. We found the multivalued imaging formula to perform considerably better only for idealized examples such as that in Figure 35. For more realistic models, like the Marmousi (Versteeg, 1994), for example, the multivalued formula did not perform considerably better

than its single valued version. The challenge in implementing the multivalued imaging formula 57 comes from implementing efficiently the expression

$$\{\mathbf{C}(\mathbf{x}_i^*, \mathbf{s}) \,|\, \mathbf{x}^* = (\mathbf{x}, \boldsymbol{\phi}_i)\}$$

in algorithm 1; that is, given a point $\mathbf{x}$, find the set of ray-cells that belong to a phase-space traveltime map $T(\mathbf{x}^*, \mathbf{s})$ from a shot positioned at $\mathbf{s}$, and such that the projection of each ray-cell in the physical-space contains the point $\mathbf{x}$.

The discussion in section I.3.6, particularly the algorithm in equation 63, has not yet been tested[1]. Therefore, at the time of writing this manuscript, we are unable to comment on how well the algorithm in equation 63 would work in practice. There are, however, some familiar expressions in equation 63: the expression in equation 63d is similar to that obtained by Plessix (2006) for shot extended wave-based differential semblance MVA, and the expression in equation 63g is similar to that obtained by Tavakoli et al. (2017). What is new in equation 63 are, to our knowledge, the expressions on the right-hand side of equations 63c, 63e, and 63f.

I.3.8 Conclusion

Image extension is a key component of modern velocity analysis and imaging. In wave-based velocity analysis, extended imaging is mainly dominated by image-domain extensions such as time-lag and space-lag extensions. In ray-based velocity analysis, extended imaging is mainly dominated by data-domain extensions such as offset and reflection angle extensions. In this paper, we propose a general Kirchhoff extension formula that handles all the extensions in a similar unified way, and both single-arrival and multi-arrival imaging are discussed. A practical advantage of the proposed method is that all the extensions are constructed using the same piece of code, only input parameters change for different extensions. Similarly, we propose a differential semblance MVA formulation that is simultaneously valid for multiple data-domain extensions.

I.3.9 Acknowledgements

Modeste Irakarama would like to thank Guillaume Caumon and Paul Cupillard for their encouragement and the permission to work on this project. Modeste Irakarama would also like to thank Melchior Schuh-senlis and Mustapha Zakari for many discussions about the algebraic manipulations presented in the appendix. Modeste Irakarama and Thomas Beraud acknowledge the industrial and academic sponsors of the RING-GOCAD Consortium managed by ASGA for their support.

[1]The discussion in section I.3.5 was implemented and tested by Thomas Beraud as part of his master's project. If we had more time, implementing and testing the discussion in section I.3.6 would have made another interesting master's project.

I.3.10 Appendix

The first term in equation 59 is familiar from Plessix (2006). The derivative of the second term in equation 59 leads to

$$\begin{aligned}
\frac{\partial}{\partial_m}\int_{\mathbf{s},\mathbf{r}}\delta\big[\beta-\zeta\big]\dot{D} &= \int_{\mathbf{s},\mathbf{r}}\dot{D}\frac{\partial}{\partial_m}\delta\big[\beta-\zeta\big]+\int_{\mathbf{s},\mathbf{r}}\delta\big[\beta-\zeta\big]\frac{\partial}{\partial_m}\dot{D}\\
&= \int_{\mathbf{s},\mathbf{r}}\dot{D}\frac{\partial}{\partial_m}\eta\frac{\partial}{\partial_\eta}\delta(\eta)+\int_{\mathbf{s},\mathbf{r}}\delta\big[\beta-\zeta\big]\ddot{D}\frac{\partial}{\partial_m}T,\text{ with }\eta=\beta-\zeta\\
&= \int_{\mathbf{s},\mathbf{r}}\dot{D}\frac{\partial}{\partial_m}\eta\frac{\partial}{\partial_\eta}\lim_{\sigma\to 0}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{\eta^2}{2\sigma^2}}+\int_{\mathbf{s},\mathbf{r}}\delta\big[\beta-\zeta\big]\ddot{D}\frac{\partial}{\partial_m}T\\
&= \int_{\mathbf{s},\mathbf{r}}\dot{D}\frac{\partial}{\partial_m}\eta\lim_{\sigma\to 0}\frac{-\eta}{\sigma^3\sqrt{2\pi}}e^{\frac{-\eta^2}{2\sigma^2}}+\int_{\mathbf{s},\mathbf{r}}\delta\big[\beta-\zeta\big]\ddot{D}\frac{\partial}{\partial_m}T\\
&= \int_{\mathbf{s},\mathbf{r}}\dot{D}\frac{\partial}{\partial_m}\zeta\lim_{\sigma\to 0}\frac{(\beta-\zeta)}{\sigma^3\sqrt{2\pi}}e^{\frac{-(\beta-\zeta)^2}{2\sigma^2}}+\int_{\mathbf{s},\mathbf{r}}\delta\big[\beta-\zeta\big]\ddot{D}\frac{\partial}{\partial_m}T. \qquad (64)
\end{aligned}$$

We are only interested in (partial) images that occur when $\beta=\zeta$ (see equation 48), at which point the first term in equation 64 vanishes. The rest of the terms in equation 59 are familiar from Taillandier et al. (2009); Tavakoli et al. (2017).

Bibliography

V. Bashkardin. *Phase-Space Imaging of Reflection Seismic Data*. PhD thesis, University of Texas at Austin, Texas, USA, 2014.

E. Baysal, D. Kosloff, and J. Sherwood. Reverse time migration. *GEOPHYSICS*, 48:1514–1524, 1983.

B. Biondi. *3D Seismic Imaging*. Investigations in GEOPHYSICS. Society of Exploration Geophysicists, 2006.

N. Bleistein, J. W. S. Jr., and J. K. Cohen. *Mathematics of Multidimensional Seismic Imaging, Migration, and Inversion*. Interdisciplinary Applied Mathematics. Springer New York, 2001.

C. Bunks, F. Saleck, S. Zaleski, and G. Chavent. Multiscale seismic waveform inversion. *GEOPHYSICS*, 60:1457–1473, 1995.

V. Cerveny. *Seismic ray theory*. Cambridge University Press, 2001.

J. Claerbout. Toward a unified theory of reflector mapping. *GEOPHYSICS*, 36(3), 1971.

J. Claerbout. *Imaging the Earth's interior*. Blackwell Science Inc, 1985.

J. F. Claerbout. *EARTH SOUNDINGS ANALYSIS: Processing versus Inversion*. Stanford University, 2004.

F. Collino and C. Tsogka. Application of perfectly machted absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. *GEOPHYSICS*, 66(1), 2001.

J. Etgen, S. Gray, and Y. Zhang. An overview of depth migration in exploration geophysics. *GEOPHYSICS*, 74(6), 2009.

J. Fatti, G. Smith, P. Vail, P. Strauss, and P. Levitt. Detection of gas in sandstone reservoirs using AVO analysis: A 3-D seismic case history using the Geostack technique. *GEOPHYSICS*, 59:1362–1376, 1994.

A. Fichtner, H. Bunge, and H. Igel. The adjoint method is seismology I. Theory. *Physics of the Earth and Planetary Interiors*, 157:86–104, 2006.

J. Gazdag. Wave equation migration with the accurate space derivative method. *Geophysical Prospecting*, 28:60–70, 1980.

A. Guitton. Amplitude and kinematic corrections of migrated images for nonunitary imaging operators. *GEOPHYSICS*, 69:1017–1024, 2004.

A. Guitton. Smoothing imaging condition for shot-profile migration. *GEOPHYSICS*, 72:S149–S154, 2007.

J. Hou and W. Symes. Accelerating extended least-squares migration with weighted conjugate gradient iteration. *GEOPHYSICS*, 81:S165–S179, 2016.

M. Irakarama, G. Laurent, J. Renaudeau, and G. Caumon. Finite Difference Implicit Modeling of Geological Structures. In *80th EAGE Conference and Exhibition*, Copenhagen, Denmark, 2018.

D. Kosloff and E. Baysal. Migration with the full acoustic wave equation. *GEOPHYSICS*, 48: 677–687, 1983.

G. McMechan. Migration by extrapolation of time-dependent boundary values. *Geophysical Prospecting*, 31:413–420, 1983.

D. Miller, M. Oristaglio, and G. Beylkin. A new slant on seismic imaging: migration and integral geometry. *GEOPHYSICS*, 52:943–964, 1987.

W. Mulder and R.-E. Plessix. A comparison between one-way and two-way wave-equation migration. *GEOPHYSICS*, 69(6), 2004.

W. Mulder and A. ten Kroode. Automatic velocity analysis by differential semblance optimization. *GEOPHYSICS*, 67(4), 2002.

T. Nemeth, ChengjunWu, and G. Schuster. Least-squares migration of incomplete reflection data. *GEOPHYSICS*, 64:208–221, 1999.

J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2006.

R. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167:495–503, 2006.

R.-E. Plessix and W. Mulder. Frequency-domain finite-difference amplitude-preserving migration. *Geophys. J. Int.*, 157:975–987, 2004.

R. Pratt, C. Shin, and G. Hicks. Gauss-newton and full newton methods in frequency-space seismic waveform inversion. *Geophys. J. Int.*, 131:341–362, 1998.

J. Rickett. Illuminationbased normalization for waveequation depth migration. *GEOPHYSICS*, 68:1371–1379, 2003.

D. Rocha, P. Sava, and A. Guitton. 3D acoustic least-squares reverse time migration using the energy norm. *GEOPHYSICS*, 83:S261–S270, 2018.

P. Sava. A comparative review of wavefield tomography methods. Technical report, Colorado School of Mines, 2014.

P. Sava and S. Fomel. Time-shift imaging condition in seismic migration. *GEOPHYSICS*, 71 (6), 2006.

P. Sava and I. Vasconcelos. Extended imaging conditions for wave-equation migration. *Geophysical Prospecting*, 59:35–55, 2010.

W. Schneider. Integral formulation for migration in two and three dimensions. *GEOPHYSICS*, 43:49–76, 1978.

J. Sethian and A. Popovici. 3d traveltime computation using the fast marching method. *GEOPHYSICS*, 64(2), 1999.

J. R. Shewchuk. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Technical report, Carnegie Mellon University, 1994.

C. Shin, S. Jang, and D.-J. Min. Improved amplitude preservation for prestack depth migration by inverse scattering theory. *Geophysical Prospecting*, 49:592–606, 2001.

C. Stork. Reflection tomography in the postmigrated domain. *GEOPHYSICS*, 57:680–692, 1992.

W. Symes. Migration velocity analysis and waveform inversion. *Geophysical Prospecting*, 56: 765–790, 2008.

W. Symes and J. Carazzone. Velocity inversion by differential semblance optimization. *GEOPHYSICS*, 56:654–663, 1991.

C. Taillandier, M. Noble, H. Chauris, and H. Calandra. First-arrival traveltime tomography based on the adjoint-state method. *GEOPHYSICS*, 74(6), 2009.

A. Tarantola. Linearized inversion of seismic reflection data. *Geophysical Prospecting*, 32: 998–1015, 1984a.

A. Tarantola. Inversion of seismic reflection data in the acoustic approximation. *GEOPHYSICS*, 49:1259–1266, 1984b.

B. Tavakoli, S. Operto, A. Ribodetti, and J. Virieux. Slope tomography based on eikonal solvers and the adjoint-state method. *Geophys. J. Int.*, 209:1629–1647, 2017.

P. Thierry, S. ephane Operto, and G. L. e. Fast 2-D ray+Born migration/inversion in complex media. *GEOPHYSICS*, 64:162–181, 1999.

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

V. Vinje, E. Iversen, and H. Gjoystdal. Traveltime and amplitude estimation using wavefront construction. *GEOPHYSICS*, 58(8):1157–1166, 1993.

J. Virieux. SH-wave propagation in heterogeneous media: Velocity-stress finite-difference method. *GEOPHYSICS*, 49:1933–1942, 1984.

J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *GEOPHYSICS*, 74(6), 2009.

M. Woodward, D. Nichols, O. Zdraveva, P. Whitfield, and T. Johns. A decade of tomography. *GEOPHYSICS*, 73(5), 2008.

S. Xu and G. Lambaré. Fast migration/inversion with multivalued rayfields: Part I-Method, validation test, and application in 2D to Marmousi. *GEOPHYSICS*, 69(5):1311–1319, 2004.

S. Xu, H. Chauris, G. Lambaré, and M. Noble. Common-angle migration: A strategy for imaging complex media. *GEOPHYSICS*, 66(6), 2001.

Y. Zhang and J. Sun. Practical issues in reverse time migration: true amplitude gathers, noise removal and harmonic source encoding. *First Break*, 27:53–59, 2009.

Y. Zhang, L. Duan, and Y. Xie. A stable and practical implementation of least-squares reverse time migration. *GEOPHYSICS*, 80:V23–V31, 2015.

Part II

Notes on Structural Modeling

Introductory Comments

This second Part of the thesis is devoted to structural modeling, particularly implicit structural modeling (e.g. Chilès et al., 2004; Caumon et al., 2013). The global implicit structural modeling workflow is illustrated in Figure 40: first, input data are interpolated to build a stratigraphic function; then, horizons are extracted from isovalues of the stratigraphic function. In this manuscript, we only discuss the interpolation part, referred to as structural interpolation hereafter.

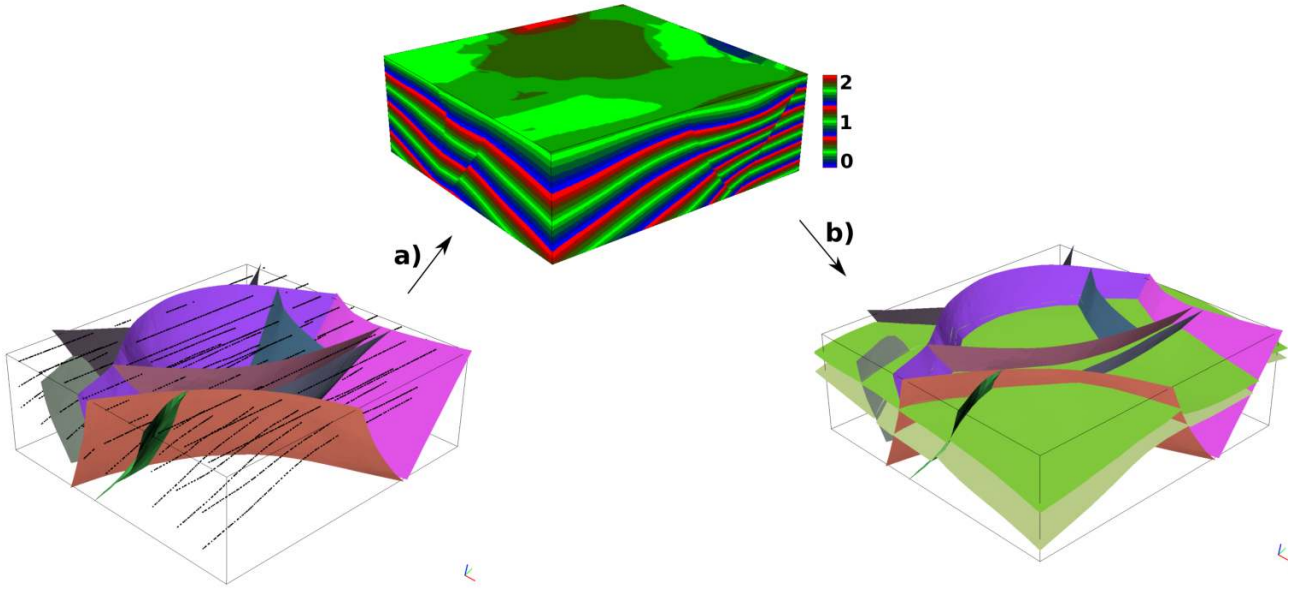

Figure 40: Implicit structural modeling workflow. **a**) Interpolation of input data to build a stratigraphic function. **b**) Extraction of horizons from the isovalues of the stratigraphic function.

Structural interpolation is usually more challenging than the standard scattered data interpolation (e.g. Turk and O'Brien, 2002; Anjyo et al., 2014). First of all, structural interpolation has to account for the presence of discontinuities (faults, unconformities, ...) in the most general case. But even in the absence of discontinuities, structural interpolation has additional challenges compared to scattered data interpolation. This is because the output of structural interpolation is a stratigraphic function and is therefore subjected to some geological constraints. The two benchmark models of Renaudeau et al. (2017a) and Renaudeau et al. (2017b) in Figure 41 are good examples to illustrate some of these challenges. Figures 42a and 43a show the results of the scattered data interpolation problem

$$\phi^*(\mathbf{x}) = \underset{\phi(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{x}} \left\{ \left[\mathscr{R}\phi(\mathbf{x})\right]^2 + \delta(\mathbf{x}-\mathbf{x}_i)\left[\phi(\mathbf{x}) - \phi_o(\mathbf{x}_i)\right]^2 \right\} \tag{65}$$

where $\mathscr{R}$ is a linear operator that measures roughness, and $\phi_o(\mathbf{x}_i)$ denotes data observed at the point $\mathbf{x}_i$. These results are not suited for implicit structural modeling. The stratigraphic function represents geological time, so it can not have saddle points; this condition is not satisfied

by the function obtained in Figure 42a. Furthermore, geological time should increase/decrease monotonically between two successive horizons; this condition is not satisfied by the function obtained in Figure 43a. The more suitable structural interpolation problem may be stated as

$$\begin{cases} \phi^*(\mathbf{x}) = \underset{\phi(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{x}} \left\{ \left[\mathscr{R}\phi(\mathbf{x})\right]^2 + \delta(\mathbf{x}-\mathbf{x}_i)\left[\phi(\mathbf{x}) - \phi_o(\mathbf{x}_i)\right]^2 \right\} & (a) \\ \text{subject to} & \\ \text{minimal variation of } \nabla\phi^*(\mathbf{x}) \text{ away from observations} & (b) \\ ||\nabla\phi^*(\mathbf{x})|| > 0 \text{ for all } \mathbf{x} & (c) \\ \frac{d}{dl}\phi^*(\mathbf{x}) \text{ is monotonic along any line joining two successive horizons} & (d) \end{cases} . \quad (66)$$

The constraint in equation 66b follows from geological field mapping practices. Some geological constraints may be nonlinear. In practice however, we often try to enforce the nonlinear geological constraints using some heuristic linear constraints. For example, the constraint in equation 66c (equation 66b as well) may be satisfied by using an appropriate roughness operator $\mathscr{R}$; this was the case for the result shown in Figure 42b where we use a roughness operator proposed in Irakarama et al. (2018) and in chapter II.1) instead of the Laplacian roughness operator (e.g. Briggs, 1974; Aster et al., 2004) used in Figure 42a. As for the constraint in equation 66d, Renaudeau et al. (2017b) propose to use a weighted roughness operator where the roughness operator has small weights near data points to allow for rapid variation of the stratigraphic function near data points. Here we explore two new alternative solutions that explicitly take into account the orientation of geological structures via linear constraints, as detailed in chapter II.3; such methods can be quite successful as illustrated in Figure 43b.

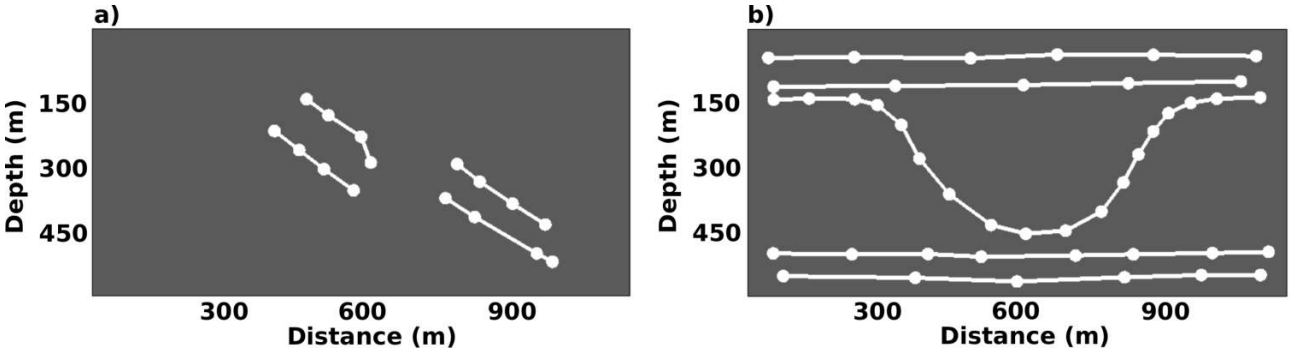

Figure 41: Benchmark models; the white points represent input data. **a**) Benchmark model proposed by Renaudeau et al. (2017a). **b**) Benchmark model proposed by Renaudeau et al. (2017b).

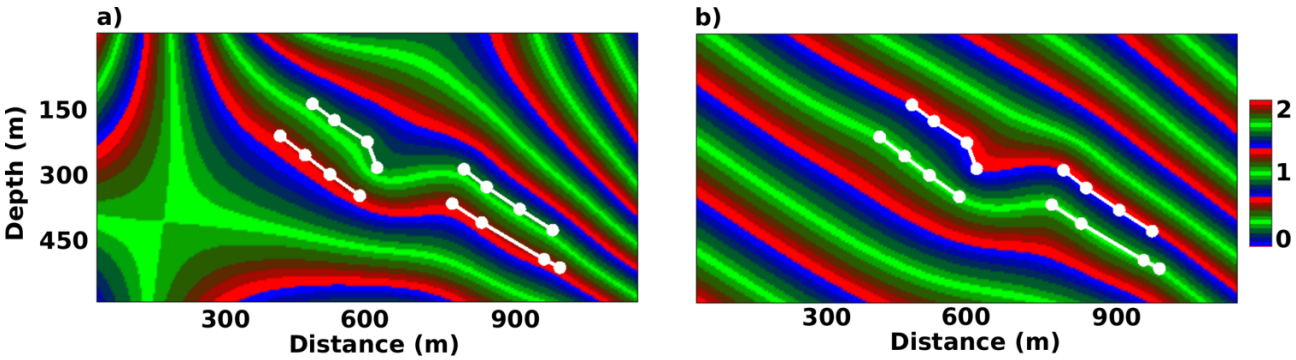

Figure 42: Interpolation of the data in Figure 41a. **a**) Results obtained using the Laplacian roughness operator. **b**) Results obtained using a more appropriate roughness operator for structural modeling.

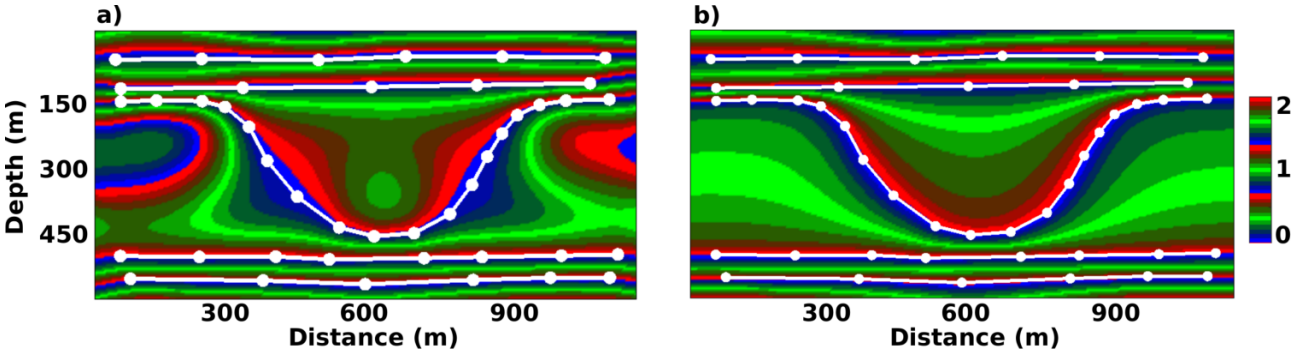

Figure 43: Interpolation of the data in Figure 41b. **a**) Results obtained using an isotropic roughness operator. **b**) Results obtained when we explicitly take into account the orientation of horizons.

Chapter II.1

Finite difference implicit structural modeling of geological structures

This chapter resulted from a technical report co-authored with Laurent Gautier, Julien Renaudeau, and Guillaume Caumon.

II.1.1 Summary

We introduce a new method for implicit structural modeling. The main development in this chapter are the new regularization operators we propose by extending inherent properties of the classic 1D discrete second derivative operator to higher dimensions. The proposed regularization operators discretize very naturally on the Cartesian grid using finite differences, owing to the highly symmetric nature of the Cartesian grid. Furthermore, the proposed regularization operators do not require any special treatment on boundary nodes, and their generalization to higher dimensions is straightforward. As a result, the proposed method has the advantage of being simple to implement. Numerical examples show that the proposed method is robust and numerically efficient.

II.1.2 Introduction

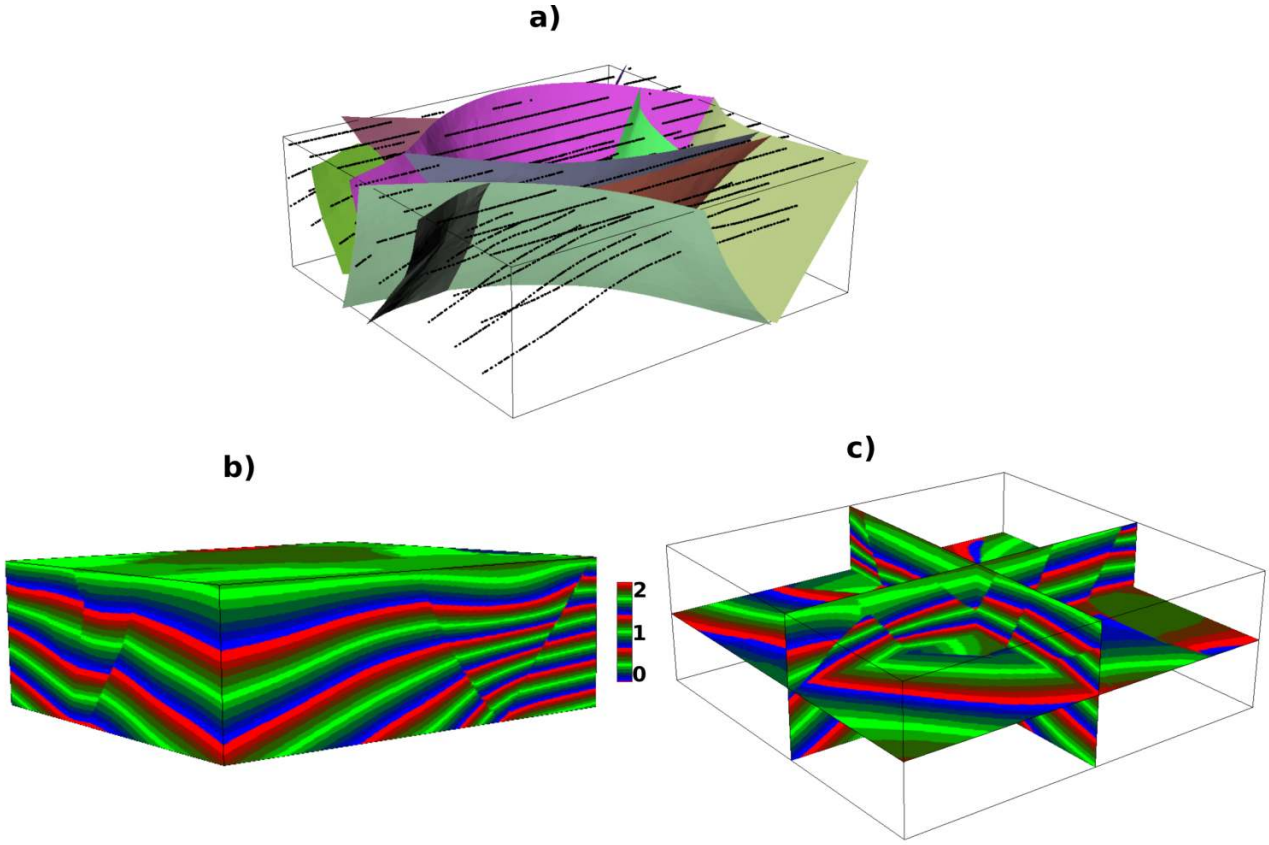

Figure 44: Implicit stratigraphic field computed using the proposed method. **a**) Input data consisting of 10 fault surfaces and horizon points picked along 3 horizons. **b**) Solid view of the resulting stratigraphic field. **c**) Section view of the resulting stratigraphic field. Data courtesy of Total.

Structural implicit modeling, in this chapter, is defined as the interpolation of randomly distributed, and possibly sparse, *structural data*. For simplicity, structural data, also referred to as *structural constraints*, will be limited to the following:

$$\text{value constraints} \quad : \quad \phi(\mathbf{x}_j) = a_j, \tag{67}$$

$$\text{orientation constraints} \quad : \quad \nabla\phi(\mathbf{x}_j) = ||\nabla\phi(\mathbf{x}_j)||\mathbf{u}_j; \tag{68}$$

where $\phi(\mathbf{x})$ is the unknown function to be interpolated, $\mathbf{x}$ is a point in space, a is some given scalar, and $\mathbf{u}$ is some given unit vector. We refer to equation 68 as the *normal form* of an orientation constraint, as opposed to the *tangential form*. Let $\mathbf{u}^1$ be a given unit vector, then there are $N-1$ unit vectors $\{\mathbf{u}^l\}_{l=2}^{l=N}$ such that $\{\mathbf{u}^l\}_{l=1}^{l=N}$ forms an orthogonal basis in $N \geq 2$ dimensions. In that case, the tangential form of the normal orientation constraint $\nabla\phi(\mathbf{x}) = ||\nabla\phi(\mathbf{x})||\mathbf{u}^1$ is

$$\mathbf{u}^l \cdot \nabla\phi(\mathbf{x}) = 0\,, \text{ for } l = 2, ..., N. \tag{69}$$

In implicit structural modeling, the object being modeled is obtained by extracting a hypersurface along an iso-value of the interpolated function $\phi(\mathbf{x})$.

Implicit structural modeling has extensively been used in geosciences for contour mapping (Briggs, 1974; Mallet, 1984; Smith and Wessel, 1990; Wessel and Bercovici, 1998), and for 3D geological modeling (Lajaunie et al., 1997; Mallet, 2004). The method presented here was developed mainly for geological modeling applications, but it is suited for contour mapping as

well. Most of the groundwork of modern structural implicit modeling was developed in the 2000's (Cowan et al., 2002; Ledez, 2003; Chilès et al., 2004; Moyen, 2005; Tertois, 2007), starting with the pioneering work of Lajaunie et al. (1997), to the theoretical framework developed by Mallet (2004), all the way to the more application-ready works of Chilès et al. (2004) and Frank (2007). Structural implicit modeling can be separated into two main classes (Caumon et al., 2013): (1) methods based on dual Kriging and radial basis interpolation (Lajaunie et al., 1997; Cowan et al., 2002; Chilès et al., 2004; Calcagno et al., 2008), (2) and methods based on domain discretization (Mallet, 2004; Frank, 2007; Caumon et al., 2013; Souche et al., 2013). Domain discretization methods are collectively referred to as *Discrete Smooth Interpolation* (DSI) methods hereafter (Mallet, 1989, 1992, 1997). The principle of DSI methods is to discretize all structural constraints on a discrete domain, and assemble them into a least-squares system of linear equations supplemented with smoothing regularization constraints (Mallet, 1989, 1992, 1997).

Implicit geological modeling is still an active area of research. Some authors have shown that existing methods still present limitations caused by the smoothing approaches and the discretization schemes currently used (Laurent, 2016; Renaudeau et al., 2017b). In addition, recent work has started investigating stochastic modeling approaches of geological structures geometry (Jessell et al., 2014; Cherpeau and Caumon, 2015; Godefroy et al., 2017) for which structural interpolation is becoming a bottleneck. Computation times that used to be acceptable for building a single "best model" are becoming far too long when considering stochastic approaches for sampling uncertainty. Other recent research and advances in implicit modeling include better modeling of folds (Laurent et al., 2016; Grose et al., 2017), automated building of models that conform to seismic data (Wu, 2017), more numerically efficient discretization schemes (Renaudeau et al., 2018), and many more (Mallet, 2014; Hillier et al., 2014; Gonçalves et al., 2017; Martin and Boisvert, 2017; Renaudeau et al., 2019). As we move towards an era of multi-realization structural modeling (Caumon, 2010), new challenges continuously emerge and motivate the quest for more robust and more efficient structural implicit modeling schemes.

In this chapter, we introduce a new approach based on finite differences; this method belongs to the DSI class of methods. Most DSI methods, as applied in geological modeling, rely on simplices (triangles in 2D, tetrahedra in 3D) for domain discretization due to their geometrical flexibility. Because simplicial meshes are unstructured and irregular, it is customary to discretize structural constraints and smoothing constraints using a piecewise linear approach (Frank, 2007; Caumon et al., 2013). The method we introduce is based on discretizing the domain on Cartesian grids, and on discretizing structural constraints and smoothing constraints using finite differences. Cartesian gridding and finite differences are arguably the simplest discretizations possible for structural interpolation; as a result, the proposed method is also easy to implement. Modern computing capabilities make the method numerically efficient as well. A 3D implicit structural modeling example using the proposed method is shown in Figure 44.

Implicit structural interpolation on Cartesian grids using finite differences is not new. Briggs (1974) solves the biharmoinc equation in 2D directly on the Carstesian grid using finite differences. In Briggs (1974)'s formulation, assignment constraints (equation 67) are treated as internal boundary conditions. Briggs also shows the equivalence between the biharmonic equation and the minimization of the global curvature of the implicit function. Wu and Hale (2015) use finite differences operators on Cartesian grids to create implicit functions on seismic images. Assuming that (1) the normal of reflectors can be estimated at each point of the seismic image, and that (2) the implicit function increases monotonically with seismic traveltime (or depth), they use the normal form of orientation constraints (equation 68) to assemble an overdetermined system solved using least-squares. However, it is usually challenging to make an accurate estimation of reflector dips at each grid point due to coherent noise in seismic data (Chauris

et al., 2002; Fehmers and Höcker, 2003). As for the second assumption, it is not always valid; the section from the Ribaute model (Caumon et al., 2009) in Figure 45 is an example of where this assumption does not hold. Wu (2017) addresses this problem by first unfaulting and unfolding the seismic image before computing the implicit function. Mallet (1989) also proposed an interpolation method on Cartesian grids using finite differences; the method proposed here resolves some issues raised by Mallet (1989), as discussed in the section about discontinuities (section II.1.7). Furthermore, unlike Mallet (1989), the method proposed here does not require input data to be located at grid points.

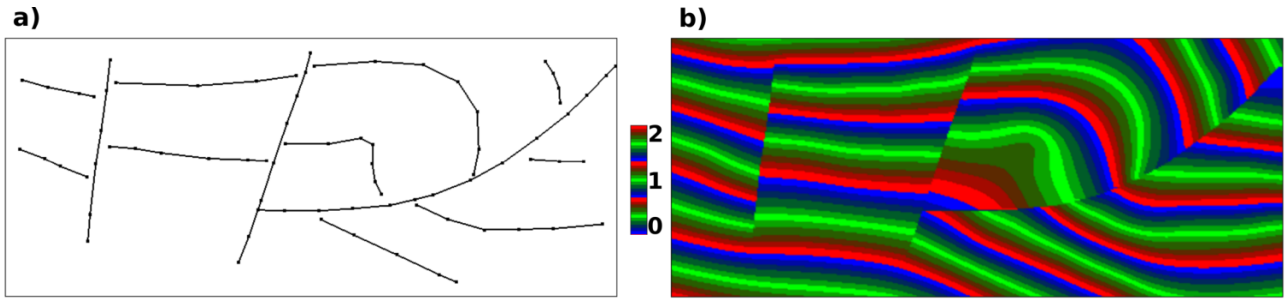

Figure 45: Implicit stratigraphic modeling of a section from the Ribaute model of Caumon et al. (2009). **a**) Input data, three faults and three horizons. **b**) Resulting stratigraphic field.

We propose an alternative approach. Briggs (1974) noted that the 1D interpolation problem had well defined properties; his strategy was then to extend these properties in 2D by solving the equivalent continuous problem (partial differential equation). Here, we notice that the 1D discrete interpolation problem has inherent properties that we wish to extend to high dimensional discrete interpolation problems. The key new elements of our method are the regularization operators we introduce. These operators discretize naturally on Cartesian grids: they do not require any special treatment at boundary nodes, and their generalization to higher dimensions is straightforward.

II.1.3 Domain and problem discretization

II.1.3.1 Domain discretization

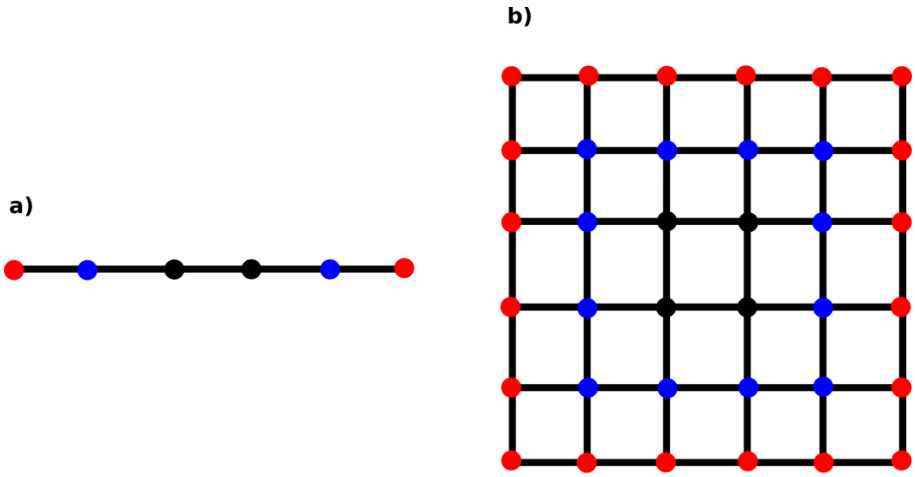

Figure 46: **a**) 1D domain discretization. **b**) 2D domain discretization. The red nodes are *external nodes*, the blue nodes are *boundary nodes*, and the black nodes are *internal nodes*.

Let Ω be the domain of definition of $\phi(\mathbf{x})$. Ω is discretized on a Cartesian grid, therefore Ω can be seen as a collection of N grid points $\{\mathbf{x}_l\}_{l=1}^{l=N}$. We distinguish three subdomains in Ω,

$$\Omega = \{\Omega_E \cup \Omega_B \cup \Omega_I\},$$

where Ω_E is the set of *external* grid points, Ω_B is the set of *boundary* grid points, and Ω_I is the set of *internal* grid points. An example of this discretization is illustrated in Figure 46. A more formal definition of these subdomains will be given shortly. The *computational domain* Ω_C is defined as

$$\Omega_C = \{\Omega_B \cup \Omega_I\}. \tag{70}$$

Each point in $\mathbf{x}_j$ in Ω_I has neighbors in Ω_C, these neighbors define the *neighborhood* $\mathscr{N}(\mathbf{x}_j)$ of $\mathbf{x}_j$. Figure 47a and Figure 47b illustrate the notion of neighborhoods in 1D and 2D; this figure also introduces *directional* vectors. For example, the 1D point x_j has two neighbors

$$\mathscr{N}(x_j) = \{x_{j-1}, x_{j+1}\} := \{x_{j-d_1}, x_{j+d_1}\};$$

that is, by definition, the 1D directional vector d_1 is [1]. In general, a point $\mathbf{x}_j$ always has two neighbors along a given direction: one neighbor in front, and one neighbor behind. For example in 2D the two neighbors of the point $\mathbf{x}_{i,j} = \mathbf{x}_k$ along the direction $\mathbf{d}_4 = [1\ 1]$ are

$$\{\mathbf{x}_{i-1,j-1}, \mathbf{x}_{i+1,j+1}\} := \{\mathbf{x}_{k-\mathbf{d}_4}, \mathbf{x}_{k+\mathbf{d}_4}\}. \tag{71}$$

In N dimensions, there are $M_n = 3^N - 1$ points in the neighborhood $\mathscr{N}(\mathbf{x}_j)$ of $\mathbf{x}_j \in \Omega_I$, and there are $M_d = \frac{M_n}{2}$ directions such that

$$\mathscr{N}(\mathbf{x}_j) = \left\{\mathbf{x}_{j\pm\mathbf{d}_k} \;\middle|\; \text{for } k = 1, ..., M_d\right\}, \tag{72}$$

where $\mathbf{d}_k$ is the k^{th} directional vector, as illustrated for example in Figure 47.

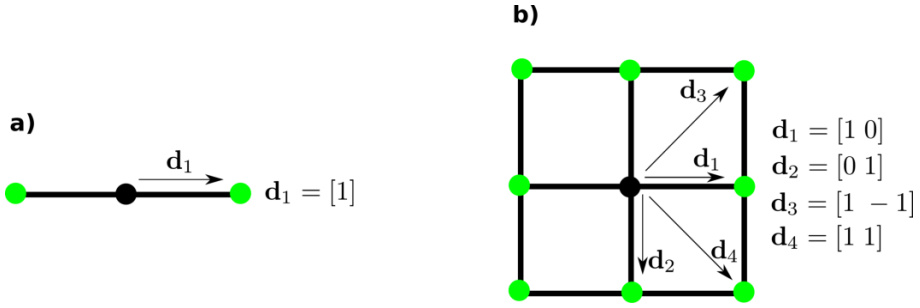

Figure 47: Notion of neighborhoods and directional vectors in 1D and 2D. **a**) 1D neighborhood: the *neighborhood* of the black node is made of the two (green) points that can be reached from it by taking one step (one forward and one backward) along the 1D *directional vector*. **b**) 2D neighborhood: the *neighborhood* of the black node is made of the eight (green) points that can be reached from it by taking one step (one forward and one backward) along the four 2D *directional vectors*.

Let us come back to the discretization $\Omega = \{\Omega_E \cup \Omega_B \cup \Omega_I\}$. Given a set of points in Ω_E, we formally define

$$\Omega_I = \left\{\mathbf{x}_j \,|\, \mathscr{N}(\mathbf{x}_j) \cap \Omega_E = \emptyset\right\}. \tag{73}$$

$$\Omega_B = \left\{\mathbf{x}_j \,|\, \mathscr{N}(\mathbf{x}_j) \cap \Omega_E \neq \emptyset \wedge \mathscr{N}(\mathbf{x}_j) \cap \Omega_I \neq \emptyset\right\}. \tag{74}$$

That is, a grid point that does not have a neighbor in Ω_E is by definition a point in Ω_I, and a grid point that has at least one neighbor in Ω_I and at least one neighbor in Ω_E is by definition a point in Ω_B. Points in Ω_E will usually be specified as inputs. It should be noted that an internal grid point is never in contact with an external grid point: there is always at least one boundary point between them.

II.1.3.2 Problem discretization

The implicit function $\phi(\mathbf{x})$ is evaluated on the discrete computational domain Ω_C by expansion into the series

$$\phi(\mathbf{x}) = \sum_{\mathbf{x}_j \in \Omega_C} B(\mathbf{x}, \mathbf{x}_j)\phi(\mathbf{x}_j) = \sum_{\mathbf{x}_j \in \Omega_C} B(\mathbf{x}, \mathbf{x}_j)\phi_j \,, \tag{75}$$

where $B(\mathbf{x}, \mathbf{x}_j)$ are some basis functions, and ϕ_j are the unknown coefficients at grid points. The basis functions $B(\mathbf{x}, \mathbf{x}_j)$ are assumed to satisfy the Kronecker delta property, that is for $\mathbf{x}_i, \mathbf{x}_j \in \Omega_C$,

$$B(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} .$$

Let $B(\mathbf{x}, \mathbf{x}_j)$ be 1^{st}-order Lagrange basis functions, simplifying equation 75 to

$$\phi(\mathbf{x}) = \sum_{\mathbf{x}_j \in \mathscr{N}'(\mathbf{x}_i)} B(\mathbf{x}, \mathbf{x}_j)\phi_j, \tag{76}$$

where $\mathbf{x}_i \in \Omega_I$ is the closest internal grid point to $\mathbf{x}$, and

$$\mathscr{N}'(\mathbf{x}_i) := \{\mathscr{N}(\mathbf{x}_i) \cup \mathbf{x}_i\};$$

it follows from definitions 73 and 74 that $\mathbf{x}_j \in \Omega_C$. Using equation 76, the first derivative along the k^{th} direction is given by

$$\partial_k \phi(\mathbf{x}) = \sum_{\mathbf{x}_j \in \mathscr{N}'(\mathbf{x}_i)} \partial_k B(\mathbf{x}, \mathbf{x}_j)\phi_j. \tag{77}$$

The constraints 67 and 69 can therefore be discretized in $N \geq 2$ dimensions as

$$\phi(\mathbf{x}) = \sum_{\mathbf{x}_j \in \mathscr{N}'(\mathbf{x}_i)} B(\mathbf{x}, \mathbf{x}_j)\phi_j = a. \tag{78}$$

$$\mathbf{u}^l \cdot \nabla\phi(\mathbf{x}) = \sum_k^N u_k^l \partial_k \phi(\mathbf{x}) = \sum_k^N u_k^l \left(\sum_{\mathbf{x}_j \in \mathscr{N}'(\mathbf{x}_i)} \partial_k B(\mathbf{x}, \mathbf{x}_j)\phi_j \right) = 0, \text{ for } l = 2, ..., N. \tag{79}$$

Some constraints may involve second derivatives; this is typically the case for smoothing regularization constraints. However, second derivatives of equation 76 cannot be computed directly, as was the case for first derivatives, because of the use of 1^{st}-order Lagrange basis functions. The second derivative at a given point $\mathbf{x}$ is therefore approximated by the finite-difference second derivative at $\mathbf{x}_i \in \Omega_I$, the closest internal grid point to $\mathbf{x}$. In particular, let d_k denote the distance between two neighboring points along the directional vector $\mathbf{d}_k$; then, using the notation in definition 71, the second derivative along the k^{th} direction is given by

$$\partial_k^2 \phi(\mathbf{x}) \approx \partial_k^2 \phi(\mathbf{x}_i) = \frac{-2\phi_i + \phi_{i+\mathbf{d}_k} + \phi_{i-\mathbf{d}_k}}{\mathrm{d}_k^2} \,, \tag{80}$$

and the mixed derivative along two orthogonal directions k, l is given by

$$\partial_k \partial_l \phi(\mathbf{x}) \approx \partial_k \partial_l \phi(\mathbf{x}_i) = \frac{\phi_{i+\mathbf{d}_k+\mathbf{d}_l} + \phi_{i-\mathbf{d}_k-\mathbf{d}_l} - \phi_{i-\mathbf{d}_k+\mathbf{d}_l} - \phi_{i+\mathbf{d}_k-\mathbf{d}_l}}{4\mathrm{d}_k\mathrm{d}_l}. \tag{81}$$

In this chapter, all constraints involving second derivatives are imposed strictly on grid points; therefore, the approximation symbol $\approx$ in equations 80 and 81 can be replaced by the equality symbol $=$.

II.1.3.3 Solving the discrete problem

The system of equations 78-79, can be written in the more compact matrix form

$$\bar{\bar{\mathbf{A}}}\bar{\Phi} = \bar{\mathbf{f}}_a\,. \tag{82}$$

$$\bar{\bar{\mathbf{U}}}\bar{\Phi} = \bar{\mathbf{0}}\,. \tag{83}$$

In practice, the system above is usually underdetermined and we have to regularize it with some smoothing constraints of the form $\bar{\bar{\mathbf{R}}}\bar{\Phi} = \bar{\mathbf{0}}$. That is, the solution $\bar{\Phi}$ is obtained by solving

$$\begin{bmatrix} \bar{\bar{\mathbf{A}}} \\ \bar{\bar{\mathbf{U}}} \\ \bar{\bar{\mathbf{R}}} \end{bmatrix} \left[\bar{\Phi}\right] = \begin{bmatrix} \bar{\mathbf{f}}_a \\ \bar{\mathbf{0}} \\ \bar{\mathbf{0}} \end{bmatrix} \tag{84}$$

in a least-squares sense. Convenient choices for $\bar{\bar{\mathbf{R}}}$ will be proposed later in the chapter. The matrices involved in the system 84 are very sparse and straightforward to build.

II.1.4 Interpolation in 1D: problem formulation

II.1.4.1 Interpolation in 1D

The structural interpolation problem can be described as a problem of finding a *smooth* function $\phi(\mathbf{x})$ that interpolates a given set of structural data. The smoothness is typically achieved by minimizing the function's *roughness* $\mathscr{R}(\phi)$ (Mallet, 1989). It is unclear what the definition of roughness is, precisely; roughness is usually understood to be related to the notion of curvature. In 1D, a widely accepted definition of roughness is the second derivate operator (Mallet, 1997)

$$\mathscr{R}(\phi) := \partial_x^2 \phi(\mathbf{x}); \tag{85}$$

Minimizing the second derivative in 1D minimizes the curvature of the function; this statement is unambiguous because there is only one way to define curvature in 1D. The 1D smoothing constraint, defined only on grid points, can therefore be discretized using equation 80 to give

$$\mathscr{R}(\phi_j) = \frac{-2\phi_j + \phi_{j+1} + \phi_{j-1}}{|dx|^2} = 0, \tag{86}$$

for every grid point $x_j \in \Omega_I$.

II.1.4.2 Problem formulation

In higher dimensions, *minimizing the roughness* of a function becomes ambiguous: there are infinitely many candidate roughness operators in higher dimensions that reduce to equation 85 in 1D. The standard way of overcoming this ambiguity is to explicitly give a physical meaning to equation 85. For example, Briggs (1974) and Levy (1999) explicitly look for a function $\phi(\mathbf{x})$ that minimizes the global (mean) curvature; in that case, it becomes clear that the extension of equation 85 in higher dimensions is

$$\mathscr{R}(\phi) = \Delta\phi(\mathbf{x}),$$

which is unambiguous since the Laplacian operator Δ has a precise definition in all dimensions. Another common choice is to seek for a function $\phi(\mathbf{x})$ that minimizes the bending energy (see Renaudeau et al., 2017a, and references therein).

In this chapter, we take a different approach to determine the equivalent of equation 85 in higher dimensions. We find inherent properties of the 1D discrete equation 86 that we consider to be practical, from an implementation point of view, then we look for an equivalent discrete equation in higher dimensions which has the same properties. The resulting discrete operator in higher dimensions can then be transformed back to the continuous version if needed.

A very practical property of sparse data interpolation using equation 86 is that it does not require boundary conditions. We define a *boundary condition* as any constraint imposed on a grid point $\mathbf{x} \in \Omega_B$. In the absence of discontinuities, a regularization matrix $\bar{\bar{\mathbf{R}}}$ built by imposing equation 86 on all grid points $\mathbf{x} \in \Omega_I$, has a $\text{rank}(\bar{\bar{\mathbf{R}}}) = M - 2$, for a 1D problem with M degrees of freedom (i.e. M grid points in Ω_C). The property $\text{rank}(\bar{\bar{\mathbf{R}}}) = M - 2$ highlights the fact that a minimum of two independent points are required to define a function in 1D. In general, $\mathscr{R}(\phi_j)$ will be said to satisfy the *maximum-rank property* if, in the absence of discontinuities, $\text{rank}(\bar{\bar{\mathbf{R}}}) = M - (N + 1)$; where $\bar{\bar{\mathbf{R}}}$ is the regularization matrix built by imposing $\mathscr{R}(\phi_j) = 0$ for all grid points $\mathbf{x}_j \in \Omega_I$, for a problem with M degrees of freedom in N dimensions. Our aim is to look for roughness operators $\mathscr{R}(\phi_j)$ that satisfy the maximum-rank property in higher dimensions.

The maximum-rank property can be extended to higher dimensions if the high dimensional equivalent of equation (86) has the following properties:

Property 1. $\mathscr{R}(\phi_j)$ should include all grid points in the neighborhood of $\mathbf{x}_j$, and $\mathbf{x}_j$ itself;
Property 2. $\mathscr{R}(\phi_j)$ should smooth independently along all directional vectors at $\mathbf{x}_j$.

The two properties above are inherent to the 1D discrete problem, as it can be observed by looking at equation (86) and Figure 47.a. Therefore, we propose to generalize these two properties to higher dimensions. As we increase dimensions, the first property will guarantee that every degree of freedom is taken into account, and the second property will guarantee that there are "enough" equations to constrain all the degrees of freedom.

II.1.4.3 Towards interpolation in higher dimensions

We propose two strategies for extending equation 86 to higher dimensions. The first one is to generalize the 1D smoothing constraint

$$\mathscr{R}(\phi_j) = \partial_x^2 \phi_j = 0$$

as a minimization of second directional derivatives of ϕ_j along all directional vectors, that is:

$$\mathscr{R}(\phi_j) = \partial_k^2 \phi_j = 0, \text{ for all directional vectors } \mathbf{d}_k \text{ at } \mathbf{x}_j. \tag{87}$$

The second approach is to generalize the 1D smoothing constraint

$$\mathscr{R}(\phi_j) = \partial_x^2 \phi_j = \partial_x(\partial_x \phi_j) = 0$$

as a minimization of mixed derivatives of ϕ_j as follows:

$$\mathscr{R}(\phi_j) = \partial_k(\partial_l \phi_j) = 0, \left| \begin{array}{l} \text{for all axis-aligned directional vectors } \mathbf{d}_k, \text{ and} \\ \text{for all directional vectors } \mathbf{d}_l \text{ such that } \mathbf{d}_l = \mathbf{d}_k \text{ or } \mathbf{d}_l \perp \mathbf{d}_k. \end{array} \right. \tag{88}$$

Both equation 87 and equation 88 satisfy the properties imposed in the last paragraph; these two equations lead to different roughness operators.

II.1.5 Interpolation in 2D

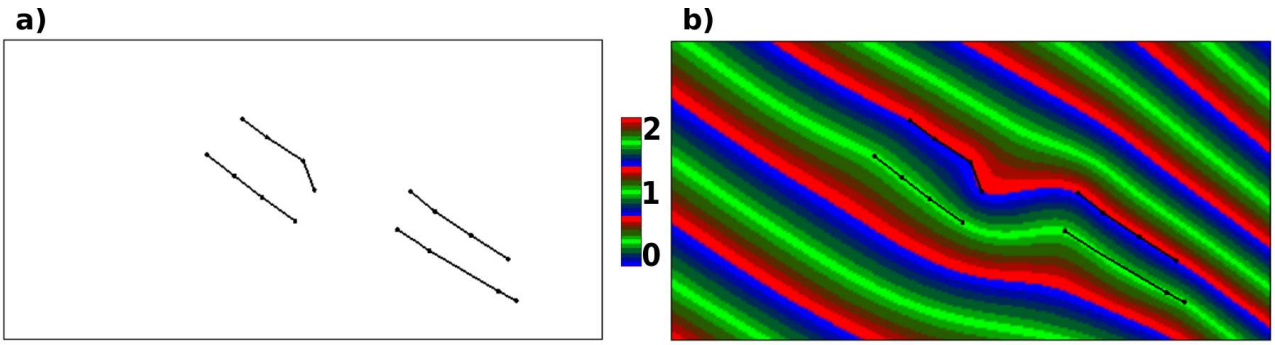

Figure 48: Implicit stratigraphic modeling of a modified section from one of the 2D benchmark models of Renaudeau et al. (2017a). **a**) Input data, two horizons. **b**) Resulting stratigraphic field.

Let us introduce the 2D *Cartesian-axes matrix* $\bar{\bar{\mathbf{C}}}_2$ and the 2D *diagonal-axes matrix* $\bar{\bar{\mathbf{D}}}_2$ defined as

$$\bar{\bar{\mathbf{C}}}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \bar{\bar{\mathbf{D}}}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}. \tag{89}$$

Each row of these matrices is a directional vector on the 2D grid as defined in Figure 47.b. The 2D *full direction matrix* is defined as

$$\bar{\bar{\mathbf{F}}}_2 = \begin{bmatrix} \bar{\bar{\mathbf{C}}}_2 \\ \bar{\bar{\mathbf{D}}}_2 \end{bmatrix}, \tag{90}$$

its rows give all the directions available, locally, on a 2D grid.

II.1.5.1 Directional second derivatives

From the directional second derivative point of view (equation 87), the 2D version of equation 86 is

$$\mathscr{R}(\phi_j) = \partial^2_k \phi_j = \frac{-2\phi_j + \phi_{j+\mathbf{d}_k} + \phi_{j-\mathbf{d}_k}}{\mathrm{d}_k^2} = 0, \text{ for all } \mathbf{d}_k \text{ rows of } \bar{\bar{\mathbf{F}}}_2. \tag{91}$$

Each of these directional derivatives $\partial^2_k \phi_j$ can be written analytically using the 2D Hessian matrix

$$\bar{\bar{\mathbf{H}}}_2 = \begin{bmatrix} \partial_x(\partial_x \phi) & \partial_x(\partial_z \phi) \\ \partial_z(\partial_x \phi) & \partial_z(\partial_z \phi) \end{bmatrix}$$

as $\partial^2_k \phi = \mathbf{d}_k \bar{\bar{\mathbf{H}}}_2 \mathbf{d}_k^T$. It follows that a 2D version of the roughness operator 85 that satisfies the maximum-rank property is

$$\mathscr{R}(\phi) = \begin{cases} \partial^2_{[1\ 0]}\phi = & \begin{bmatrix} 1 & 0 \end{bmatrix} \bar{\bar{\mathbf{H}}}_2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} & = \partial^2_x \phi \\ \partial^2_{[0\ 1]}\phi = & \begin{bmatrix} 0 & 1 \end{bmatrix} \bar{\bar{\mathbf{H}}}_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} & = \partial^2_z \phi \\ \partial^2_{[1\ -1]}\phi = & \frac{1}{2} \begin{bmatrix} 1 & -1 \end{bmatrix} \bar{\bar{\mathbf{H}}}_2 \begin{bmatrix} 1 \\ -1 \end{bmatrix} & = \frac{1}{2}(\partial^2_x + \partial^2_z - 2\partial_x \partial_z)\phi \\ \partial^2_{[1\ 1]}\phi = & \frac{1}{2} \begin{bmatrix} 1 & 1 \end{bmatrix} \bar{\bar{\mathbf{H}}}_2 \begin{bmatrix} 1 \\ 1 \end{bmatrix} & = \frac{1}{2}(\partial^2_x + \partial^2_z + 2\partial_x \partial_z)\phi \end{cases}. \tag{92}$$

The roughness operator 92 is discretized exactly as the 1D version. Let $\bar{\bar{\mathbf{R}}}$ denote the regularization matrix resulting from equation 91/92, then the j^{th} row of the least-squares matrix $\bar{\bar{\mathbf{R}}}^t\bar{\bar{\mathbf{R}}}$ is a finite-difference approximation of

$$\Big[\big(\partial^2_{[1\ 0]}\big)^2 + \big(\partial^2_{[0\ 1]}\big)^2 + \big(\partial^2_{[1\ -1]}\big)^2 + \big(\partial^2_{[1\ 1]}\big)^2\Big]\phi_j. \tag{93}$$

It is interesting to note that equation 93 equates to

$$\Big[\big(\partial^2_x\big)^2 + \big(\partial^2_z\big)^2 + 2\big(\partial_x\partial_z\big)^2\Big]\phi_j + \frac{1}{2}\Big[\big(\partial^2_x\big)^2 + \big(\partial^2_z\big)^2 + 2\partial^2_x\partial^2_z\Big]\phi_j, \tag{94}$$

where the first term on the right hand side is the bending energy of Enriquez et al. (1983); Turk and O'Brien (2002); Renaudeau et al. (2019), and the second term is half the squared (mean) curvature of Briggs (1974), also known as the squared Laplacian (Levy, 1999). Figure 48 shows an example using the roughness operator 91/92 to a modified version of a 2D benchmark model proposed by Renaudeau et al. (2017a).

II.1.5.2 Mixed derivatives

From the mixed derivative point of view (equation 88), another roughness operator $\mathscr{R}(\phi)$ that satisfies the maximum-rank property is

$$\mathscr{R}(\phi) = \begin{cases} \partial_{[1\ 0]}\big(\partial_{[1\ 0]}\phi\big) = \begin{bmatrix}1 & 0\end{bmatrix}\bar{\bar{\mathbf{H}}}_2\begin{bmatrix}1\\0\end{bmatrix} = \partial_x\big(\partial_x\phi\big) \\ \partial_{[0\ 1]}\big(\partial_{[0\ 1]}\phi\big) = \begin{bmatrix}0 & 1\end{bmatrix}\bar{\bar{\mathbf{H}}}_2\begin{bmatrix}0\\1\end{bmatrix} = \partial_z\big(\partial_z\phi\big) \\ \partial_{[1\ 0]}\big(\partial_{[0\ 1]}\phi\big) = \begin{bmatrix}1 & 0\end{bmatrix}\bar{\bar{\mathbf{H}}}_2\begin{bmatrix}0\\1\end{bmatrix} = \partial_x\big(\partial_z\phi\big) \\ \partial_{[0\ 1]}\big(\partial_{[1\ 0]}\phi\big) = \begin{bmatrix}0 & 1\end{bmatrix}\bar{\bar{\mathbf{H}}}_2\begin{bmatrix}1\\0\end{bmatrix} = \partial_z\big(\partial_x\phi\big) \end{cases}, \tag{95}$$

which, using $\partial_x(\partial_z) = \partial_z(\partial_x)$, becomes

$$\mathscr{R}(\phi) = \begin{cases} \partial_{[1\ 0]}\big(\partial_{[1\ 0]}\phi\big) = \partial^2_x\phi \\ \partial_{[0\ 1]}\big(\partial_{[0\ 1]}\phi\big) = \partial^2_z\phi \\ \sqrt{2}\partial_{[1\ 0]}\big(\partial_{[0\ 1]}\phi\big) = \sqrt{2}\partial_x\big(\partial_z\phi_i\big) \end{cases}. \tag{96}$$

The factor $\sqrt{2}$ comes from noticing that

$$\Big(\partial_{[1\ 0]}\big(\partial_{[0\ 1]}\big)\Big)^2\phi_j + \Big(\partial_{[0\ 1]}\big(\partial_{[1\ 0]}\big)\Big)^2\phi_j = \Big(\sqrt{2}\partial_{[1\ 0]}\big(\partial_{[0\ 1]}\big)\Big)^2\phi_j;$$

as in the previous paragraph, the reason for equating squares of operators is because the problem is solved by least-squares. The complete sum of squares of terms in this operator equates to the first term on the right hand side of equation 94. The first two terms of equation 95 are discretized using equation 80 and the remaining terms are discretized using equation 81.

II.1.6 Interpolation in 3D

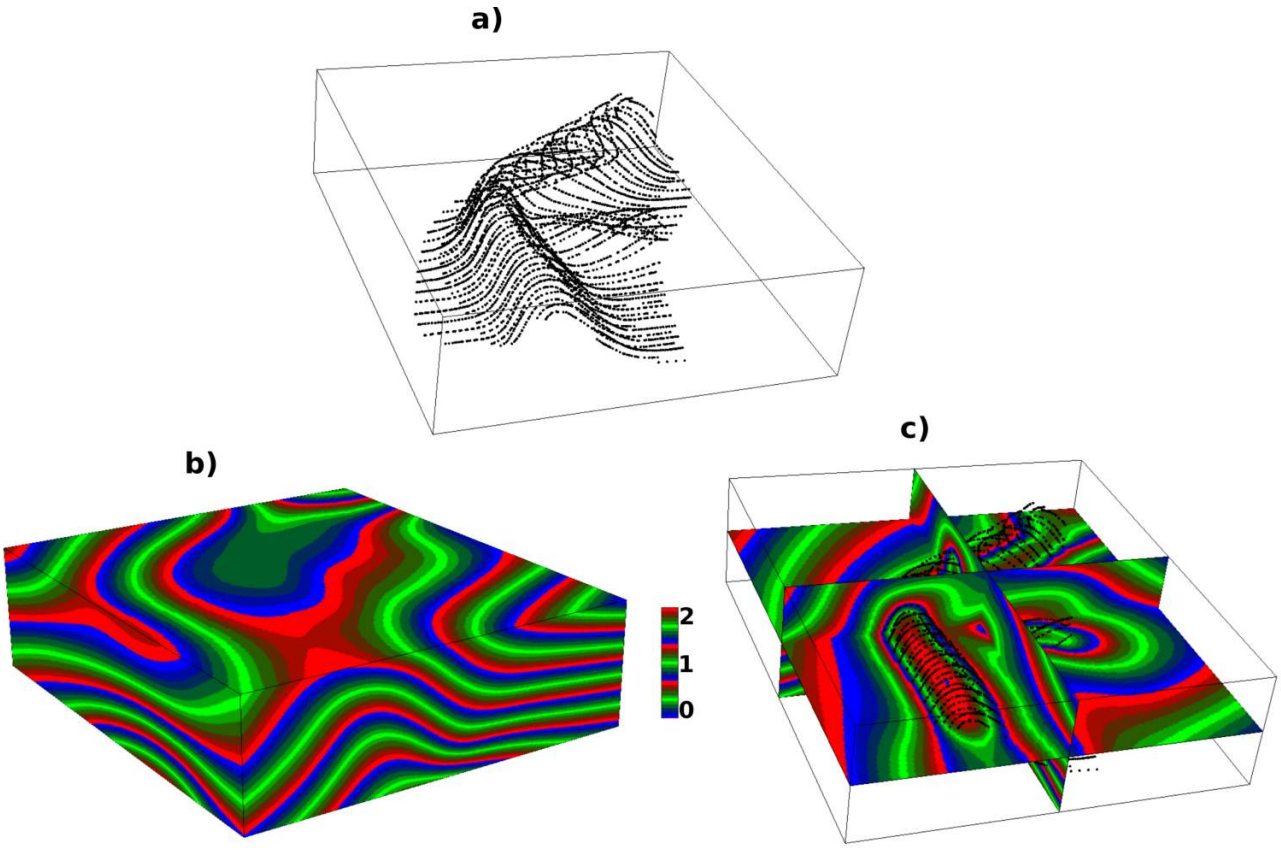

Figure 49: Implicit stratigraphic modeling of the Balzes fold model of Ramón et al. (2015). **a**) Input data consisting of horizon points picked along two horizons. **b**) Solid view of the resulting stratigraphic field. **c**) Section view of the resulting stratigraphic field.

The 3D Cartesian-axes matrix $\bar{\bar{\mathbf{C}}}_3$ and diagonal-axes matrix $\bar{\bar{\mathbf{D}}}_3$ are

$$\bar{\bar{\mathbf{C}}}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \bar{\bar{\mathbf{D}}}_3 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{97}$$

We also introduce the 3D *extended-diagonal-axes matrix* $\bar{\bar{\mathbf{E}}}_3$ defined as

$$\bar{\bar{\mathbf{E}}}_3 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & -1 \\ 0 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 1 & 0 \end{bmatrix}. \tag{98}$$

The 3D extended-diagonal matrix $\bar{\bar{\mathbf{E}}}_3$ is obtained by inserting an extra zero at different positions of each row of the 2D diagonal matrix $\bar{\bar{\mathbf{D}}}_2$. The 3D full direction matrix is defined as

$$\bar{\bar{\mathbf{F}}}_3 = \begin{bmatrix} \bar{\bar{\mathbf{C}}}_3 \\ \bar{\bar{\mathbf{D}}}_3 \\ \bar{\bar{\mathbf{E}}}_3 \end{bmatrix}; \tag{99}$$

its rows give all the directions available, locally, on a 3D grid.

Following the same reasoning as in section II.1.5, we use the direction matrix $\bar{\bar{\mathbf{F}}}_3$ to propose the following choices for the roughness operator:

$$\mathscr{R}(\phi) = \begin{cases} \partial^2_{[1\ 0\ 0]}\phi \\ \partial^2_{[0\ 1\ 0]}\phi \\ \partial^2_{[0\ 0\ 1]}\phi \\ \partial^2_{[0\ 1\ -1]}\phi \\ \partial^2_{[0\ 1\ 1]}\phi \\ \partial^2_{[1\ 0\ -1]}\phi \\ \partial^2_{[1\ 0\ 1]}\phi \\ \partial^2_{[1\ -1\ 0]}\phi \\ \partial^2_{[1\ 1\ 0]}\phi \\ \partial^2_{[1\ -1\ -1]}\phi \\ \partial^2_{[1\ -1\ 1]}\phi \\ \partial^2_{[1\ 1\ -1]}\phi \\ \partial^2_{[1\ 1\ 1]}\phi \end{cases}, \text{ and } \mathscr{R}(\phi) = \begin{cases} \partial^2_{[1\ 0\ 0]}\phi_i\big) \\ \partial^2_{[0\ 1\ 0]}\phi_i\big) \\ \partial^2_{[0\ 0\ 1]}\phi_i\big) \\ \sqrt{2}\partial_{[1\ 0\ 0]}\big(\partial_{[0\ 1\ 0]}\phi\big) \\ \sqrt{2}\partial_{[1\ 0\ 0]}\big(\partial_{[0\ 0\ 1]}\phi\big) \\ \sqrt{2}\partial_{[0\ 1\ 0]}\big(\partial_{[0\ 0\ 1]}\phi\big) \\ \sqrt{2}\partial_{[1\ 0\ 0]}\big(\partial_{[0\ 1\ 1]}\phi\big) \\ \sqrt{2}\partial_{[1\ 0\ 0]}\big(\partial_{[0\ 1\ -1]}\phi\big) \\ \sqrt{2}\partial_{[0\ 1\ 0]}\big(\partial_{[1\ 0\ 1]}\phi\big) \\ \sqrt{2}\partial_{[0\ 1\ 0]}\big(\partial_{[1\ 0\ -1]}\phi\big) \\ \sqrt{2}\partial_{[0\ 0\ 1]}\big(\partial_{[1\ 1\ 0]}\phi\big) \\ \sqrt{2}\partial_{[0\ 0\ 1]}\big(\partial_{[1\ -1\ 0]}\phi\big) \end{cases}. \tag{100}$$

The first operator comes from the directional second derivative approach (equation 87), while the second one comes the mixed derivatives approach (equation 88). It is also possible to combine these two approaches to get a third operator that satisfies the maximum-rank property (section II.1.4.2) ; in particular

$$\mathscr{R}(\phi) = \begin{cases} \partial^2_{[1\ 0\ 0]}\phi \\ \partial^2_{[0\ 1\ 0]}\phi \\ \partial^2_{[0\ 0\ 1]}\phi \\ \sqrt{2}\partial_{[1\ 0\ 0]}\big(\partial_{[0\ 1\ 0]}\phi\big) \\ \sqrt{2}\partial_{[1\ 0\ 0]}\big(\partial_{[0\ 0\ 1]}\phi\big) \\ \sqrt{2}\partial_{[0\ 1\ 0]}\big(\partial_{[0\ 0\ 1]}\phi\big) \\ \partial^2_{[1\ -1\ -1]}\phi \\ \partial^2_{[1\ -1\ 1]}\phi \\ \partial^2_{[1\ 1\ -1]}\phi \\ \partial^2_{[1\ 1\ 1]}\phi \end{cases}. \tag{101}$$

The analytic expressions of each term in these operators are given in the Appendix. While these operators may look complex as we increase dimensions, their discretization remains straightforward: all the second derivative terms are discretized using 80, and all the mixed derivatives terms are discretized using 81. This means that first operator in equation 100 is implemented exactly as its 2D (equation 91), and its 1D (equation 86) version. The only thing that changes are the number of directional vectors and their dimension. Figure 49 shows an example using this smoothing operator to the Balzes model of Ramón et al. (2015).

II.1.7 Handling discontinuities

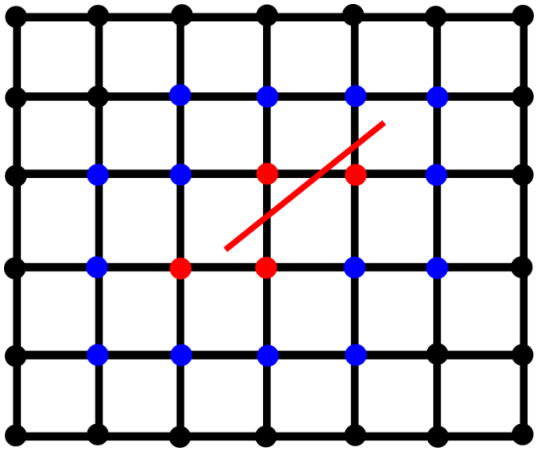

Figure 50: Domain discretization in the presence of a discontinuity. The red line and nodes represent a discontinuity and its rasterization. Red nodes are tagged as external nodes. Blue nodes are boundary nodes, and black nodes are internal nodes.

Discontinuities are often encountered in geological structural modeling. They usually consist of faults, unconformities and intrusive bodies. We propose a simple way to handle discontinuities: grid points affected by a discontinuity, after its rasterization, are tagged as external grid points (grid points in Ω_E). Consider for example the illustration in Figure 50: the discontinuity represented by the red line is rasterized to obtain the red external grid points, then boundary grid points are obtained from definition 74. In fact, handling discontinuities in this manner was the main motivation for most of the formalism presented earlier such as the definitions 72, 73, 74, and the maximum-rank property of Section II.1.4.2. This formalism resolves some issues raised by Mallet (1989); Mallet (1989) cautioned that the discretization of the roughness operator $\mathscr{R}(\phi)$ may require special attention at model boundaries and near discontinuities. No special treatment is required at boundaries here, since the roughness operator is only discretized on internal grid points. For the illustration in Figure 50, the solution would be obtained everywhere except at the red grid points. Values at the red grid points may be estimated by extrapolation as a post-processing step. As an example, tagging grid points rasterized by the fault in Figure 51.a as points in Ω_E, changes the stratigraphic field from Figure 48.b into the stratigraphic field in Figure 51.b. This straightforward approach has proven capable of handling quite complex fault networks as shown in Figure 52. It also generalizes to higher dimensions as demonstrated by the 3D example in Figure 53.

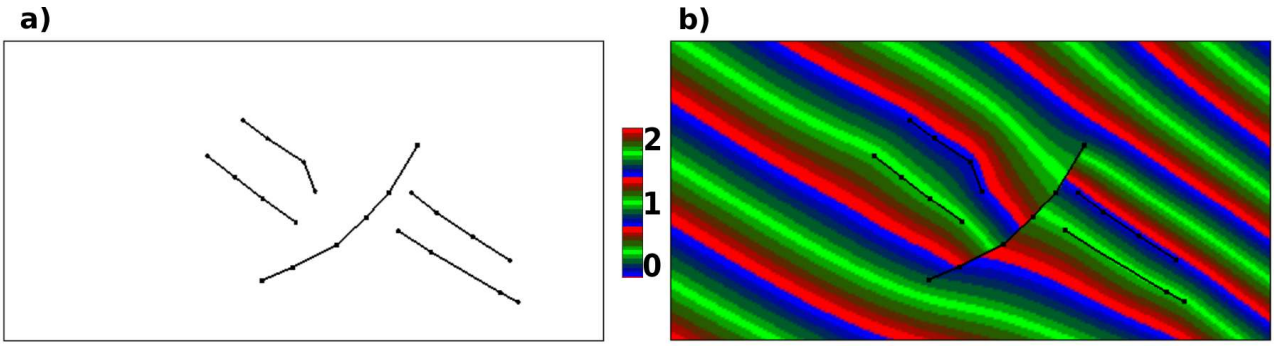

Figure 51: Implicit stratigraphic modeling of model in Figure 48, in the presence a discontinuity. **a**) Input data, two horizons and one fault. **b**) Resulting stratigraphic field.

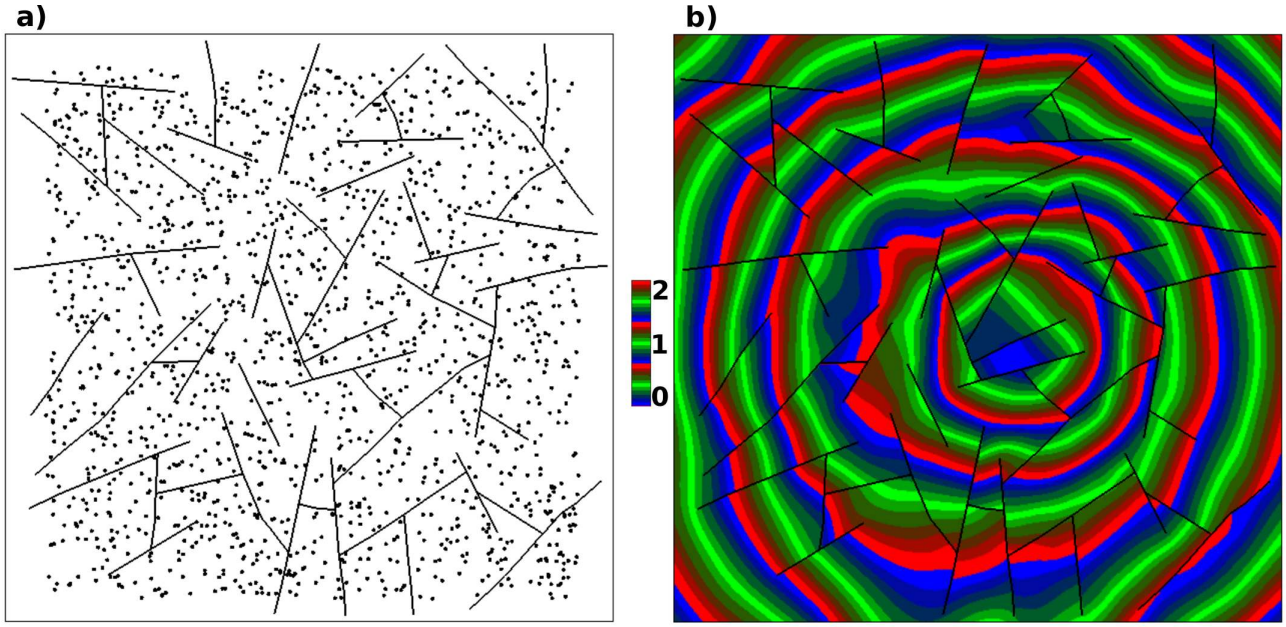

Figure 52: Implicit stratigraphic modeling in a highly faulted model. **a**) Input data, 1400 data points and 50 faults. **b**) Resulting stratigraphic field. Synthetic data courtesy of ExxonMobil.

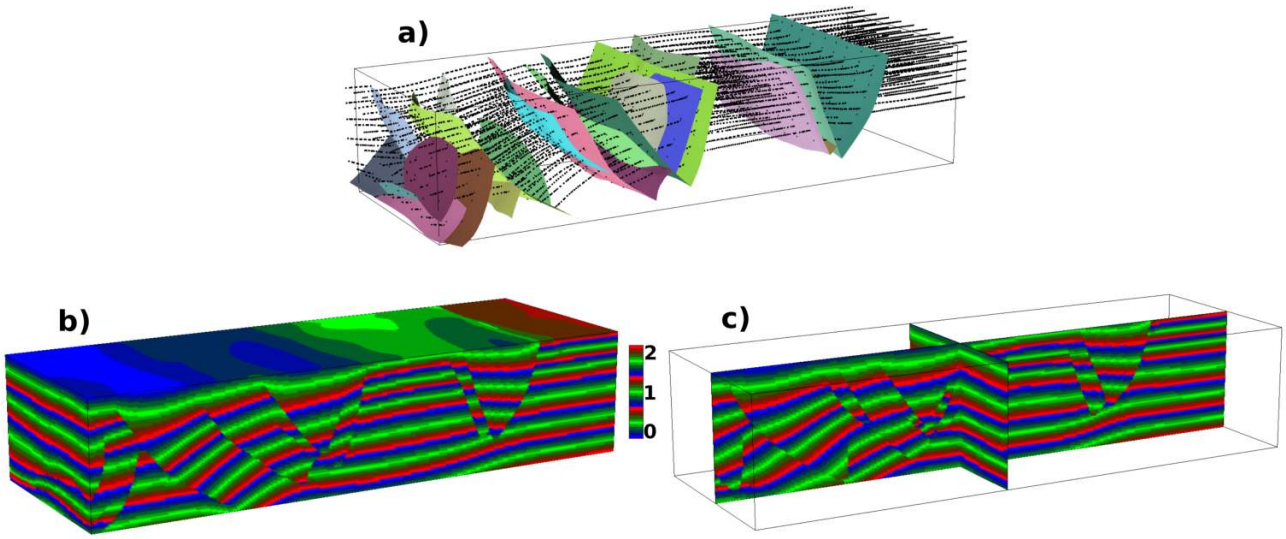

Figure 53: Implicit stratigraphic modeling of the 3D sandbox model of Chauvin et al. (2018). **a**) Input data consisting of 27 fault surfaces and horizon points picked along 6 horizons. **b**) Solid view of the resulting stratigraphic field. **c**) Section view of the resulting stratigraphic field. Data courtesy of IFPEN and C&C Reservoirs.

II.1.8 Limitations and discussion

II.1.8.1 Large thickness variations

When a layer exhibits large thickness variations, the resulting implicit field can have strong artifacts. This is a problem common to most implicit modeling techniques. According to Smith and Wessel (1990), this problem seems to come from our need to look for functions with continuous second derivatives everywhere. Current research in implicit structural modeling includes finding a solution to this problem suited for geological modeling applications (Laurent, 2016; Renaudeau et al., 2017b).

II.1.8.2 Discontinuities

The method proposed to handle discontinuities in Section II.1.7 can be numerically inefficient in some cases. If two faults are very close to one another, and there are some data between them, we are forced to use a fine grid in order to take into account those points. This can dramatically increase both the computational time and the computational memory required to solve the system depending on how close the faults are to each other. Current research in finite difference implicit modeling involves finding ways to handle arbitrarily complex fault networks without having to resort to finer grids (Renaudeau et al., 2018).

II.1.8.3 Performance

We use a multi-grid conjugate gradient solver running on GPU on a computer equipped with a 3.5 GHz core and a Quadro M4000 GPU. The 2D problem takes about a second for a problem with 250K grid points, and 3-4 seconds for a problem with 1M grid points. The 3D problem takes about 7-8 seconds for a problem with 1M grid points, and about 40 seconds for a problem with 5M grid points. Future research may include trying to improve performance by preconditioning our solver; some preconditioners for such a problem can be found in Wu and Hale (2015). One limitation of the proposed method is the high memory requirement for 3D problems; if the first roughness operator from equation 100 is used, each grid point has a memory cost of at least

$$13 \text{ equations} \times 3 \text{ nonzeros} \times (1 \text{ row id} + 1 \text{ column id} + 1 \text{ value }) \times 4 \text{ bytes} = 468 \text{ bytes}.$$

II.1.8.4 Beyond 3D

Sections II.1.5 and II.1.6 gave particular applications of equations 87 and 88 in 2D and in 3D. While the formulas for the roughness operator $\mathscr{R}(\phi)$ appeared to become more complex as we went from 1D, to 2D, to 3D, the discrete version of the roughness operator did not change. The only change was in the size of the full directional matrix $\bar{\bar{\mathbf{F}}}$. It is reasonable to expect that this remains true for all higher dimensions $\mathrm{N} > 3$. That is, all we need for an ND interpolation problem is to determine the ND directional matrix $\bar{\bar{\mathbf{F}}}_{\mathrm{N}}$. In general, the directional matrix is given by

$$\bar{\bar{\mathbf{F}}}_{\mathrm{N}} = \begin{bmatrix} \bar{\bar{\mathbf{C}}}_{\mathrm{N}} \\ \bar{\bar{\mathbf{D}}}_{\mathrm{N}} \\ \bar{\bar{\mathbf{E}}}_{\mathrm{N}} \end{bmatrix},$$

with the understanding that the diagonal-axis matrix $\bar{\bar{\mathbf{D}}}_{\mathrm{N}}$ only exists for $\mathrm{N} > 1$, and the extented-diagonal-axis matrix $\bar{\bar{\mathbf{E}}}_{\mathrm{N}}$ only exists for $\mathrm{N} > 2$. The Cartesian-axis and diagonal-axis matrices are given by

$$\begin{aligned} \bar{\bar{\mathbf{C}}}_{\mathrm{N}}(i,j) &= \delta_{ij}. \\ \bar{\bar{\mathbf{D}}}_{\mathrm{N}}(i,j) &= (-1)^{\mathrm{f}(i,j)}, \text{ with } \mathrm{f}(i,j) = (2 \cdot i)^{(j+1-\mathrm{N})}. \end{aligned}$$

The extended-diagonal-axis matrix $\bar{\bar{\mathbf{E}}}_{\mathrm{N}}$ is obtained by inserting a zero at different positions of each row of the matrix

$$\begin{bmatrix} \bar{\bar{\mathbf{D}}}_{\mathrm{N\text{-}1}} \\ \bar{\bar{\mathbf{E}}}_{\mathrm{N\text{-}1}} \end{bmatrix},$$

in a similar way that we obtained $\bar{\bar{\mathbf{E}}}_3$ from $\bar{\bar{\mathbf{D}}}_2$ in the previous section. $\bar{\bar{\mathbf{F}}}_{\mathrm{N}}$ has $\frac{3^{\mathrm{N}}-1}{2}$ rows and N columns.

II.1.9 Conclusion

We have introduced a new technique for implicit modeling on Cartesian grids. We identified inherent practical properties of the well behaved 1D discrete second derivative operator classically used to regularize interpolation in 1D, and then we designed higher dimensional discrete regularization operators with the same properties. In doing so, we obtained regularization operators that discretize very naturally on the Cartesian grid: the operators do not require special treatment on boundary nodes, and they generalize to higher dimensions easily. Discarding boundary conditions allowed us to handle discontinuities by simply surrounding them with boundary nodes. As a result, our method is easy to implement, even in the presence of discontinuities. Numerical experiments demonstrate the robustness and numerical efficiency of the proposed method.

II.1.10 Acknowledgements

The authors would like to thank Total for the data used in Figure 44, Hao Huang (ExxonMobil) for the data used in Figure 52, and Benjamin Chauvin (previously RING, currently Harvard), IFPEN, C&C Reservoirs for the data used in Figure 53.

Modeste Irakarama would also like to thank Paul Cupillard (RING) for his encouragement to publish this work.

This work was done in the frame of the RING project at Université de Lorraine. We would therefore like to thank the sponsors of the RING-GOCAD Consortium managed by ASGA for their support. Software corresponding to this chapter is available to consortium members.

II.1.11 Appendix

The analytic expressions of terms appearing in the formulas for $\mathscr{R}(\phi)$, as proposed in Section II.1.6, are given below:

$$
\begin{aligned}
\partial^2_{[1\,0\,0]}\phi &= \partial^2_x\phi,\\
\partial^2_{[0\,1\,0]}\phi &= \partial^2_y\phi,\\
\partial^2_{[0\,0\,1]}\phi &= \partial^2_z\phi,\\
\sqrt{2}\partial_{[1\,0\,0]}\big(\partial_{[0\,1\,0]}\phi\big) &= \sqrt{2}\partial_x\big(\partial_y\phi\big),\\
\sqrt{2}\partial_{[1\,0\,0]}\big(\partial_{[0\,0\,1]}\phi\big) &= \sqrt{2}\partial_x\big(\partial_z\phi\big),\\
\sqrt{2}\partial_{[0\,1\,0]}\big(\partial_{[0\,0\,1]}\phi\big) &= \sqrt{2}\partial_y\big(\partial_z\phi\big),\\
\sqrt{2}\partial_{[1\,0\,0]}\big(\partial_{[0\,1\,1]}\phi\big) &= \partial_x\big(\partial_y\phi + \partial_z\phi\big),\\
\sqrt{2}\partial_{[1\,0\,0]}\big(\partial_{[0\,1\,-1]}\phi\big) &= \partial_x\big(\partial_y\phi - \partial_z\phi\big),\\
\sqrt{2}\partial_{[0\,1\,0]}\big(\partial_{[1\,0\,1]}\phi\big) &= \partial_y\big(\partial_x\phi + \partial_z\phi\big),\\
\sqrt{2}\partial_{[0\,1\,0]}\big(\partial_{[1\,0\,-1]}\phi\big) &= \partial_y\big(\partial_x\phi - \partial_z\phi\big),\\
\sqrt{2}\partial_{[0\,0\,1]}\big(\partial_{[1\,1\,0]}\phi\big) &= \partial_z\big(\partial_x\phi + \partial_y\phi\big),\\
\sqrt{2}\partial_{[0\,0\,1]}\big(\partial_{[1\,-1\,0]}\phi\big) &= \partial_z\big(\partial_x\phi - \partial_y\phi\big),\\
\partial^2_{[0\,1\,-1]}\phi &= \tfrac{1}{2}\big(\partial^2_y + \partial^2_z - 2\partial_y\partial_z\big)\phi,\\
\partial^2_{[0\,1\,1]}\phi &= \tfrac{1}{2}\big(\partial^2_y + \partial^2_z + 2\partial_y\partial_z\big)\phi,\\
\partial^2_{[1\,0\,-1]}\phi &= \tfrac{1}{2}\big(\partial^2_x + \partial^2_z - 2\partial_x\partial_z\big)\phi,\\
\partial^2_{[1\,0\,1]}\phi &= \tfrac{1}{2}\big(\partial^2_x + \partial^2_z + 2\partial_x\partial_z\big)\phi,\\
\partial^2_{[1\,-1\,0]}\phi &= \tfrac{1}{2}\big(\partial^2_x + \partial^2_y - 2\partial_x\partial_y\big)\phi,\\
\partial^2_{[1\,1\,0]}\phi &= \tfrac{1}{2}\big(\partial^2_x + \partial^2_y + 2\partial_x\partial_y\big)\phi,\\
\partial^2_{[1\,-1\,-1]}\phi &= \tfrac{1}{3}\big(\partial^2_x + \partial^2_y + \partial^2_z - 2\partial_x\partial_y - 2\partial_x\partial_z + 2\partial_y\partial_z\big)\phi,\\
\partial^2_{[1\,-1\,1]}\phi &= \tfrac{1}{3}\big(\partial^2_x + \partial^2_y + \partial^2_z - 2\partial_x\partial_y + 2\partial_x\partial_z - 2\partial_y\partial_z\big)\phi,\\
\partial^2_{[1\,1\,-1]}\phi &= \tfrac{1}{3}\big(\partial^2_x + \partial^2_y + \partial^2_z + 2\partial_x\partial_y - 2\partial_x\partial_z - 2\partial_y\partial_z\big)\phi,\\
\partial^2_{[1\,1\,1]}\phi &= \tfrac{1}{3}\big(\partial^2_x + \partial^2_y + \partial^2_z + 2\partial_x\partial_y + 2\partial_x\partial_z + 2\partial_y\partial_z\big)\phi.
\end{aligned}
$$

Chapter II.2

Finite element implicit structural modeling of geological structures

This chapter resulted from an ongoing technical report co-authored with Morgan Thierry-Coudon, Mustapha Zakari, Paul Cupillard, and Guillaume Caumon.

II.2.1 Summary

We introduce a new method for implicit structural modeling. The method is based on a finite element discretization of recently proposed regularization operators for implicit modeling. A finite element implicit modeling scheme is believed to offer some geometrical flexibility as it is readily implemented on both structured and unstructured grids. While implicit modeling on unstructured grids is not new, the method proposed here is, to our knowledge, the first implementation based on finite elements. The finite element method is routinely used to solve boundary value problems. However, because boundary conditions are usually unknown in implicit structural modeling, the traditional finite element method requires minor adjustments in order to be suitable for implicit modeling.

The work presented in this chapter is preliminary and is part of an ongoing master's project by Morgan Thierry-Coudon.

II.2.2 Introduction

Implicit structural modeling is one of the main techniques used in geomodeling (Lajaunie et al., 1997; Cowan et al., 2002; Frank, 2007; Calcagno et al., 2008; Caumon et al., 2013; Souche et al., 2013; Mallet, 2014). Structural implicit modeling has traditionally been separated into two main classes (Caumon et al., 2013; Wellmann and Caumon, 2018): (1) meshfree methods (Lajaunie et al., 1997; Cowan et al., 2002; Chilès et al., 2004; Calcagno et al., 2008; Renaudeau et al., 2019), (2) and mesh-based methods (Mallet, 2004; Frank, 2007; Caumon et al., 2013; Souche et al., 2013; Renaudeau et al., 2018). Mesh-based methods are collectively referred to as *Discrete Smooth Interpolation* (DSI) methods hereafter (Mallet, 1989, 1992, 1997). The method presented here belongs to the DSI class of methods. The principle of DSI methods is to discretize all structural constraints on a discrete domain, and assemble them into a least-squares system of linear equations supplemented with smoothing regularization constraints (Mallet, 1989, 1992, 1997). In particular, the DSI problem can be stated as that of finding a scalar function $\phi(\mathbf{x})$ that satisfies a system of linear equations of the form

$$\begin{bmatrix} \bar{\bar{\mathbf{D}}} \\ \bar{\bar{\mathbf{R}}} \end{bmatrix} \left[\bar{\Phi}\right] = \begin{bmatrix} \bar{\mathbf{f}_a} \\ \bar{\mathbf{0}} \end{bmatrix}, \tag{102}$$

where the *data system* $\bar{\bar{\mathbf{D}}}\bar{\Phi} = \bar{\mathbf{f}_a}$ is assembled from geological data collected from the field or from seismic images. Geological data are often sparse; as a result, the data system is usually underdetermined and has to be regularized by the *regularization system* $\bar{\bar{\mathbf{R}}}\bar{\Phi} = \bar{\mathbf{0}}$. We only focus on the regularization system in this chapter.

The Laplacian has been found to be an appropriate regularization operator for some inverse problems (e.g. Clapp et al., 2003; Aster et al., 2004). Laplacian regularization is not appropriate for implicit modeling however. This is because the implicit function $\phi(\mathbf{x})$ represents stratigraphic time and is therefore subjected to some physical constraints. For example, the gradient of $\phi(\mathbf{x})$ should be non-zero everywhere in the domain of study. This constraint is usually not satisfied by Laplacian regularization when interpolating sparse data as the resulting $\phi(\mathbf{x})$ may exhibit saddle points. Another issue with Laplacian regularization is that it requires boundary conditions — in this chapter, *boundary conditions* are defined as any constraint defined on a boundary point. Often, one sets the derivatives of $\phi(\mathbf{x})$ to zero at boundaries. As a result, the iso-values of the function $\phi(\mathbf{x})$ tend to be perpendicular to boundaries, which is also not desirable in implicit modeling: the iso-values of the function $\phi(\mathbf{x})$ represent geological horizons so it is not reasonable to have them artificially terminate perpendicularly on boundaries and geological discontinuities. To avoid the aforementioned shortcomings of Laplacian regularization in implicit modeling, alternative regularization operators have been proposed. For example, Frank (2007); Caumon et al. (2013) present an implicit modeling method on unstructured meshes where they assemble a regularization system by minimizing second directional derivatives across tetrahedron faces. In particular, let $\nabla\phi(\mathbf{x})_1$, $\nabla\phi(\mathbf{x})_2$ be the gradients of two neighboring tetrahedrons T_1, T_2, and $\bar{n}$ the normal vector of the face shared by T_1, T_2, the regularization system is built by imposing

$$\left(\nabla\phi_1 - \nabla\phi_2\right) \cdot \bar{n} = 0 = \partial_{\bar{n}}\phi_1 - \partial_{\bar{n}}\phi_2 \Rightarrow \partial^2_{\bar{n}}\phi = 0 \tag{103}$$

across all faces shared by tetrahedrons. The notation $\partial_{\bar{n}}$ denotes the directional derivative along the direction vector $\bar{n}$. The regularization operator 103 therefore minimizes second directional derivatives along all the normals of tetrahedrons' faces; this is conceptually similar to the regularization operator proposed in chapter II.1, which minimizes second directional derivatives along all the directions on a Cartesian grid. For the general unstructured mesh, the directions of derivation in equation 103 may change from one tetrahedron to another. The directions of derivation on Cartesian grids do not change on the other hand, making it easier

to propose an analytical formula for the resulting regularization operator. For example, one of the regularization operators proposed in chapter II.1 in 2D is

$$\mathscr{R}(\phi) = \begin{cases} \partial^2_{[1\ 0]}\phi & = \partial^2_x\phi \\ \partial^2_{[0\ 1]}\phi & = \partial^2_z\phi \\ \partial^2_{[1\ -1]}\phi & = \frac{1}{2}(\partial^2_x + \partial^2_z - 2\partial_x\partial_z)\phi \\ \partial^2_{[1\ 1]}\phi & = \frac{1}{2}(\partial^2_x + \partial^2_z + 2\partial_x\partial_z)\phi \end{cases} . \tag{104}$$

In this chapter, we present a finite element discretization of the regularization operator 104. In practice, both equation 103 and 104 work equally well, so we are merely proposing an alternative method. The finite element method is familiar from solving boundary value problems. However, because boundary conditions are usually unknown in implicit modeling, the traditional finite element method requires minor adjustments in order to be suitable for implicit modeling. The subject of this chapter is to discuss these adjustments.

II.2.3 Finite elements for boundary value problems

In this section, we consider first the case where the regularization system is built by imposing the Laplacian

$$\mathscr{R}(\phi) = \begin{cases} \Delta\phi & = 0 \text{ in } \Omega_I \\ \nabla\phi\cdot\bar{n} & = 0 \text{ on } \Omega_B \end{cases} , \tag{105}$$

where Ω_I is the interior of the domain of interest Ω, Ω_B its boundary, such that

$$\Omega = \{\Omega_I \cup \Omega_B\}, \{\Omega_I \cap \Omega_B\} = \varnothing;$$

$\bar{n}$ is the normal on Ω_B. The short review of finite elements presented here follows the text of Hughes (2000), to which the reader is referred to for more details. One reduces the order of derivation on ϕ in equation 105 using integration by parts and the divergence theorem. In particular, let w be a *test function*, then equation 105 can be transformed to

$$\int_\Omega w(\mathbf{x})\Delta\phi(\mathbf{x}) = -\int_\Omega \nabla w(\mathbf{x})\nabla\phi(\mathbf{x}) + \int_{\Omega_B} w(\mathbf{x})\nabla\phi(\mathbf{x})\cdot\bar{n} = 0 \tag{106}$$

$$= -\int_\Omega \nabla w(\mathbf{x})\nabla\phi(\mathbf{x}) = 0. \tag{107}$$

In practice, the domain of interest Ω is discretized; we will occasionally think of Ω as a set of discrete points $\mathbf{x}_i$. Let $N(\mathbf{x}, \mathbf{x}_i)$ denote the standard piecewise linear finite element basis function such that

$$N(\mathbf{x}_j, \mathbf{x}_i) = \delta_{ji}, \tag{108}$$

and express $w(\mathbf{x})$ and $\phi(\mathbf{x})$ as

$$w(\mathbf{x}) = \sum_{\mathbf{x}_i\in\Omega} N(\mathbf{x}, \mathbf{x}_i)w_i, \tag{109}$$

$$\phi(\mathbf{x}) = \sum_{\mathbf{x}_j\in\Omega} N(\mathbf{x}, \mathbf{x}_j)\phi_j, \tag{110}$$

allowing equation 107 to be expressed as

$$0 = \sum_{\mathbf{x}_i\in\Omega} w_i \sum_{\mathbf{x}_j\in\Omega} \int_\Omega \nabla N(\mathbf{x}, \mathbf{x}_i)\nabla N(\mathbf{x}, \mathbf{x}_j)\phi_j$$

$$= \sum_{\mathbf{x}_i\in\Omega} w_i \sum_{\mathbf{x}_j\in\Omega} K_{ij}\phi_j = \sum_{\mathbf{x}_i\in\Omega} w_i G_i. \tag{111}$$

Equation 111 should be valid for all w_i; because the coefficients w_i are arbitrary, it follows that

$$G_i = \sum_{\mathbf{x}_j \in \Omega} K_{ij}\phi_j = \sum_{\mathbf{x}_j \in \Omega} \int_\Omega \nabla N(\mathbf{x}, \mathbf{x}_i) \nabla N(\mathbf{x}, \mathbf{x}_j)\phi_j = 0, \text{ for all } \mathbf{x}_i \in \Omega. \tag{112}$$

The matrix K_{ij} in equation 112 can then be assembled by integrating over each element independently as is customary in finite elements (Hughes, 2000).

II.2.4 Finite elements for structural interpolation

We now consider the finite element discretization of equation 104. We detail only the first term; the rest of the terms are discretized in a similar way. Consider the integral

$$\begin{aligned} \int_\Omega w(\mathbf{x})\partial_x^2\phi(\mathbf{x}) &= \int_z \left(\int_x w(\mathbf{x})\partial_x^2\phi(\mathbf{x}) \right) \\ &= \int_z \left(-\int_x \partial_x w(\mathbf{x})\partial_x\phi(\mathbf{x}) + \left[w(\mathbf{x})\partial_x\phi(\mathbf{x}) \right]_{\mathbf{x}_1 \in \Omega_B}^{\mathbf{x}_2 \in \Omega_B} \right), \end{aligned} \tag{113}$$

for some two boundary points $\mathbf{x}_1, \mathbf{x}_2$. In implicit modeling, the derivatives of $\phi(\mathbf{x})$ are not known at Ω_B, so the boundary term in equation 113 is undesirable. A straightforward way to get rid of that boundary term is to set the derivatives of $\phi(\mathbf{x})$ to zero as was done in section II.2.3 for Laplacian regularization. This leads to the matrix

$$K^1_{ij}\phi_j = \int_\Omega \partial_x N(\mathbf{x}, \mathbf{x}_i)\partial_x N(\mathbf{x}, \mathbf{x}_j)\phi_j = 0, \text{for all } \mathbf{x}_i, \mathbf{x}_j \in \Omega, \tag{114}$$

where the superscript 1 in K^1_{ij} refers to the 1st term of the operator 104. The complete regularization matrix is obtained by gathering the four matrices, one for each term; particularly

$$\bar{\bar{\mathbf{K}}} = \begin{bmatrix} \bar{\bar{\mathbf{K}}}^1 \\ \bar{\bar{\mathbf{K}}}^2 \\ \bar{\bar{\mathbf{K}}}^3 \\ \bar{\bar{\mathbf{K}}}^4 \end{bmatrix}. \tag{115}$$

However, the regularization system built this way is not appropriate for implicit modeling. Setting the derivatives of $\phi(\mathbf{x})$ to zero implies that geological horizons are perpendicular to the model boundary and geological discontinuities; this is too strong of an assumption.

We propose a simple strategy to get rid of the boundary term in equation 113. Instead of imposing heuristic boundary conditions on $\phi(\mathbf{x})$, which is subjected to physical constraints, we propose to impose that $w(\mathbf{x})$ vanishes at boundaries. In traditional finite elements, a similar argument is used to get rid of the boundary term on Dirichlet boundaries (Hughes, 2000). Here we simply extend the argument to the entire boundary. In practice, because the basis functions satisfy the Kronecker delta property 108, imposing $w(\mathbf{x})$ to vanish at boundaries means replacing equation 109 by

$$w(\mathbf{x}) = \sum_{\mathbf{x}_i \in \Omega_I} N(\mathbf{x}, \mathbf{x}_i) w_i, \tag{116}$$

leading to

$$K^1_{ij}\phi_j = \int_\Omega \partial_x N(\mathbf{x}, \mathbf{x}_i)\partial_x N(\mathbf{x}, \mathbf{x}_j)\phi_j = 0, \text{for all } \mathbf{x}_i \in \Omega_I, \mathbf{x}_j \in \Omega. \tag{117}$$

Note that the only difference between equations 109 and 116 is the domain of summation. Furthermore, the matrix in equation 117 is not square and symmetric, unlike the matrix in equation 114, since $|\Omega| > |\Omega_I|$. Note also that using equation 116 with Laplacian regularization would not work for sparse data interpolation because the resulting regularization matrix would not have enough equations (rows); it works for the operator 104 because this operator has multiple linearly independent terms for each grid point. Figure 54 compares interpolation results obtained using the regularization operator 112 with results obtained using the regularization operator 117. Results obtained using the regularization operator 114 are not shown as they are similar to the results shown in Figure 54a.

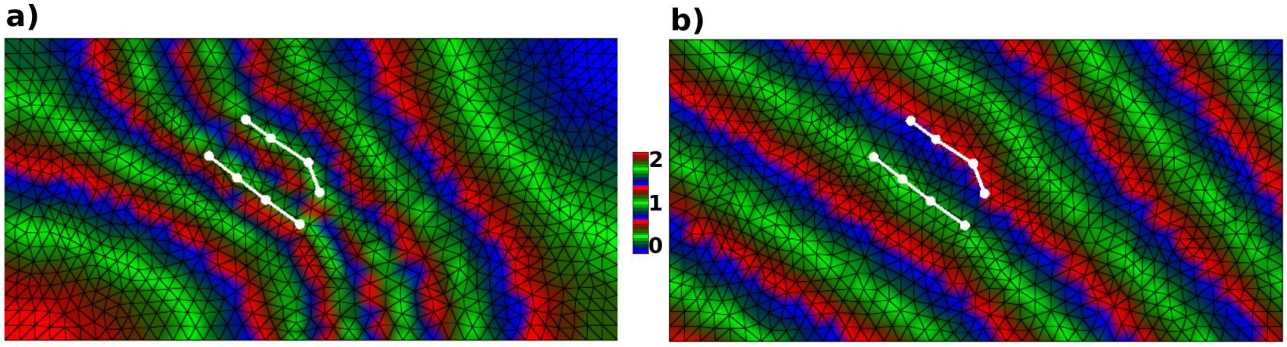

Figure 54: Implicit stratigraphic modeling of the 2D benchmark model of Renaudeau et al. (2017a). The input data, the white curves, represent two different iso-values. **a**) Result obtained on a triangle mesh using Laplacian regularization (eq. 112). **b**) Result obtained on a triangle mesh using the proposed regularization operator (eq. 117).

Equation 116 imposes a minor restriction on mesh generation. Consider for example the illustrative square domain in Figure 55a discretized with nine nodes and eight elements. There is only one internal node: node 5. Because there is no edge connecting the boundary nodes 3 and 7 to node 5, it follows from equation 117 that the 3$^{\text{rd}}$ and 7$^{\text{th}}$ columns of the matrix $\bar{\bar{\mathbf{K}}}$ (equation 115) will contain only zeros. In order to avoid such cases, each element of the mesh should have at least one internal node as shown in Figure 55b.

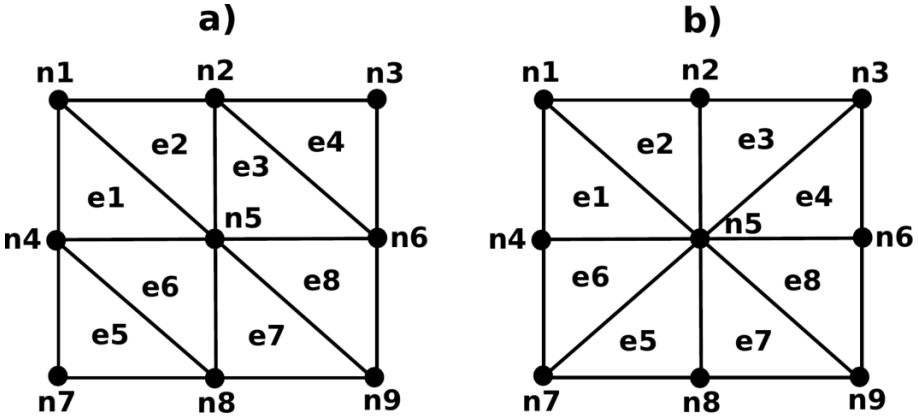

Figure 55: Example of valid and invalid meshing for structural interpolation. **a**) Invalid mesh: nodes 3 and 7 will not be constrained because there is no edge connecting them to an internal node. **b**) Valid mesh: all the elements have at least one internal node.

II.2.5 Numerical experiments

We now test the different regularization strategies on a section of the Ribaute model of Caumon et al. (2009). The input data is made of three faults, and iso-values picked along three horizons (Figure 56). Faults, and other discontinuities, are handled by duplicating nodes on fault-surfaces as proposed by Caumon et al. (2013). Figure 56a shows results obtained using Laplacian regularization 112 on a triangle mesh; the stratigraphic field tends to be perpendicular to boundaries as may be observed clearly on the bottom boundary. Figure 56b shows results obtained using the regularization operator 114 on a triangle mesh; the artifacts, due to imposing the derivative of the stratigraphic field along boundary normals to zero, are more severe compared to Laplacian regularization. Figures 56c and 56d show results obtained using the proposed regularization operator 117 respectively on a triangle mesh and a quadrilateral mesh. The quadrilateral mesh has fewer elements than the triangle mesh for the same resolution. As a result, interpolation on quadrilateral meshes has shown to converge slightly faster. However, for some examples it was not possible to generate a quadrilateral mesh of the entire domain of interest. In particular, the mesh generator was unable to generate quadrilaterals at some fault intersections.

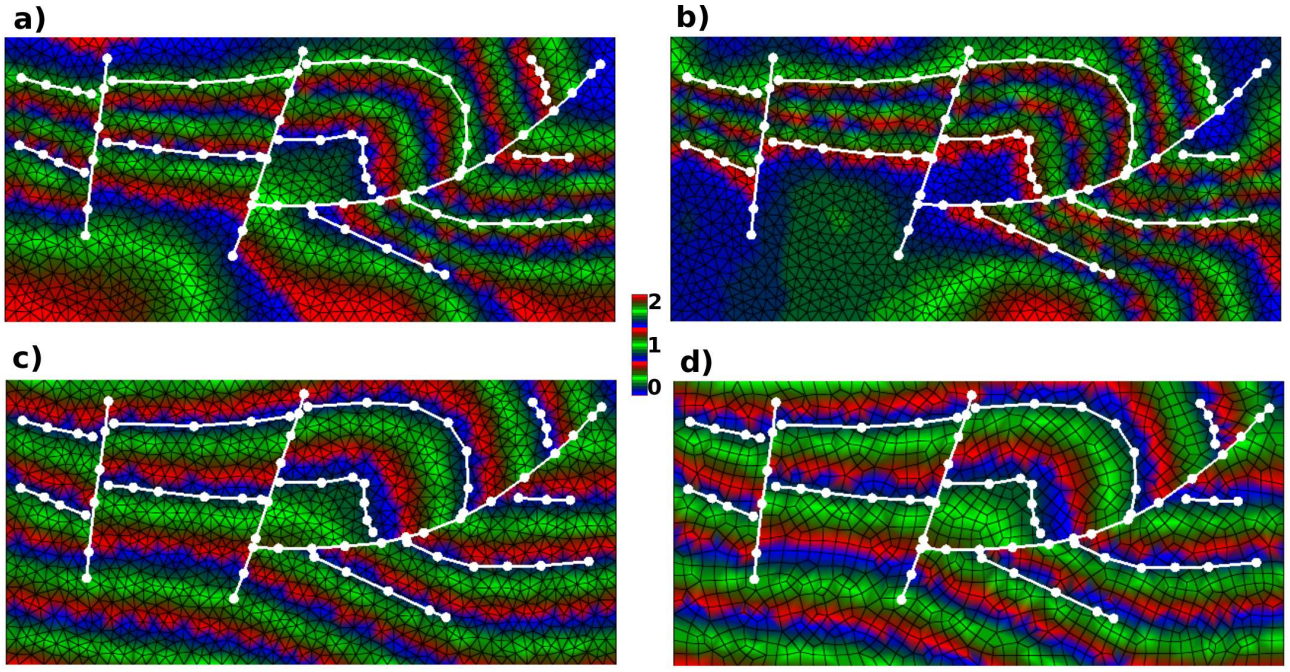

Figure 56: Implicit stratigraphic modeling of a section from the Ribaute model of Caumon et al. (2009). **a**) Result obtained on a triangle mesh using Laplacian regularization (eq. 112). **b**) Result obtained on a triangle mesh using the regularization operator (eq. 114). **c**) Result obtained on a triangle mesh using the proposed regularization operator (eq. 117). **d**) Result obtained on a quadrilateral mesh using the proposed regularization operator (eq. 117).

II.2.6 Discussion & Perspectives

One major challenge of the proposed method is being able to handle complex fault intersections. The current standard practice of handling faults in implicit modeling on irregular meshes is by duplicating nodes on fault-surfaces. This imposes an explicit computation of fault-intersection points. Furthermore, it imposes nodes on fault-surfaces to carry some topological information, such as which faults does a given node belong to. Such restrictions do not exist in finite difference implicit structural modeling (chapter II.1). When interpolating on unstructured meshes, a preprocessing step is usually required to address those restrictions. The preprocessing step

can be long and tedious; ongoing research includes automating this preprocessing step (Anquez et al., 2018, 2019).

Interpolation on quadrilateral meshes has shown to converge slightly faster than interpolation of triangle meshes; this is in part because a quadrilateral mesh typically has fewer elements than a triangle mesh for the same resolution. However, in some cases, it has proven challenging to generate an entirely quadrilateral mesh in the presence of intersecting faults. One way to take advantage of quadrilateral-elements may be to use a mixed-elements mesh (eg. Botella et al., 2016; Botella, 2016); the method proposed here is readily implemented on mixed-elements meshes.

II.2.7 Conclusion

We present a new implementation of implicit structural modeling based on finite elements of the recently proposed regularization operators (chapter II.1). A finite element implementation is believed to offer some geometrical flexibility as it is readily available for regular and iregular meshes. We show that finite elements for implicit structural modeling requires a minor modification of the standard finite elements for boundary value problems. These modifactions are imposed by the fact that we rarely have boundary conditions in structural modeling; this also imposes a constraint on mesh generation: every boundary node should have at least one edge connecting it to an internal node.

II.2.8 Acknowledgements

Modeste Irakarama's knowledge of implicit modeling has benefited greatly from discussions with Gautier Laurent and Julien Renaudeau. This work also benefited from discussions with François Bonneau, Pierre Anquez and Melchior Schuh-Senlis. Part of the code used in this chapter was written in collaboration with Maxim Torgonskiy and Olivier Giles as a school project while we were still students at École Nationale Supérieure de Géologie. Mesh generation was performed using the open source software Gmsh (Geuzaine and Remacle, 2009).

This work was done in the frame of the RING project at Université de Lorraine. We would therefore like to thank the sponsors of the RING-GOCAD Consortium managed by ASGA for their support.

Chapter II.3

Discussion

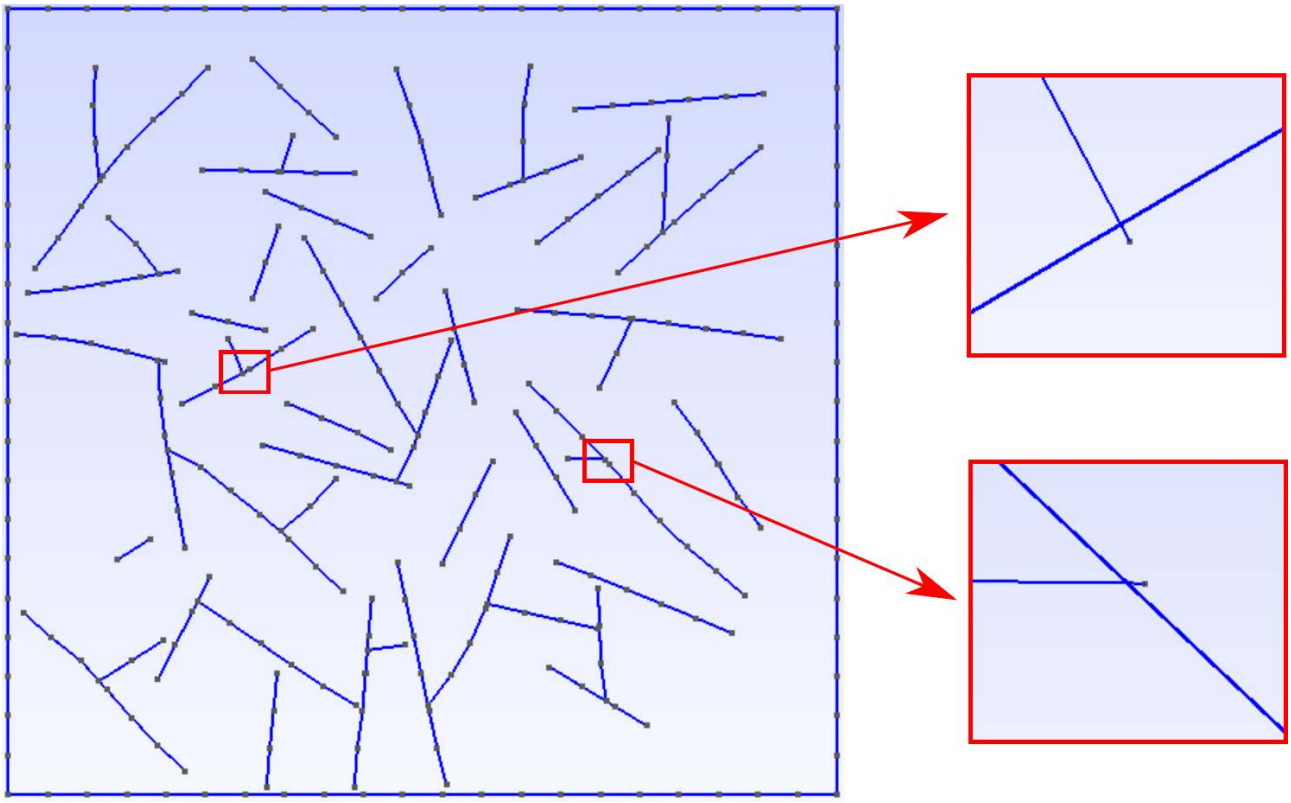

Figure 57: Example of a highly faulted model. A closer look at fault intersections shows that branching faults may overshot beyond the intersection point in practice. This eventually makes it difficult to generate a mesh that is suitable for implicit modeling. Synthetic data courtesy of ExxonMobil.

In this second Part of the manuscript, we presented new perspectives on implicit structural modeling. In addition to proposing new regularization operators, we introduced Finite Difference Implicit Structural Modeling and Finite Element Implicit Structural Modeling. The main motivation for the work presented in this Part was to develop a simple method for implicit modeling: a method based on finite differences on Cartesian grids. The finite element method was developed later following the interest of some colleagues in implementing the new regularization operators on irregular meshes. Current standard grid-based implicit modeling methods rely on irregular meshes. The conceptual advantage of irregular meshes over regular meshes is that irregular meshes require much less elements, and therefore less computer memory, to accurately model geometries with curves. The disadvantage of irregular meshes is that it becomes a challenge to generate a mesh in complex models with discontinuities. Consider for example the highly faulted model in Figure 57. A closer look at fault-fault intersections

shows that some branching faults overshoot beyond the branch point. Because current practice for handling faults in implicit modeling on irregular meshes involves duplicating fault nodes (e.g. Caumon et al., 2013), one has to explicitly compute the fault-fault intersection point and discard the overshooting part in order to generate a suitable mesh. This preprocessing step usually involves a lot of manual manipulations and can be quite fastidious, especially in 3D. Furthermore, computing fault branch lines in 3D is a challenging task on its own. The preprocessing step is not needed for the finite difference implicit modeling proposed in chapter II.1 (see Figure 52 for an example using the model in Figure 57).

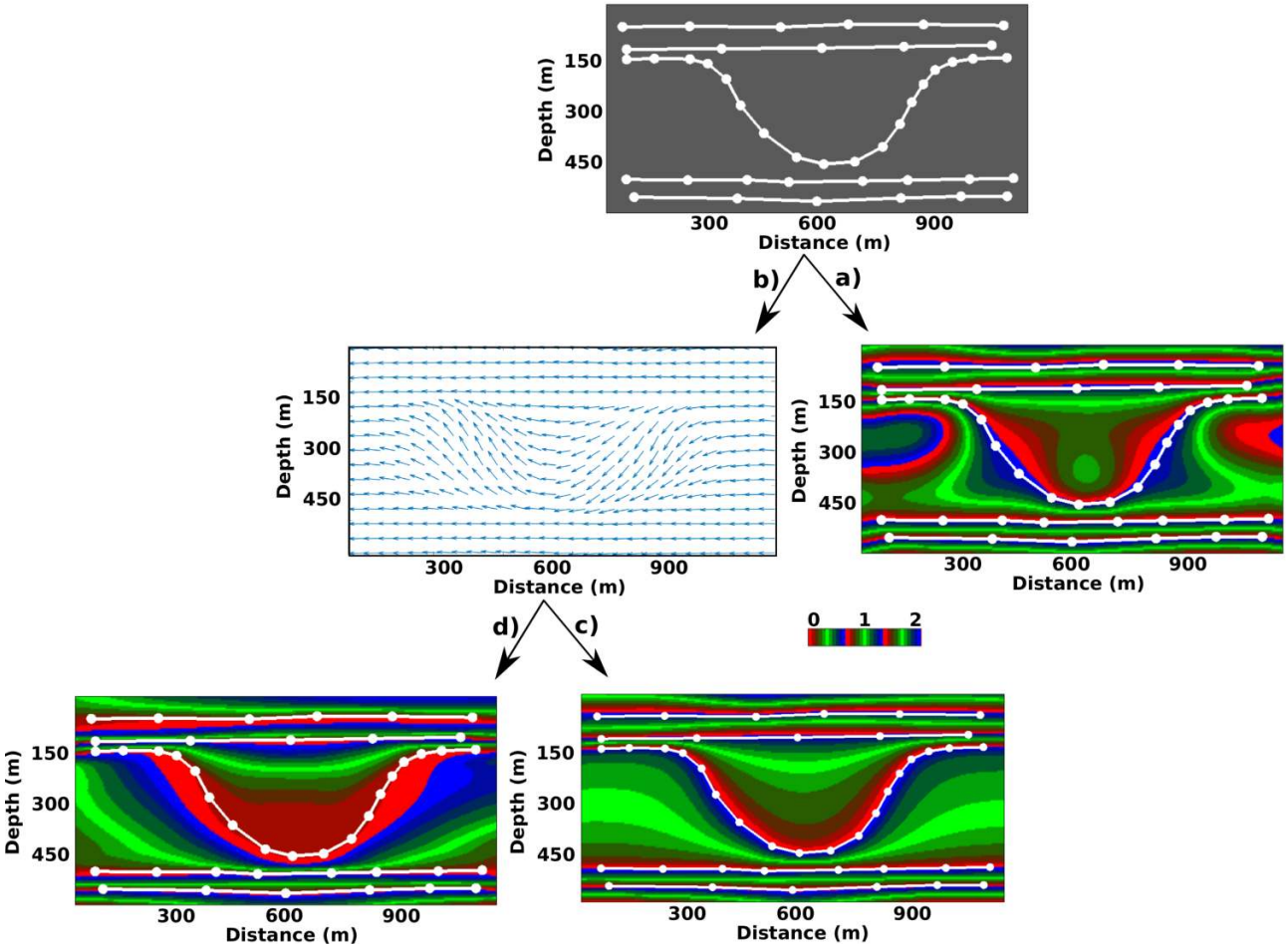

Figure 58: Handling large thickness variations by explicitly taking into account the orientation of horizons. This benchmark model is from Renaudeau et al. (2017b). **a**) Interpolation result obtained using the isotropic regularization operator proposed in chapter II.1. **b**) Estimation of the stratigraphic vector field from input data. **c**) Interpolation result obtained using the isotropic regularization operator proposed in chapter II.1 in addition to constraining the stratigraphic field to follow the estimated vector field. **d**) Interpolation result obtained using an anisotropic version of the regularization operator proposed in chapter II.1; anisotropic smoothing weights are determined from the estimated vector field.

Current implicit modeling methods perform poorly in the presence of large thickness variations (e.g. Collon and Caumon, 2017; Renaudeau et al., 2017b). In particular the constraint in equation 66d is usually violated in the presence of large thickness variation as illustrated by the results in Figure 58a. We now discuss two new strategies to address this problem; both methods use the orientation of horizons to guide the interpolation. The first step is common to both methods, it consists in estimating the orientation of the stratigraphic field in the entire domain from input data as shown in Figure 58b. The first method we propose is simply to constrain

the stratigraphic field to follow the estimated vector field using equation 69; this results in the stratigraphic field in Figure 58c. The second method we propose is to use the estimated vector field to build a regularization operator that smooths preferentially along the estimated vector field; in particular, the regularization operator in equation 92 is modified to

$$\mathscr{R}\phi(\mathbf{x}) = \begin{cases} w_1(\mathbf{x})\partial^2_{\mathbf{d}1}\phi = & w_1(\mathbf{x})\partial^2_{[1\ 0]}\phi & = w_1(\mathbf{x})\partial^2_x\phi \\ w_2(\mathbf{x})\partial^2_{\mathbf{d}2}\phi = & w_2(\mathbf{x})\partial^2_{[0\ 1]}\phi & = w_2(\mathbf{x})\partial^2_z\phi \\ w_3(\mathbf{x})\partial^2_{\mathbf{d}3}\phi = & w_3(\mathbf{x})\partial^2_{[1\ -1]}\phi & = w_3(\mathbf{x})\frac{1}{2}(\partial^2_x + \partial^2_z - 2\partial_x\partial_z)\phi \\ w_4(\mathbf{x})\partial^2_{\mathbf{d}4}\phi = & w_4(\mathbf{x})\partial^2_{[1\ 1]}\phi & = w_4(\mathbf{x})\frac{1}{2}(\partial^2_x + \partial^2_z + 2\partial_x\partial_z)\phi \end{cases}, \tag{118}$$

with

$$w_i(\mathbf{x}) = \left(\frac{\mathbf{d}i \cdot \mathbf{s}(\mathbf{x})}{||\mathbf{d}i||||\mathbf{s}(\mathbf{x})||}\right)^{\alpha} + \epsilon, \tag{119}$$

where $\mathbf{d}i$ is any of the directions of derivation in equation 118, $\mathbf{s}(\mathbf{x})$ is the value of the estimated vector field (Figure 58b) at the point $\mathbf{x}$, ϵ is a small value to avoid zero weights, and α is a parameter that determines the degree of anisotropic smoothing. The results in Figure 58d were obtained using $\alpha = 10$.

The first method proposed (Figure 58c) is conceptually similar to that of Laurent (2016) who also proposed to estimate a vector field from input data and use the vector field to constrain the interpolation. Our approach differs from that of Laurent (2016) at least in two ways:

1. Laurent (2016) estimates the stratigraphic orientation vector field (e.g. Figure 58b) from the gradient of an initial scalar field estimated from input data (e.g. Figure 58a) and then updates both the vector field and the scalar field iteratively; this approach is not guaranteed to converge and is computationally intensive since it is iterative. Here, we use every two consecutive points along input isovalue data to define vectors. The resulting vectors are then interpolated componentwise in the entire domain.

2. Laurent (2016) uses the estimated vector field to constrain the norm of the gradient of the stratigraphic field while we use estimated vector field to constrain the orientation of the stratigraphic field. Constraining the norm of the gradient is a nonlinear problem whereas constraining the orientation is a linear problem.

The differences between our method and that of Laurent (2016) can be justified by the difference in the problems we are seeking to solve: Laurent (2016) seeks to enforce the constraint in equation 66b while we seek to enforce the constraint in equation 66d.

The second method proposed (Figure 58d) reduces the artifacts observed in Figure 58a considerably but it does not eliminate them entirely: a closer look shows that artifacts are still present in Figure 58d (e.g. near the point 170,1100). It should also be noted that equation 119 is a specific case of the more general

$$w_i(\mathbf{x}) = M\big[\mathbf{d}i, \mathbf{s}(\mathbf{x})\big],$$

where $M(\cdot,\cdot)$ is an operator that measures the difference between two vectors. Current research includes defining an appropriate operator M that may eliminate the remaining artifacts in Figure 58d.

One disadvantage common to the two methods proposed is the need to specify the orientation of each input isovalue curve used to estimate the stratigraphic vector field (e.g. Figure 58b); this may not always be straightforward, especially in 3D.

Bibliography

K. Anjyo, J. Lewis, and F. Pighin. Scattered data interpolation for computer graphics. In *ACM SIGGRAPH 2014 Courses*, Vancouver, Canada, 2014.

P. Anquez, J. Pellerin, and G. Caumon. A Graph-Based Method to Detect and Correct Invalid Features in Subsurface Structural Models. In *80th EAGE Conference and Exhibition*, Copenhagen, Denmark, 2018.

P. Anquez, J. Pellerin, M. Irakarama, P. Cupillard, B. Lévy, and G. Caumon. Automatic correction and simplification of geological maps and cross-sections for numerical simulations. *Comptes Rendus Geoscience*, 351:48–58, 2019.

R. Aster, B. Borchers, and C. Thurber. *Parameter Estimation and Inverse Problems*. Academic Press, 2004.

A. Botella. *Génération de maillages non structurés volumiques de modèles géologiques pour la simulation de phénom'enes physiques*. PhD thesis, Université de Lorraine-ENSG, Nancy, France, 2016.

A. Botella, B. Lévy, and G. Caumon. Indirect unstructured hex-dominant mesh generation using tetrahedra recombination. *Computational Geosciences*, 20:437–451, 2016.

I. Briggs. Machine countouring using minimum curvature. *GEOPHYSICS*, 39(1):39–48, 1974.

P. Calcagno, J. Chilès, G. Courrioux, and A. Guillen. Geological modelling from the field data and geological knowledge Part I. Modeling method coupling 3D potential-field interpolation and geological rules. *Physics of the Earth and Planetary Interiors*, 17:147–157, 2008.

G. Caumon. Towards stochastic time-varying geological modeling. *Mathematical Geosciences*, 42:555–569, 2010.

G. Caumon, P. Collon, C. L. C. de Veslud, S. Viseur, and J. Sausse. Surface-Based 3D Modeling of Geological Structures. *Mathematical Geosciences*, 41:927–945, 2009.

G. Caumon, G. Gray, C. Antoine, and M.-O. Titeux. Three-dimensional implicit stratigraphic model building from remote sensing data on tetrahedral meshes: theory and application to regional model of La Popa Basin, NE Mexico. *IEEE Transactions on Geoscience and Remote Sensing*, 51(3), 2013.

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), 2002.

B. Chauvin, P. Lovely, J. Stockmeyer, A. Plesch, G. Caumon, and J. Shaw. Validating novel boundary conditions for three-dimensional mechanics-based restoration: An extensional sandbox model example. *AAPG Bulletin*, 102:245–266, 2018.

N. Cherpeau and G. Caumon. Stochastic structural modelling in sparse data situations. *Petroleum Geoscience*, 21:233–247, 2015.

J.-P. Chilès, C. Aug, A. Guillen, and T. Less. Modeling the geometry of geological units and its uncertainty in 3d form structural data: the potential-field method. In *Orebody Modeling and Strategic Mine Planning*, volume Spectrum 14, Perth, WA, 2004.

R. Clapp, B. Biondi, and J. Claerbout. Incorporating geologic information into reflection tomography. *GEOPHYSICS*, 69(2), 2003.

P. Collon and G. Caumon. 3D Geomodelling in Structurally Complex Areas - Implicit vs. Explicit representations. In *79th EAGE Conference and Exhibition*, Paris, France, 2017.

E. Cowan, R. Beatson, W. Fright, T. McLennan, and T. Mitchell. Rapid geological modeling. In *Applied Structural Geology for Mineral Exploration and Mining*, Kalgoorlie, Australia, September 2002.

J. Enriquez, J. Thomann, and M. Goupillot. Applications of bidimensional spline functions to geophysics. *GEOPHYSICS*, 48(9):1268–1273, 1983.

G. Fehmers and C. Höcker. Fast structural intepretation with structure-oriented filtering. *GEOPHYSICS*, 68(4), 2003.

T. Frank. 3D reconstruction of complex geological interfaces from irregularly distributed and noisy point data. *Computers and Geosciences*, 33:932–943, 2007.

C. Geuzaine and J.-F. Remacle. Gmsh: A 3d finite element mesh generator with builtin preand postprocessing facilities. *International journal for numerical methods in engineering*, 79(11): 1309–1331, 2009.

G. Godefroy, F. Bonneau, G. Caumon, and G. Laurent. Stochastic association of fault evidences using graph theory and geological rules. In *2017 RING meeting*, Université de Lorraine-ENSG, Nancy, France, September 2017. ASGA.

I. G. Gonçalves, S. Kumaira, and F. Guadagnin. A machine learning approach to the potential-field method for implicit modeling of geological structures. *Computers and Geosciences*, 103: 173–182, 2017.

L. Grose, G. Laurent, L. Ailleres, R. Armit, M. Jessell, and G. Caumon. Structural data constraints for implicit modeling of folds. *Journal of Structural Geology*, 104:80–92, 2017.

M. Hillier, E. Schetselaar, E. Kemp, and G. Perron. Three-Dimensional Modeling of Geological Surfaces Using Generelized Interpolation with Radial Basis Functions. *Mathematical Geosciences*, 46:931–953, 2014.

T. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2000.

M. Irakarama, G. Laurent, J. Renaudeau, and G. Caumon. Finite Difference Implicit Modeling of Geological Structures. In *80th EAGE Conference and Exhibition*, Copenhagen, Denmark, 2018.

M. Jessell, L. Ailleres, E. de Kemp, M. Lindsay, F. Wellmann, M. Hillier, G. Laurent, T. Carmichael, and R. Martin. Next generation three-dimensional geologic modeling and inversion. *Society of Economic Geologists Special Publication*, 18:261–272, 2014.

C. Lajaunie, G. Courrioux, and L. Manuel. Foliation fields and 3D cartography in geology: Principles of a method based on potential interpolation. *Mathematical Geology*, 29(4), 1997.

G. Laurent. Iterative thickness regularization of stratigraphic layers in discrete implicit modeling. *Mathematical Geosciences*, 48:811–833, 2016.

G. Laurent, L. Ailleres, L. Grose, G. Caumon, M. Jessell, and R. Armit. Implicit modeling of folds and overprinting deformation. *Earth and Planetary Science Letters*, 456:26–38, 2016.

D. Ledez. *Modélisation d'objets naturels par formulation implicite*. PhD thesis, Université de Lorraine-ENSG, Nancy, France, 2003.

B. Levy. *Topologie Algorithmique, Combinatoire et Plongement*. PhD thesis, Université de Lorraine-ENSG, Nancy, France, 1999.

J. Mallet. Automatic contouring in presence of discontinuities. In G. Verly, M. David, A. G. Journel, and M. Marechal, editors, *Geostatistics for natural resources characterization, Part 2*, pages 669–677. D. Reidel Publishing Company, 1984.

J.-L. Mallet. Discrete smooth interpolation. *ACM Transactions on Graphics*, 8:121–144, 1989.

J.-L. Mallet. Discrete smooth interpolation in geometric modeling. *Computer-Aided Design*, 24:178–191, 1992.

J.-L. Mallet. Discrete Modeling for Natural Objects. *Mathematical Geology*, 29(1), 1997.

J.-L. Mallet. Space-time mathematical framework for sedimentary geology. *Mathematical Geology*, 36(1), 2004.

J.-L. Mallet. *Elements of mathematical sedimentary geology: The GeoChron model*. EAGE, 2014.

R. Martin and J. Boisvert. Iterative refinement of implicit boundary models for improved geological feature reproduction. *Computers and Geosciences*, 109:1–15, 2017.

R. Moyen. *"Paramétrisation 3D de l'espace en géologie sédimentaire: le modèle GeoChron"*. PhD thesis, Université de Lorraine-ENSG, Nancy, France, 2005.

M. J. Ramón, E. L. Pueyo, G. Caumon, and J. L. Briz. Parametric unfolding of flexural folds using palaeomagnetic vectors. *Geological Society of London*, 425:247–258, 2015.

J. Renaudeau, E. Malvesin, F. Maerten, and G. Caumon. Towards continuous equations behind implicit stratigraphic modeling. In *2017 RING meeting*, Université de Lorraine-ENSG, Nancy, France, September 2017a. ASGA.

J. Renaudeau, E. Malvesin, F. Maerten, and G. Caumon. The weighted minimization criterion as a physical way to handle thickness variation in implicit structural modeling. In *2017 RING meeting*, Université de Lorraine-ENSG, Nancy, France, September 2017b. ASGA.

J. Renaudeau, M. Irakarama, G. Laurent, F. Maerten, and G. Caumon. Implicit modeling of geological structures: A Cartesian grid method handling discontinuities with ghost points. In *41st International Conference on Boundary Elements and other Mesh Reduction Methods*, New Forest, UK, 2018.

J. Renaudeau, E. Malvesin, F. Maerten, and G. Caumon. Implicit Structural Modeling by Minimization of the Bending Energy with Moving Least Squares Functions. *Mathematical Geosciences*, xxx:xxx–xxx, 2019.

W. Smith and P. Wessel. Gridding with continous curvature splines in tension. *GEOPHYSICS*, 55(3):293–305, 1990.

L. Souche, F. Lepage, and G. Iskenova. Volume Based Modeling - Automated construction of complex structural models. In *75th EAGE Conference and Exhibition*, 2013.

A.-L. Tertois. *"Création et modification de modéles géologiques par champs de potentiel. Application au modèle GeoChron"*. PhD thesis, Université de Lorraine-ENSG, Nancy, France, 2007.

G. Turk and J. O'Brien. Modelling with Implicit Surfaces that Interpolate. *ACM Transactions on Graphics*, 21:855–873, 2002.

F. Wellmann and G. Caumon. 3D Structural geological models: Concepts, methods, and uncertainties. *Advances in Geophysics*, 59:1–121, 2018.

P. Wessel and D. Bercovici. Interpolation with splines in tension: a Green's function approach. *Mathematical Geology*, 30(1), 1998.

X. Wu. Building 3D subsurface models conforming to seismic structural and stratigraphic features. *GEOPHYSICS*, 82:IM21–IM30, 2017.

X. Wu and D. Hale. Horizon volumes with interpreted constraints. *GEOPHYSICS*, 80:IM21–IM33, 2015.

Part III

Reducing Structural Uncertainties using Seismic Data

Introductory Comments

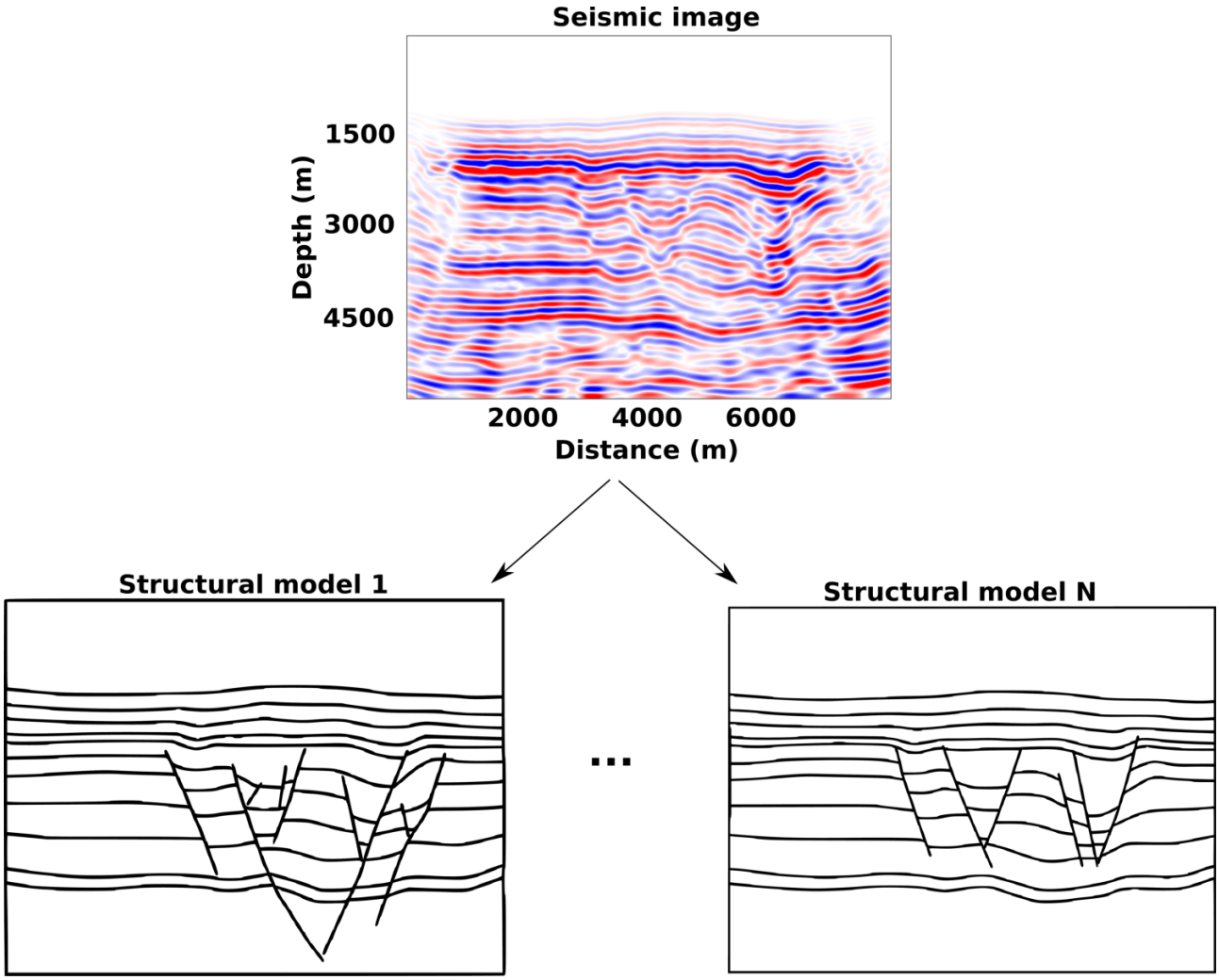

Figure 59: Illustration of structural interpretation uncertainties. The seismic image above supports N geologically consistent structural interpretations. The problem of appraising structural interpretations using seismic data consists in determining which among the N structural models are more likely than others using seismic data.

This third Part of the manuscript is devoted to reducing structural uncertainties by appraising structural models using seismic data. This problem is illustrated in Figure 59: a single seismic image supports multiple structural interpretations; can we use seismic data to determine which models are more likely than others?

Let $\mathbf{M}$ denote a structural model, and let $\mathscr{M}$ denote an arbitrary set of candidate structural models. Consider the optimization problem

$$\begin{cases} \mathbf{M}^* = \underset{\mathbf{M}}{\operatorname{argmin}}\, \Phi^d(\mathbf{M}),\ \mathbf{M} \in \mathscr{M} & (a) \\ \text{subject to} & \\ \mathbf{M}^* \text{ is geologically consistent} & (b) \\ \mathbf{M}^* \text{ is the closest model in } \mathscr{M} \text{ to the ``true subsurface geology''} & (c) \end{cases}, \tag{120}$$

where $\Phi^d(\mathbf{M})$ denotes a seismic data misfit value associated to the structural model $\mathbf{M}$. A solution to problem 120 implies a solution to the appraising structural models using seismic data problem: let $\Phi^m(\mathbf{M})$, the model misfit, be a measure of how close the structural model $\mathbf{M}$ is to the true subsurface geology; then, if problem 120 has a solution, it follows that

$$\infty > \frac{\Phi^d(\mathbf{M}_j) - \Phi^d(\mathbf{M}_i)}{\Phi^m(\mathbf{M}_j) - \Phi^m(\mathbf{M}_i)} > 0 \text{ when } i \neq j, \tag{121}$$

for any pair of models $\mathbf{M}_i, \mathbf{M}_j \in \mathscr{M}$ (this is certainly true for a set that has only two candidate models $\{\mathbf{M}_1, \mathbf{M}_2\}$; because these two models are arbitrary, it follows that it's also true for any two pairs of models $\mathbf{M}_i, \mathbf{M}_j \in \mathscr{M}$). Given a definition of Φ^m, the problem of appraising structural models using seismic data will consist in defining seismic data misfit functions Φ^d that satisfy equation 121 for all pairs of models $\mathbf{M}_i, \mathbf{M}_j \in \mathscr{M}$. Equation 121 implies that ranking models using Φ^m is equivalent to ranking models using Φ^d; therefore, if we can satisfy that equation we can rank models using seismic data (Φ^d) without any knowledge of the true subsurface geology needed to define Φ^m.

Problem 120 has several challenges:

1. The data misfit Φ^d in equation 120a implies that there exists a forward modeling operator that generates synthetic seismic data from a structural model. In this work, structural models are populated with velocities estimated from the migration velocity model, which are then used to run numerical simulations. Populating structural models with velocities from a smooth migration velocity model is not a trivial problem.

2. The constraint in equation 120c implies that we can define a distance between two models, denoted Φ^m; this is also not a trivial problem.

3. Given a definition of Φ^d and Φ^m, we still have the problem of satisfying the constraint in equation 120c, or equivalently satisfying equation 121 for any pair of models $\mathbf{M}_i, \mathbf{M}_j \in \mathscr{M}$; this is also not a trivial problem.

In this thesis, we will only propose straightforward solutions to the first and second challenge so that we can focus on the third challenge. The constraint in equation 120b will be satisfied by requiring that all structural models in $\mathscr{M}$ be constructed from structural interpretations proposed by geologists.

Problem 120 is conceptually similar to methods that have been proposed in reservoir modeling to determine reservoir models that fit production data (e.g. Suzuki et al., 2008; Cherpeau et al., 2012). What is new in problem 120, other than the use of seismic data, is the constraint in equation 120c. It is usually implicitly assumed that a model that minimizes data misfit is also the "most correct" model. However, this assumption is not necessarily true, especially in geology where we usually solve nonlinear physical problems in complex models. For example, Carter et al. (2006) show that reservoir models that match available production data may have poor predictive values.

Chapter III.1

Structural Interpretation Uncertainties

We speak of structural uncertainties when the position and/or the presence of a structural surface is uncertain on a seismic image (Thore et al., 2002). Sources of uncertainties in seismic imaging include data uncertainty and velocity uncertainty (e.g. Li et al., 2015; Pawelec, 2018), acquisition geometry (e.g. Lecomte et al., 2016), and geological complexity (e.g. Suiter et al., 2005). As an example of uncertainties due to geological complexity, consider the *Caumon salt model* in Figure 60a. The model is relatively simple from a structural geology point of view but the presence of salt and thin layer structures makes the model challenging for seismic imaging. The model was built by adding thin layer structures in the macrolayered velocity model in Figure 60b. The seismic image computed from the thinlayered model (Figure 60c) has a more realistic texture than the seismic image computed in the macrolayered model (Figure 60d); therefore, the true subsurface geology is expected to look more like the model in Figure 60a than the model in Figure 60b (Landa and Thore, 2007). When building structural models however, we are usually only interested in major layers (see for example the structural model in Figure 62). The problem of structural interpretation is therefore that of identifying the structural surfaces (faults, horizons, ...) observed in the macrolayered model in Figure 60b from the seismic image in Figure 60c. This is not entirely straightforward even for this experiment where we used the exact velocity model for seismic imaging. Consider for example the well illuminated portion of the model between the distances x = 2000m and x = 6000m in Figure 60c. While it is fairly straightforward to identify the fault, it is less obvious how to identify the major horizons and to correlate them across the fault.

When the positions and/or the presence of structural surfaces are uncertain on a given seismic image, it is usually possible to propose multiple structural interpretations from the same seismic image (e.g. Bond et al., 2007). Consider for example the seismic image in Figure 61; this image was generated from the sandbox model of Colletta et al. (1991) (see Figure 70 in chapter III.2) using the fine layering technique described in the previous paragraph. The image was then given to different interpreters who provided 7 different structural interpretations, shown in Figure 63 to Figure 69. The structural interpretation in Figure 62 was performed directly on the reference velocity model (Figure 70) in order to have one structural interpretation that we are sure is the most accurate, for benchmarking purposes. The different interpretations are geologically consistent, but because they are different, we can expect some interpretations to be more consistent than others. In this third Part of the manuscript, we are going to study the problem of determining which of those structural interpretations are more consistent than others using seismic data. In particular, we will generate synthetic data for each structural interpretation, and then compare the synthetic data with observed data in order to assign a misfit value to each interpretation.

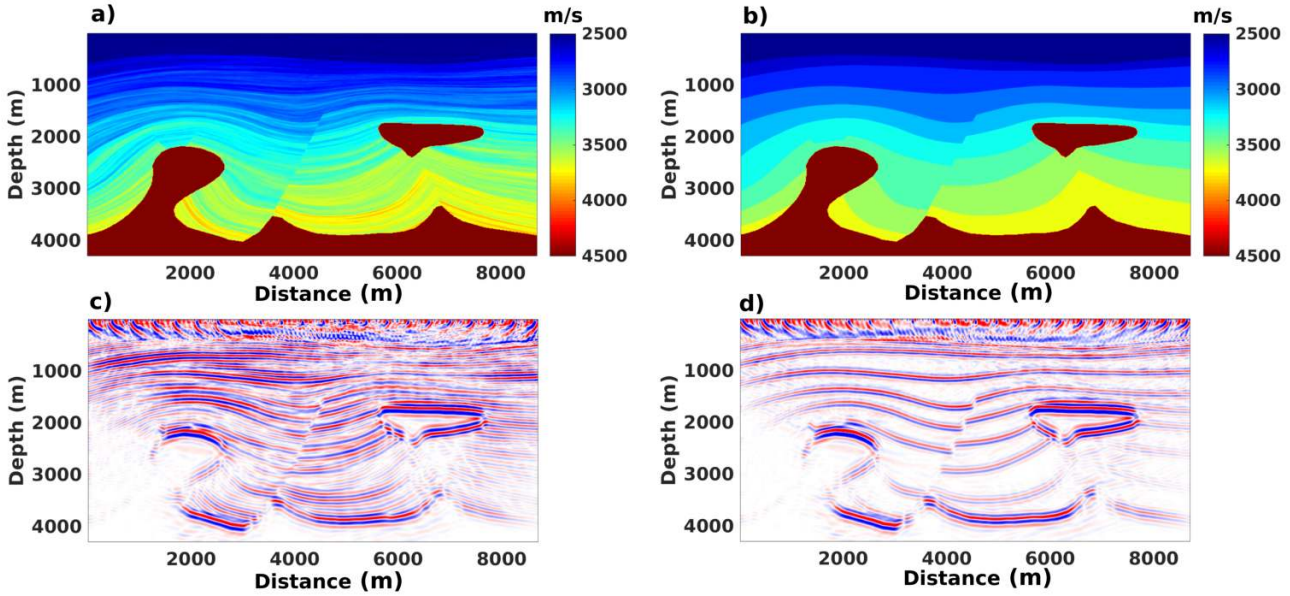

Figure 60: Seismic uncertainties due to complex geology. **a**) Thinlayered velocity model. **b**) Macrolayered velocity model used to build the thinlayered model. **c**) Seismic image (reverse time migration) computed from data generated in the thinlayered model. **d**) Seismic image (reverse time migration) computed from data generated in the macrolayered model. The problem of structural interpretation is that of identifying the structural surfaces observed in the macrolayered model b) from the seismic image in c).

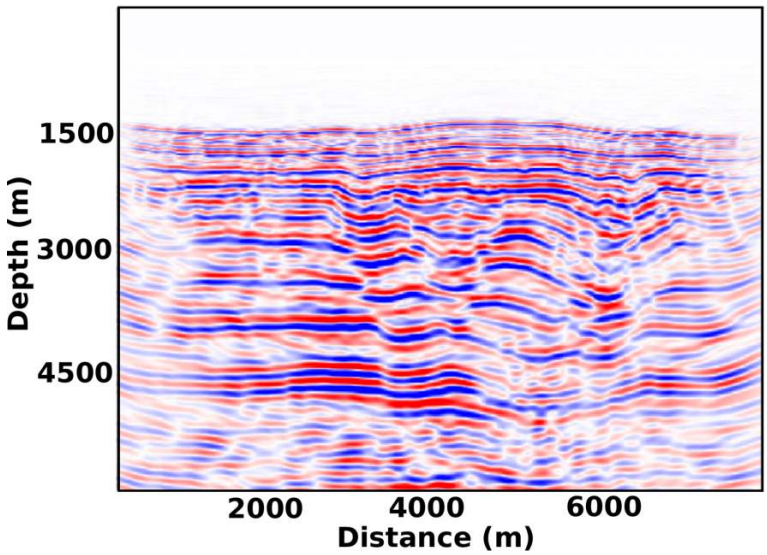

Figure 61: Seismic image generated from the sandbox model of Colletta et al. (1991)

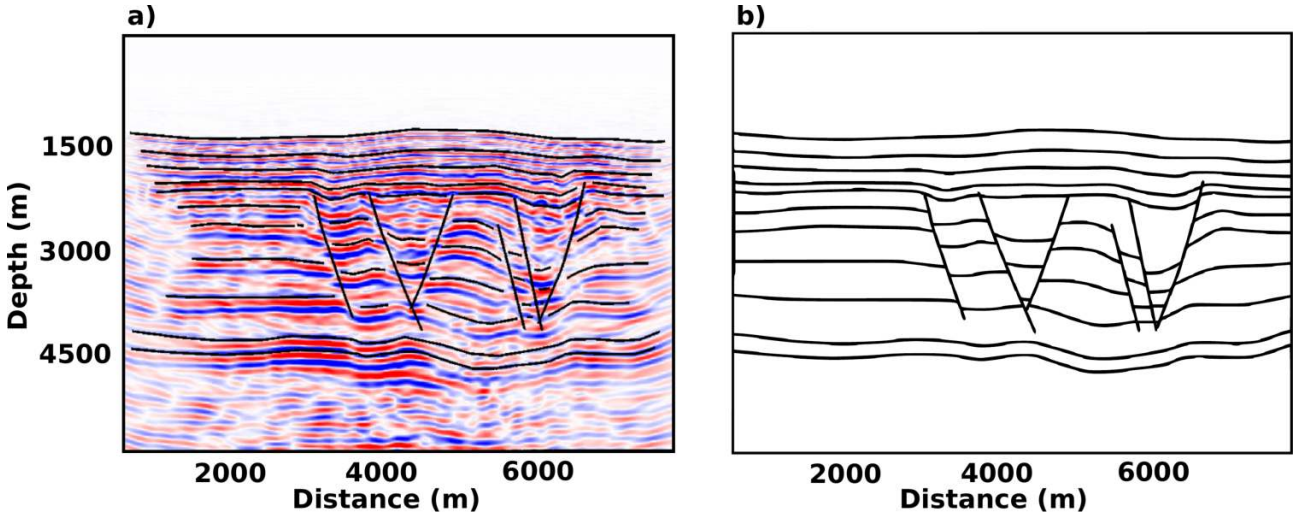

Figure 62: **a**) Structural interpretation 1. **b**) Structural model 1.

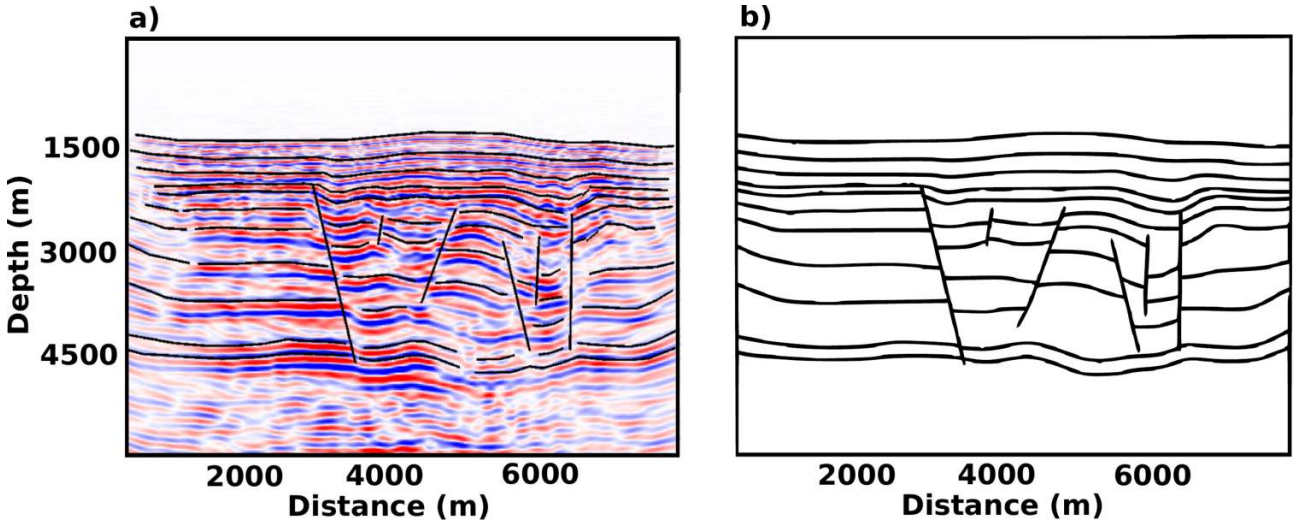

Figure 63: **a**) Structural interpretation 2. **b**) Structural model 2.

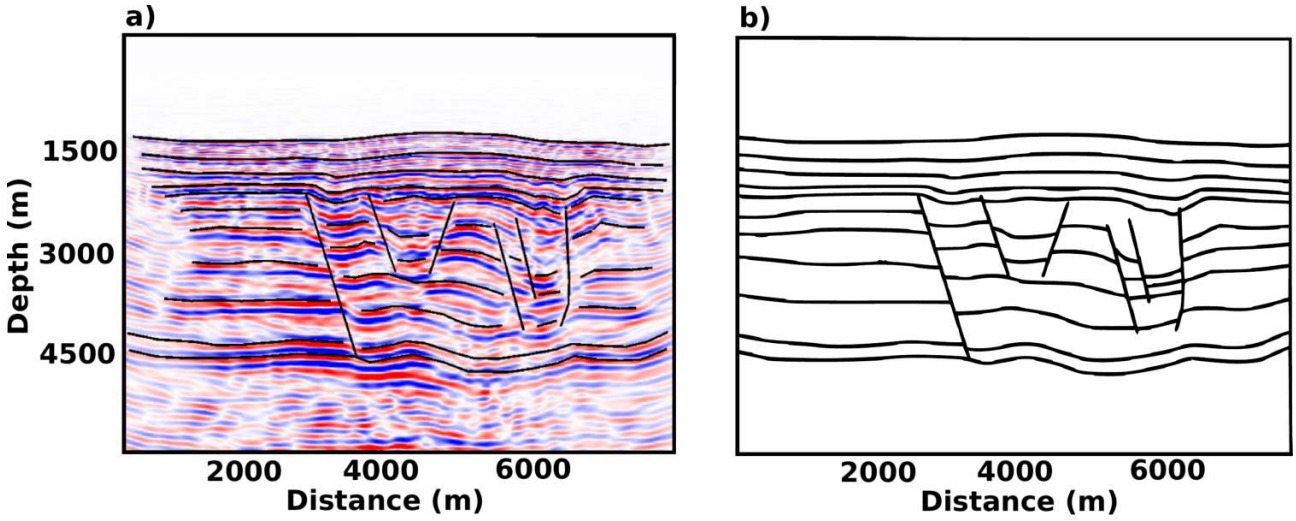

Figure 64: **a**) Structural interpretation 3. **b**) Structural model 3.

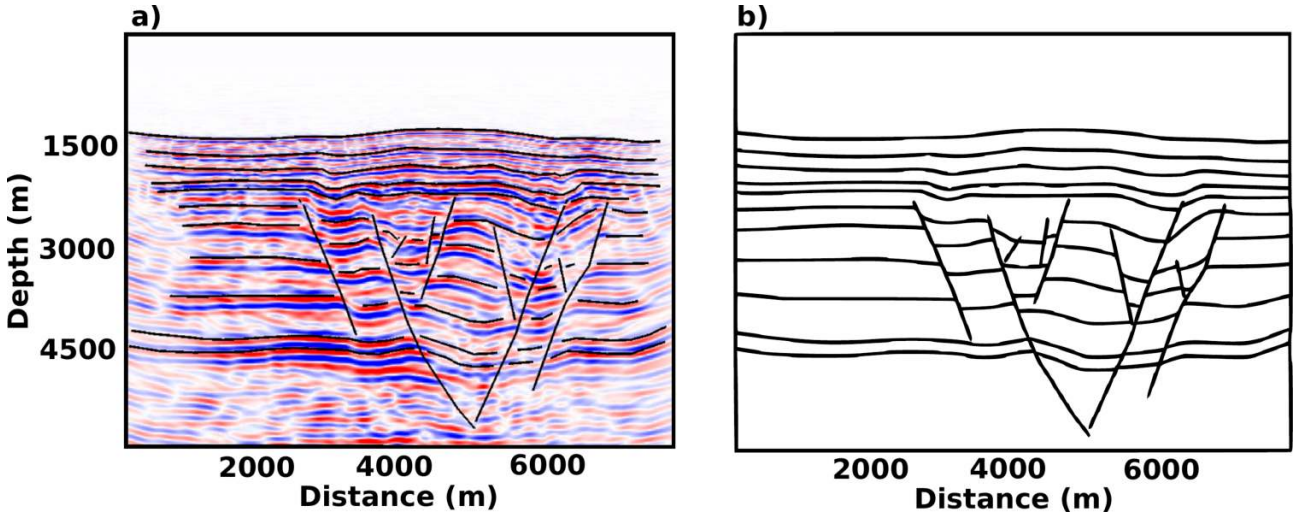

Figure 65: **a**) Structural interpretation 4. **b**) Structural model 4.

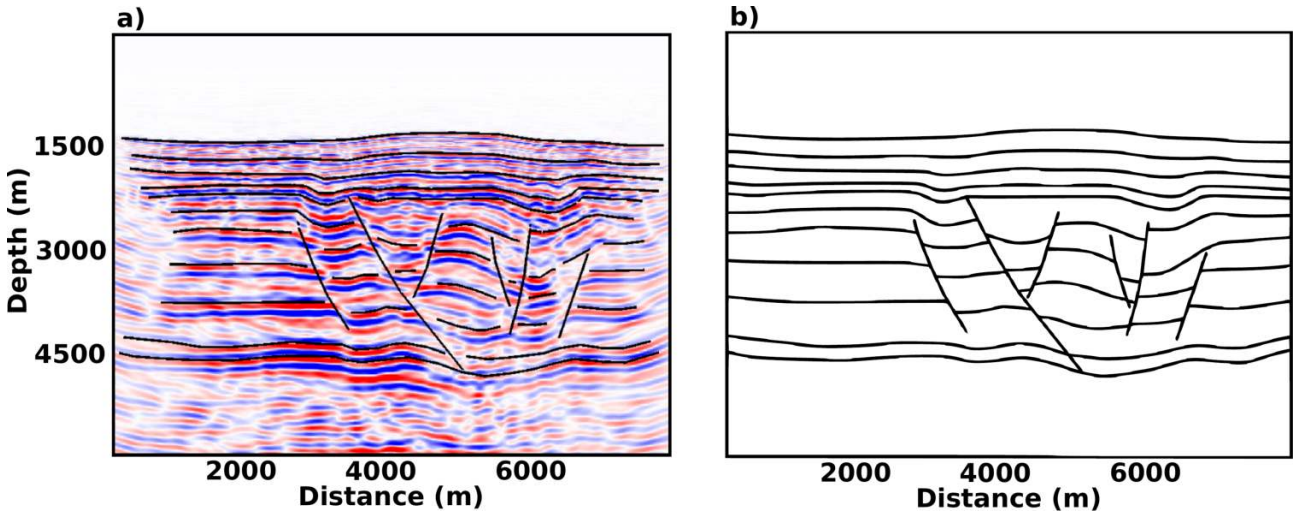

Figure 66: **a**) Structural interpretation 5. **b**) Structural model 5.

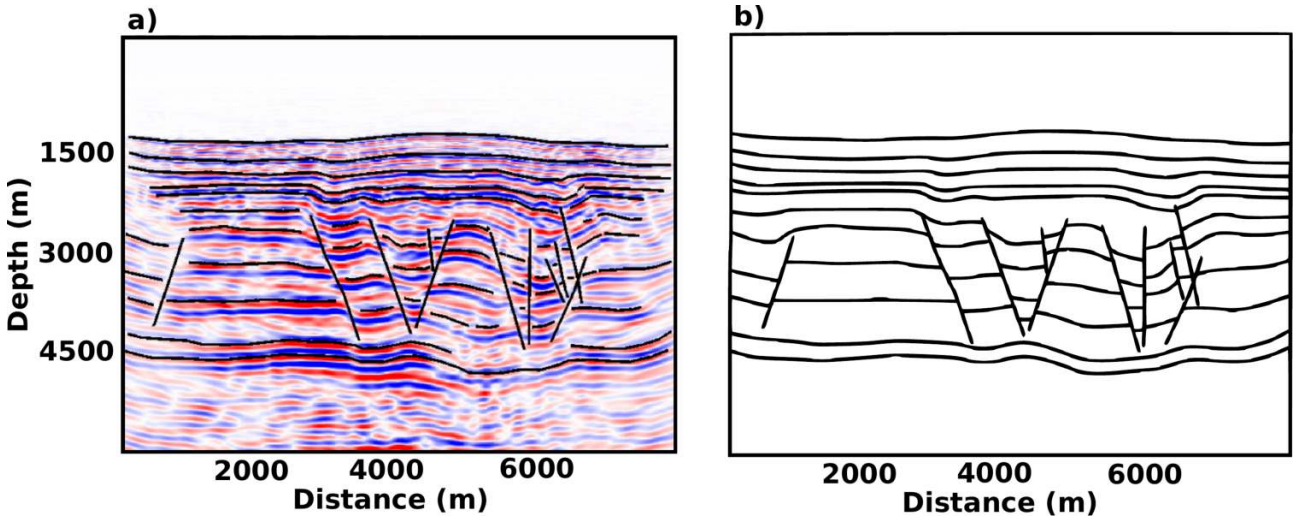

Figure 67: **a**) Structural interpretation 6. **b**) Structural model 6.

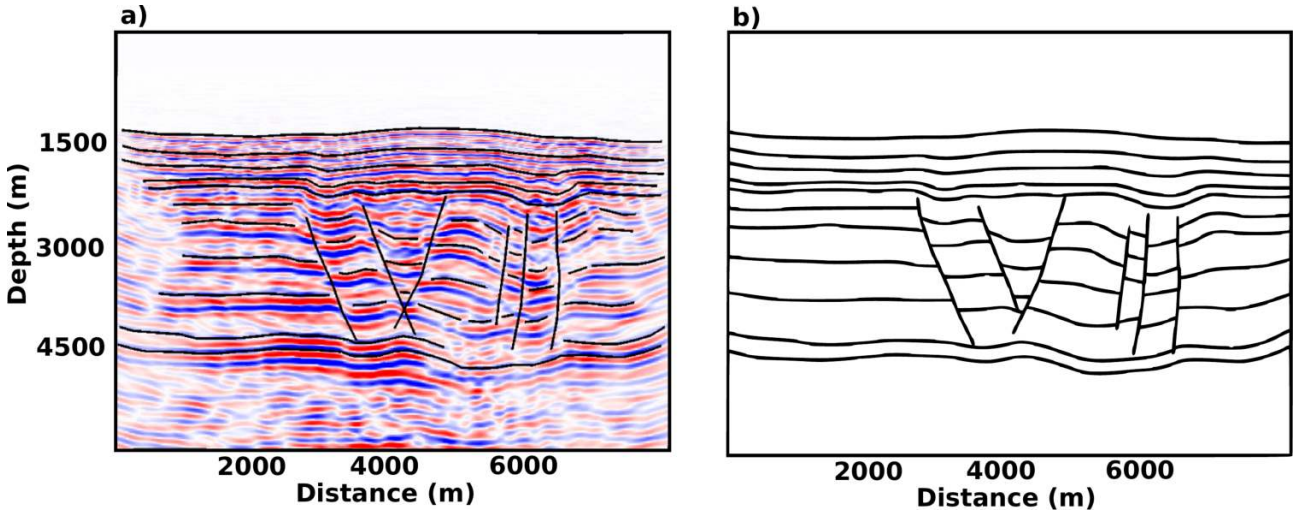

Figure 68: **a**) Structural interpretation 7. **b**) Structural model 7.

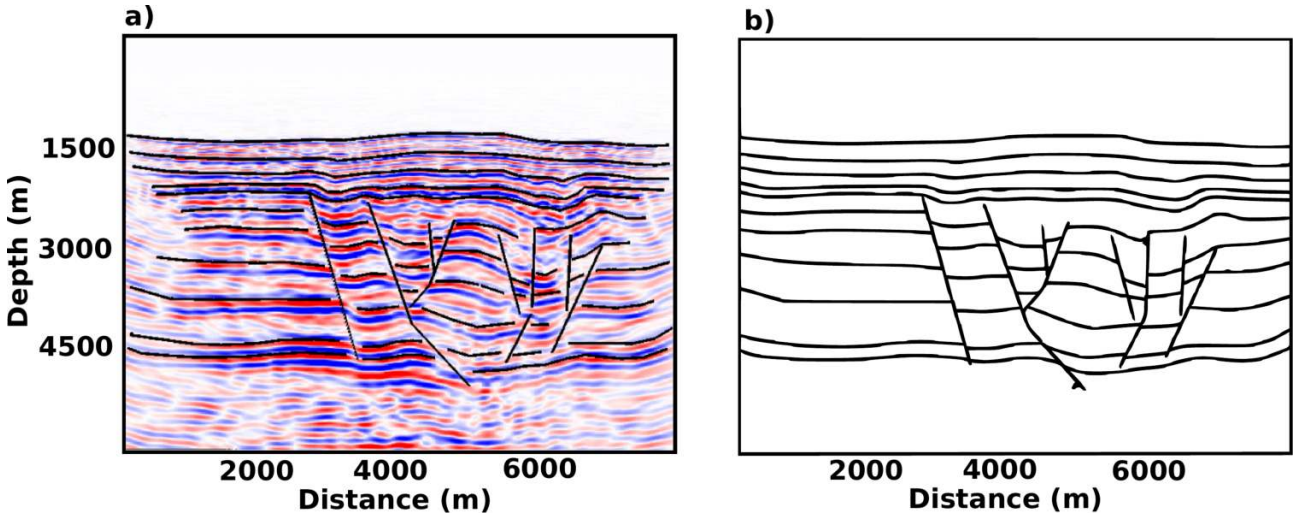

Figure 69: **a**) Structural interpretation 8. **b**) Structural model 8.

III.1.1 Acknowledgements

I would like to thank Nicolas Clausolles for building the Caumon salt model in Figure 60. I would also like to thank Christelle Butault, Francois Bonneau, Guillaume Caumon, Jonathan Edwards, and Paul Angrand, for some of the structural interpretations presented in this chapter.

Chapter III.2

Appraising Structural Models using Seismic Data — Theory

This chapter resulted from a technical report co-authored with Paul Cupillard, Guillaume Caumon, Paul Sava, and Jonathan Edwards.

III.2.1 Summary

Structural interpretation of seismic images can be highly subjective, especially in complex geological settings. A single seismic image will often support multiple geologically valid interpretations. However, it is usually difficult to determine which of those interpretations are more likely than others. We refer to this problem as *structural model appraisal* herein. In this chapter, we use misfit functions to rank and appraise multiple interpretations of a given seismic image. Given a set of possible interpretations, we compute synthetic data for each structural interpretation, then compare these synthetic data against observed seismic data; this allows us to assign a data-misfit value to each structural interpretation. Our aim is to find data-misfit functions which enable a ranking of interpretations. To do so, we formalize the problem of appraising structural interpretations using seismic data and derive a set of conditions to be satisfied by the data-misfit function for a successful appraisal. We investigate both vertical seismic profiling and surface seismic configurations. An application of the proposed method to a realistic synthetic model shows promising results for appraising structural interpretations using vertical seismic profiling data, provided the target region is well illuminated. However, we find appraising structural interpretations using surface seismic data to be more challenging, mainly due to the difficulty of computing phase-shift data-misfits.

III.2.2 Introduction

Uncertainties in structural interpretations of seismic images have been of interest to geoscientists for a long time. Earlier studies were mainly focused on migration velocity errors and seismic resolution; some examples can be found in Hajnal and Sereda (1981) and Beylkin et al. (1985). Structural uncertainties propagated from velocity errors usually translate to subtle displacement or flexing of horizons and faults away from the reference position of those horizons and faults (Thore and Haas, 1996; Bube et al., 2004; Pon and Lines, 2005; Fomel and Landa, 2014). By the *reference* position of a given horizon or fault, one often means the position obtained from the tomographic velocity model (e.g. Messud et al., 2017). Geological realizations obtained by such perturbations of a reference structural model often underestimate structural uncertainties, since interpretation uncertainties are not taken into account. Interpretation uncertainties arise where reflectors cannot be tracked deterministically; this can be a result of an inaccurate imaging velocity model (Li et al., 2015), poor illumination and/or poor resolution (Lecomte et al., 2016). In these situations, multiple geologically possible structural models can be interpreted from the same seismic image (Bond et al., 2007; Bond, 2015; Alcalde et al., 2017).

Generating multiple structural models from a single data set was one of the earliest methods proposed to evaluate structural uncertainties in geomodeling (Thore et al., 2002). This method opened ways to new challenges, two of which are :

1. *Sampling structural uncertainties.* This is essentially a structural modeling problem (Abrahamsen, 1993; Lecour et al., 2001; Holden et al., 2003; Wellmann et al., 2010, 2014; Cherpeau and Caumon, 2015; Julio et al., 2015). In particular, how can we generate multiple structural models respecting both uncertainty bounds and geological rules efficiently?

2. *Appraising structural models.* This usually amounts to evaluating the model's response for some observed physical phenomenon. In most studies on structural model appraisal, the physical phenomenon that has been used to evaluate the model is fluid flow (Suzuki et al., 2008; Cherpeau et al., 2012). Potential fields (gravity and magnetic) have also been used to validate model scenarios (Foss et al., 2008; Fullagar et al., 2008).

With standard geomodeling workflows, we are able to generate multiple geological scenarios from a seismic image, but we rarely go back to check if these models are consistent with the initial data; doing so would be a way to reduce structural uncertainty. Here, initial data refers to the recorded pre-migrated seismic traces. An interesting approach proposed by Lecomte et al. (2003) to address this issue is to generate synthetic seismic images from different geological scenarios and compare them to the initial seismic image; a similar approach was adopted by Lallier et al. (2012). A major challenge of this approach is that it is not clear how to objectively *compare* seismic images. An alternative way for reducing interpretation uncertainties in structural modeling is to use "data-driven" interpretation approaches. Here, data-driven interpretation includes all the methods that have been proposed to automatically track horizons and faults, and to build a relative-geologic-time function from a seismic image (Stark, 2004; Pauget et al., 2009; Wu and Hale, 2015; Wu, 2017). Data-driven techniques, however, require good quality seismic data (Hoyes and Cheret, 2011); where the geology is complex or lacks reflectivity contrasts, manual interpretation is still necessary.

In this chapter, we aim to reduce interpretation uncertainties by appraising structural interpretations using seismic data. Our approach is conceptually similar to that proposed by Lecomte et al. (2003): we generate synthetic data from a set of candidate geological models, then we compare the synthetic data against observed data. The candidate geological models are obtained from different interpretations of the same seismic image. There are two main differences

between the method proposed here and the method proposed by Lecomte et al. (2003). First, we compare initial pre-migrated data instead of seismic images; second, we propose a quantitative comparison instead of a qualitative comparison. The objective of this chapter is to investigate how exactly we should compare synthetic data against observed data. We mainly focus on theoretical aspects of the problem and defer the more practical issues for further investigations.

In what follows, we will introduce the problem of appraising structural interpretations using seismic data, and propose a mathematical formulation to describe the process. We then propose some strategies to build a velocity model with structural discontinuities from a structural interpretation and a (smooth) migration velocity model; the resulting velocity model with structural discontinuities is then used to generate synthetic data for the given structural interpretation. Finally, we propose a workflow to design data misfit functions in order to evaluate and rank the different structural interpretations. The proposed method is then applied on a realistic synthetic vertical seismic profiling (VSP) case.

III.2.3 The structural interpretation appraisal problem

We use the model in Figure 70 to illustrate an example of interpretation uncertainty. The model was built by assigning constant velocity values in each major layer of the sandbox model of Colletta et al. (1991). Additional thin-layering structures were added inside each major layer in order to generate more realistic synthetic seismic data (Landa and Thore, 2007). The resulting velocity model was then used to simulate "observed" seismic data using an acoustic, constant density, staggered finite-difference scheme (Virieux, 1984) with perfectly matched absorbing boundaries (Collino and Tsogka, 2001). All subsequent seismic modeling mentioned in this chapter were performed with the same code. The reference velocity model in Figure 70 was also used to obtain the migration velocity model in Figure 71a by Gaussian smoothing of the slowness; this migration velocity model was then used to migrate the "observed" data to obtain the seismic image in Figure 71b using Kirchhoff depth migration (e.g. Etgen et al., 2009).

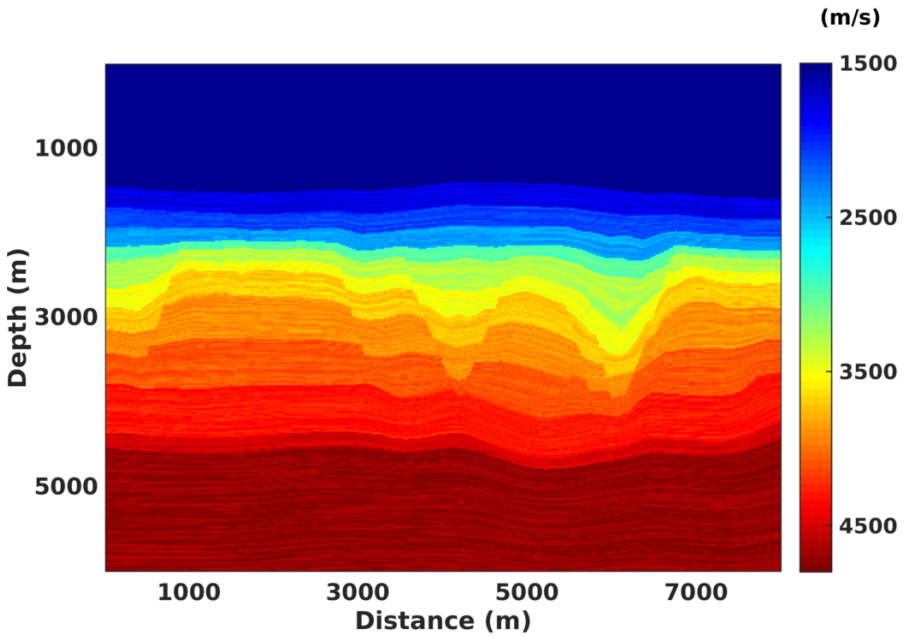

Figure 70: The reference velocity model. It is derived from the sandbox model of Colletta et al. (1991).

Figures 72a and 72b show two different structural interpretations of the seismic image in Figure

71b from two different interpreters. Both interpretations are geologically possible, but they are different in some regions. Assuming that one is more accurate than the other in the regions where they differ, can we use seismic data to determine which of the two models is more accurate in those regions? To answer this question, we rely on *macro-layered velocity models*. Macro-layered velocity models are defined as velocity models with discontinuities from structural models and velocity values from the migration model (e.g. Figure 72c and Figure 72d). The migration velocity model is assumed to be kinematically accurate throughout this chapter. We use these macro-layered velocity models to compute synthetic data, which are compared against the observed data to appraise the different geological scenarios.

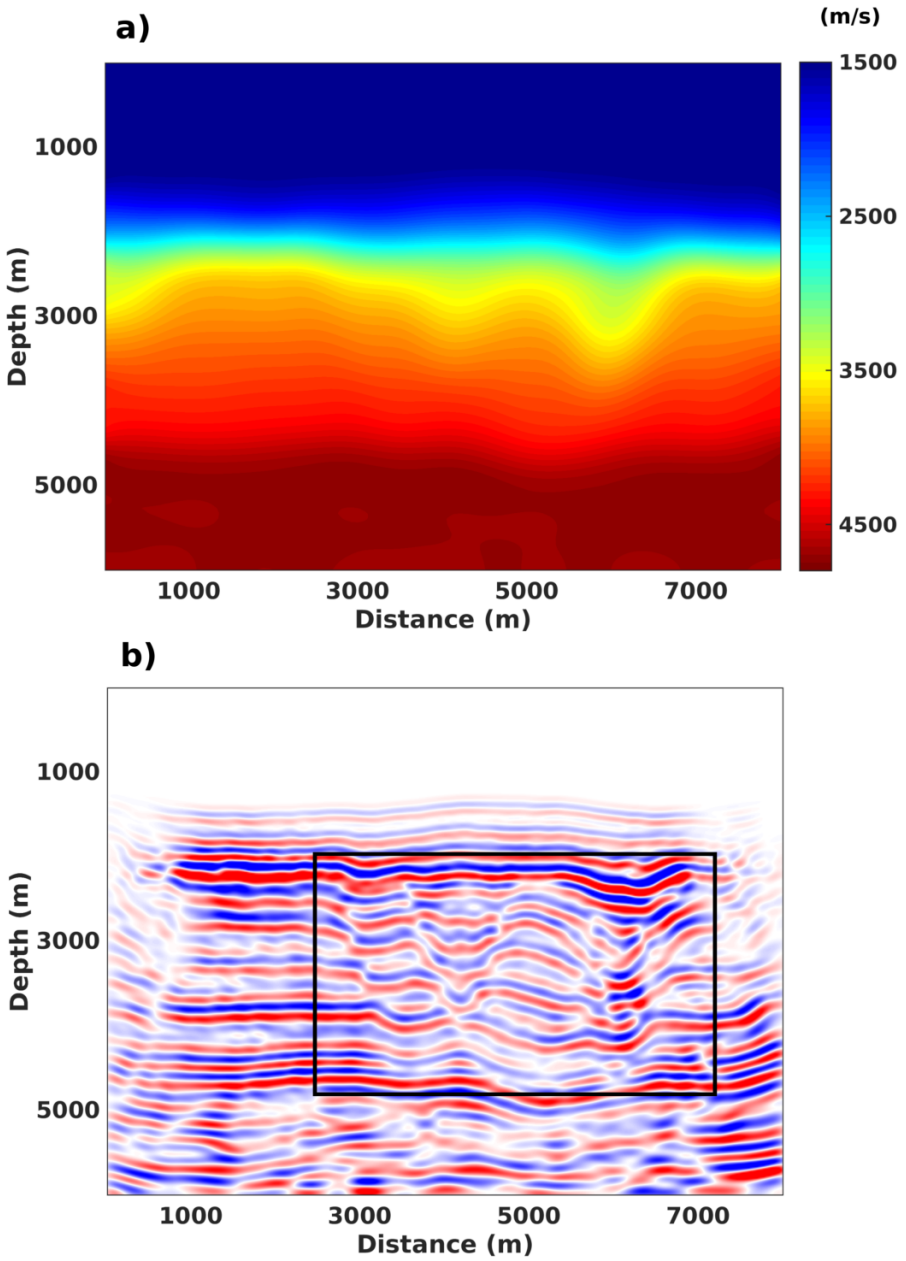

Figure 71: **a**) Migration velocity model obtained by smoothing the slowness of the reference model in Figure 70. **b**) Depth migrated image (Kirchhoff depth migration) of data computed from the reference model. The black box is the region of interest.

Structural interpretations are typically segments or curves picked by interpreters along struc-

tural discontinuities on a seismic image. A structural model is a set of consistent structural surfaces, such as horizons and faults, that represent a geological model (Caumon et al., 2009) (e.g. Figure 72a and Figure 72b). In this chapter, structural models are built from structural interpretations and there is a one-to-one relation from structural interpretations to structural models. Therefore, appraising structural interpretations is synonymous with appraising structural models herein. The relation from structural models to macro-layered velocity models is assumed to be one-to-one as well.

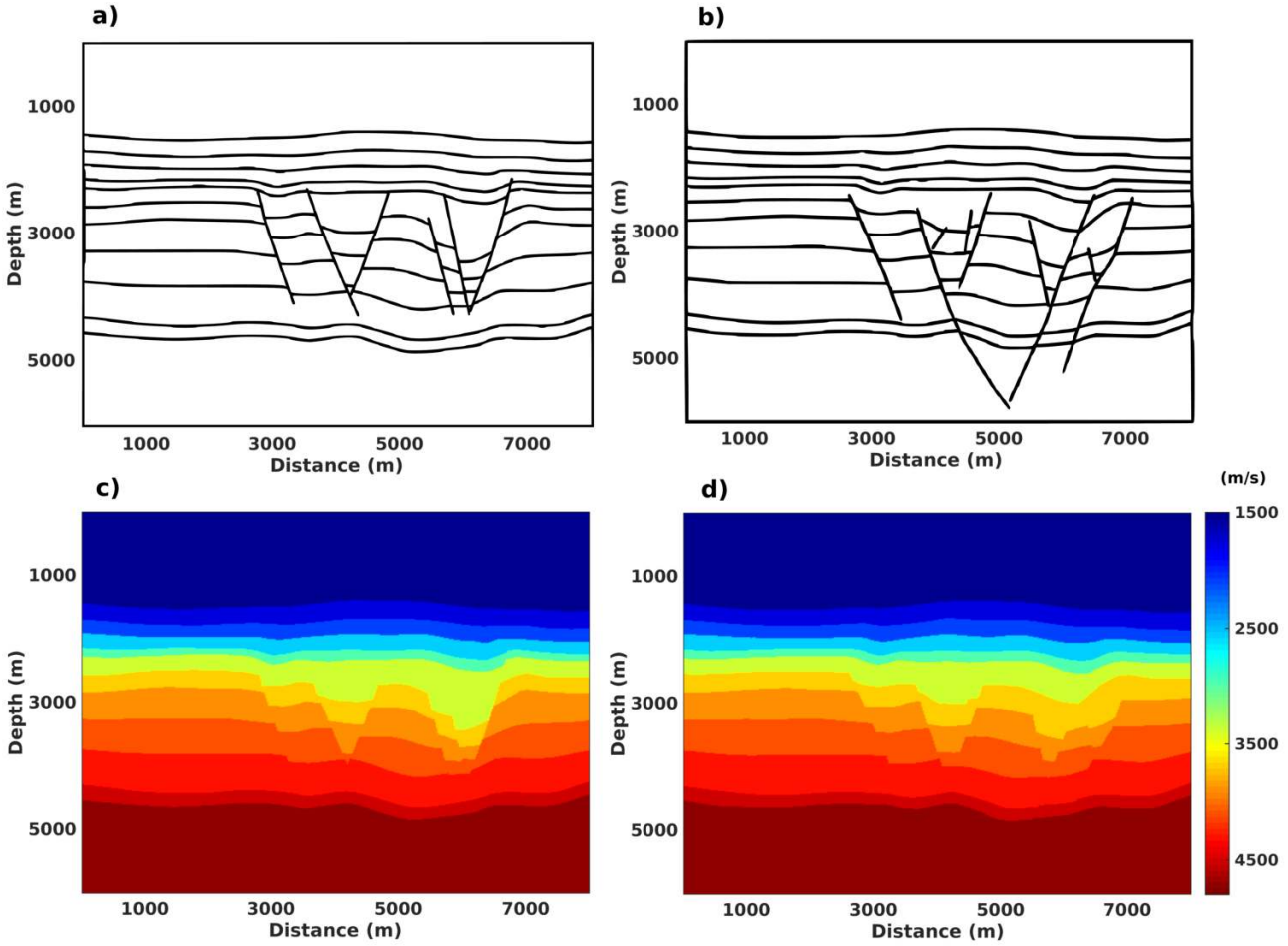

Figure 72: **a**, **b**) Two possible structural interpretations/models of Figure 71b from different interpreters. **c**, **d**) Macro-layered velocity models built from the migration velocity model and the structural models.

Let us define a *model-misfit* as a number that quantifies the mismatch between the reference structural model and a candidate structural model. The reference structural model is the "true" structural model; it is unknown in practice. However, for the sake of argument, we assume the reference model to be known for now. Let us also assume that we can determine this model-misfit value for each model in a given set of candidate structural models. Furthermore, we assume that different models will have different model-misfit values, allowing us to rank those models from the best model to the worst model. This ranking of structural models using model-misfit values will be referred to as *model-space ranking* of structural models. Objects in the model-space are structural interpretations, or alternatively structural models. We also define a *data-misfit* as a number that quantifies the mismatch between observed data and the synthetic data generated from candidate structural models. Data-misfit values will allow us to rank structural interpretations in the data-space; we refer to this as *data-space ranking*. The

data-space here refers to the pre-migration domain; i.e., objects in the data-space are seismic traces. We assume that the observed data are the wave equation's response of the reference model (true earth) subjected to some acquisition geometry, just like synthetic data are the wave equation's response of a given candidate (synthetic) model. If the ranking in data-space is the same as the ranking in model-space, then we do not need to know the reference model to determine which among our candidate models are more probable than others. For this reasoning to be applicable, the following conditions have to be satisfied:

1. Different candidate structural models should, overall, have different data-misfit values.
2. Data-misfit values should, overall, be rank-correlated to model-space misfit values.

In practice, a pair of different structural models may exhibit similar seismic response at receivers; in this case, the first condition can be satisfied by ignoring one of the models. The second condition means that if we plot model-misfits and data-misfits on a scatter plot, we should be able to find a monotonic curve that fits the data.

In general, it is straightforward to design data-misfit functions and compute data misfit-values. However, it is much harder to design data-misfits that satisfy the conditions mentioned above. The objective of this chapter is to investigate what it takes to define data-misfit functions that honor the conditions mentioned above as much as possible, and thus run the process with real data, where the reference (true) model truly is unknown.

III.2.4 Mathematical formulation

Let $\mathbf{M}_i$ denote a candidate structural model (e.g. Figure 72a and Figure 72b), the index i taking different values for different interpretation scenarios. Each structural model $\mathbf{M}_i$ can be used, together with the migration velocity model, to build a macro-layered velocity model, also denoted by $\mathbf{M}_i$. Examples of macro-layered velocity models will be presented in the next section (see also Figure 72c and Figure 72d).

Let us introduce the *model-space misfit function* $\Phi^m_{i,j}$ that measures the difference between two structural models $\mathbf{M}_i$ and $\mathbf{M}_j$, and denote the mismatch between the reference model and a candidate model $\mathbf{M}_i$ as $\Phi^m_i = \Phi^m_{i,ref}$. Similarly, we introduce a *data-space misfit* $\Phi^d_{i,j}$ that measures the difference between data computed in two macro-layered models $\mathbf{M}_i$ and $\mathbf{M}_j$, and denote the mismatch between data computed in the macro-layered model $\mathbf{M}_i$ and the observed (reference) data as $\Phi^d_i = \Phi^d_{i,ref}$. Two models $\mathbf{M}_i$ and $\mathbf{M}_j$ are said to be *data-different* if $|\Phi^d_i - \Phi^d_j| > \epsilon$, for some "small" user-defined threshold $\epsilon > 0$.

Let $\mathscr{M}$ be a set of data-different structural models. If Φ^d_i and Φ^m_i are perfectly rank-correlated for all models in $\mathscr{M}$, then ranking models in $\mathscr{M}$ using Φ^d_i is the same as ranking models in $\mathscr{M}$ using Φ^m_i. In that case, by definition, the structural models in $\mathscr{M}$ can be appraised (i.e. ranked) using seismic data. Therefore, given a set of macro-layered velocity models $\mathscr{M}$, we consider the problem of appraising structural models using seismic data to be solvable in $\mathscr{M}$ if the following conditions are satisfied:

Condition 1. $|\Phi^d_j - \Phi^d_i| > \epsilon$ *when* $i \neq j$.

Condition 2. $\infty > \frac{\Phi^d_j - \Phi^d_i}{\Phi^m_j - \Phi^m_i} > 0$ *when* $i \neq j$.

Given a set of random structural models $\mathscr{R}$, we can build a set of data-different structural models $\mathscr{M}$ by progressively moving each model $\mathbf{M}_i$ from $\mathscr{R}$ to $\mathscr{M}$, when $\mathbf{M}_i$ is data-different from all the models already in $\mathscr{M}$. Therefore, condition 1 can be satisfied in practice. We

focus on condition 2 hereafter. Let us define $\Phi_i^d(\Omega)$ and $\Phi_i^m(\Omega)$ respectively as the data-space and model-space misfit values corresponding to a specific region Ω in the candidate model $\mathbf{M}_i$. The reason for localizing misfits in space will be justified later. In that case, two necessary conditions, although not sufficient, for (the localized version of) condition 2 to be satisfied are:

Condition 3. *A change of* $\Phi_i^d(\Omega) \Rightarrow$ *a change of* $\Phi_i^m(\Omega)$.

Condition 4. *A change of* $\Phi_i^m(\Omega) \Rightarrow$ *a change of* $\Phi_i^d(\Omega)$.

The symbol $\Rightarrow$ stands for "implies". Furthermore, suppose that Φ^m satisfies the following condition:

Condition 5. *A change of* $\Phi_i^m(\Omega) \Leftrightarrow$ *a change of the structural model* $\mathbf{M}_i$ *in* Ω.

Then conditions 3 and 4 become:

Condition 6. *A change of* $\Phi_i^d(\Omega) \Rightarrow$ *a change of the structural model* $\mathbf{M}_i$ *in* Ω.

Condition 7. *A change of the structural model* $\mathbf{M}_i$ *in* $\Omega \Rightarrow$ *a change of* $\Phi_i^d(\Omega)$.

At this point, we have transformed the problem of appraising structural models using seismic data to that of satisfying one condition on Φ^m (condition 5) and three conditions on Φ^d (conditions 1, 6, and 7). The conditions imposed on Φ^d can be checked in practice as they do not depend on the (unknown) reference model. Note that condition 2 implies that we assume the relation between the model and data to be weakly nonlinear [1] , i.e., we assume the perturbations in conditions 3 and 4 to be "small". In other words, we assume, to some extent, that all the candidate structural models are "close" to the (unknown) reference model.

Our goal will be to design Φ^d such that conditions 6 and 7 are honored as much as possible. We will argue that for VSP data-misfits, condition 7 is relatively easy to satisfy whereas condition 6 is practically impossible to satisfy in general. As for surface seismic data-misfit functions, we will argue that if all the candidate macro-layered velocity models are kinematically equivalent to the migration velocity model, itself assumed to be kinematically accurate, both conditions 6 and 7 can be satisfied in theory for stratified (i.e. layered) models; however, it remains challenging to design and compute misfit function satisfying these conditions. These difficulties make the ideal condition 2 too strict in practice; we replace it with the less ambitious condition:

Condition 8. Φ^d *is statistically rank-correlated with* Φ^m.

Condition 2 means that the rank correlation coefficient of the variables Φ^m and Φ^d is 1, while condition 8 means that it is "high enough".

III.2.5 Macro-layered velocity models

We propose several options for building macro-layered velocity models from a structural interpretation and a migration model. In this section, we use the illustrative model in Figure 73a as the reference model, and the model in Figure 73b as the migration model; the migration model was obtained by smoothing the slowness of the reference model.

[1] Condition 2 implies that the data misfit function has no local minima; therefore, the relation between data and model cannot be highly nonlinear. However, because the ratio in condition 2 need not be constant, the relation between data and model need not be linear either.

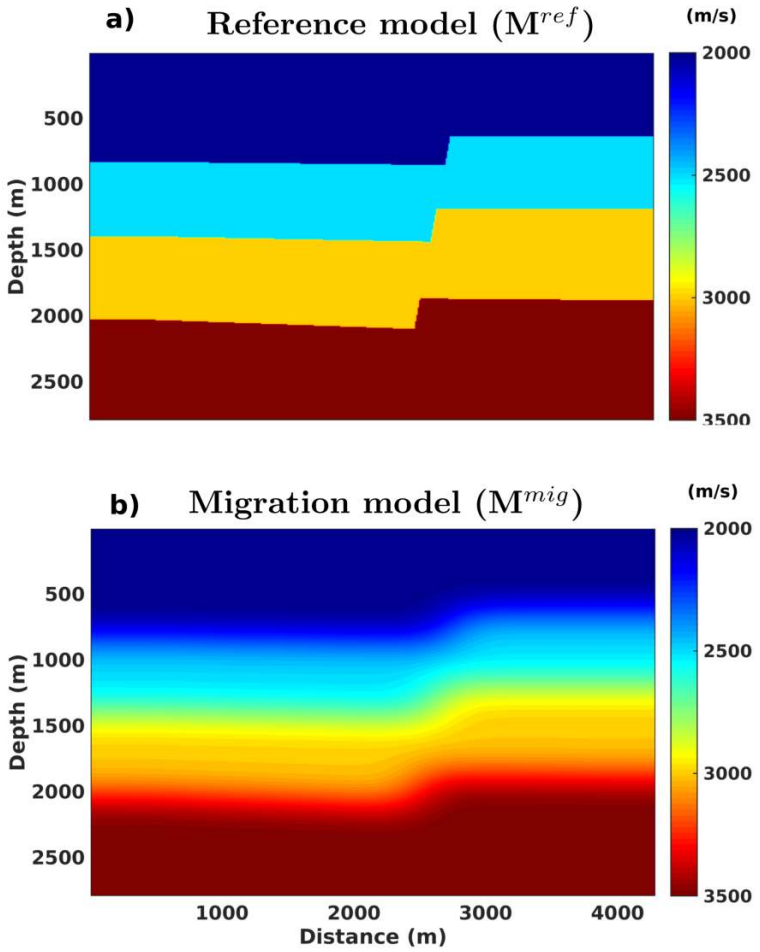

Figure 73: **a**) Reference model ($\mathbf{M}^{ref}$). **b**) Migration model ($\mathbf{M}^{mig}$) obtained by smoothing the slowness of the reference model.

III.2.5.1 Block macro-layered velocity model

The first type of macro-layered model we propose is obtained by averaging migration slowness values (Figure 73b) in each layer of a structural model. Averaging slowness values preserves traveltimes, as opposed to averaging velocities. The velocity of the i^{th} layer is therefore given by

$$v_i = \text{mean}\left[\left\{\frac{1}{\mathbf{M}^{mig}(\mathbf{x})}\,\middle|\,\mathbf{x} \text{ in layer } i\right\}\right]. \tag{122}$$

This results in a *block* velocity model, denoted $\mathbf{M}^b$ hereafter, as illustrated in Figure 74a. Figure 74b shows velocity model residuals obtained by subtracting the reference model from the block macro-layered model. Well-log information, when available, can also be used to constrain velocity values assigned in each layer of block macro-layered models. Block macro-layered velocity models are not in general kinematically equivalent to the migration velocity model.

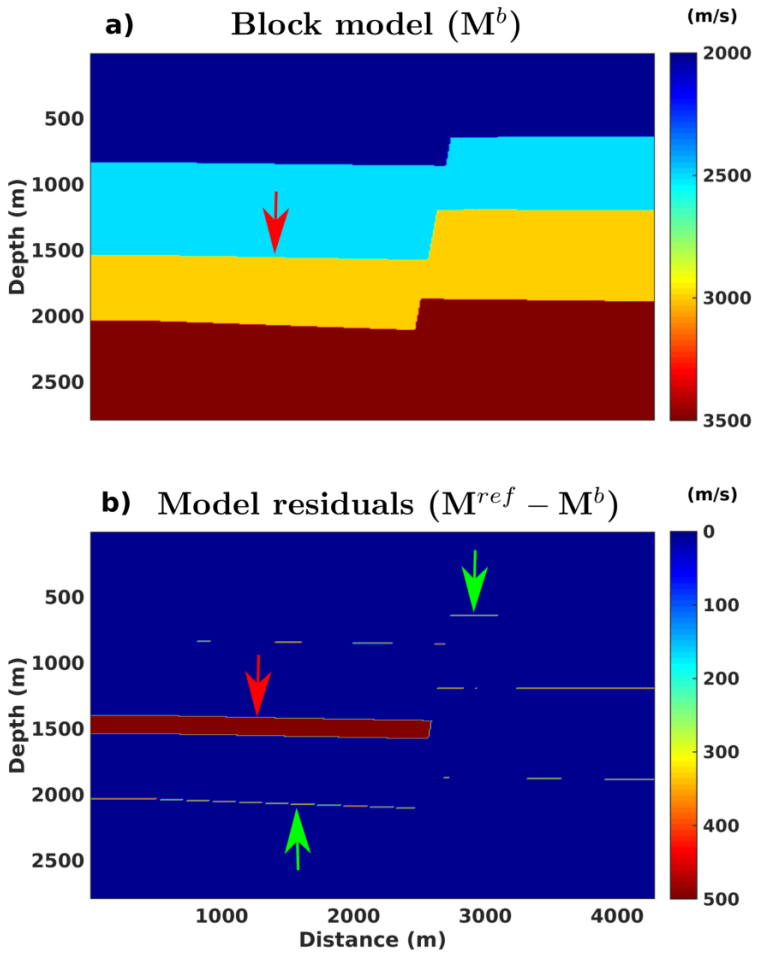

Figure 74: **a**) A candidate interpretation represented as a block macro-layered velocity model ($\mathbf{M}^b$) built using the migration model in Figure 73b, as described by equation 122. The red arrow identifies the part of the horizon that was mispositioned during interpretation.**b**) Model residual, difference between a) and the reference model in Figure 73a. The red arrow identifies the interpretation error while the green arrows identify picking errors.

III.2.5.2 Wire macro-layered velocity model

The second type of macro-layered model is the *wire* velocity model, $\mathbf{M}^w$, illustrated in Figure 75. This model is obtained by

$$\mathbf{M}^w(\mathbf{x}) = \mathbf{M}^{mig}(\mathbf{x}) + \Delta\mathbf{M}^b(\mathbf{x}), \tag{123}$$

where $\Delta\mathbf{M}^b(\mathbf{x})$ denotes the velocity contrast at point $\mathbf{x}$; it is equal to zero everywhere except at block boundaries where it is equal to the velocity jump between the velocities in adjacent blocks. Wire macro-layered velocity models are kinematically equivalent to the migration velocity model everywhere expect at geological interfaces.

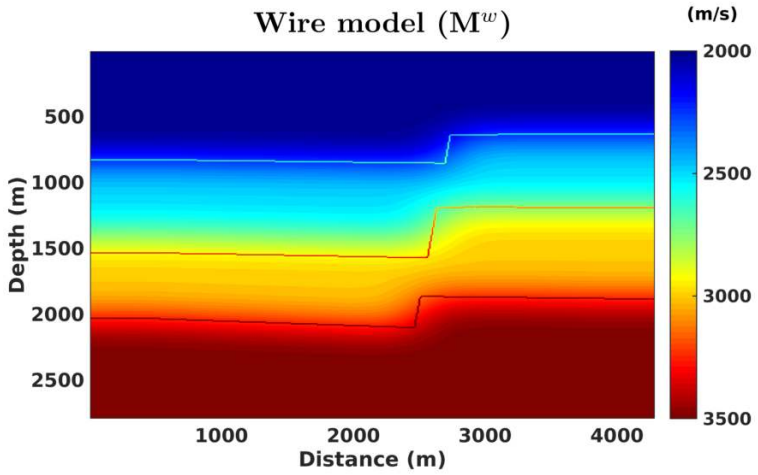

Figure 75: The same candidate interpretation from Figure 74a represented using a wire macro-layered model ($\mathbf{M}^w$), as described by equation 123.

III.2.5.3 Reflectivity macro-layered velocity model

We also propose the *reflectivity* velocity model, $\mathbf{M}^r$, illustrated in Figure 76 and obtained by

$$\mathbf{M}^r(\mathbf{x}) = \mathbf{M}^{mig}(\mathbf{x}) + \left[\, \mathbf{M}^b(\mathbf{x}) - \bar{\mathbf{M}}^b(\mathbf{x}) \,\right], \tag{124}$$

where $\bar{\mathbf{M}}^b$ is a smooth velocity model obtained by smoothing the slowness of $\mathbf{M}^b$. The idea behind the reflectivity macro-layered model comes from the Born approximation which approximates the true velocity model by the sum of a smooth background model plus a *reflectivity function* (Bleistein et al., 2001, chapter 2); the method proposed here attempts to replace the interpretation-dependent smooth background model $\bar{\mathbf{M}}^b$ with the migration velocity model $\mathbf{M}^{mig}$.

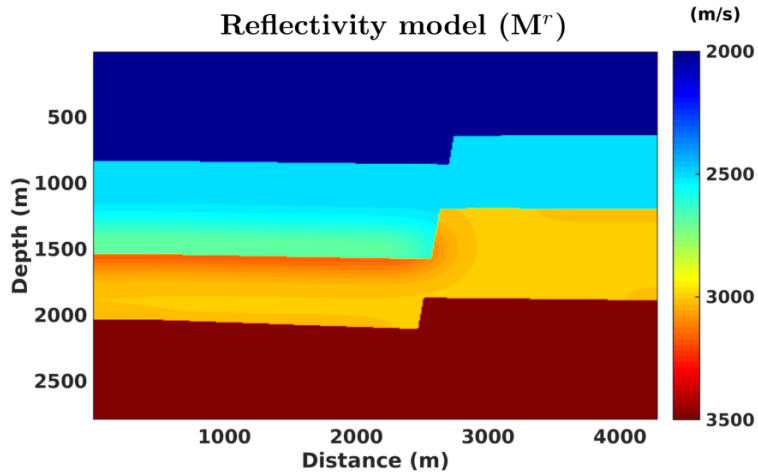

Figure 76: The same candidate interpretation from Figure 74a represented using a reflectivity macro-layered model ($\mathbf{M}^r$), as described by equation 124.

III.2.5.4 Density macro-layered velocity models

Finally, we propose a variable density macro-layered model, $\mathbf{M}^d$, given by the pair $(\mathbf{M}^{mig}, \mathbf{M}^*)$ where $\mathbf{M}^*$ is a density model estimated from $\mathbf{M}^b$, by Gardner's law for example (Gardner et al., 1974). The variable density model allows us to propose different macro-layered velocity models

that are kinematically equivalent to the migration model by proposing different impedance models.

III.2.6 Data-space misfit functions

Seismic data can only contain structural information about subsurface regions that have been illuminated by the wavefield recorded at receivers. Therefore, a data-misfit value computed for a given structural interpretation will only be representative of the illuminated region, and not of the entire model. Furthermore, a misfit value is a scalar, so it summarizes all the interpretation errors in the model in just one value. As a consequence, the best information a global misfit can offer is to determine which of two models is better than the other, but not why (i.e., where the errors are coming from). This motivates the need to look for misfit functions that are localized in space, i.e., data-misfit functions that are a function of space. It is important that we know where the errors are coming from in our model because we might be able to update our structural interpretation in that region to lower the misfit. With this in mind, we propose a data-misfit function of the form

$$\Phi_i^d(\Omega) = \sum_{\mathbf{s},\mathbf{r}} \sum_{\mathbf{x}\in\Omega} \omega(\mathbf{s},\mathbf{r},\mathbf{x}) \chi\big[f_o(\mathbf{s},\mathbf{r},t), f_i(\mathbf{s},\mathbf{r},t)\big], \tag{125}$$

where Ω is the region in the model we would like to evaluate, ω is a weighting function depending on the illumination in Ω, $f_o(\mathbf{s},\mathbf{r})$ is an observed seismogram from a shot fired at $\mathbf{x}_s = \mathbf{s}$ and recorded at a receiver at $\mathbf{x}_r = \mathbf{r}$, $f_i(\mathbf{s},\mathbf{r})$ is the corresponding synthetic seismogram computed in the candidate macro-layered velocity model $\mathbf{M}_i$, and χ is a *data residual function* computing the mismatch between observed and synthetic data. We will consider one region of interest Ω at a time; this region can have an arbitrary shape, provided it is illuminated by at least one source-receiver raypath. The data-misfit 125 can be written in the more compact form

$$\Phi_i^d(\Omega) = \sum_{\mathbf{x}\in\Omega} \mathbf{W}\Big[\chi\big[f_o(\mathbf{s},\mathbf{r},t), f_i(\mathbf{s},\mathbf{r},t)\big]\Big], \tag{126}$$

where it is now more apparent that the data-misfit function 126 acts in two steps: first, the residual function χ maps the data plane $\mathbb{D}\times\mathbb{D}$ to a *reduced data-space* $\mathbb{D}'$; second, the weighting operator $\mathbf{W}$ maps the reduced data-space $\mathbb{D}'$ to the *image-space* $\mathbb{I}$. Here, the image-space refers to the range of any seismic imaging operator; it is usually referred to as the *image-domain*. The reduced data-space is, by definition, the range of the residual function χ; a concrete example of a reduced data-space will be presented in the following paragraph. Therefore, if we can find an appropriate linear relation $\mathbf{L}$ that maps $\mathbb{I}$ to $\mathbb{D}'$, the weight function ω can be determined from $\mathbf{L}^* : \mathbb{D}' \to \mathbb{I}$, where $\mathbf{L}^* = \mathbf{W}$ is either the adjoint or the pseudo-inverse of $\mathbf{L}$. In this chapter, we limit ourselves to the case where $\mathbf{L}^*$ is the adjoint and defer the possibility of $\mathbf{L}^*$ being the pseudo-inverse for further investigations. In summary, we propose the following steps for using the localized data-misfit function 126 defined above:

1. Find an appropriate residual function $\chi : \mathbb{D}\times\mathbb{D} \to \mathbb{D}'$.
2. Find an appropriate linear map $\mathbf{L} : \mathbb{I} \to \mathbb{D}'$.
3. Find the adjoint $\mathbf{L}^* : \mathbb{D}' \to \mathbb{I}$.
4. Determine the weight function ω from $\mathbf{L}^* = \mathbf{W}$.

III.2.6.1 Example for vertical seismic profiling data

A common choice of $\chi : \mathbb{D} \times \mathbb{D} \rightarrow \mathbb{D}'$ for VSP data is the L^2-norm of data windowed around the first-arrival pick (Pratt and Shipp, 1999):

$$\chi\big[f_o(\mathbf{s},\mathbf{r},t), f_i(\mathbf{s},\mathbf{r},t)\big] = \sum_{t=\tau(\mathbf{s},\mathbf{r})+\Delta t}^{t=\tau(\mathbf{s},\mathbf{r})-\Delta t} \big[f_o(\mathbf{s},\mathbf{r},t) - f_i(\mathbf{s},\mathbf{r},t)\big]^2 = f(\mathbf{s},\mathbf{r})\,, \tag{127}$$

where $\tau(\mathbf{s},\mathbf{r})$ is the traveltime from $\mathbf{s}$ to $\mathbf{r}$, and Δt is half the time-window size. In this case, an object in the reduced data-space $\mathbb{D}'$ is a function of the source and receiver coordinates $(\mathbf{s},\mathbf{r})$; therefore, we need to look for $\mathbf{L} : f(\mathbf{x}) \rightarrow f(\mathbf{s},\mathbf{r})$. Let us choose $\mathbf{L}$ to be the integration of traveltime along a ray from the source to the receiver:

$$\mathbf{L}[m(\mathbf{x})] = \sum_{\mathbf{x}} dl(\mathbf{s},\mathbf{r},\mathbf{x})[m(\mathbf{x})] = \tau(\mathbf{s},\mathbf{r})\,,$$

where $dl(\mathbf{s},\mathbf{r},\mathbf{x})$ is a ray-segment centered at $\mathbf{x}$ along the ray, and $m(\mathbf{x})$ is the slowness model at $\mathbf{x}$. The adjoint $\mathbf{L}^*$ is readily available (e.g. Christensen, 2010, chapter 4):

$$\begin{aligned}
\Big\langle \mathbf{L}\big[f(\mathbf{x})\big], f(\mathbf{s},\mathbf{r}) \Big\rangle_{\mathbb{D}'} &= \sum_{\mathbf{s},\mathbf{r}} \Big(\sum_{\mathbf{x}} dl(\mathbf{s},\mathbf{r},\mathbf{x}) f(\mathbf{x}) \Big) f(\mathbf{s},\mathbf{r}) \\
&= \sum_{\mathbf{x}} f(\mathbf{x}) \Big(\sum_{\mathbf{s},\mathbf{r}} dl(\mathbf{s},\mathbf{r},\mathbf{x}) f(\mathbf{s},\mathbf{r}) \Big) \\
&= \Big\langle f(\mathbf{x}), \mathbf{L}^*\big[f(\mathbf{s},\mathbf{r})\big] \Big\rangle_{\mathbb{I}}\,;
\end{aligned} \tag{128}$$

$< \cdot,\cdot >_{\mathbb{I}}$ denotes an inner product defined in $\mathbb{I}$. By comparing $\mathbf{L}^*$ as defined in equation 128 to equation 125, we can identify $\omega(\mathbf{s},\mathbf{r},\mathbf{x}) = dl(\mathbf{s},\mathbf{r},\mathbf{x})$. Because the rays can always be interpolated such that all ray-segments $dl(\mathbf{s},\mathbf{r},\mathbf{x}) = 1$, we conclude that

$$w(\mathbf{s},\mathbf{r},\mathbf{x}) = \left| \begin{array}{ll} 1 & \text{if } \mathbf{x} \text{ is along the ray connecting } \mathbf{s} \text{ and } \mathbf{r}, \\ 0 & \text{otherwise .} \end{array} \right. \tag{129}$$

This implies that, for VSP, we can also rewrite the data-misfit 125 as

$$\Phi_i^d(\Omega) = \sum_{\mathbf{s},\mathbf{r}} \omega(\mathbf{s},\mathbf{r},\Omega)\chi\big[f_o(\mathbf{s},\mathbf{r},t), f_i(\mathbf{s},\mathbf{r},t)\big]\,, \tag{130}$$

where $\omega(\mathbf{s},\mathbf{r},\Omega)$ is the length of the ray-segment that actually goes through Ω.

The cartoon in Figure 77 shows the geometrical interpretation of using the data-misfit 125 along with equation 127 and equation 129. The data-misfit is projecting data residuals along rays that illuminate the region of interest Ω; this projection (of data residuals into the image-space) is equivalent to a generalized Radon transform back-projection (e.g. Toft, 1996, chapter 4). Figure 77a shows rays emanating from sources at the free surface to receivers in a well. The green raypaths illuminate the region of interest Ω, while the black raypaths do not; only the green raypaths will actually contribute to the data-misfit 125. However, the data-misfit is still not localized enough: errors are projected along entire rays since every point $\mathbf{x}$ along the green rays has $\omega(\mathbf{s},\mathbf{r},\mathbf{x}) = 1$, according to equation 129. Therefore, for χ and ω as given by equation 127 and equation 129, respectively, it is impossible to limit the contribution of the data-misfit 125 to points $\mathbf{x} \in \Omega$ in the most general case. This is a violation of condition 6. Figure 77b shows how we can localize the misfit better by adding an additional well in order to illuminate

the region of interest from a different direction. In this case, points $\mathbf{x}$ along the green rays have $\omega(\mathbf{s},\mathbf{r},\mathbf{x}) = 2$ if $\mathbf{x} \in \Omega$ (i.e. where the two family of raypaths cross each other), otherwise $\omega(\mathbf{s},\mathbf{r},\mathbf{x}) = 1$.

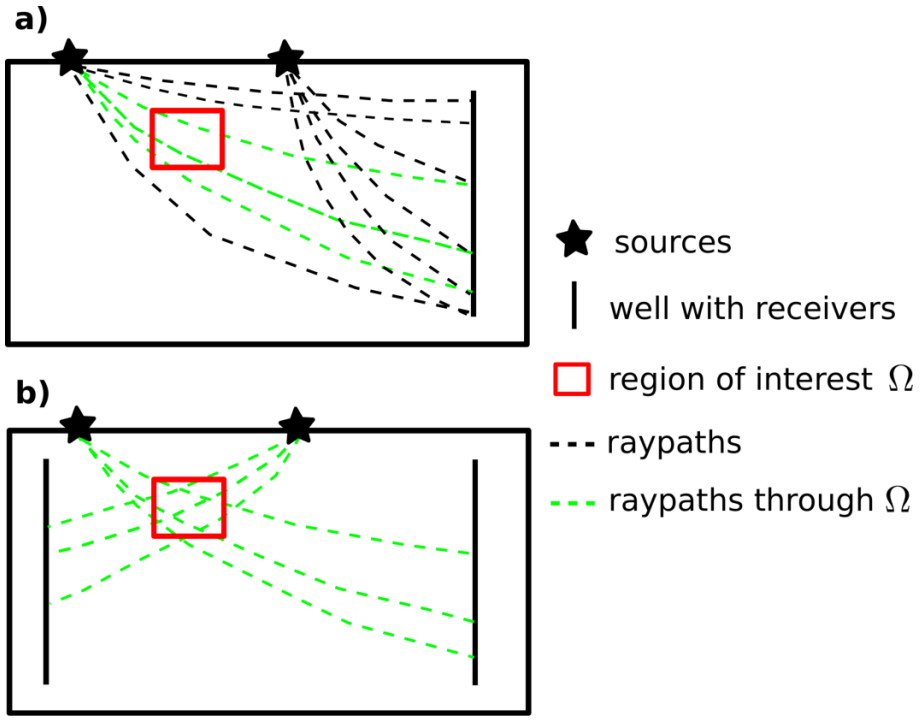

Figure 77: Geometrical interpretation of the VSP misfit weight function. **a**) Receivers are positioned only in one well. The green rays correspond to source-receiver pairs that will actually contribute to the data-misfit 125. According to equation 129, data-misfit values are projected along entire rays since every point $\mathbf{x}$ along the green rays has $\omega(\mathbf{s},\mathbf{r},\mathbf{x}) = 1$. **b**) Receivers are positioned in two wells, thus illuminating the region of interest from different directions. Now points $\mathbf{x}$ along the green rays have $\omega(\mathbf{s},\mathbf{r},\mathbf{x}) = 2$ if $\mathbf{x} \in \Omega$, otherwise $\omega(\mathbf{s},\mathbf{r},\mathbf{x}) = 1$.

III.2.6.2 Example for surface seismic data

A common choice of $\chi : \mathbb{D} \times \mathbb{D} \to \mathbb{D}'$ for surface data is the $L^p(p \in [1,2])$ difference data residuals (Engquist and Froese, 2014), i.e.

$$\chi\big[f_o(\mathbf{s},\mathbf{r},t), f_i(\mathbf{s},\mathbf{r},t)\big] = |f_o(\mathbf{s},\mathbf{r},t) - f_i(\mathbf{s},\mathbf{r},t)|^p. \tag{131}$$

Our numerical experiments suggest that this is not the best choice of χ for appraising structural interpretations using reflection data. Consider for example the reference and interpreted models in Figure 73a and Figure 74a, respectively; reflection shot-gathers from each of these models are shown in Figure 78a and Figure 78b. In Figure 74b, we distinguish interpretation errors from picking errors. Interpretation errors typically result from mispositioning geological interfaces due to poor resolution of a seismic image, while picking errors result from mispositioning geological interfaces due to the fact that it is rarely possible to perfectly track manually a reflector on a seismic image. Looking at the model-space residuals (Figure 74b), picking errors are practically negligible, while in the data-space (Figure 78c) picking errors are as strong as interpretation errors. This is problematic because picking errors are inevitable in practice. In Figure 78c, the picking errors identified by the top green arrow are strong because of the high amplitude of early arrivals; the picking errors identified by the bottom green arrow are important because of the velocity error introduced by the interpretation error above them. This second point motivates the use of macro-layered velocity models that are kinematically equivalent to the migration velocity model when appraising structural interpretations using surface

data: if a macro-layered velocity model introduces a velocity error, then all reflectors below this error will be mispositioned in the data-space even if they were positioned at the right place in the image-space by the interpreter. This eventually leads to a violation of condition 6 after back-projecting data errors into the image-space. An illustration of these problems is presented in Figure 79 where we compare individual traces. Figure 79a compares a trace from Figure 78a against a trace from Figure 78b; the difference between these two traces is shown in Figure 79b, where it can be noted that picking errors (δp) are as strong as errors due to mispositioning of horizons (δh). Figure 79c shows a similar experiment but using synthetic data computed in the more kinematically accurate reflectivity macro-layered model in Figure 76; in this case, the additional velocity errors are removed as highlighted by the green circle in Figure 79c.

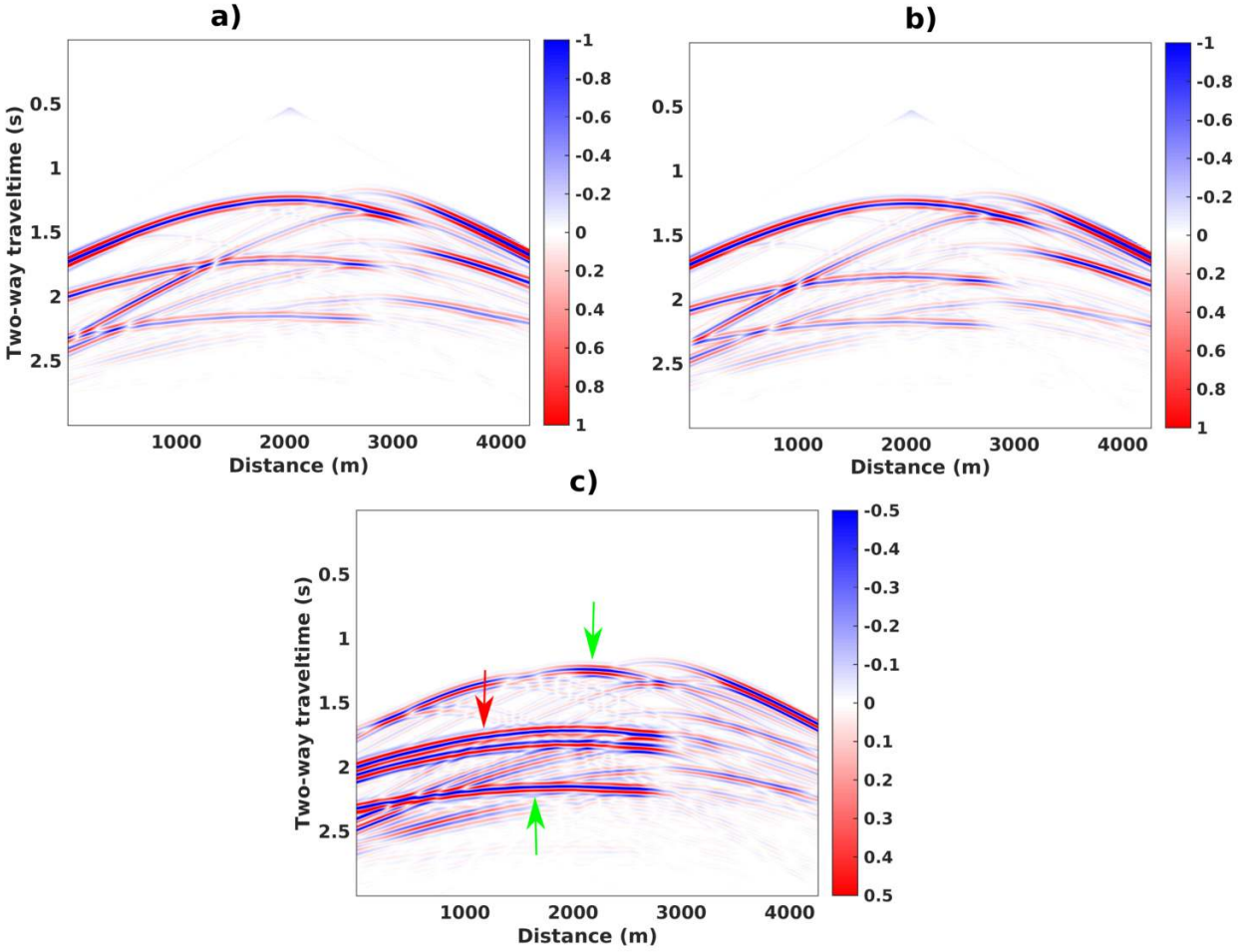

Figure 78: **a**) "Observed" data simulated in the model of Figure 73a. **b**) Synthetic data simulated in the model of Figure 74a. **c**) Data residuals, difference between observed and synthetic data. The red arrow identifies residuals due to the interpretation error shown in Figure 74b; the green arrows identify residuals due to picking errors, also shown in Figure 74b.

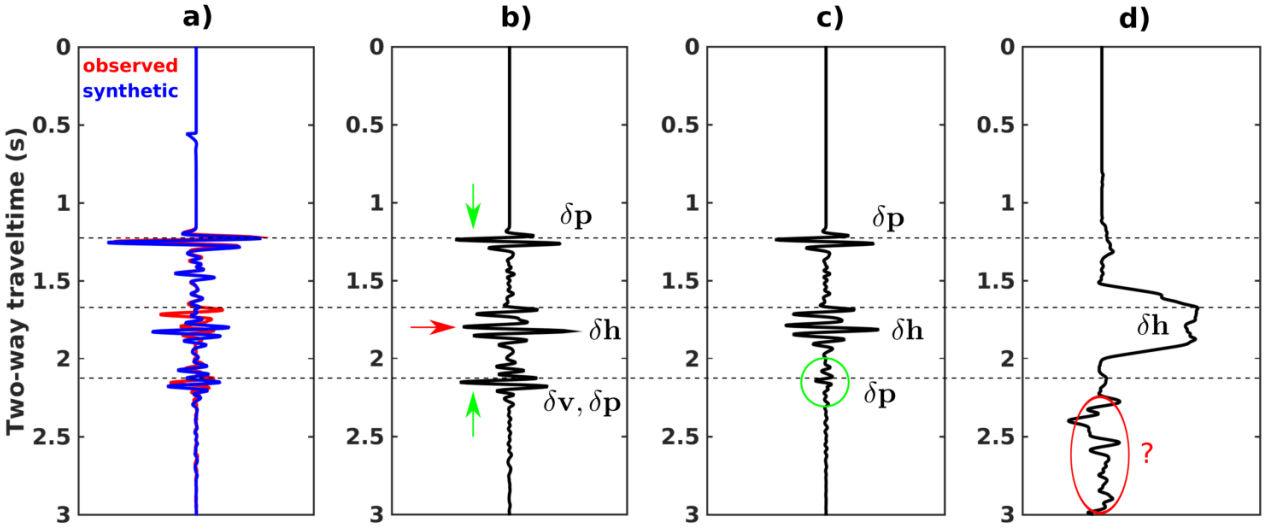

Figure 79: Data residuals of two traces from Figure 78. **a**) Comparison between observed (red) against synthetic (blue) traces. **b**) Difference between observed and synthetic traces. The top green arrow identifies picking errors (δp), the bottom green arrow identifies picking errors in addition to velocity errors (δv) in the macro-layered model in Figure 74a, and the red arrow identifies horizon mispositioning error (δh). **c**) Same as b) but using synthetic data from the more kinematically accurate macro-layered model in Figure 76. Velocity errors are removed as highlighted by the green circle. **d**) Dynamic-time-warping phase residual using synthetic data from the model in Figure 76.

In addition to being very sensitive to picking errors, equation 131 also suffers from cycle-skipping: if a horizon's position is progressively shifted away from its true position, there is a limit beyond which a data-misfit computed using equation 131 will no longer depend on the horizon's position (Engquist and Froese, 2014). This is a violation of condition 7. For instance, the data-misfit $\phi_i^d(\mathbf{s},\mathbf{r}) = \sum_t \chi[f_o(\mathbf{s},\mathbf{r},t), f_i(\mathbf{s},\mathbf{r},t)]$ between the two traces in Figure 79a would not change significantly if the mispositioned horizon in Figure 74a was shifted further down because the corresponding reflection-pulse no longer overlaps with the reference one. We therefore conclude that the $L^p(p \in [1,2])$ seismogram difference is not the best choice for χ. In principle, phase data residuals can overcome the aforementioned problems. Here, a phase residual refers to any function of time that measures phase-shifts between two signals. Such residual functions include dynamic-warping-based techniques (Hale, 2013), and optimal-transport-based techniques (Engquist and Froese, 2014). The dynamic-time-warping (DTW) phase residual $\delta\phi(\mathbf{s},\mathbf{r},t)$ of $f_o(\mathbf{s},\mathbf{r},t)$ and $f_i(\mathbf{s},\mathbf{r},t)$ is defined as (Hale, 2013):

$$\delta\phi(\mathbf{s},\mathbf{r},t) \text{ such that } \sum_t \left| f_o\big(\mathbf{s},\mathbf{r},t\big) - f_i\big(\mathbf{s},\mathbf{r},t+\delta\phi(\mathbf{s},\mathbf{r},t)\big)\right|^2 \text{ is minimized.} \tag{132}$$

Figure 79d shows the data residuals of the two signals in Figure 79a using this choice of χ. We note that such a misfit is more sensitive to horizon mispositioning errors, and less sensitive to picking errors. However, such a misfit can have some undesired effects as highlighted by the red ellipse. When using equation 131 or 132, an object in the reduced data-space $\mathbb{D}'$ is a function of time as well as source and receiver coordinates, that is $\mathbb{D}' = \mathbb{D}$; therefore we need to look for $\mathbf{L} : f(\mathbf{x}) \rightarrow f(\mathbf{s},\mathbf{r},t)$. For simplicity, let us choose $\mathbf{L}$ to be a Kirchhoff-type modeling operator

$$\mathbf{L}\big[m(\mathbf{x})\big] = \sum_{\mathbf{x}} \alpha(\mathbf{s},\mathbf{x},\mathbf{r}) S\big(t-\tau(\mathbf{s},\mathbf{x},\mathbf{r})\big)\big[m(\mathbf{x})\big] = f(\mathbf{s},\mathbf{r},t)\,, \tag{133}$$

where $f(\mathbf{s},\mathbf{r},t)$ is the computed trace recorded at receiver $\mathbf{r}$ from a shot at $\mathbf{s}$; $S(t)$ is the source wavelet; $\alpha(\mathbf{s},\mathbf{x},\mathbf{r})$ are "appropriate" migration weights as may be found, for example, in Cohen

et al. (1986); $\tau(\mathbf{s},\mathbf{x},\mathbf{r})$ is the traveltime for a ray originating from $\mathbf{s}$ scattered off $\mathbf{x}$ and recorded at $\mathbf{r}$; m is the reflectivity model. The adjoint $\mathbf{L}^*$ is readily available:

$$\begin{aligned}\Big\langle \mathbf{L}\big[f(\mathbf{x})\big], f(\mathbf{s},\mathbf{r},t)\Big\rangle_{\mathbb{D}'=\mathbb{D}} &= \sum_{\mathbf{s},\mathbf{r},t}\Big(\sum_{\mathbf{x}}\alpha(\mathbf{s},\mathbf{x},\mathbf{r})S\big(t-\tau(\mathbf{s},\mathbf{x},\mathbf{r})\big)f(\mathbf{x})\Big)f(\mathbf{s},\mathbf{r},t)\\ &= \sum_{\mathbf{x}} f(\mathbf{x})\Big(\sum_{\mathbf{s},\mathbf{r},t}\alpha(\mathbf{s},\mathbf{x},\mathbf{r})S\big(t-\tau(\mathbf{s},\mathbf{x},\mathbf{r})\big)f(\mathbf{s},\mathbf{r},t)\Big)\\ &= \Big\langle f(\mathbf{x}), \mathbf{L}^*\big[f(\mathbf{s},\mathbf{r},t)\big]\Big\rangle_{\mathbb{I}}. \end{aligned} \tag{134}$$

$\mathbf{L}^*$ as defined in equation 134 differs from the standard Kirchhoff depth migration (KDM) in that it does not have a derivative along the vertical axis. One often defines

$$\mathrm{KDM}\big[f(\mathbf{s},\mathbf{r},t)\big] = \frac{\partial}{\partial z}\mathbf{L}^*\big[f(\mathbf{s},\mathbf{r},t)\big]$$

(Schneider, 1978), or

$$\mathrm{KDM}\big[f(\mathbf{s},\mathbf{r},t)\big] = \mathbf{L}^*\big[\frac{\partial}{\partial t}f(\mathbf{s},\mathbf{r},t)\big] \tag{135}$$

(Santos et al., 2000); we use this latter definition of KDM hereafter.

By comparing $\mathbf{L}^*$ as defined in equation 134 to equation 125, we can identify

$$\omega(\mathbf{s},\mathbf{r},\mathbf{x}) = \sum_t \alpha(\mathbf{s},\mathbf{x},\mathbf{r})S\big(t-\tau(\mathbf{s},\mathbf{x},\mathbf{r})\big). \tag{136}$$

The interpretation of this result is that the localized data-misfit 125 merely amounts to migrating data residuals for surface data. Figure 80a shows the $\mathbf{L}^*$ projection of L^1 data residuals of waveforms computed in the model in Figure 73a and waveforms computed in the model in Figure 74a; notable contribution of picking errors (green arrows) and velocity errors (red arrow) are observed. Figure 80b shows the $\mathbf{L}^*$ projection of L^1 data residuals of waveforms computed in the model in Figure 73a and waveforms computed in the model in Figure 76; velocity errors are dramatically reduced since the model in Figure 76 is more kinematically accurate than the model in Figure 74a. Figure 81a shows the KDM of DTW data residuals of waveforms computed in the model in Figure 73a and waveforms computed in the model in Figure 76. We observe that high-frequency noise is present and that the residual map has two poles: positive contribution above the black line, negative contribution below the black line. Both the high-frequency noise and the bipolarity of projected residuals are a footprint of the derivative along the vertical axis in KDM, and can be removed by using the $\mathbf{L}^*$ projection as shown in Figure 81b.

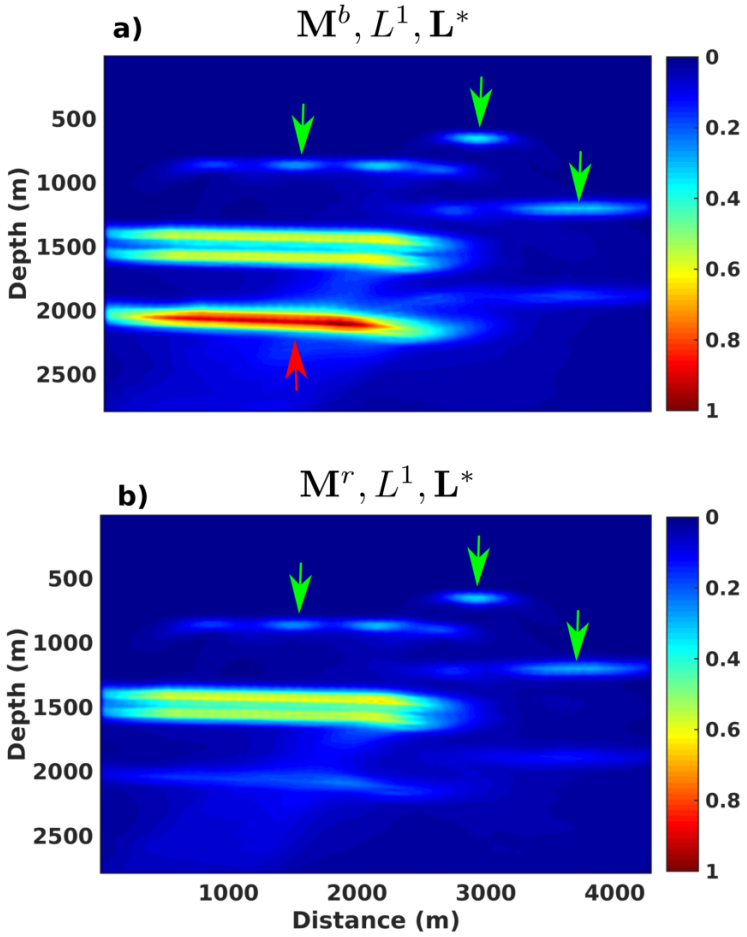

Figure 80: **a**) Projection of L^1 data residuals (equation 131, with p=1) into the image-space using $\mathbf{L}^*$ (defined in equation 134). Synthetic data were computed in the macro-layered model in Figure 74a. Green arrows identify picking errors while the red arrow identifies velocity errors. **b**) Projection of L^1 data residuals into the image-space using $\mathbf{L}^*$. Synthetic data were computed in the more accurate macro-layered model in Figure 76. The effect of velocity errors is removed.

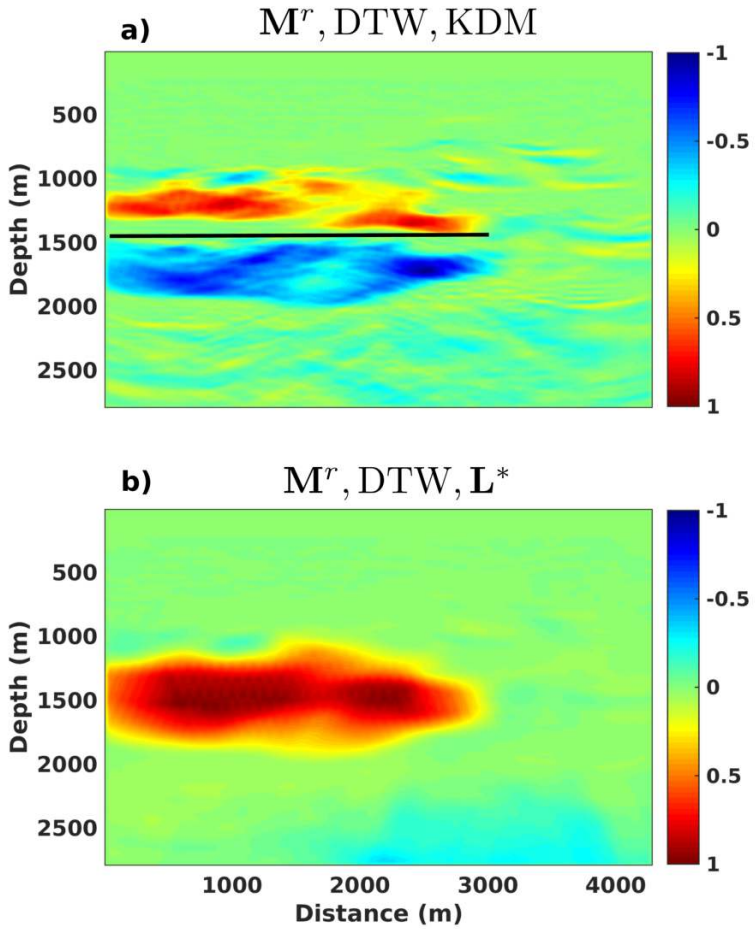

Figure 81: **a**) Projection of Dynamic-time-warping data residuals (equation 132) into the image-space using KDM (defined in equation 162). Synthetic data were computed in the macro-layered model in Figure 76. The black line highlights the bipolarity of the residuals by separating positive residuals above from negative residuals below. **b**) Projection of Dynamic-time-warping data residuals into the image-space using $\mathbf{L}^*$ (defined in equation 134). Synthetic data were computed in the macro-layered model in Figure 76.

III.2.7 Application & Discussion

Appraising structural interpretations quantitatively is a challenging problem. Consider the eight different structural models in Figure 82 interpreted from the seismic image in Figure 71b by different interpreters. We would like to rank those structural models using VSP data from the most likely to the least likely. For each structural model, we built a block macro-layered velocity model, allowing us to compute VSP synthetic data for each structural model. Observed VSP data were generated in the reference model in Figure 70, with sources covering the free surface and receivers positioned in a vertical well at $x = 7500$ m. First, we tried to rank the structural models using a data-misfit defined as the L^1-norm of the entire data set, for each model. Figure 83a shows that the data-space ranking obtained this way is not consistent with the ranking expected in the model-space (the model-space misfits were computed as the L^1-norm of block macro-layered velocity models and the reference velocity model). We then tried to rank the structural models using the method proposed in this chapter. Figure 83b shows the results obtained by limiting the model-misfit and the data-misfit to the region of interest, i.e. the black box in Figure 71b, and by relying on equations 125, 127 and 129 to define the data-misfit. We observe that the scatter plot in Figure 83b satisfies condition 8,

while the scatter plot in Figure 83a does not. The model-misfit axis in Figures 83a and 83b show that model 1 is the best model. The data-misfit in 83b was able to indentify model 1 as the best model, whereas the data-misfit in Figure 83a identified model 3 as the best model. In fact, the data-misfit in Figure 83b was able to successfully rank all the models except model 7.

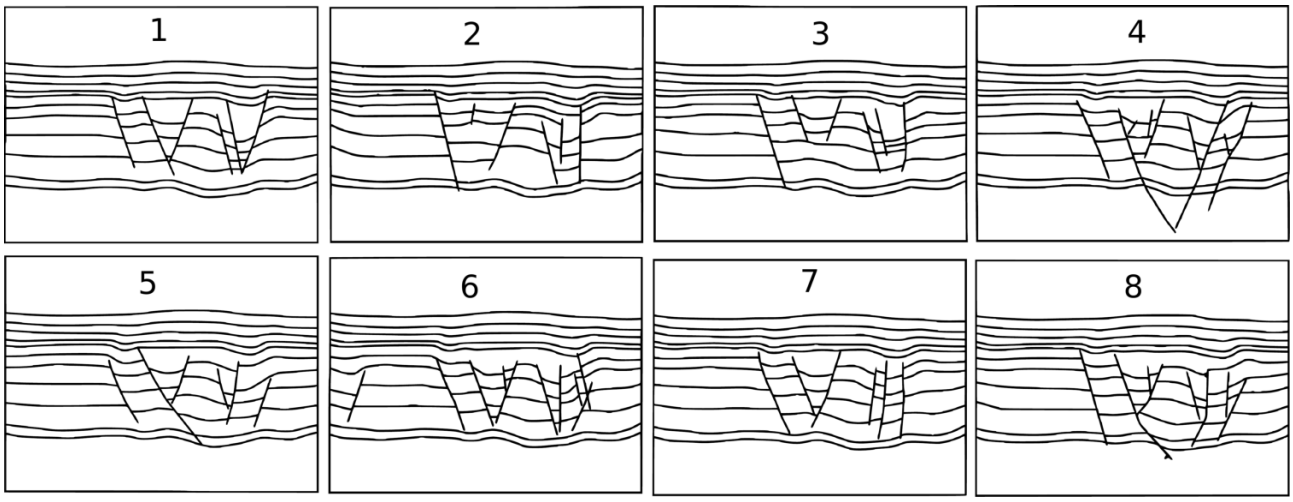

Figure 82: Eight possible structural interpretations/models of the seismic image in Figure 71b from different interpreters.

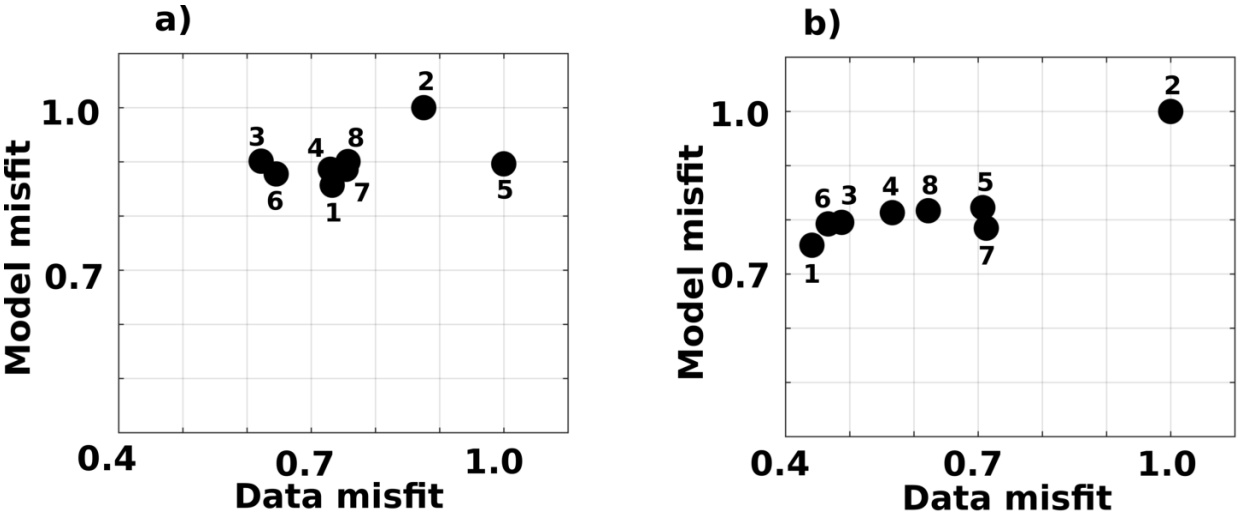

Figure 83: Appraising the structural models in Figure 82 using VSP data. Each point represents a structural model from Figure 82. The "observed" data were computed in the reference model in Figure 70, with sources positioned at the free surface and receivers positioned in a vertical well at $x = 7500\,m$. **a**) The data-space ranking obtained using the L^1-norm of the entire data set is not consistent with the ranking expected in the model-space. **b**) The data-space ranking obtained using the localized data-misfit 125, along with equation 127 and equation 129, is consistent with the ranking expected in the model-space. The region of interest Ω is the black box in Figure 71b.

The present chapter focuses on misfit functions with the aim of determining the possible reasons as to why the straightforward approach implemented in Figure 83a fails. First, we argue that one should use localized data-misfits $\Phi^d(\Omega)$. It follows that conditions 6 and 7 are necessary conditions for being able to rank structural models using $\Phi^d(\Omega)$, if we expect data-space ranking to be consistent with model-space ranking. These conditions are expected to be valid if the available candidate structural models are "close enough" to the (unknown) reference model.

Condition 6 mandates that a perturbation of the data-misfit value $\Phi_i^d(\Omega)$ in the region Ω of the i^{th} model should only occur if that model is modified in that specific region Ω. Condition 7 mandates that if the i^{th} model is modified in the region Ω, the data misfit $\Phi_i^d(\Omega)$ corresponding to that region should change as well.

For VSP data, condition 7 can readily be satisfied by computing Φ^d using only source-receiver pairs that illuminate Ω (e.g. using equation 129) and windowing data, along the time axis, around the first arrival (e.g. using equation 127). The difficulty then comes from satisfying condition 6, which is impossible in the most general case as any perturbation of the structural model at any point along any raypath through Ω will lead to a perturbation of Φ^d, not just points in Ω. This difficulty can be alleviated, in principle, by adding data that illuminate Ω from different directions as illustrated in Figure 77.

For surface seismic data, using $L^p(p \in [1,2])$ data residuals, we show that velocity errors introduced in macro-layered velocity models can lead to a perturbation of $\Phi^d(\Omega)$ for a region Ω below the velocity error even in the absence of any structural interpretation errors in Ω (e.g. Figure 80a), thereby violating condition 6. We also argue that if a reflector in the structural model is shifted away from its true position until its reflection-pulse no longer overlaps with true reflection-pulse along the time axis, shifting the reflector further in the model would not necessarily affect the data misfit, thereby violating condition 7. However, it is possible, in principle, to approximate condition 7 by using more appropriate data residuals like phase-shift residuals. The challenge then becomes how to compute phase-shifts for complex data sets; this is a subject of ongoing investigations.

The proposed method assumes that the region of interest is well illuminated. We can therefore expect the method to perform poorly for more challenging seismic imaging targets, such as subsalt areas for example. Moreover, appraising structural models using surface seismic data requires a kinematically accurate migration velocity model, which is not easy to obtain in practice. Challenging our approach with a realistic migration velocity model obtained by migration velocity analysis is a subject of ongoing investigations.

III.2.8 Conclusion

In this chapter, we propose a theoretical formulation and some general solutions to the problem of appraising structural interpretations using seismic data. Assuming that the different structural models are close to the unknown true model, we propose a set of conditions imposed on data-space misfit functions needed for a reliable model appraisal. We argue that misfit functions should be able to localize interpretation errors in the image-space. This localization of errors is achieved by back-projection of data-residuals into the image-space. It follows that, because it is not possible to truly localize errors using VSP data, one cannot predict which interpretations are more probable than others using VSP data in the most general case. However, it is expected that VSP data can always be used to statistically rank structural interpretations if the data illuminates the target from different directions. As for appraising structural models using surface seismic data, we expect a better localization of errors, compared to VSP, and therefore a better chance for ranking structural interpretations. The challenge when using surface seismic data is to define appropriate data residual functions. We argue that phase-shift residual functions, such as dynamic-warping and optimal-transport residuals, are good candidates; our current work involves finding robust ways to compute such phase-shift residuals for data acquired in complex geological settings.

III.2.9 Acknowledgements

The authors would like to thank Ludovic Métivier and Pierre Thore for fruitful discussions about the subject. This work also greatly benefited from constructive discussions with Thomas Bodin and Yann Capdeville in the frame of the HIWAI Project. We are very thankful to Elizabeth L'Heureux, Isabelle Lecomte, Stan Davis, and Xinming Wu for the time they dedicated to review this chapter. We would also like to thank Christelle Butault, Francois Bonneau, and Paul Angrand, for some seismic interpretations used in this work. The traveltime maps needed to evaluate Kirchhoff integral operators were computed using the Madagascar software freely available from www.ahay.org.

This work was done in the frame of the RING project at Université de Lorraine. The sponsors of the RING-GOCAD Consortium managed by ASGA are hereby acknowledged for their support. We would also like to thank Paradigm for providing the SKUA-GOCAD software used for structural modeling.

Chapter III.3

Appraising Structural Models using Seismic Data — Practice

This chapter resulted from an ongoing technical report co-authored with Paul Cupillard, Guillaume Caumon, and Paul Sava.

III.3.1 Summary

Structural models built from interpretation of seismic images inherit uncertainties propagated from interpretation uncertainties. As a result, it is often possible to propose multiple structural models from a single seismic image. We propose a method for appraising different geologically valid structural models built from a single seismic image. The main idea of the proposed method is to simulate synthetic data for each structural model, and then to compare the resulting synthetic data to observed data. In this chapter, we discuss the practical aspects of appraising structural models using surface seismic data.

III.3.2 Introduction

Structural models built from interpretation of seismic images inherit uncertainties propagated from interpretation uncertainties. As a result, it is often possible to propose multiple structural models from a single seismic image. In this chapter we consider the problem of determining which structural models, from a set of structural models, are more likely than others using seismic data. This problem is referred to as *appraising structural models* hereafter. A detailed introduction to the problem of appraising structural models using seismic data is given in chapter III.2. In chapter III.2, an application of appraising structural models using vertical seismic profiling data is presented. Here, we discuss the challenges of appraising structural models using surface seismic data. The main idea of the proposed method is to simulate synthetic data for each candidate structural model available, and then compare the resulting synthetic data to observed data. Candidate structural models are assumed to result from structural interpretations of a single seismic image. As discussed in chapter III.2, the data-misfit $\Phi_i(\Omega)$ of the i^{th} structural model in the region of interest Ω can be expressed as

$$\Phi_i(\Omega) = \sum_{\mathbf{x}\in\Omega} \mathbf{P}\big[\mathbf{R}(d_i, d_o)\big] = \sum_{\mathbf{x}\in\Omega} \mathbf{E}(\mathbf{x}), \tag{137}$$

where d_o is the observed data, d_i is the synthetic data generated from the i^{th} structural model. The data residual operator $\mathbf{R}$ maps the data-plane $\mathbb{D}\times\mathbb{D}$ to a reduced-data-space $\mathbb{D}'$, and the weighting operator $\mathbf{P}$ maps the reduced-data-space $\mathbb{D}'$ to the image-domain $\mathbb{I}$. The *interpretation error map* $\mathbf{E}$ defined in the data misfit 137 is therefore computed in two steps: first, the residual operator $\mathbf{R}$ computes data residuals; second, the weight operator $\mathbf{P}$ projects the data residuals in the image-domain. We will investigate only residual operators $\mathbf{R}$ for which $\mathbb{D}' = \mathbb{D}$ (i.e. an object in $\mathbb{D}'$ is a function of source and receiver coordinates and time) so that one may consider $\mathbf{P} = \mathbf{F}^\dagger$, where $\mathbf{F}^\dagger$ is either the adjoint (e.g. section III.3.3) or the inverse (e.g. section III.3.4) of any appropriate operator

$$\mathbf{F} : \mathbb{I} \to \mathbb{D}. \tag{138}$$

In section III.3.3, we consider the case where $\mathbf{R}$ is the dynamic image warping (DIW) operator (Hale, 2013) that computes phase-shift residuals of seismic data, and $\mathbf{P}$ is a linear imaging operator. In this section, we study only candidate interpreted structural models that have the same number of geological layers as the reference (observed) structural model. This is an ideal case where the assumptions of DIW, and therefore its validity within this framework, can be expected to hold. In practice, however, we can expect candidate models to have a different number of layers from the reference structural model, especially if the reference model has some fine layered structures (see for example chapter III.1); in this case, the assumptions of DIW do not hold and its validity within this framework is questionable. In section III.3.4, we consider the case where $\mathbf{R}$ is the L1-difference and $\mathbf{P}$ is a waveform inversion operator. This approach does not constrain the number of layers in candidate structural models; however, it inherits all the computational challenges of waveform inversion. We will only study macro-layered models in this chapter, that is, models without any fine layer structures, even though this is not necessary for the approach presented in section III.3.4.

III.3.3 Appraising models with the same number of layers

Let us consider the case where $\mathbf{F}$ in equation 138 is the Born modeling operator (see for example section I.1.1)

$$\mathbf{L}(m)P_i(\mathbf{x},t;\mathbf{s}) = \left[\frac{1}{m^2(\mathbf{x})}\frac{\partial^2}{\partial t^2} - \Delta\right]P_i(\mathbf{x},t;\mathbf{s}) = m_i(\mathbf{x})P(\mathbf{x},t;\mathbf{s}), \tag{139}$$

where P_i is the output synthetic wavefield, m_i is the input model, P is the solution to

$$\mathbf{L}(m)P(\mathbf{x},t;\mathbf{s}) = \delta(\mathbf{x}-\mathbf{s})w(t), \tag{140}$$

m is the smooth background velocity model (assumed to be kinematically accurate), and w is the source wavelet. Synthetic data $d_i(t,\mathbf{s},\mathbf{r})$ are obtained by sampling the synthetic wavefield

$$P_i(\mathbf{x},t;\mathbf{s}) = d_i(\mathbf{x},\mathbf{s},t) = \int_{\mathbf{x}',t'} G(\mathbf{x},t;\mathbf{x}',t')P(\mathbf{x}',t';\mathbf{s})m_i(\mathbf{x}') = \mathbf{B}m_i(\mathbf{x}) \tag{141}$$

at receivers; that is

$$\begin{aligned} d_i(t,\mathbf{s},\mathbf{r}) &= \int_{\mathbf{x}} \delta(\mathbf{x}-\mathbf{r})d_i(\mathbf{x},\mathbf{s},t) = \int_{\mathbf{x}} \delta(\mathbf{x}-\mathbf{r}) \int_{\mathbf{x}',t'} G(\mathbf{x},t;\mathbf{x}',t')P(\mathbf{x}',t';\mathbf{s})m_i(\mathbf{x}') \\ &= \int_{\mathbf{x},t'} G(\mathbf{x},t-t';\mathbf{r})P(\mathbf{x},t';\mathbf{s})m_i(\mathbf{x}) \\ &= \mathbf{F}m_i(\mathbf{x}), \end{aligned} \tag{142}$$

(see the appendices in section I.1.5 for more details) where $G(\mathbf{x},t;\mathbf{r})$ is the solution to

$$\mathbf{L}(m)G(\mathbf{x},t;\mathbf{r}) = \delta(\mathbf{x}-\mathbf{r})\delta(t).$$

For $\mathbf{F}$ as defined in equation 142, one may equate its adjoint to the weight operator $\mathbf{P}$ in equation 137. For example, let $f(\mathbf{s},\mathbf{r},t)$ denote the DIW data residual

$$f(\mathbf{s},\mathbf{r},t) = \mathbf{R}\big[d_o(\mathbf{s},\mathbf{r},t), d_i(\mathbf{s},\mathbf{r},t)\big] = \underset{l(\mathbf{s},\mathbf{r},t)}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},t} \big[d_o(\mathbf{s},\mathbf{r},t-l) - d_i(\mathbf{s},\mathbf{r},t)\big]^2, \tag{143}$$

then the adjoint of equation 142 leads to $\mathbf{P} : f(\mathbf{s},\mathbf{r},t) \to f(\mathbf{x})$ being the reverse time migration (RTM) operator

$$\mathbf{P}[f(\mathbf{s},\mathbf{r},t)] = \int_{\mathbf{r},\mathbf{s},t,t'} G(\mathbf{x},t-t';\mathbf{r})P(\mathbf{x},t';\mathbf{s})[f(\mathbf{s},\mathbf{r},t)], \tag{144}$$

or more customarily

$$\begin{cases} \mathbf{P}[f(\mathbf{s},\mathbf{r},t)] = \int_{\mathbf{s},t} R(\mathbf{x},t;\mathbf{r})P(\mathbf{x},t;\mathbf{s}) \\ \mathbf{L}^*(m)R(\mathbf{x},t;\mathbf{r}) = f(\mathbf{s},\mathbf{r},t) \end{cases}, \tag{145}$$

(see the appendices in section I.1.5 for more details) where the asterisk in $\mathbf{L}^*(m)$ denotes adjoincy; that is, the equation is solved backwards in time (e.g. Fichtner et al., 2006).

In practice, the straightforward application of equation 145 to project DIW data residuals in the image space does not work well. Consider for example the illustrative reference model in Figure 84a, the corresponding candidate synthetic model is shown in Figure 84b; the difference between the two models is shown in Figure 84c. Figure 85a shows an "observed" shot gather computed in the reference model (Figure 84a), Figure 85b shows a synthetic shot gather computed in the synthetic model (Figure 84b), and Figure 85c shows the difference between the observed and synthetic data. Ideally, given a migration velocity model (Figure 86a), we would like to construct an interpretation error map $\mathbf{E}(\mathbf{x})$ similar to Figure 84c using the the observed data (Figure 85a) and synthetic data (Figure 85b). Figure 86b shows the DIW data residuals obtained from the observed and synthetic data in Figure 85 and Figure 86c shows the result of applying equation 145 to the DIW data residuals using the migration model in Figure 86a. The resulting error map is noisy and does not reveal the interpretation error observed in Figure 84c. An improved error map, shown in Figure 86d, is obtained by replacing G and P in equation 144

by their ray-based approximations (i.e. Kirchhoff depth migration). For this simple example, the error map obtained by Kirchhoff migration (KDM) reveals the interpretation error in Figure 84c. For more complex models however, the KDM-based error map does not work well either.

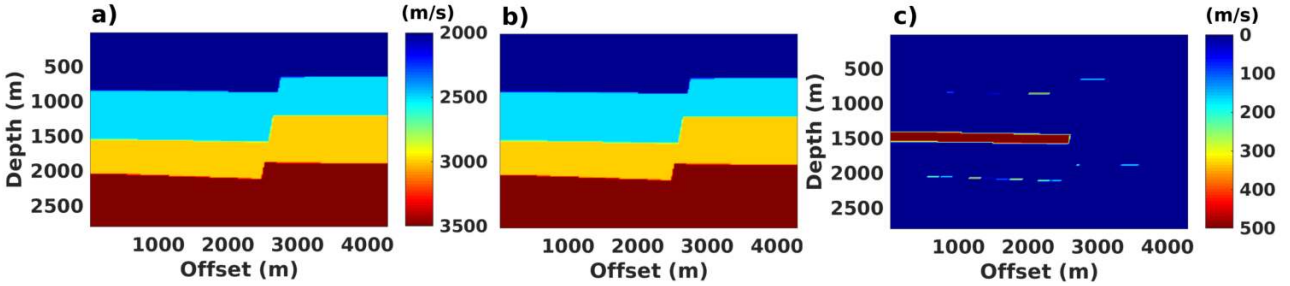

Figure 84: **a**) Reference model. **b**) Candidate model from interpretation. **c**) Difference between reference (a) and candidate model (b).

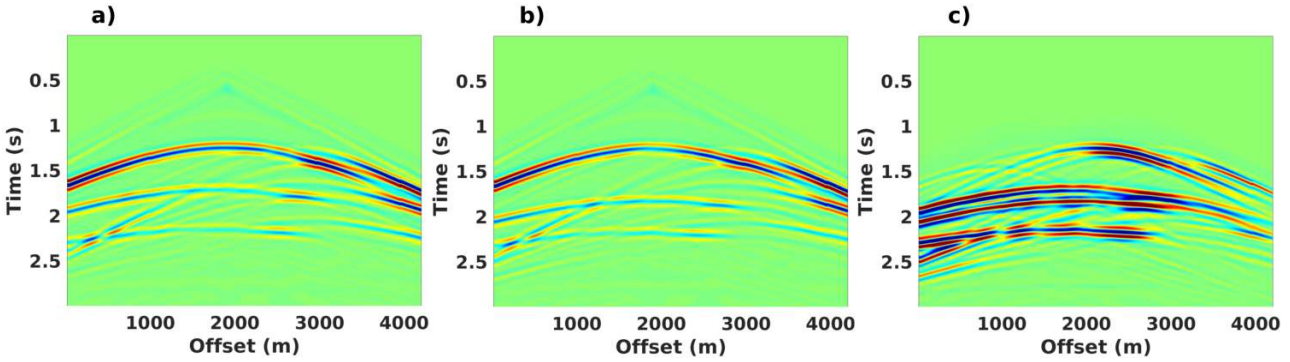

Figure 85: **a**) "Observed data" computed in the reference model in Figure 84a. **b**) Synthetic data computed in the candidate model in Figure 84b. **c**) Difference between observed (a) and synthetic data (b).

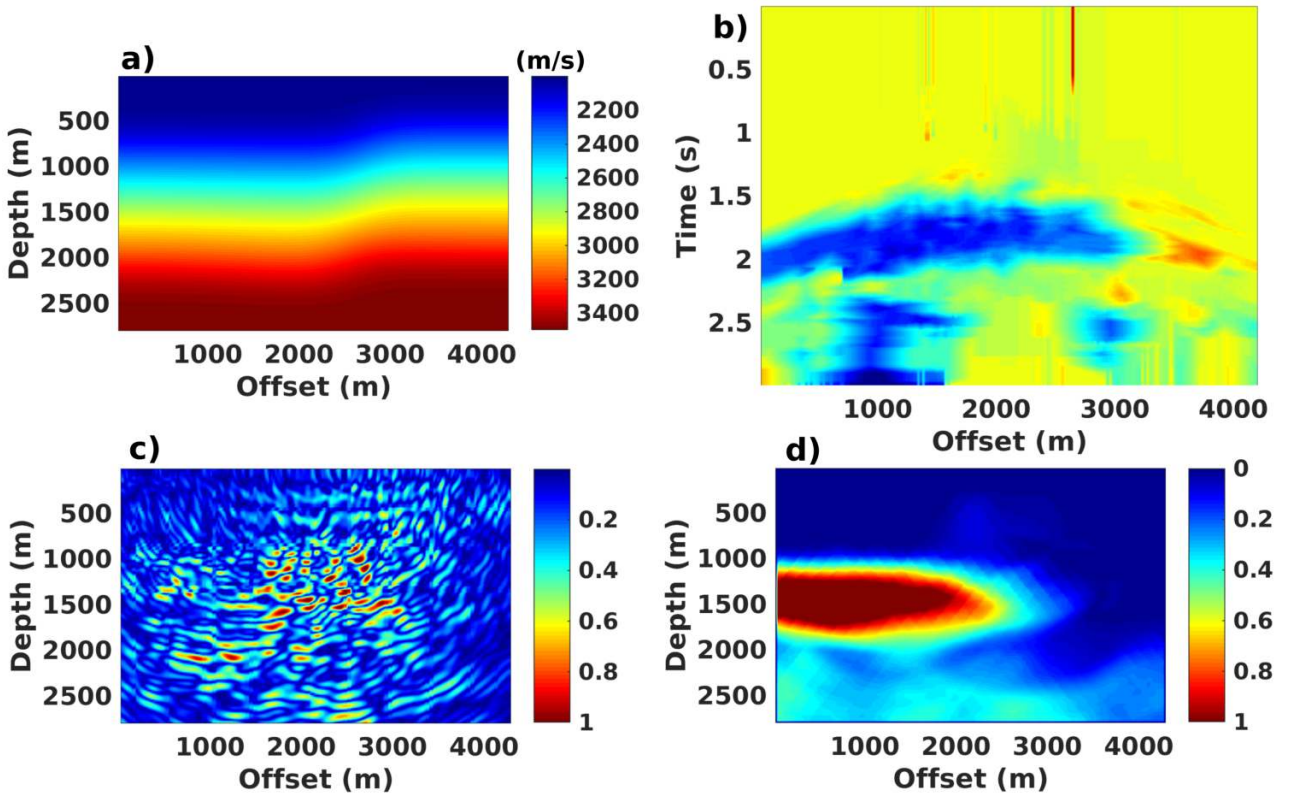

Figure 86: **a**) Migration velocity model obtained by smoothing the reference model (Figure 84)a. **b**) Dynamic image warping (DIW) residuals computed from the observed and synthetic data in Figure 85a and Figure 85b. **c**) Reverse time migration of the DIW data residuals (b). **d**) Kirchhoff depth migration of the DIW data residuals (b).

Let us first consider the slightly more complex model in Figure 87. The reference and the candidate model from interpretation are shown in Figures 87a and 87b; the difference between the two models is shown in Figure 87c. The "observed" and synthetic data are shown in Figures 88a and 88b; the difference between the observed and synthetic data is shown in Figure 88c. The DIW data residuals computed from the observed (Figure 88a) and synthetic (Figure 88b) are shown in Figure 89a. The KDM projection of DIW data residual in the image-space, using the migration model in Figure 89b, is shown in Figure 89c. The interpretation error in blue about 1000m deep in Figure 87c is detected, as clearly seen in Figure 89c. However, the rest of the interpretation errors observed beyond 1000m are not visible in Figure 87c.

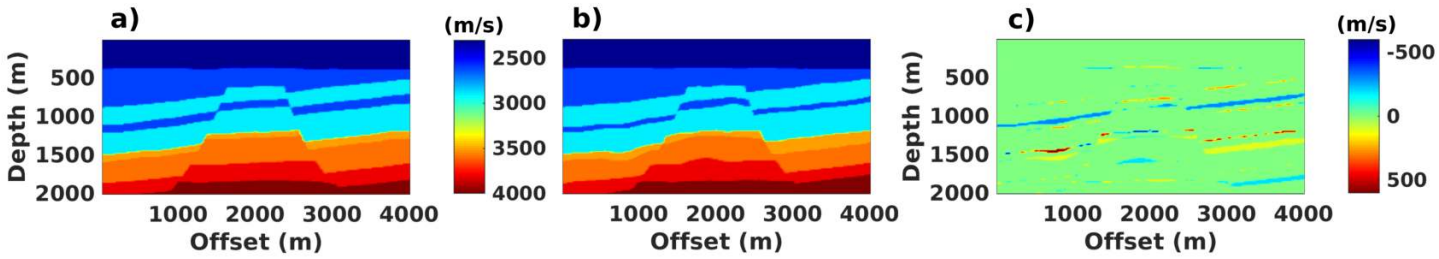

Figure 87: **a**) Reference model. **b**) Candidate model from interpretation. **c**) Difference between reference (a) and candidate model (b).

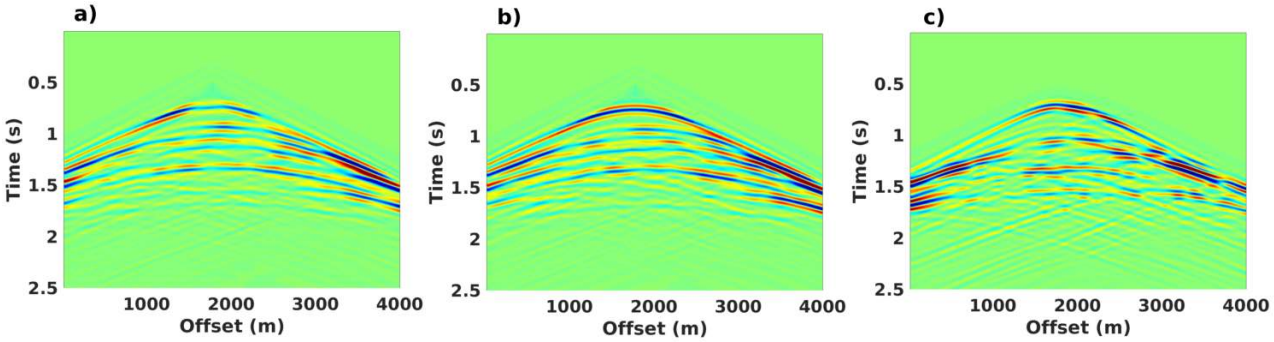

Figure 88: **a**) "Observed data" computed in the reference model in Figure 87a. **b**) Synthetic data computed in the candidate model in Figure 87b. **c**) Difference between observed (a) and synthetic data (b).

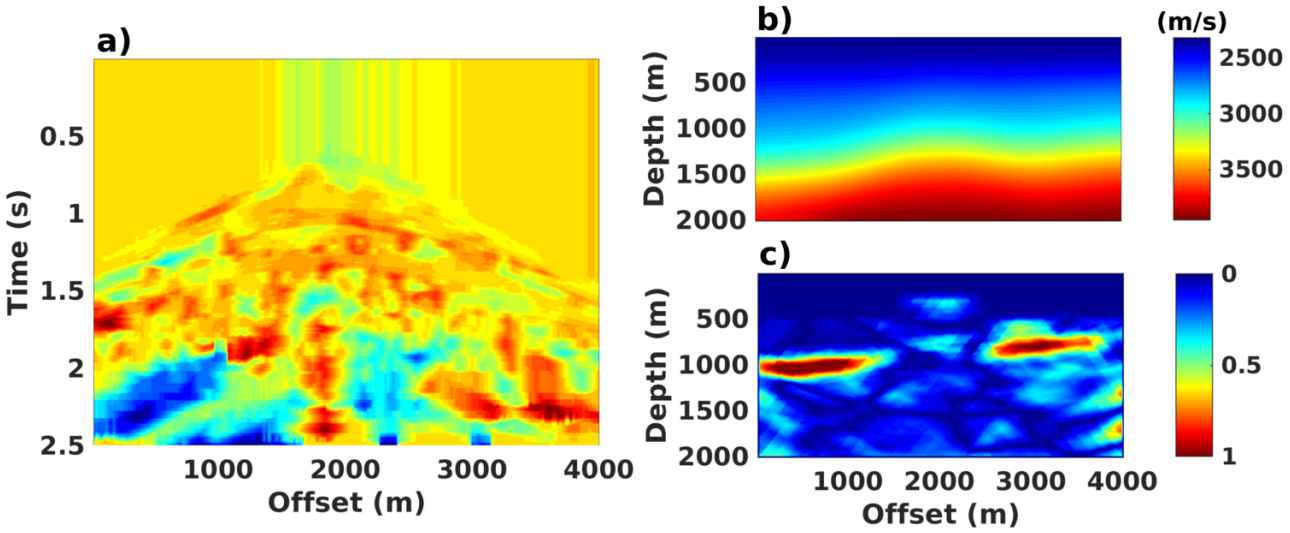

Figure 89: **a**) Dynamic image warping (DIW) residuals computed from the observed and synthetic data in Figure 88a and Figure 88b. **b**) Migration velocity model obtained by smoothing the reference model (Figure 87)a. **c**) Kirchhoff depth migration of the DIW data residuals (a).

Now let us consider the more complex reference model in Figure 90a; the corresponding interpreted model is shown in Figure 90b, and the difference between the two is shown in Figure 90c. The "observed" and synthetic data are shown in Figures 91a and 91b; the difference between the observed and the reference data is shown in Figure 91c. The DIW data residuals computed from the observed (Figure 91a) and synthetic (Figure 91b) are shown in Figure 92a. The KDM projection of DIW data residual in the image-space, using the migration model in Figure 92b, is shown in Figure 92c. The resulting error map was clearly unable to detect the interpretation errors observed in Figure 90c.

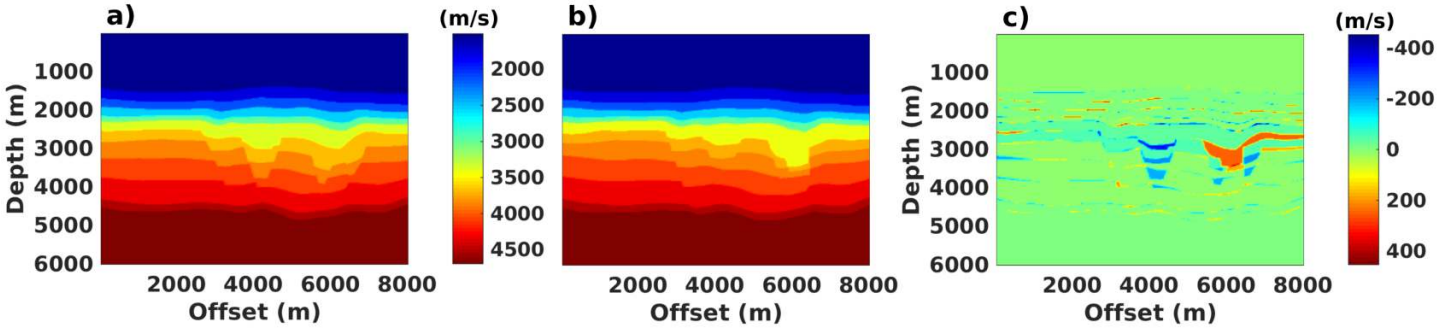

Figure 90: **a**) Reference model. **b**) Candidate model from interpretation. **c**) Difference between reference (a) and candidate model (b).

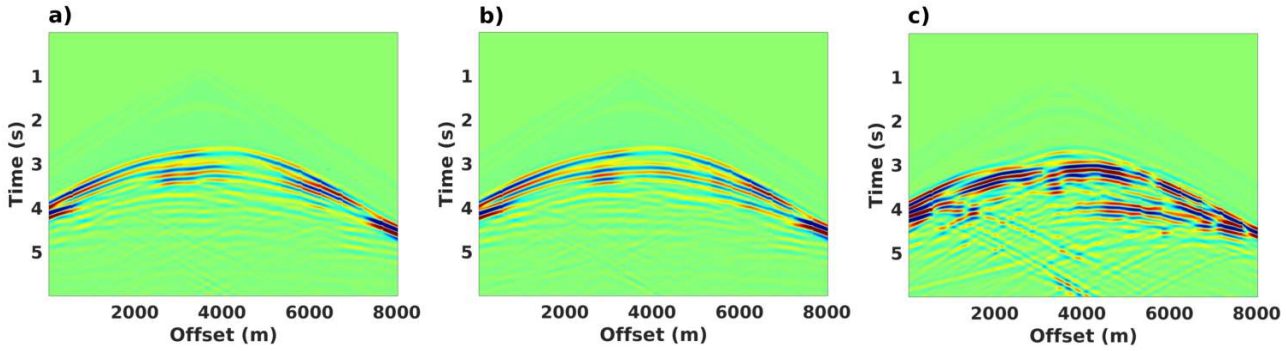

Figure 91: **a**) "Observed data" computed in the reference model in Figure 90a. **b**) Synthetic data computed in the candidate model in Figure 90b. **c**) Difference between observed (a) and synthetic data (b).

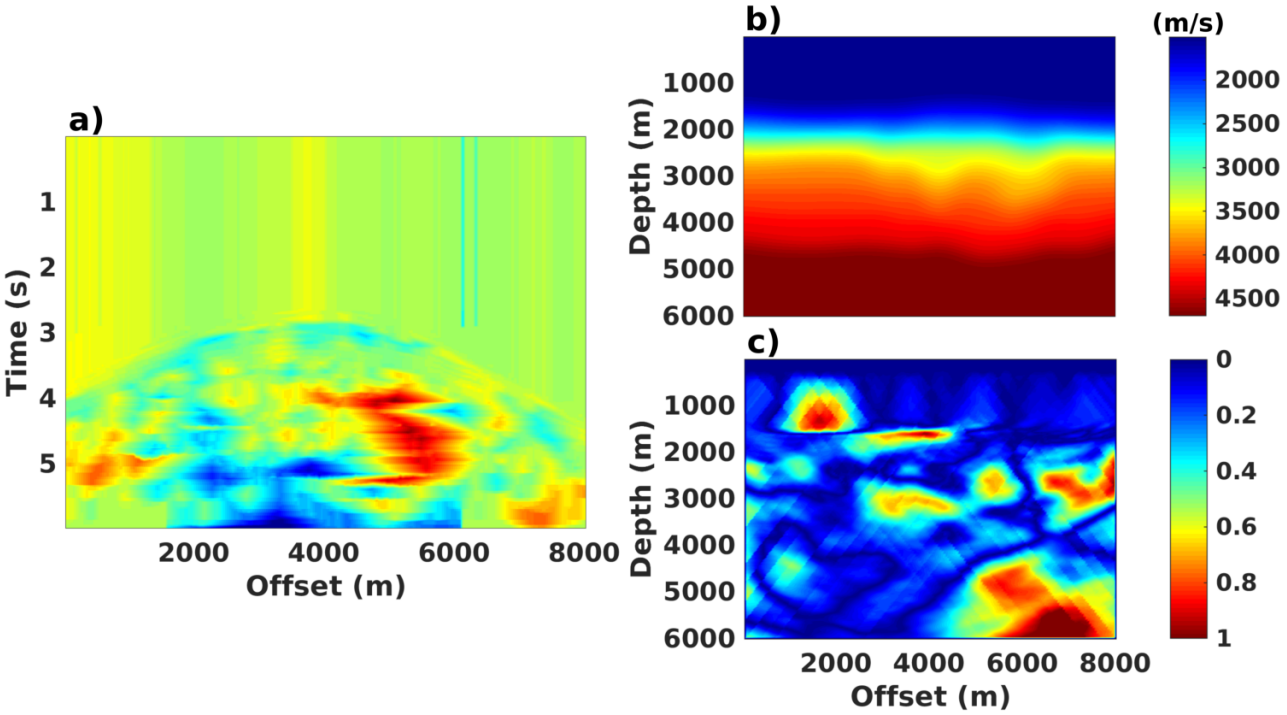

Figure 92: **a**) Dynamic image warping (DIW) residuals computed from the observed and synthetic data in Figure 91a and Figure 91b. **b**) Migration velocity model obtained by smoothing the reference model (Figure 90)a. **c**) Kirchhoff depth migration of the DIW data residuals (a).

While it is still unclear why equation 145 does not work as expected, there are at least two issues that we think may be leading to the unsatisfactory results observed:

1. The shifts computed by DIW may not be accurate; this seems to be especially true towards the end of seismic traces dominated by multiples. Even in the ideal case where the reference and synthetic model have the same number of layers, as investigated in this section, the synthetic data will generally not satisfy the DIW assumption requiring the synthetic data to be a warped version of the observed data. Furthermore, DIW, as presented by Hale (2013), has a couple of free parameters. DIW data-residuals have shown to strongly depend on these parameters and it has proven challenging to determine the optimal values for these parameters.

2. The output data residuals from equation 143 are usually very low frequency data, as can be observed in Figures 86b, 89a, and 92a. Both our intuition and numerical experiments suggest that it is unlikely to reconstruct interpretation errors using equation 145 because interpretation errors usually contain both high and low frequency as observed in Figures 84c, 87c, and 90c.

The first issue raised above, that of computing accurate phase-shift residuals, remains an open problem, at least within the scope of this work.

We now discuss the second issue raised above: the low frequency content of DIW phase-shift residuals. An alternative way to think of $\mathbf{P}$ in equation 144 is as the first iteration of the least-square inverse of equation 142 (i.e. least-squares RTM, see section I.1.2 for details). The advantage of this approach is that it automatically accounts for the low frequency nature of residuals obtained using 143. Consider the optimization problem

$$\begin{cases} m(\mathbf{x}) = \underset{m(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},t} f(\mathbf{s},\mathbf{r},t,m)^2 = \int_{\mathbf{s},\mathbf{r},\mathbf{x},t} \delta(\mathbf{x}-\mathbf{r}) f(\mathbf{s},\mathbf{x},t,m)^2 \\ \text{subject to} \\ f(\mathbf{s},\mathbf{r},t,m) = \underset{l(\mathbf{s},\mathbf{r},t,m)}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},\mathbf{x},t} \delta(\mathbf{x}-\mathbf{r}) \big[d_o(\mathbf{s},\mathbf{x},t-l) - d_i(\mathbf{s},\mathbf{x},t,m)\big]^2 \\ d_i(\mathbf{s},\mathbf{x},t,m) = \mathbf{B} m_i \end{cases}, \tag{146}$$

resulting from equations 141 and 143. Ma and Hale (2013) note from equation 143 that

$$\frac{\partial}{\partial l}\mathbf{R}|_f = 0 = -2 \int_{\mathbf{s},\mathbf{r},t} \big[d_o(\mathbf{s},\mathbf{r},\tau) - d_i(\mathbf{s},\mathbf{r},t)\big] \frac{\partial}{\partial \tau} d_o(\mathbf{s},\mathbf{r},\tau), \text{ with } \tau = t - f, \tag{147}$$

(see the appendix in section III.3.7 for details) transforming problem 146 to

$$\begin{cases} m(\mathbf{x}) = \underset{m(\mathbf{x})}{\operatorname{argmin}} = \int_{\mathbf{s},\mathbf{r},\mathbf{x},t} \delta(\mathbf{x}-\mathbf{r}) f(\mathbf{s},\mathbf{x},t,m)^2 \\ \text{subject to} \\ 0 = \int_{\mathbf{s},\mathbf{r},\mathbf{x},t} \delta(\mathbf{x}-\mathbf{r}) \big[d_o(\mathbf{s},\mathbf{x},\tau) - d_i(\mathbf{s},\mathbf{x},t,m)\big] \frac{\partial}{\partial \tau} d_o(\mathbf{s},\mathbf{x},\tau), \text{ with } \tau = t - f(\mathbf{s},\mathbf{x},t,m) \\ d_i(\mathbf{s},\mathbf{x},t,m) = \mathbf{B} m_i \end{cases}. \tag{148}$$

The Lagrangian of problem 148 is given by

$$\begin{aligned} \mathscr{L} &= \Big\langle \delta(\mathbf{x}-\mathbf{r}), f(\mathbf{s},\mathbf{x},t,m)^2 \Big\rangle_{\mathbf{s},\mathbf{r},\mathbf{x},t} \\ &+ \Big\langle \lambda_1 \delta(\mathbf{x}-\mathbf{r}), \big[d_o(\mathbf{s},\mathbf{x},\tau) - d_i(\mathbf{s},\mathbf{x},t,m)\big] \frac{\partial}{\partial \tau} d_o(\mathbf{s},\mathbf{x},\tau) \Big\rangle_{\mathbf{s},\mathbf{r},\mathbf{x},t} \\ &+ \Big\langle \lambda_2, d_i(\mathbf{s},\mathbf{x},t,m) - \mathbf{B} m_i \Big\rangle_{\mathbf{s},\mathbf{x},t}. \end{aligned} \tag{149}$$

The derivative of the Lagrangian with respect to the model is given by (see the appendix in section III.3.7 for details)

$$\begin{aligned}\frac{\partial}{\partial m}\mathscr{L} &= \left\langle \delta(\mathbf{x}-\mathbf{r})\frac{\partial}{\partial m}f, 2f - \lambda_1\left\{\dot{d}_o(\mathbf{s},\mathbf{x},\tau)^2 + \big[d_o(\mathbf{s},\mathbf{x},\tau) - d_i(\mathbf{s},\mathbf{x},t,m)\big]\ddot{d}_o(\mathbf{s},\mathbf{x},\tau)\right\}\right\rangle_{\mathbf{s},\mathbf{r},\mathbf{x},t} \\ &+ \left\langle \frac{\partial}{\partial m}d_i(\mathbf{s},\mathbf{x},t,m), \lambda_2 - \int_{\mathbf{r}}\lambda_1\delta(\mathbf{x}-\mathbf{r})\dot{d}_o(\mathbf{s},\mathbf{x},\tau)\right\rangle_{\mathbf{s},\mathbf{x},t} \\ &- \left\langle \mathbf{B}^*\lambda_2, 1\right\rangle_{\mathbf{x}}, \end{aligned} \tag{150}$$

or more conveniently

$$\begin{cases} \lambda_1 = 2\frac{f(\mathbf{s},\mathbf{x},t,m)}{\dot{d}_o(\mathbf{s},\mathbf{x},\tau)^2+\left[d_o(\mathbf{s},\mathbf{x},\tau)-d_i(\mathbf{s},\mathbf{x},t,m)\right]\ddot{d}_o(\mathbf{s},\mathbf{x},\tau)} \\ \lambda_2 = \int_{\mathbf{r}}\lambda_1\delta(\mathbf{x}-\mathbf{r})\dot{d}_o(\mathbf{s},\mathbf{x},\tau) \\ \frac{\partial}{\partial m}\mathscr{L} = -\left\langle \mathbf{B}^*\lambda_2, 1\right\rangle_{\mathbf{x}} \end{cases}, \tag{151}$$

where $\mathbf{B}^*$ is the adjoint of the Born modeling operator defined in equation 141 (i.e. $\mathbf{B}^*$ is the RTM operator). It follows from equation 151 that the first iteration of least-squares RTM, using the DIW equation 143, leads to

$$\begin{cases} \mathbf{P}[f(\mathbf{s},\mathbf{r},t)] = \int_{\mathbf{s},t} R(\mathbf{x},t;\mathbf{r})P(\mathbf{x},t;\mathbf{s}) & (a) \\ \mathbf{L}^*(m)R(\mathbf{x},t;\mathbf{r}) = \frac{f(\mathbf{s},\mathbf{r},t,m)\dot{d}_o(\mathbf{s},\mathbf{r},\tau)}{\dot{d}_o(\mathbf{s},\mathbf{r},\tau)^2+\left[d_o(\mathbf{s},\mathbf{r},\tau)-d_i(\mathbf{s},\mathbf{r},t,m)\right]\ddot{d}_o(\mathbf{s},\mathbf{r},\tau)} & (b) \\ \tau = t - f(\mathbf{s},\mathbf{r},t) & (c) \end{cases}. \tag{152}$$

The term on the right-hand side of equation 152b will be referred to as the adjoint-source. According to equation 152b, the projection operator 152a overcomes the low-frequency issue by scaling the low-frequency DIW phase-shifts by the time derivative of observed traces. However, the error map obtained by the projection operator 152a will still be unreliable in general because it still strongly depends on our ability to compute accurate DIW phase-shifts. Consider for example adjoint-source in Figure 93a computed from the traces in Figure 85 and the phase-shifts in Figure 86b; the RTM projection, shown in Figure 93b, was able to detect the interpretation error in Figure 84c. For comparison, Figure 93c shows the RTM projection of the L^1 difference in Figure 85c. The L^1 data-difference error map misleadingly amplifies the picking error in Figure 86c around the depth of 2000m. Such shortcomings of $L^p(p \in [1,2])$ differences were among the motivations for investigating phase-shift residuals (see chapter III.2). These theoretical advantages of phase-shift residuals over data-difference residuals become less convincing in practice as models get more complex. For example, given the adjoint-source in Figure 94a computed from the traces in Figure 88 and the phase-shifts in Figure 89a, the superiority of the adjoint-source error map in Figure 94b over the L^1 difference error map in Figure 94c in detecting the interpretation errors in Figure 87c is less obvious. In fact, for the more complex example where the adjoint-source in Figure 95a is computed from the traces in Figure 91 and the phase-shifts in Figure 92a, the adjoint-source error map in Figure 95b appears to give a poorer estimate of the interpretation errors in Figure 90c than the L^1 data-difference error map in Figure 95c.

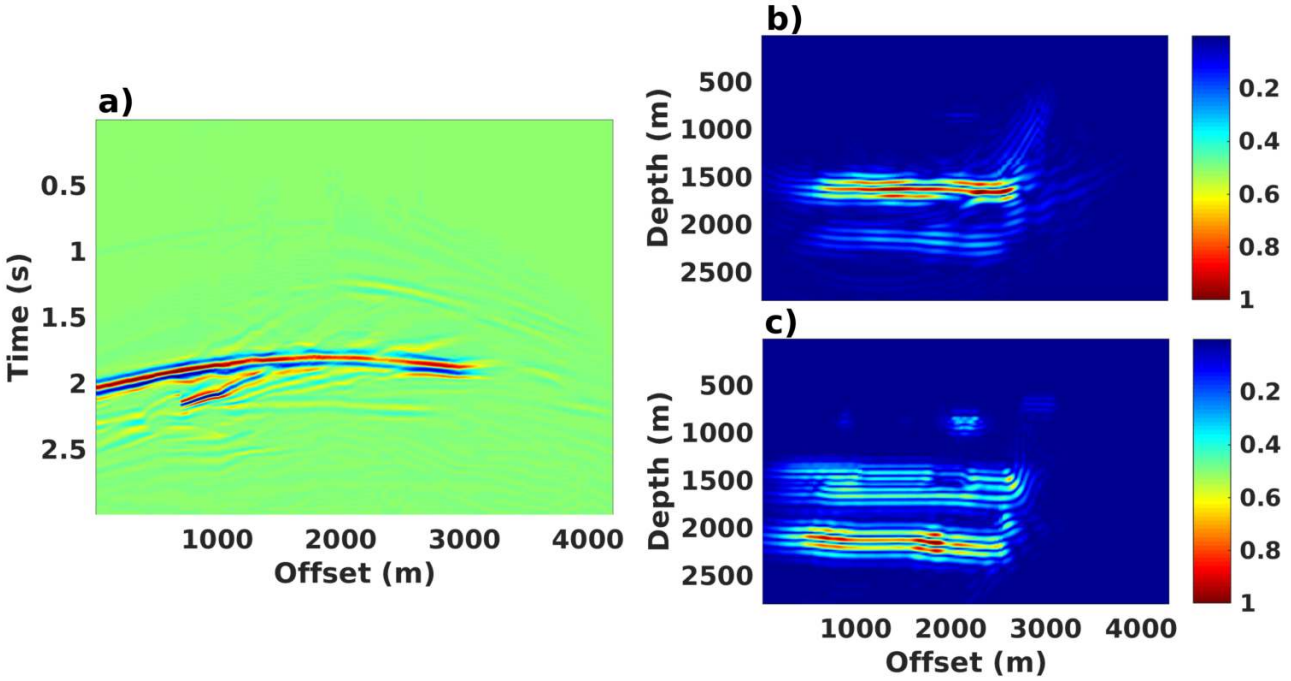

Figure 93: **a**)Adjoint-source (right-hand side of equation 152b) computed from the traces in Figure 85 and the phase-shifts in Figure 86b. **b**)Reverse time migration of the adjoint-source in a). **c**)Reverse time migration of the data-difference in Figure 86c.

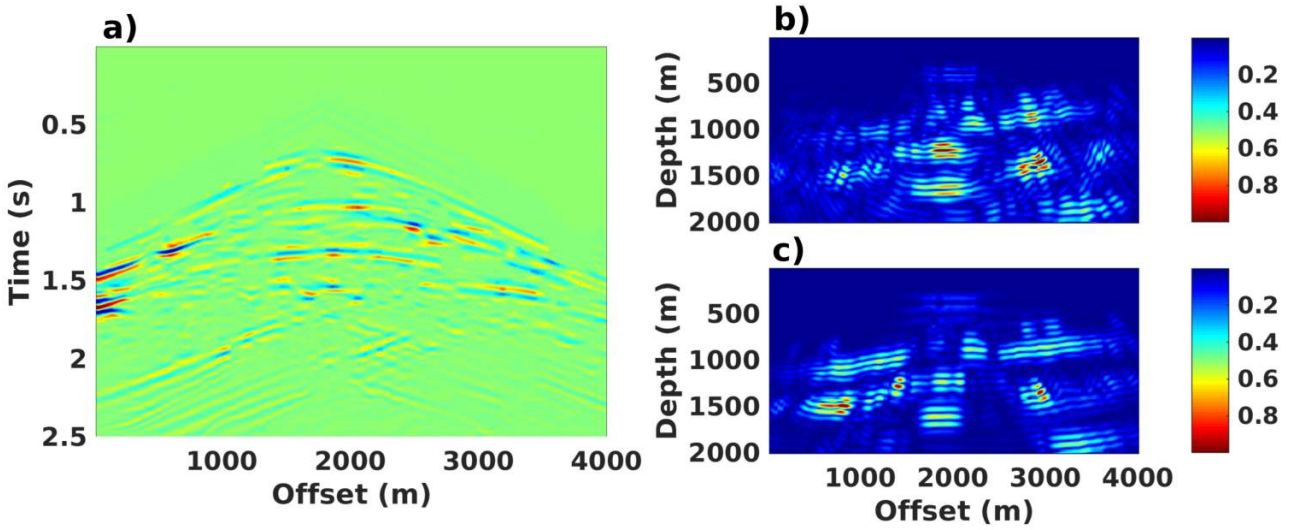

Figure 94: **a**)Adjoint-source (right-hand side of equation 152b) computed from the traces in Figure 88 and the phase-shifts in Figure 89b. **b**)Reverse time migration of the adjoint-source in a). **c**)Reverse time migration of the data-difference in Figure 89c.

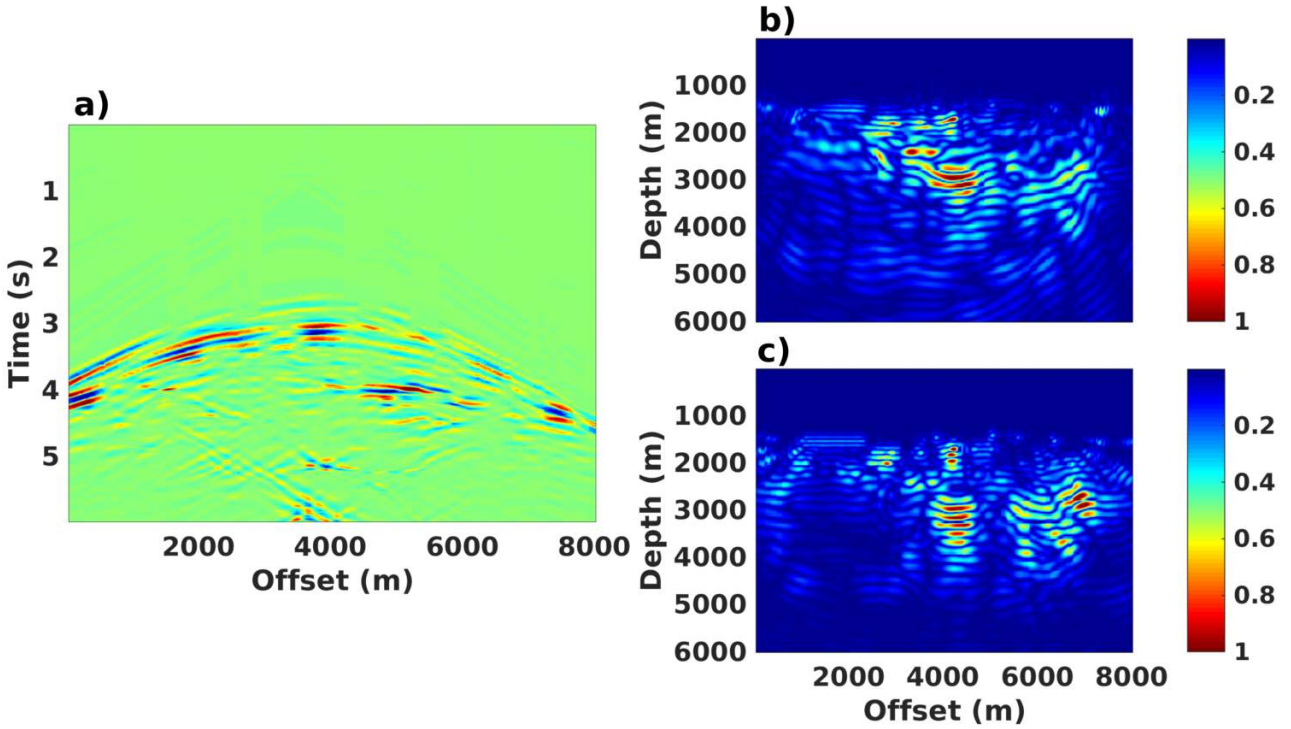

Figure 95: **a**)Adjoint-source (right-hand side of equation 152b) computed from the traces in Figure 91 and the phase-shifts in Figure 92b. **b**)Reverse time migration of the adjoint-source in a). **c**)Reverse time migration of the data-difference in Figure 92c.

We know from theoretical arguments and from numerical experiments that $L^p(p \in [1,2])$ data-difference residuals are not suited for appraising structural interpretations using a linear projection operator $\mathbf{P}$ (see chapter III.2). As we have just shown in this section, while DIW phase-shift residuals offer some theoretical advantages, they appear not to be suited for appraising structural interpretations using a linear projection operator $\mathbf{P}$ as well in practice. We investigate the use of a nonlinear projection operator $\mathbf{P}$ in the following section.

III.3.4 Appraising models with different number of layers

Consider the experiment shown in Figure 96. The velocity model in Figure 96a, obtained by adding the migration velocity model in Figure 86a to the error map in Figure 84c, was used to generate the synthetic data in Figure 96b. A waveform inversion (see section I.1.3) of this data set results in the velocity error map in Figure 96c, whose absolute value is presented in Figure 96d. In some sense, the error map in Figure 96d represents the best estimate of the error map we can expect to retrieve from data residuals using waveform inversion because it was obtained from the true error map (Figure 84c). It follows from this experiment that if we can use the observed data (Figure 85a) and synthetic data (e.g. Figure 85b) to compute data residuals that approximate the data in Figure 96b "well enough", then these data residuals can be used to estimate the error map by waveform inversion.

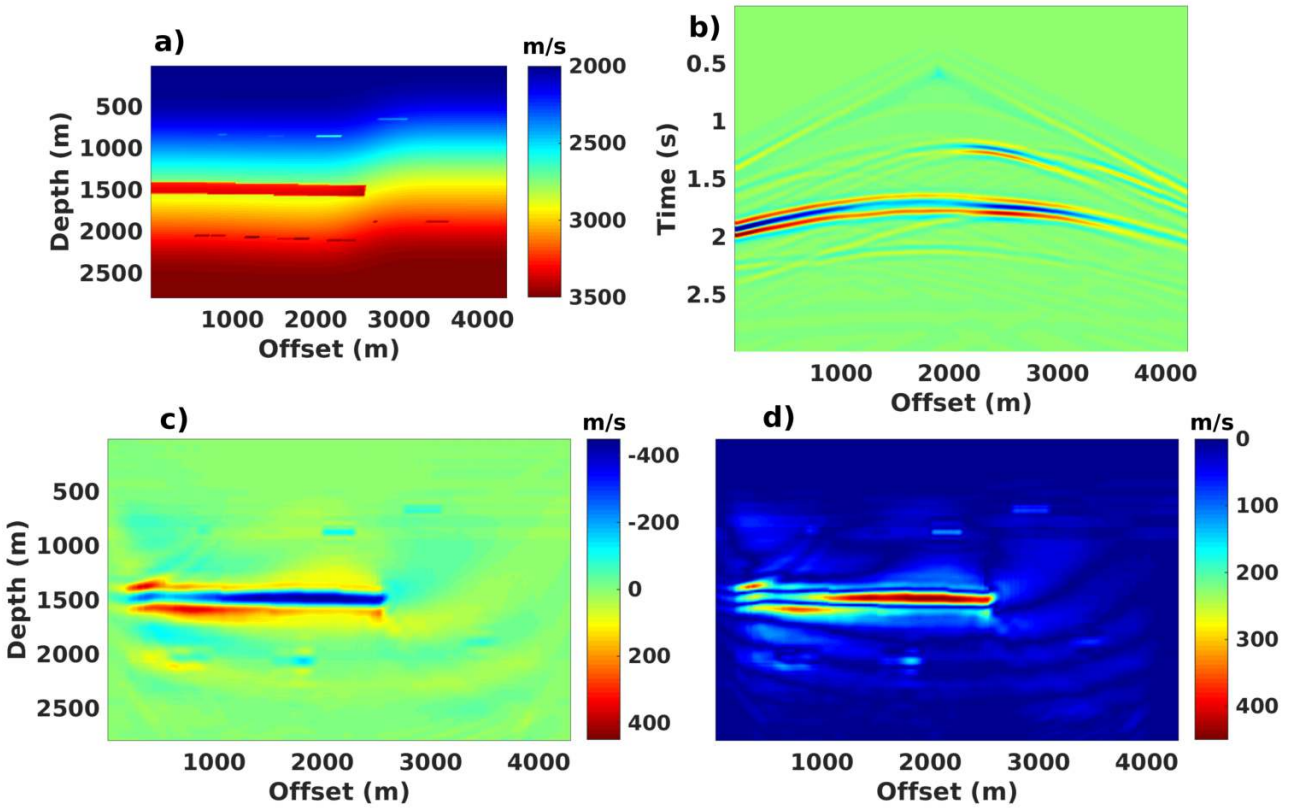

Figure 96: **a**) Velocity model obtained by adding the migration velocity model in Figure 86a to the error map in Figure 84c. **b**) Synthetic data generated from the model in a). **c**) Velocity error obtained by waveform inversion of the synthetic data set. **d**) Absolute value of the velocity error map in c.

Following the discussion in the previous paragraph, let

$$a_o(\mathbf{s},\mathbf{r},t) = \mathbf{R}(d_i, d_o)$$

denote data residuals computed using the observed data d_o and synthetic data d_i obtained from the i$^{\text{th}}$ structural model, and let us assume that $a_o(\mathbf{s},\mathbf{r},t)$ is a "good" approximation of data obtained by means of the experiment in Figure 96. Then the interpretation error map corresponding to the i$^{\text{th}}$ structural model can be estimated by solving the following waveform inversion problem

$$\begin{cases} \delta m_i &= \underset{b_i(\mathbf{x})}{\text{argmin}} \int_{\mathbf{s},\mathbf{r},t} \big[a_o(\mathbf{s},\mathbf{r},t) - a_i(\mathbf{s},\mathbf{r},t,b_i)\big]^2 \\ &= \underset{b_i(\mathbf{x})}{\text{argmin}} \int_{\mathbf{s},\mathbf{r},\mathbf{x},t} \delta(\mathbf{x}-\mathbf{r})\big[a_o(\mathbf{s},\mathbf{x},t) - a_i(\mathbf{s},\mathbf{x},t,b_i)\big]^2 \\ &\text{subject to } \mathbf{L}(m+b_i)a_i(\mathbf{x},t,b_i;\mathbf{s}) = \delta(\mathbf{x}-\mathbf{s})w(t) \end{cases} . \tag{153}$$

The Lagrangian of problem 153 is given by

$$\begin{aligned} \mathscr{L} &= \Big\langle \delta(\mathbf{x}-\mathbf{r}), \big[a_o(\mathbf{s},\mathbf{x},t) - a_i(\mathbf{s},\mathbf{x},t,b_i)\big]^2 \Big\rangle_{\mathbf{s},\mathbf{r},\mathbf{x},t} \\ &+ \Big\langle \lambda, \mathbf{L}(m+b_i)a_i(\mathbf{x},t,b_i;\mathbf{s}) - \delta(\mathbf{x}-\mathbf{s})w(t) \Big\rangle_{\mathbf{s},\mathbf{x},t} . \end{aligned} \tag{154}$$

The derivative of the Lagrangian with respect to the model $b(\mathbf{x})$ is given by

$$\begin{aligned}\frac{\partial}{\partial b}\mathscr{L} &= \left\langle \frac{\partial}{\partial b}a_i(\mathbf{s},\mathbf{x},t,b_i), \mathbf{L}^*(m+b_i)\lambda - 2\int_{\mathbf{r}}\delta(\mathbf{x}-\mathbf{r})\big[a_o(\mathbf{s},\mathbf{x},t)-a_i(\mathbf{s},\mathbf{x},t,b_i)\big]\right\rangle_{\mathbf{s},\mathbf{x},t} \\ &+ \left\langle \frac{\partial}{\partial t}\lambda, \frac{2}{(m+b_i)^3}\frac{\partial}{\partial t}a_i(\mathbf{x},t,b_i;\mathbf{s})\right\rangle_{\mathbf{s},\mathbf{x},t}, \end{aligned} \tag{155}$$

(see the appendix in section III.3.7 for details), or more conveniently

$$\begin{cases} \mathbf{L}^*(m+b_i)\lambda = 2\int_{\mathbf{r}}\delta(\mathbf{x}-\mathbf{r})\big[a_o(\mathbf{s},\mathbf{x},t)-a_i(\mathbf{s},\mathbf{x},t,b_i)\big] \\ \frac{\partial}{\partial b}\mathscr{L} = \left\langle \frac{\partial}{\partial t}\lambda, \frac{2}{(m+b_i)^3}\frac{\partial}{\partial t}a_i(\mathbf{x},t,b_i;\mathbf{s})\right\rangle_{\mathbf{s},\mathbf{x},t} \end{cases}. \tag{156}$$

The derivative of the Lagrangian, equation 156, can be used to solve problem 153 using an iterative algorithm of the form

$$\begin{cases} \mathbf{L}(m+b_i)a_i(\mathbf{x},t,b_i;\mathbf{s}) = \delta(\mathbf{x}-\mathbf{s})w(t) & (a) \\ \mathbf{L}^*(m+b_i)R(\mathbf{x},t,b_i;\mathbf{r}) = \int_{\mathbf{r}}\delta(\mathbf{x}-\mathbf{r})\big[a_o(\mathbf{s},\mathbf{x},t)-a_i(\mathbf{s},\mathbf{x},t,b_i)\big] & (b) \\ \Delta b_i = \frac{1}{(m+b_i)^3}\int_{\mathbf{s},t}\frac{\partial}{\partial t}R(\mathbf{x},t,b_i;\mathbf{r})\frac{\partial}{\partial t}a_i(\mathbf{x},t,b_i;\mathbf{s}) & (c) \\ b_i^{n+1} = b_i^n + \Delta b_i^n & (d) \end{cases}. \tag{157}$$

Let us consider the case where the data residuals are obtained using a simple data difference

$$a_o(\mathbf{s},\mathbf{r},t) = d_i - d_o. \tag{158}$$

Figure 97 shows an estimate of the error map in Figure 84c obtained by waveform inversion when the synthetic data d_i in equation 158 is generated using a *block* macro-layered velocity model (section III.2.5.1).

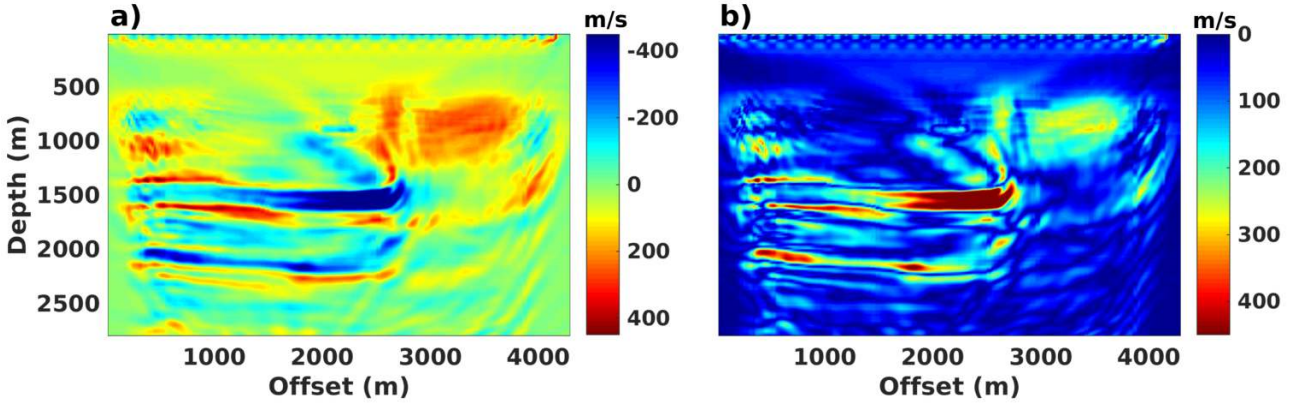

Figure 97: Estimating the error map in Figure 84c by waveform inversion. **a**) Velocity error map obtained using synthetic data generated in a block macro-layered velocity model. **b**) Absolute value of the error map in a.

We note that the error map in Figure 97b differs from the error map in Figure 96d at least in two aspects:

1. The error map in Figure 97b shows an additional interpretation error around the depth of 2000m.
2. The error map in Figure 97b shows additional random oscillatory noise scattered throughout the error map.

The first issue comes from the fact that block macro-layered models are not kinematically accurate; this issue can be alleviated by using macro-layered models that are more kinematically accurate. For example, Figure 98 shows the error map obtained by waveform inversion when the synthetic data d_i in equation 158 is generated using a *reflectivity* macro-layered velocity model (section III.2.5.3). Clearly the error map in Figure 98b is a better estimate of the error map 96d than the error map in Figure 97b.

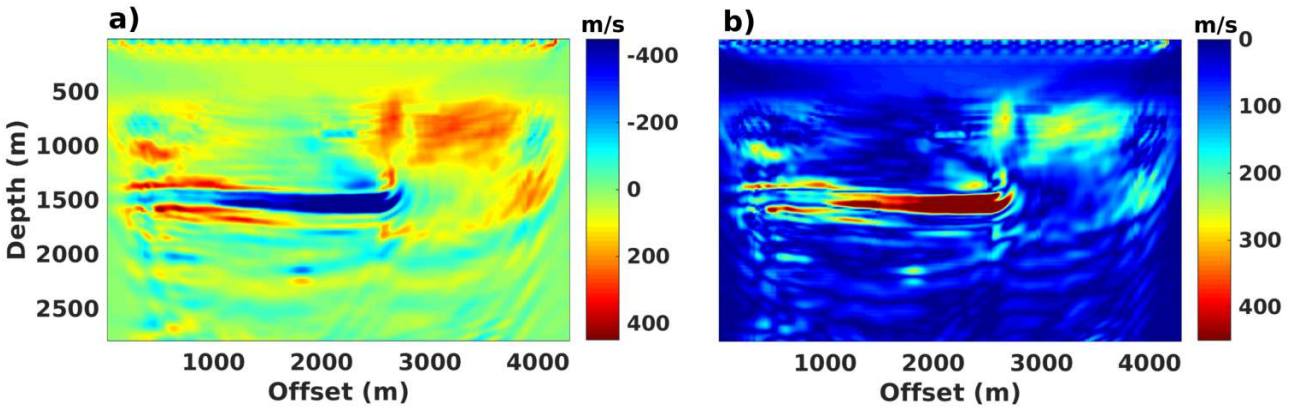

Figure 98: Estimating the error map in Figure 84c by waveform inversion. **a**) Velocity error map obtained using synthetic data generated in a reflectivity macro-layered velocity model. **b**) Absolute value of the error map in a.

The second issue raised above (random oscillatory noise) probably comes from the fact that data residuals obtained by the data difference 158 differ from data obtained by means of the experiment in Figure 96. One way to reduce this oscillatory noise is to use a regularization operator that penalizes oscillations such as the total variation (TV) regularization (e.g. Anagaw and Sacchi, 2012; Lopez, 2014; Esser et al., 2016). The TV regularized version of problem 153 is

$$\left\{\begin{array}{l} \delta m_i = \underset{b_i(\mathbf{x})}{\operatorname{argmin}}\left\{ \int_{\mathbf{s},\mathbf{r},\mathbf{x},t} \delta(\mathbf{x}-\mathbf{r})\big[a_o(\mathbf{s},\mathbf{x},t) - a_i(\mathbf{s},\mathbf{x},t,b_i)\big]^2 + \beta \int_x \sqrt{(\frac{\partial}{\partial_{x1}}b_i)^2 + (\frac{\partial}{\partial_{x2}}b_i)^2} \right\} . \\ \text{subject to } \mathbf{L}(m+b_i)a_i(\mathbf{x},t,b_i;\mathbf{s}) = \delta(\mathbf{x}-\mathbf{s})w(t) \end{array}\right. \tag{159}$$

where $\mathbf{x} = (x_1, x_2)$, and $\beta \in \mathbb{R}$ is a user-defined regularization weight. It can be shown that problem 159 can be solved iteratively using an algorithm of the form

$$\left\{\begin{array}{rl} \mathbf{L}(m+b_i)a_i(\mathbf{x},t,b_i;\mathbf{s}) = \delta(\mathbf{x}-\mathbf{s})w(t) & (a) \\ \mathbf{L}^*(m+b_i)R(\mathbf{x},t,b_i;\mathbf{r}) = \int_{\mathbf{r}} \delta(\mathbf{x}-\mathbf{r})\big[a_o(\mathbf{s},\mathbf{x},t) - a_i(\mathbf{s},\mathbf{x},t,b_i)\big] & (b) \\ \Delta b_i = \frac{1}{(m+b_i)^3}\int_{\mathbf{s},t} \frac{\partial}{\partial t}R(\mathbf{x},t,b_i;\mathbf{r})\frac{\partial}{\partial t}a_i(\mathbf{x},t,b_i;\mathbf{s}) + \beta\nabla\cdot\left[\frac{1}{\sqrt{(\frac{\partial}{\partial_{x1}}b_i)^2+(\frac{\partial}{\partial_{x2}}b_i)^2}}\nabla b_i\right] & (c) \\ b_i^{n+1} = b_i^n + \Delta b_i^n & (d) \end{array}\right. . \tag{160}$$

The algorithm in equation 160 is derived from problem 159 in the same way the algorithm in equation 157 was derived from problem 153; the derivation of the extra term in equation 160c is given in the appendices (section III.3.7). The TV regularized waveform inversion estimate of the error map in Figure 84c is shown in Figure 99, the random oscillatory noise has been considerably reduced.

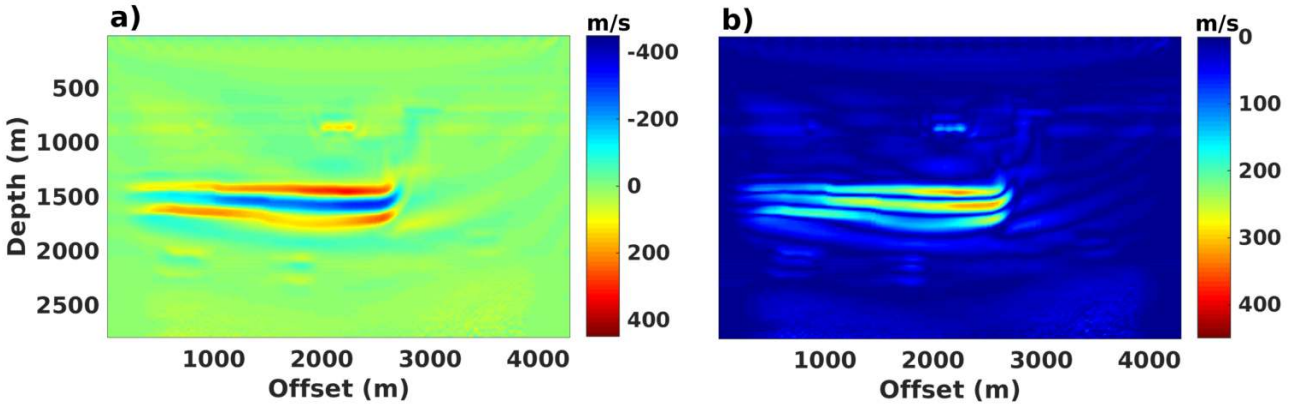

Figure 99: Estimating the error map in Figure 84c by waveform inversion with total variation regularization. **a**) Velocity error map obtained using synthetic data generated in a reflectivity macro-layered velocity model. **b**) Absolute value of the error map in a.

We now test the waveform inversion method on the slightly more complex model shown in Figure 87. Figure 100a shows the error map estimated by means of the experiment in Figure 96; this error map is the best we can hope to get without knowledge of the reference model by waveform inversion as described in this section. Figure 100b shows the error map estimated from synthetic data generated in a block macro-layered velocity model, while Figure 100c shows the error map estimated from synthetic data generated in a reflectivity macro-layered velocity model. The error map from the block macro-layered model (Figure 100b) visually looks closer to the target error map (Figure 100a) than the error map from the reflectivity macro-layered model (100c). However, because we know that the block macro-layered model may introduce velocity errors as discussed previously, we will always prefer the error map from the reflectivity macro-layered model and try to improve it by regularization. The TV regularized version of the error map in Figure 100c is shown in Figure 100d. We find that the error map in Figure 100d is a reasonable estimate of the target error map in Figure 100a, itself a reasonable estimate of the true error map in Figure 87c.

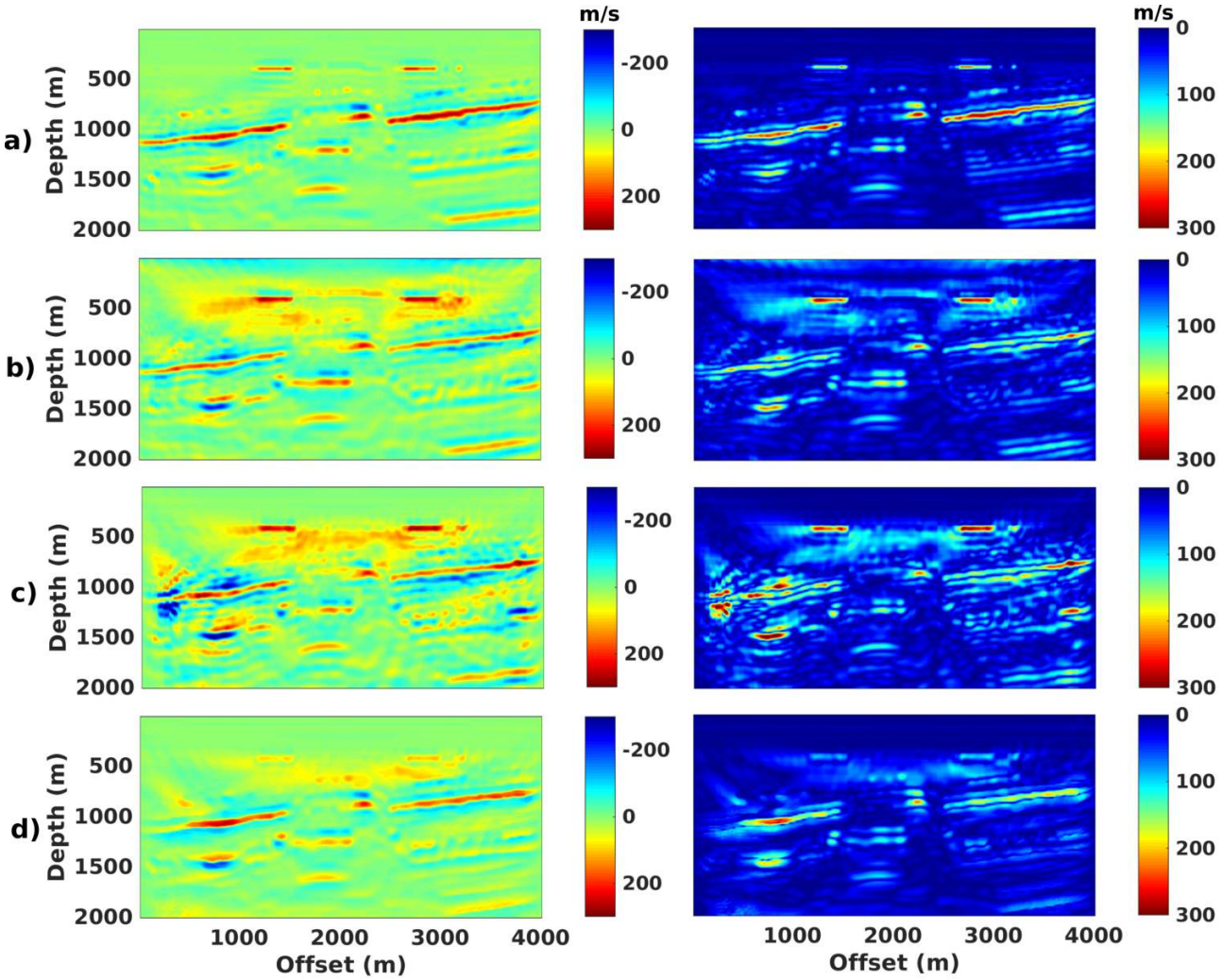

Figure 100: Estimating the error map in Figure 87c by waveform inversion. The figures in the first column are velocity error maps; the second column is the absolute value of the first column. **a**) Target error map obtained by means of the experiment in Figure 96. **b**) Error map obtained using synthetic data generated in a block macro-layered velocity model. **c**) Error map obtained using synthetic data generated in a reflectivity macro-layered velocity model. **d**) Error map obtained using synthetic data generated in a reflectivity macro-layered velocity model plus total variation regularization.

III.3.5 Conclusion & Discussion

In this chapter, we study the practical aspects of appraising structural interpretations using surface seismic data. The global workflow we use for appraising structural interpretations is to generate synthetic data for each interpretation, and then use the synthetic data to compute a misfit value for each interpretation. We use misfit functions that are localized in space (i.e. a misfit function that is a function of space). The localized misfit functions are composed of two main components: (1) a residual operator that computes the difference between synthetic and observed data, (2) and a projection operator that projects the data residuals into the image-domain. We present two strategies for computing the localized misfit functions:

- In the first strategy (section III.3.3), the residual operator does most of the work in the misfit function by computing phase-shifts between observed and synthetic data; these phase shifts are intended to be proportional to horizon shifts between the reference and interpreted structural models. A linear projection operator is then used to project the phase-shifts to the image-domain. The challenge in this first strategy is to propose ap-

propriate residual operators that can compute phase-shifts accurately. We conclude that phase-shift residual operators based on dynamic warping are not appropriate mainly because they rely on the assumption that observed data are a warped version of synthetic data (e.g. Hale, 2013). This assumption implies that synthetic structural models (models from seismic interpretation) should have the same number of layers as the true structural model, which is usually not the case. Furthermore, the assumption is usually also not valid for faulted complex structural models even when they have the same number of layers as the reference model. A natural perspective for this first strategy is to test other phase-shift residual operators, particularly operators based on optimal transport (e.g. Engquist and Froese, 2014; Métivier et al., 2016).

- In the second strategy (section III.3.4), the residual operator is based on a simple L1-difference; in this case, the projection operator does most of the work in the misfit function by projecting the data residual in the image domain using an iterative process. Here, the projection operator is based on waveform inversion. This strategy therefore inherits all the computational challenges already familiar from the waveform inversion literature. The advantage of this strategy is that it does not require synthetic structural models to have the same number of layers as the reference model. Current research on this second strategy involves finding appropriate regularization schemes to stabilize the waveform inversion of data residuals. A promising perspective to investigate is the regularization scheme presented by Esser et al. (2016); their regularization scheme seems to be able to recover low wavenumber velocity components from inversion. Such an ability would certainly be valuable for our method.

While we concluded that phase-shift residuals based on dynamic warping were not appropriate for the first strategy, we expect them to be useful in the second strategy. In particular, consider the problem

$$\left\{\begin{array}{rll} a_o(\mathbf{s},\mathbf{r},t) & = \mathbf{R}_1\big[d_i(\mathbf{s},\mathbf{r},t), d_o(\mathbf{s},\mathbf{r},t)\big] & (a) \\ \delta m_i & = \underset{b_i(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},t} \mathbf{R}_2\big[a_i(\mathbf{s},\mathbf{r},t), a_o(\mathbf{s},\mathbf{r},t)\big]^2 & (b) \\ & = \underset{b_i(\mathbf{x})}{\operatorname{argmin}} \int_{\mathbf{s},\mathbf{r},\mathbf{x},t} \delta(\mathbf{x}-\mathbf{r})\mathbf{R}_2\big[a_i(\mathbf{s},\mathbf{x},t), a_o(\mathbf{s},\mathbf{x},t)\big]^2 & (c) \\ \text{subject to} & \mathbf{L}(m+b_i)a_i(\mathbf{x},t,b_i;\mathbf{s}) = \delta(\mathbf{x}-\mathbf{s})w(t) & (d) \end{array}\right. ; \qquad (161)$$

the method presented in section III.3.4 is a specific case of problem 161 where both $\mathbf{R}_1$ and $\mathbf{R}_2$ are L1-difference data residual operators. A promising perspective is to solve problem 161 where $\mathbf{R}_1$ is an L1-difference data residual operator and $\mathbf{R}_2$ is a phase-shift data residual operator (based on dynamic warping for example). It should be noted in problem 161 that $\mathbf{R}_1$ is the residual operator $\mathbf{R}$ in equation 137 while $\mathbf{R}_2$ is part of the projection operator $\mathbf{P}$ in equation 137.

III.3.6 Acknowledgements

The authors would like to thank Ludovic Métivier and Pierre Thore for fruitful discussions about the subject. This work also greatly benefited from constructive discussions with Thomas Bodin and Yann Capdeville within the framework of the HIWAI Project.

This work was done in the frame of the RING project at Université de Lorraine. The sponsors of the RING-GOCAD Consortium managed by ASGA are hereby acknowledged for their support. We would also like to thank Paradigm for providing the SKUA-GOCAD software used for structural modeling.

III.3.7 Appendix

Derivation of equation 147

$$\begin{aligned}
\frac{\partial}{\partial l}\mathbf{R}|_f &= \int_{\mathbf{s},\mathbf{r},t} \frac{\partial}{\partial f}\big[d_o(\mathbf{s},\mathbf{r},t-f) - d_i(\mathbf{s},\mathbf{r},t)\big]^2 \\
&= \int_{\mathbf{s},\mathbf{r},t} \frac{\partial}{\partial f}\big[d_o(\mathbf{s},\mathbf{r},g) - d_i(\mathbf{s},\mathbf{r},t)\big]^2, \text{ with } g(f) = t - f \\
&= \int_{\mathbf{s},\mathbf{r},t} \frac{\partial}{\partial f} h(g)^2, \text{ with } h(g) = d_o(\mathbf{s},\mathbf{r},g) - d_i(\mathbf{s},\mathbf{r},t) \\
&= \int_{\mathbf{s},\mathbf{r},t} \frac{\partial}{\partial h} h^2 \frac{\partial}{\partial g} h(g) \frac{\partial}{\partial f} g(f) \\
&= -2 \int_{\mathbf{s},\mathbf{r},t} \big[d_o(\mathbf{s},\mathbf{r},t) - d_i(\mathbf{s},\mathbf{r},g)\big] \frac{\partial}{\partial g} d_o(\mathbf{s},\mathbf{r},g)
\end{aligned}$$

Derivation of equation 150

The derivative of the first term in equation 149 with respect to the model m is given by

$$\frac{\partial}{\partial m}\Big\langle \delta(\mathbf{x}-\mathbf{r}), f(\mathbf{s},\mathbf{x},t,m)^2 \Big\rangle_{\mathbf{s},\mathbf{r},\mathbf{x},t} = \Big\langle \delta(\mathbf{x}-\mathbf{r}), 2f(\mathbf{s},\mathbf{x},t,m)\frac{\partial}{\partial m} f(\mathbf{s},\mathbf{x},t,m) \Big\rangle_{\mathbf{s},\mathbf{r},\mathbf{x},t}.$$

The derivative

$$\begin{aligned}
&\frac{\partial}{\partial m}\Big(\big[d_o(\mathbf{s},\mathbf{x},\tau) - d_i(\mathbf{s},\mathbf{x},t,m)\big]\dot{d}_o(\mathbf{s},\mathbf{x},\tau) \Big) \\
= \;& \dot{d}_o(\mathbf{s},\mathbf{x},\tau)\frac{\partial}{\partial m}\big[d_o(\mathbf{s},\mathbf{x},\tau) - d_i(\mathbf{s},\mathbf{x},t,m)\big] + \big[d_o(\mathbf{s},\mathbf{x},\tau) - d_i(\mathbf{s},\mathbf{x},t,m)\big]\frac{\partial}{\partial m}\dot{d}_o(\mathbf{s},\mathbf{x},\tau) \\
= \;& \dot{d}_o(\mathbf{s},\mathbf{x},\tau)\big[\dot{d}_o(\mathbf{s},\mathbf{x},\tau)\frac{\partial}{\partial m}\tau - \frac{\partial}{\partial m} d_i(\mathbf{s},\mathbf{x},t,m)\big] + \big[d_o(\mathbf{s},\mathbf{x},\tau) - d_i(\mathbf{s},\mathbf{x},t,m)\big]\ddot{d}_o(\mathbf{s},\mathbf{x},\tau)\frac{\partial}{\partial m}\tau \\
= \;& \Big\{\dot{d}_o(\mathbf{s},\mathbf{x},\tau)^2 + \big[d_o(\mathbf{s},\mathbf{x},\tau) - d_i(\mathbf{s},\mathbf{x},t,m)\big]\ddot{d}_o(\mathbf{s},\mathbf{x},\tau)\Big\}\frac{\partial}{\partial m}\tau - \dot{d}_o(\mathbf{s},\mathbf{x},\tau)\frac{\partial}{\partial m} d_i(\mathbf{s},\mathbf{x},t,m) \\
= \;& -\Big\{\dot{d}_o(\mathbf{s},\mathbf{x},\tau)^2 + \big[d_o(\mathbf{s},\mathbf{x},\tau) - d_i(\mathbf{s},\mathbf{x},t,m)\big]\ddot{d}_o(\mathbf{s},\mathbf{x},\tau)\Big\}\frac{\partial}{\partial m} f - \dot{d}_o(\mathbf{s},\mathbf{x},\tau)\frac{\partial}{\partial m} d_i(\mathbf{s},\mathbf{x},t,m)
\end{aligned}$$

is used to compute the derivative of the second term in equation 149. The derivative of the last term in equation 149 is given by

$$\begin{aligned}
\frac{\partial}{\partial m}\Big\langle \lambda_2, d_i(\mathbf{s},\mathbf{x},t,m) - \mathbf{B}m_i(\mathbf{x}) \Big\rangle_{\mathbf{s},\mathbf{x},t} &= \Big\langle \lambda_2, \frac{\partial}{\partial m} d_i(\mathbf{s},\mathbf{x},t,m) \Big\rangle_{\mathbf{s},\mathbf{x},t} - \Big\langle \lambda_2, \mathbf{B}\frac{\partial}{\partial m} m_i(\mathbf{x}) \Big\rangle_{\mathbf{s},\mathbf{x},t} \\
&= \Big\langle \lambda_2, \frac{\partial}{\partial m} d_i(\mathbf{s},\mathbf{x},t,m) \Big\rangle_{\mathbf{s},\mathbf{x},t} - \Big\langle \mathbf{B}^*\lambda_2, 1 \Big\rangle_{\mathbf{x}};
\end{aligned}$$

that $\Big\langle f(\mathbf{s},\mathbf{x},t), \mathbf{B}f(\mathbf{x}) \Big\rangle_{\mathbf{s},\mathbf{x},t} = \Big\langle \mathbf{B}^* f(\mathbf{s},\mathbf{x},t), f(\mathbf{x}) \Big\rangle_{\mathbf{x}}$ follows from equation 19 and equation 20: $\mathbf{B}$ is the Born modeling operator, $\mathbf{B}^*$ is the reverse time migration operator, and $\mathbf{B}^* f(\mathbf{s},\mathbf{x},t) = \int_{\mathbf{s},t} \mathbf{B}f(\mathbf{s},\mathbf{x},t)$.

Derivation of equation 155

The derivative of the first term in equation 154 with respect to the model parameter b is given by

$$\begin{aligned} & \frac{\partial}{\partial b}\left\langle \delta(\mathbf{x}-\mathbf{r}), \left[a_o(\mathbf{s},\mathbf{x},t) - a_i(\mathbf{s},\mathbf{x},t,b_i)\right]^2 \right\rangle_{\mathbf{s},\mathbf{r},\mathbf{x},t} \\ = & \left\langle \delta(\mathbf{x}-\mathbf{r}), -2\left[a_o(\mathbf{s},\mathbf{x},t) - a_i(\mathbf{s},\mathbf{x},t,b_i)\right]\frac{\partial}{\partial b}a_i(\mathbf{s},\mathbf{x},t,b_i) \right\rangle_{\mathbf{s},\mathbf{r},\mathbf{x},t}. \end{aligned}$$

The derivative of the second term in equation 154 is given by

$$\begin{aligned} & \frac{\partial}{\partial b}\left\langle \lambda, \mathbf{L}(m+b_i)a_i(\mathbf{x},t,b_i;\mathbf{s}) - \delta(\mathbf{x}-\mathbf{s})w(t) \right\rangle_{\mathbf{s},\mathbf{x},t} \\ = & \left\langle \lambda, \left[\frac{\partial}{\partial b}\mathbf{L}(m+b_i)\right]a_i(\mathbf{x},t,b_i;\mathbf{s}) \right\rangle_{\mathbf{s},\mathbf{x},t} + \left\langle \lambda, \mathbf{L}(m+b_i)\frac{\partial}{\partial b}a_i(\mathbf{x},t,b_i;\mathbf{s}) \right\rangle_{\mathbf{s},\mathbf{x},t} \\ = & \left\langle \lambda, -\frac{2}{(m+b_i)^3}\frac{\partial^2}{\partial t^2}a_i(\mathbf{x},t,b_i;\mathbf{s}) \right\rangle_{\mathbf{s},\mathbf{x},t} + \left\langle \mathbf{L}^*(m+b_i)\lambda, \frac{\partial}{\partial b}a_i(\mathbf{x},t,b_i;\mathbf{s}) \right\rangle_{\mathbf{s},\mathbf{x},t} \\ = & \left\langle \frac{\partial}{\partial t}\lambda, \frac{2}{(m+b_i)^3}\frac{\partial}{\partial t}a_i(\mathbf{x},t,b_i;\mathbf{s}) \right\rangle_{\mathbf{s},\mathbf{x},t} + \left\langle \mathbf{L}^*(m+b_i)\lambda, \frac{\partial}{\partial b}a_i(\mathbf{x},t,b_i;\mathbf{s}) \right\rangle_{\mathbf{s},\mathbf{x},t}, \end{aligned}$$

where $\mathbf{L}^*$ is the adjoint of $\mathbf{L}$.

Derivation of equation 160c

The first term in equation 160c is already familiar from equation 157c. The second term in equation 160c comes from the derivative of the regularization term in problem 159, in particular

$$
\begin{aligned}
& \frac{\partial}{\partial b}\left\langle 1, \sqrt{(\frac{\partial}{\partial_{x1}}b)^2 + (\frac{\partial}{\partial_{x2}}b)^2}\right\rangle_{\mathbf{x}} \\
= & \left\langle 1, \frac{\partial}{\partial b}\sqrt{(\frac{\partial}{\partial_{x1}}b)^2 + (\frac{\partial}{\partial_{x2}}b)^2}\right\rangle_{\mathbf{x}} \\
= & \left\langle 1, \frac{\partial}{\partial b}\sqrt{f}\right\rangle_{\mathbf{x}}, \text{ with } f = (\frac{\partial}{\partial_{x1}}b)^2 + (\frac{\partial}{\partial_{x2}}b)^2 \\
= & \left\langle 1, \frac{\partial}{\partial f}\sqrt{f}\frac{\partial}{\partial b}f\right\rangle_{\mathbf{x}} = \left\langle \frac{1}{2\sqrt{f}}, \frac{\partial}{\partial b}f\right\rangle_{\mathbf{x}} \\
= & \left\langle \frac{1}{2\sqrt{f}}, \frac{\partial}{\partial b}\big[(\frac{\partial}{\partial_{x1}}b)^2 + (\frac{\partial}{\partial_{x2}}b)^2\big]\right\rangle_{\mathbf{x}} \\
= & \left\langle \frac{1}{2\sqrt{f}}, \frac{\partial}{\partial b}f_1^2 + \frac{\partial}{\partial b}f_2^2\right\rangle_{\mathbf{x}}, \text{ with } f_1 = \frac{\partial}{\partial_{x1}}b \text{ and } f_2 = \frac{\partial}{\partial_{x2}}b \\
= & \left\langle \frac{1}{2\sqrt{f}}, \frac{\partial}{\partial f_1}f_1^2\frac{\partial}{\partial b}f_1 + \frac{\partial}{\partial f_2}f_2^2\frac{\partial}{\partial b}f_2\right\rangle_{\mathbf{x}} \\
= & \left\langle \frac{1}{\sqrt{f}}, f_1\frac{\partial}{\partial b}f_1 + f_2\frac{\partial}{\partial b}f_2\right\rangle_{\mathbf{x}} \\
= & \left\langle \frac{1}{\sqrt{f}}, \frac{\partial}{\partial_{x1}}b\left(\frac{\partial}{\partial b}\frac{\partial}{\partial_{x1}}b\right) + \frac{\partial}{\partial_{x2}}b\left(\frac{\partial}{\partial b}\frac{\partial}{\partial_{x2}}b\right)\right\rangle_{\mathbf{x}} \\
= & \left\langle \frac{1}{\sqrt{f}}, \nabla b\cdot\left(\frac{\partial}{\partial b}\nabla b\right)\right\rangle_{\mathbf{x}} = \left\langle \frac{1}{\sqrt{f}}, \nabla b\cdot\nabla\big[\frac{\partial}{\partial b}b\big]\right\rangle_{\mathbf{x}} \\
= & \left\langle 1, \frac{1}{\sqrt{f}}\nabla b\cdot\nabla\big[1\big]\right\rangle_{\mathbf{x}} \\
= & -\left\langle \nabla\cdot\left[\frac{1}{\sqrt{f}}\nabla b\right], 1\right\rangle_{\mathbf{x}} = -\left\langle \nabla\cdot\left[\frac{1}{\sqrt{(\frac{\partial}{\partial_{x1}}b)^2 + (\frac{\partial}{\partial_{x2}}b)^2}}\nabla b\right], 1\right\rangle_{\mathbf{x}}
\end{aligned}
$$

The last line follows from the divergence theorem and neglecting boundary contributions.

Bibliography

P. Abrahamsen. Bayesian kriging for seismic depth conversion of a multi-layer reservoir. In A. Soares, editor, *Geostatistics Troia 92*, pages 385–398. 1993.

J. Alcalde, C. Bond, G. Johnson, J. Ellis, and R. Butler. Impact of seismic image quality on fault interpretation uncertainty. *GSA Today*, 27:4–10, 2017.

A. Anagaw and M. Sacchi. Edge-preserving seismic imaging using the total variation method. *Journal of Geophysics and Engineering*, 9(2), 2012.

G. Beylkin, M. Oristaglio, and D. Miller. Spatial resolution of migration algorithms. In A. Berkhout, J. Ridder, and L. V. der Wal, editors, *Acoustical Imaging*, pages 155–168. 1985.

N. Bleistein, J. W. Stockwell, and J. K. Cohen. *Mathematics of Multidimensional Seismic Imaging, Migration, and Inversion*. Interdisciplinary Applied Mathematics. Springer New York, 2001.

C. Bond. Uncertainty in structural interpretation: Lessons to be learnt. *Journal of Structural Geology*, 74:185–200, 2015.

C. Bond, Z. Shipton, and S. Jones. What do you think this is? Conceptual uncertainty in geoscience interpretation. *GSA Today*, 17:4–10, 2007.

K. Bube, J. Kane, T. Nemeth, D. Medwede, and O. Mikhailov. The influence of stacking velocity uncertainties on structural uncertainties. In *74th SEG Annual Meeting*, Denver, Colorado, 2004.

J. Carter, P. Ballester, Z. Tavassoli, and P. King. Our calibrated model has poor predictive value: An example from the petroleum industry. *Reliability Engineering and System Safety*, 91:1373–1381, 2006.

G. Caumon, P. Collon-Drouaillet, C. Le Carlier de Veslud, S. Viseur, and J. Sausse. Surface-Based 3D Modeling of Geological Structures. *Mathematical Geosciences*, 41:927–945, 2009.

N. Cherpeau and G. Caumon. Stochastic structural modelling in sparse data situations. *Petroleum Geoscience*, 21:233–247, 2015.

N. Cherpeau, G. Caumon, J. Caers, and B. Lévy. Method for Stochastic Inverse Modeling of Fault Geometry and Connectivity Using Flow Data. *Mathematical Geosciences*, 44:147–168, 2012.

O. Christensen. *Functions, Spaces, and Expansions: Mathematical Tools in Physics and Engineering*. Applied and Numerical Harmonic Analysis. Birkhauser Basel, 2010.

J. Cohen, F. Hagin, and N. Bleistein. Three-dimensional Born inversion with an arbitrary reference. *GEOPHYSICS*, 51:1552–1558, 1986.

B. Colletta, J. Letouzey, R. Pinedo, J. F. Ballard, and P. Balé. Computerized X-ray tomography analysis of sandbox models: Examples of thin-skinned thrust systems. *Geology*, 19:1063–1067, 1991.

F. Collino and C. Tsogka. Application of the perfectly matched absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. *GEOPHYSICS*, 66: 294–307, 2001.

B. Engquist and B. Froese. Application of the Wasserstein metric to seismic signals. *Communications in Mathematical Sciences*, 12:979–988, 2014.

E. Esser, L. Guasch, T. van Leeuwen, A. Aravkin, and F. Herrmann. Total-variation regularization strategies in full-waveform inversion. *arXiv preprint*, 2016.

J. Etgen, S. Gray, and Y. Zhang. An overview of depth migration in exploration geophysics. *GEOPHYSICS*, 74(6):WCA5–WCA17, 2009.

A. Fichtner, H. Bunge, and H. Igel. The adjoint method is seismology I. Theory. *Physics of the Earth and Planetary Interiors*, 157:86–104, 2006.

S. Fomel and E. Landa. Structural uncertainty of time-migrated seismic images. *Journal of Applied Geophysics*, 101:27–30, 2014.

S.-K. Foss, M. Rhodes, B. Dalstrom, C. Gram, and A. Welbon. Geologically constrained seismic imaging - Workflow. *GEOPHYSICS*, 73(5):VE313–VE319, 2008.

P. Fullagar, G. Pears, and B. McMonnies. Constrained inversion of geologic surfaces - pushing the boundaries. *The Leading Edge*, 28:98–105, 2008.

G. Gardner, L. Gardner, and A. Gregory. Formation velocity and density-the diagnostic basics for stratigraphic traps. *GEOPHYSICS*, 69:770–780, 1974.

Z. Hajnal and I. Sereda. Maximum uncertainty of interval velocity estimates. *GEOPHYSICS*, 46:1543–1547, 1981.

D. Hale. Dynamic warping of seismic images. *GEOPHYSICS*, 78(2):S105–S115, 2013.

L. Holden, P. Mostad, B. F. Nielsen, J. Gjerde, C. Townsend, and S. Ottesen. Stochastic Structural Modeling. *Mathematical geology*, 35:899–914, 2003.

J. Hoyes and T. Cheret. A review of global interpretation methods for automated 3d horizon picking. *The Leading Edge*, 30:38–47, 2011.

C. Julio, G. Caumon, and M. Ford. Sampling uncertainty about segmented normal fault interpretation using a stochastic downscaling method. *Tectonophysics*, 639:56–67, 2015.

F. Lallier, G. Caumon, J. Borgomano, S. Viseur, F. Fournier, C. Antoine, and T. Gentilhomme. Relevance of the stochastic stratigraphic well correlation approach for the study of complex carbonate settings: application to the Malampaya buildup (Offshore Palawan, Philippines). *Geological Society, Londonm Special Publications*, 370:265–275, 2012.

E. Landa and P. Thore. Realistic Finite Differences Modeling - A Case Study. In *69th EAGE Conference and Exhibition incorporating SPE EUROPEC 2007*, London, UK, 2007.

I. Lecomte, H. Gjoystdal, and A. Drottning. Simulated Prestack Local Imaging: a robust and efficient interpretation tool to control illumination, resolution, and time-lapse properties of reservoirs. In *73rd SEG Annual Meeting*, Dallas, Texas, 2003.

I. Lecomte, P. Lavadera, C. Botter, I. Anell, S. Buckley, C. Eide, A. Grippa, V. Mascolo, and S. Kjoberg. 2(3)D convolution modeling of complex geological targets - Beyond 1D convolution. *First Break*, 34:99–107, 2016.

M. Lecour, R. Cognot, I. Duvinage, P. Thore, and J.-C. Dulac. Modelling of stochastic faults and fault networks in a structural uncertainty study. *Petroleum Geoscience*, 7:S31–S42, 2001.

L. Li, J. Caers, and P. Sava. Assessing seismic uncertainty via geostatistical velocity-model perturbation and image registration: An application to subsalt imaging. *The Leading Edge*, 34:1064–1066, 2015.

C. C. Lopez. *Speed up and regularization techniques for seismic full waveform inversion*. PhD thesis, Université Nice Sophia Antipolis, France, 2014.

Y. Ma and D. Hale. Wave-equation reflection traveltime inversion with dynamic warping and full-waveform inversion. *GEOPHYSICS*, 78:R223–R233, 2013.

J. Messud, M. Reinier, H. Prigent, P. Guillaume, T. Coléou, and S. Masclet. Extracting seismic uncertainties from tomographic velocity inversion and their use in reservoir analysis. *The Leading Edge*, 36:127–132, 2017.

L. Métivier, R. Brossier, Q. Mérigot, E. Oudet, and J. Virieux. Measuring the misfit between seismograms using an optimal transport distance: Application to full waveform inversion. *Geophys. J. Int.*, 205:345–377, 2016.

F. Pauget, S. Lacaze, and T. Valding. A global approach in seismic interpretation based on cost function minimization. In *79th SEG Annual Meeting*, Houston, Texas, 2009.

I. Pawelec. UNCERTAINTY QUANTIFICATION IN SEISMIC IMAGING. Master's thesis, Colorado School of Mines, Colorado, USA, 2018.

S. Pon and L. R. Lines. Sensitivity analysis of seismic depth migrations. *GEOPHYSICS*, 70 (2):S39–S42, 2005.

G. Pratt and R. Shipp. Seismic waveform inversion in the frequency domain, part 2: Fault delineation in sediments using crosshole data. *GEOPHYSICS*, 64:902–914, 1999.

L. Santos, J. Schleicher, M. Tygel, and P. Hubral. Seismic modeling by demigration. *GEOPHYSICS*, 65:1281–1286, 2000.

W. Schneider. Integral formulation for migration in two and three dimensions. *GEOPHYSICS*, 43:49–76, 1978.

T. Stark. Relative geologic time (age) volumes-Relating every seismic sample to a geologically reasonable horizon. *The Leading Edge*, 23:928–932, 2004.

J. Suiter, R. Romani, J. Arnaud, S. Hollingworth, and K. Hawkins. Reducing structural uncertainties through anisotropic pre-stack depth imaging: examples from the Elgin/Franklin/Glenelg HP/HT fields area, Central North Sea. *Geological Society of London*, 6:1435–1448, 2005.

S. Suzuki, J. Caers, and G. Caumon. Dynamic data integration for structural modeling: Model screening approach using a distance-based model parameterization. *Computational Geosciences*, 12:105–119, 2008.

P. Thore and A. Haas. A practical formulation of migration errors due velocity uncertainties. In *58th EAGE Conference and Exhibition*, Amsterdam, The Netherlands, 1996.

P. Thore, A. Shtuka, M. Lecour, T. Ait-Ettajer, and R. Cognot. Structural uncertainties: Determination, management, and applications. *GEOPHYSICS*, 67:840–852, 2002.

P. Toft. *The Radon Transform - Theory and implementation.* PhD thesis, Technical University of Denmark, Denmark, 1996.

J. Virieux. SH-wave propagation in heterogeneous media: Velocity-stress finite-difference method. *GEOPHYSICS*, 49:1933–1942, 1984.

J. Wellmann, F. G.Horowitz, E. Schill, and K. Regenauer-Lieb. Towards incorporating uncertainty of structural data in 3D geological inversion. *Tectonophysics*, 490:141–151, 2010.

J. Wellmann, M. Lindsay, J. Poh, and M. Jessell. Validating 3-D structural models with geological knowledge for improved uncertainty evaluations. *Energy Procedia*, 59:374–381, 2014.

X. Wu. Building 3D subsurface models conforming to seismic structural and stratigraphic features. *GEOPHYSICS*, 82(3):IM21–IM30, 2017.

X. Wu and D. Hale. Horizon volumes with interpreted constraints. *GEOPHYSICS*, 80(2): IM21–IM33, 2015.

General Conclusion

Subsurface structural models find applications in natural resource estimation, risk management, and numerical simulations; it is therefore important that structural models represent the geometry of subsurface geological objects accurately. In this manuscript, we study the problem of reducing structural uncertainties using seismic data. In particular, we study the problem of using seismic data to determine which structural models are more likely than others in a set of geologically plausible structural models. We refer to this problem as *appraising structural models using seismic data*. Appraising structural models using seismic data is an integrative workflow that involves seismic imaging and structural modeling. Consequently, this manuscript also investigates some problems in seismic imaging and structural modeling.

The first Part of the manuscript is devoted to seismic imaging. Seismic imaging based on waveform inversion produces optimal quantitative seismic images of the subsurface by minimizing the difference between observed and synthetic data. Waveform inversion is however computationally intensive. In chapter I.2, I propose to use reverse-time migration (RTM) as a preconditioner for waveform inversion. Numerical experiments show that the proposed preconditioner accelerates both linearized waveform inversion (least squares reverse time migration) and nonlinear waveform inversion (full waveform inversion) by at least an order of magnitude. I justify the positive numerical performance of the proposed preconditioner by showing algebraically that a low pass filter of the RTM image can approximate the diagonal elements of the Hessian matrix of the objective function under appropriate assumptions. However, I am still unable to propose a physical meaning of the low pass filtering and how it relates the RTM image to the elements of the diagonal of the Hessian matrix; this is still a subject of ongoing investigation. All seismic imaging methods require an accurate imaging velocity model. In chapter I.3, we propose a generalized extended Kirchhoff imaging operator for velocity modeling; the operator is generalized in the sense that it describes multiple data-domain extensions (e.g. shot, offset, and angle extensions) and image-domain extensions (e.g. time-lag and space-lag extensions) simultaneously. The advantages of the proposed generalized extended operator are twofold: firstly, it allows a unified implementation for multiple extensions (i.e. a single implementation that is valid for multiple extensions); secondly, the operator leads to a unified gradient-based migration velocity analysis (MVA) scheme. We confirm the ability of the proposed generalized extended operator to capture image distortion caused by inaccurate velocity by applying it to a ray-based MVA experiment. We then use the proposed operator to derive a diferential semblance gradient-based MVA that is valid simultaneously for all data-domain extensions; the practical implementation of this unified gradient-based MVA has been postponed for further investigation.

The second Part of the manuscript is devoted to structural modeling, particularly implicit structural interpolation. In chapter II.1, we introduce Finite Difference Structural Implicit Modeling (FDSIM). The advantages of FDSIM are twofold: firstly, it is relatively easy to implement and to optimize since it is based on finite differences on regular grids; secondly, because it handles discontinuities by rasterization, FDSIM has shown to easily handle very complex fault-networks. The main disadvantage of the method is that it may require a very

fine resolution depending on the complexity of the fault-network, sometimes leading to memory limits. We also propose new regularization operators in chapter II.1; the particularity of these operators is that they do not need to be implemented on boundary nodes, a property which is very appealing in implicit modeling where boundary conditions are usually unknown. In chapter II.2, we introduce Finite Element Structural Implicit Modeling (FESIM). FESIM is based on a finite element implementation of the regularization operators proposed in chapter II.1. We show that the conventional finite element familiar for solving boundary value problems has to be slightly modified for implicit modeling where boundary conditions are usually unknown. One advantage of FESIM over FDSIM is that it requires less memory for complexly faulted structural models. The disadvantage of FESIM over FDSIM is that it can be quite difficult to model structural models with complex fault-networks due to meshing challenges. Current research in FESIM includes finding more automated strategies to generate a mesh for structural models with complex fault-networks. In chapter II.3, we propose two new strategies for handling large thickness variations in implicit structural modeling: the first strategy constrains the output implicit function (stratigraphic field) to follow the orientation of geological layers estimated from input data, while the second strategy uses an anisotropic regularization operator that is consistent with the orientation of geological layers. These strategies are still under investigation and they show promising results.

The third Part of the manuscript is devoted to appraising structural models/interpretations using seismic data. In chapter III.1, we discuss about interpretation uncertainties and show how a single seismic image may support multiple geologically consistent structural models. In chapter III.2, we introduce and formalize the problem of appraising structural interpretations using seismic data. We propose to solve the problem by generating synthetic data for each structural interpretation and then compute misfit values for each interpretation; this allows us to rank the different structural interpretations. The main challenge of appraising structural models using seismic data is to propose appropriate data misfit functions. We derive a set of conditions that have to be satisfied by the data misfit function for a successful appraisal of structural models. We argue that since it is not possible to satisfy these conditions using vertical seismic profile (VSP) data, it is not possible to appraise structural interpretation using VSP data in the most general case. However, we show that it is always possible to get a better approximation of those conditions using VSP data that illuminate the target region from different directions. We present an application of appraising structural interpretations using VSP data to a realistic synthetic case. The conditions imposed on the data misfit function can in principle be satisfied for surface seismic data. In practice however, it remains a challenge to propose and compute data misfit functions that satisfy those conditions. In chapter III.3, we focus on the practical aspects of appraising structural interpretations using surface seismic data. We propose a general data misfit formula that is made of two main components: (1) a residual operator that computes data residuals, that is, the difference between observed and synthetic data, and (2) a projection operator that projects the data residuals to the image-domain. This misfit function is therefore localized in space, as it outputs data misfit values in the image-domain. We present two strategies for implementing the localized data misfit function:

- In the first strategy, the residual operator does most of the work in the misfit function by computing phase shifts between synthetic and observed data; phase shifts are, in principle, correlated with horizon shifts between the true structural model and the interpreted structural model. A linear operator is then used to project the phase-shift residuals into the image-domain to give an indication of horizon shifts in the subsurface. We conclude that phase-shift based on dynamic-warping are not appropriate for appraising structural models, when using a linear projection operator, mainly because they require observed data to be a warped version of synthetic data, a requirement usually not satisfied for this particular problem.

- In the second strategy, the residual operator does most of the work in the misfit function by iteratively projecting to the image-domain data residuals based on L1-differences. In this case, the projection operator is based on nonlinear waveform inversion and therefore inherits all the computational challenges familiar from the full waveform inversion literature. Waveform inversion of data residuals has shown to be quite unstable and often requires regularization.

We believe that future research on appraising structural models using surface seismic data should include investigating other phase-shift residual operators (e.g. optimal transport) for the first strategy, and investigating robust regularization techniques and phase-shift based waveform inversion projection operators for the second strategy.

The work presented in this manuscript is a small part of the larger workflow shown in Figure 101 for reducing structural interpretation uncertainties using seismic data. The long term objective is to use seismic data to update a structural model built from seismic images. There are three main problems in this global workflow: (1) generating seismic data from structural models, (2) computing appropriate data misfit values, (3) and updating the structural model in a manner that reduces data misfit values. Each of these three problems has its own challenges. The challenge in generating synthetic data from structural models is to create macro-layared velocity models that have structural discontinuities while keeping the velocity kinematically accurate as much as possible (section III.2.5); this is still a problem under investigation. The main focus of this manuscript was to study the second problem of proposing appropriate data misfit functions. The connection of the work presented here and the global workflow in Figure 101 can be seen by the following experiment: consider a sequence of data misfit minimizing structural models generated by the loop in Figure 101; if the data misfit satisfies equation 121, then that sequence converges to the "true structural model", at least in theory. As we have seen in this manuscript, the problem of designing such data misfit functions is also challenging and we still consider it unsolved. The third problem, which we have not yet begun to investigate, is also non-trivial: how can we update structural models from a scalar field error map?

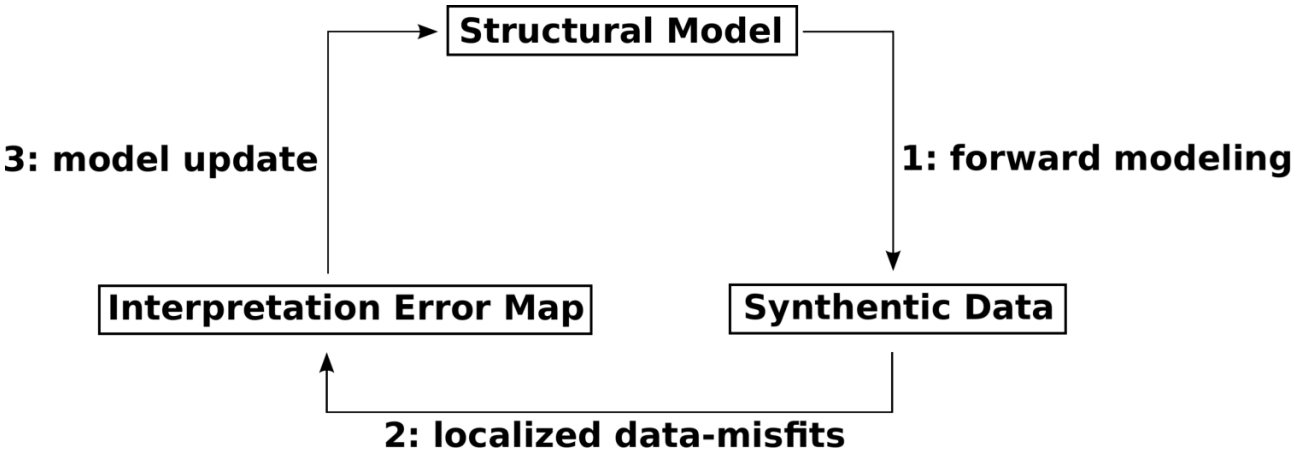

Figure 101: Long term global workflow for reducing structural interpretation uncertainties using seismic data. **1**: generating synthetic seismic data from structural models. **2**: computing data misfit values using synthetic and observed seismic data. **3**: updating the structural model in a manner that reduces misfit values.

Part IV

Seismic Imaging for GeoModel Analysis - SIGMA

Chapter IV.1

Introducing the SIGMA library

IV.1.1 Introduction & Motivation

The SIGMA (Seismic Imaging for GeoModel Analysis) library is a collection of programs that my students (students I supervised for a master's project) and I developed over the course of my thesis. The goal of the package is to develop a suite of programs, i.e. a numerical laboratory, to make my research easier. All the results presented in this thesis are reproducible and were obtained using SIGMA.

I started writing the library when I was learning about seismic imaging and back then I was often asked, "Why are you writing seismic imaging code when there are so many open-source seismic imaging packages? Why reinvent the wheel?". Here are some of the main reasons:

- Programming an algorithm is a reliable way for me to evaluate how well I have understood the algorithm. For example, I don't consider myself to know how seismic imaging works until I have written a seismic imaging code that works.
- Sometimes it takes me about the same amount of time to write a new code as to read and understand an existing code.
- Sometimes the existing code was written to solve a problem similar to mine but not exactly. So I find myself making ad hoc modifications to adapt the code to my problem. The ad hoc code can be quite substantial and usually results in a badly organized code.
- I like to code!

Some chapters in this thesis come from my "reinventing the wheel". For example, the generalized Kirchhoff imaging operator proposed in chapter I.3 was motivated by the need to avoid code duplication when implementing different Kirchhoff extended operators (i.e. one code for shot imaging, one code for angle imaging, ...), and the need to avoid shooting rays from each point in the subsurface for Kirchhoff angle imaging, as is common practice. The regularization operators proposed in chapter II.1 was motivated by the need to avoid implementing boundary conditions as this led to a lot of "if statements" (remember that faults are also boundaries), resulting in an untidy code. Had I not "reinvented the wheel", chapters I.3, II.1, II.2, and II.3 would be missing from this manuscript.

IV.1.2 Hardware requirements

The current version of the library runs on Linux, so we'll need a Linux computer. The computer should preferably be equipped with a CUDA-enabled GPU, some programs in the tutorials

presented in the following sections run on GPU [1]. Finally, the computer should support OpenGL 4.5 or higher, needed by SIGMA's viewer.

IV.1.3 Getting started

There are a couple of ways to benefit from SIGMA's services. First, developers can include it directly in their own projects. However, for recurring practice-oriented tasks, it may be more convenient to compile a set of stand-alone programs to use as commands. We will follow the latter approach in this tutorial.

Let us start by opening a terminal and then launch the SIGMA viewer by invoking the command "S2GoViewer". We then open a second terminal and place it below the viewer. Our screen should look as shown in Figure 102.

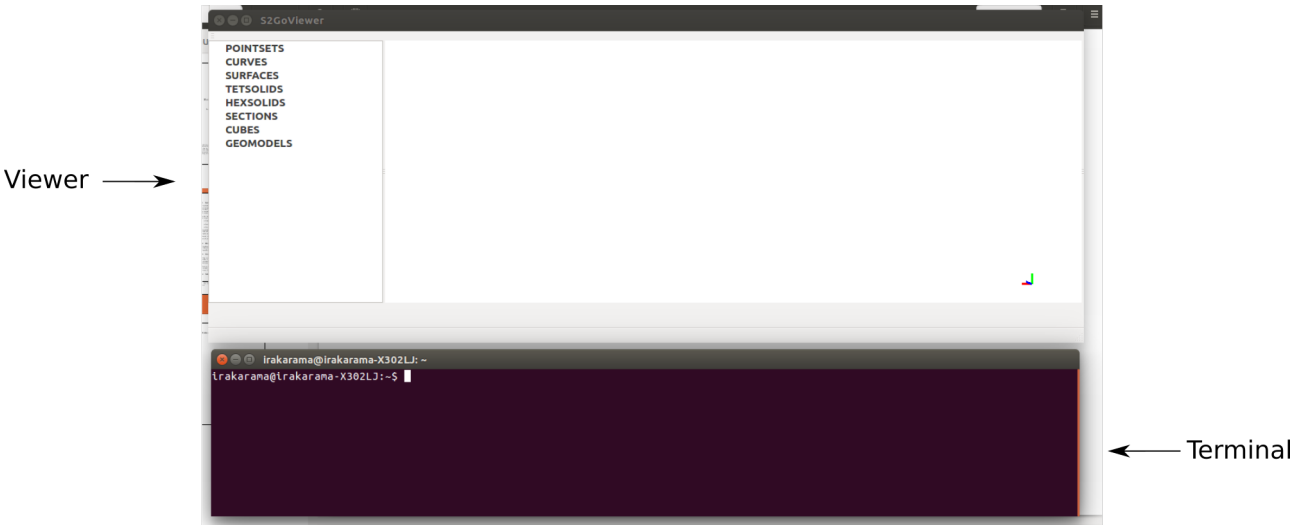

Figure 102: Getting started with SIGMA for the user. We launch the viewer, and then open a Linux terminal below it. The terminal is used not only to run stand-alone commands but also to interact with the viewer.

The terminal will be used not only to run stand-alone programs but also to interact with the viewer. That's right, the viewer can receive commands from the decoupled Linux terminal! This has proven to be very convenient, as it eliminates the need to transfer data into third-party viewers at the end of each computation to confirm that results are good.

To conclude this section, it should be pointed out that commands addressed to stand-alone programs are prefixed with "S2Go_",i.e. "S2Go_CommandName", while commands addressed to the viewer are prefixed with "S2GoV_", i.e. "S2GoV_CommandName". Commands addressed to the viewer are prefixed with "S2GoV_", i.e. "S2GoV_CommandName".

IV.1.4 The RSF file format

We will use a Regularly Sampled File (RSF) file-format very similar to the one used by Madagascar (Fomel et al., 2013). An RSF file is simply an nd-cube. For instance, a 2d RSF file is a section (2d regular grid), and a 3d RSF file is a cube (3d regular grid, a voxet). I will present the RSF file format as used by SIGMA. I distinguish three parts of an RSF file: (1)

[1] CPU code is available for some commands as well but is considerably slower.

the geometry file, (2) the header file, and (3) the data file.

1) The geometry file has a ".hd" extension. "hd" is short for header. The geometry file describes the axes of the grid. An axis is described by the following parameters:

- oi : origin of the i-th axis.
- di : sampling step of the i-th axis.
- ni : number of samples on the i-th axis.
- labeli : label (name) of the i-th axis, optional.
- uniti : units of the i-th axis, optional.

Where "i" is an integer starting from 1. An example of a simple geometry file called "geom_example.hd" is shown in Figure 103.a. This geometry file describes a cube of size 100x100x100, with d1=dx=10, d2=dy=5, d3=dz=5, and an origin at the point (0,0,0); Figure 103.b shows one command that can be used to view this cube.

a)

geom_example.hd

```
n1=100
n2=100
n3=100
o1=0
o2=0
o3=0
d1=10
d2=5
d3=5
```

b)

S2GoV_LoadFile geom_example.hd

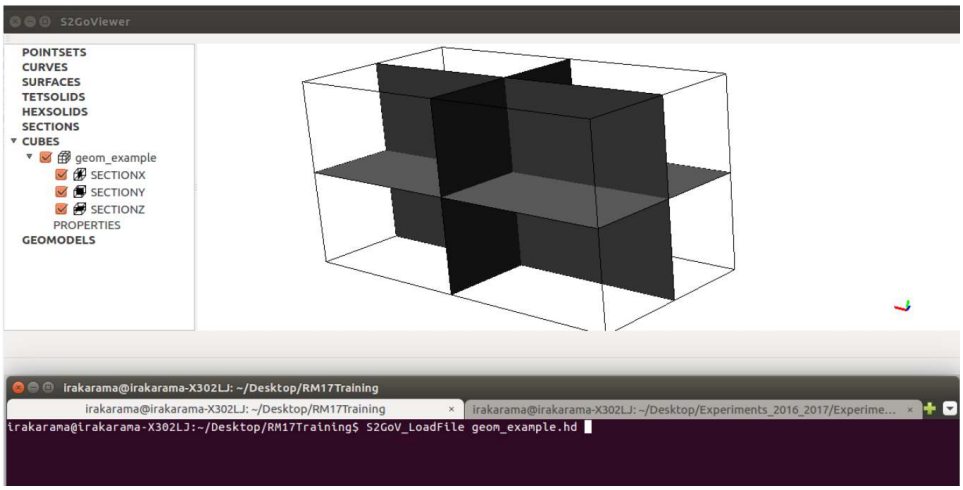

Figure 103: **a**) Example of a simple geometry file describing a cube. **b**) Loading the geometry file using the command shown above.

2) The header file also has a ".hd" extension. "hd" is short for header. The header has all the fields of the geometry file and additional ones including:

- in : path to the actual (binary) data file.
- esize : size, in bytes, of each element in the (binary) data file.

An example of a header file named "vel_section.hd", and how to view the corresponding data, are shown in Figure 104.a and Figure 104.b respectively. The velocity model in Figure 104.b is derived from the sandbox of model of Colletta et al. (1991).

3) The data file has a ".dt" extension. "dt" is short for data. The data file is in binary format and contains property values of every grid point. It is found from the "in" field of the associated header file (Figure 104).

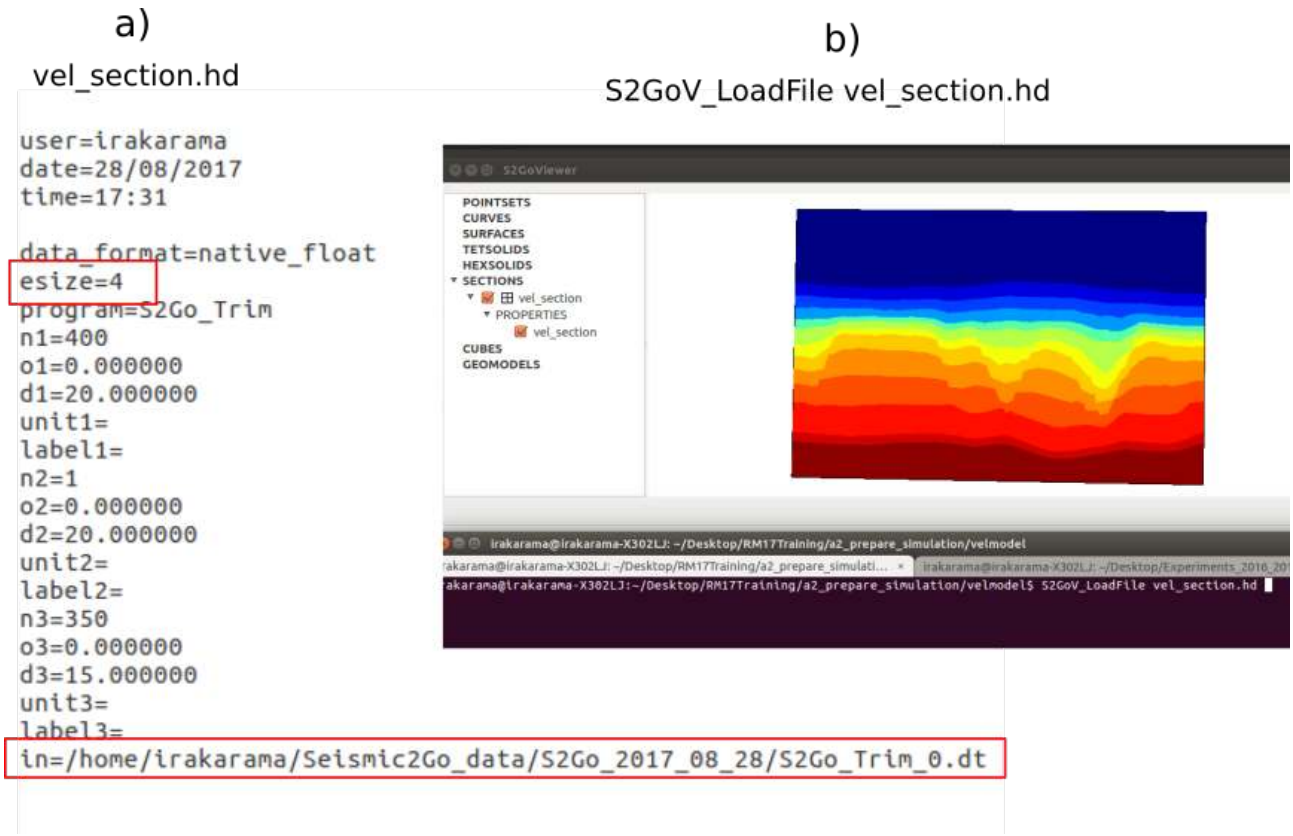

Figure 104: **a**) Example of a complete header file. The red boxes highlight the only required fields, in addition to those describing the geometry. **b**) Loading the RSF file. The velocity property values shown in the model are read from the data file pointed to by the "in" field of the header file.

Chapter IV.2

Tutorial I: Seismic Modeling & Imaging

IV.2.1 In this tutorial

In this tutorial, we will test SIGMA's 2D seismic modeling and imaging programs. In particular, we are going to perform the following activities:

- Modeling of acoustic seismic data by finite difference wave simulation.
- Building extended images, i.e. common image gathers (CIGs), from seismic data.
- Stacking CIGs to produce depth migrated images.

IV.2.2 Seismic modeling

IV.2.2.1 Preparing for wave simulation

Source and receiver coordinates

First, we need to create an array of source and receiver coordinates before running the simulation. We can create an array of points by defining its geometry file and then invoking the command "S2Go_GeomToPointSet", as shown in Figure 105. It is good practice to always view the resulting points to confirm that sources and receivers are positioned where we want them before running simulations.

a)

geom_1src.hd

```
n1=1
n2=1
n3=1
d1=400
d2=20
d3=15
o1=4000
o2=0
o3=0
```

b)

S2Go_GeomToPointSet geom=geom_1src.hd > arr_1src.hd; S2GoV_LoadFile arr_1src.hd

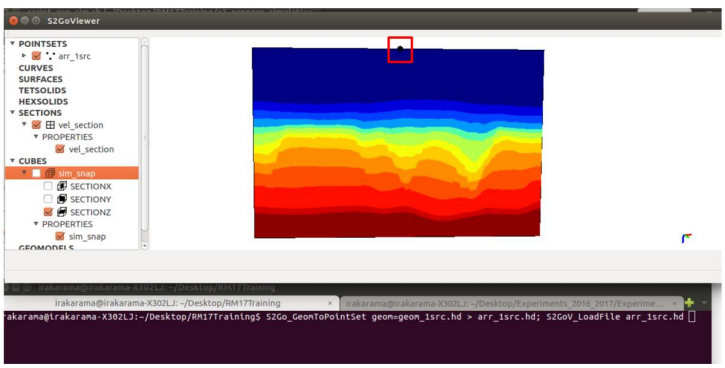

Figure 105: **a**) Example of a geometry file for one point. **b**) Viewing the point (highlighted by the red box) described by the given geometry file.

Source wavelet

Now, we need to create a source wavelet. This can be achieved by invoking the command:
"S2Go_GetRickerWavelet fmax=15 dt=0.004 tsim=5 tdelay=0.4 > wlt.hd",
which creates a Ricker wavelet signal that is 5 seconds long, sampled at 0.004 seconds, centered at 0.4 seconds, with a dominant frequency of 15 Hertz, and saves it in a file called wlt.hd.

Running the simulation

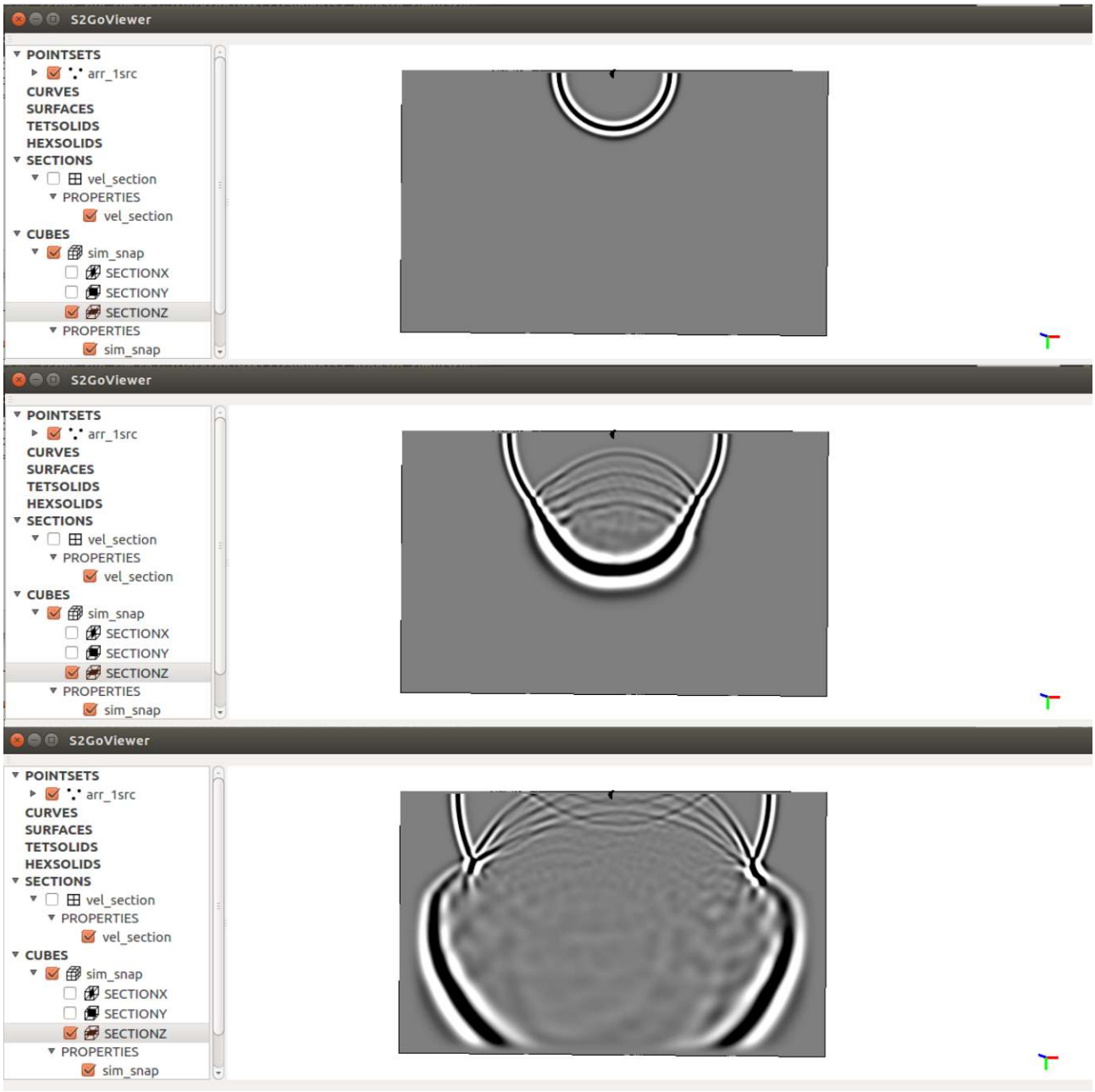

Figure 106: Three snapshots of the wavefield simulated from a shot positioned at the point shown in Figure 105. (To change the colormap: S2GoV_SetColormap cubes.cubename.propertyname colormap=gray)

The acoustic wave simulation commands are implemented using staggered finite differences (Virieux, 1984), perfectly matched layers boundary conditions (Collino and Tsogka, 2001), and run on GPU (Micikevicius, 2009); they implement 8^{th} order difference operators in space and

2nd order in time. One of such commands is "S2Go_2dAWECD_gpu" (short for 2d Acoustic Wave Equation with Constant Density, runs on GPU); a CPU version of the command is also available: "S2Go_2dAWECD_cpu". The result of this command using the velocity model and source in Figure 105.b, and the source wavelet created in the last section, are shown in Figure 106; the experiment can be reproduced using the bash script in Figure 107.

```
# (relative) path to vel_section header (the "in" path should have been set by now)
vel_header=../VelModel/vel_section.hd


# Delete earlier files (if any)
S2Go_DeleteRSFFile  arr_1src.hd wlt.hd

#create 1 source geom file
echo "n1=1" > geom_1src.hd
echo "n2=1" >> geom_1src.hd
echo "n3=1" >> geom_1src.hd
echo "o1=4000" >> geom_1src.hd
echo "o2=0" >> geom_1src.hd
echo "o3=0" >> geom_1src.hd
echo "d1=1" >> geom_1src.hd
echo "d2=1" >> geom_1src.hd
echo "d3=1" >> geom_1src.hd

#create array of src coordinates and load it to the viewer
S2Go_GeomToPointSet geom=geom_1src.hd > arr_1src.hd

#create source wavelet
S2Go_GetRickerWavelet fmax=15 dt=0.004 tsim=5 tdelay=0.4 > wlt.hd

#run simulation
S2Go_2dAWECD_gpu vel=${vel_header} src=arr_1src.hd rec=arr_1src.hd wlt=wlt.hd ofile=sim_data snapfile=sim_snap
```

Figure 107: Bash script used to produce the example in Figure 106.

IV.2.2.2 A seismic survey

a)

geom_src.hd

```
n1=77
n2=1
n3=1
d1=100
d2=20
d3=15
o1=200
o2=0
o3=0
```

b)

S2Go_GeomToPointSet geom=geom_src.hd > arr_src.hd; S2GoV_LoadFile arr_src.hd

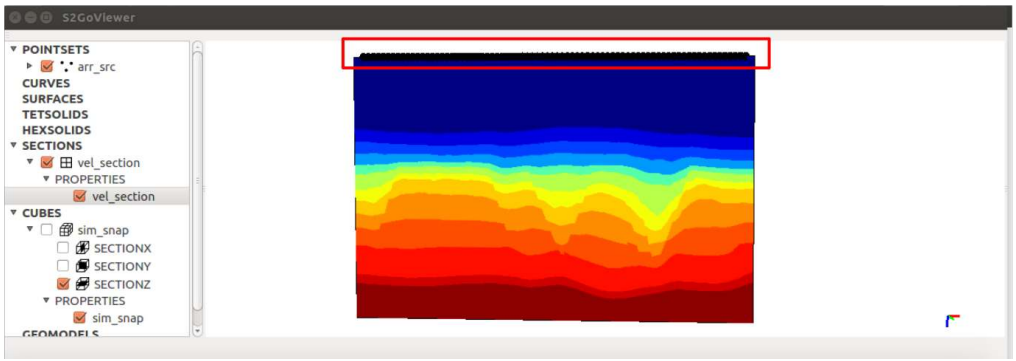

Figure 108: **a**) Geometry file describing an array of source coordinates. **b**) Viewing the source coordinates (highlighted by the red box) described the given geometry file.

a)

geom_rec.hd

```
n1=400
n2=1
n3=1
d1=20
d2=20
d3=15
o1=0
o2=0
o3=0
```

b)

S2Go_GeomToPointSet geom=geom_rec.hd > arr_rec.hd; S2GoV_LoadFile arr_rec.hd

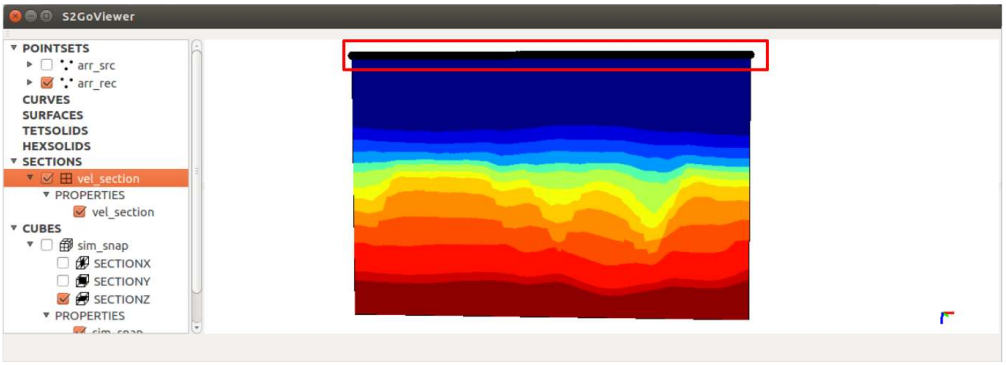

Figure 109: **a**) Geometry file describing an array of receiver coordinates. **b**) Viewing the receiver coordinates (highlighted by the red box) described the given geometry file.

Now that we have all the tools needed to run simulations, let us simulate a seismic survey. We start by creating an array of sources and receivers as shown in Figure 108 and in Figure 109 respectively. Then, using the source wavelet created earlier, we invoke the "S2Go_2dAWECD_gpu" command discussed earlier. The shot gathers computed using the source-receiver configuration and command from this paragraph are shown in Figure 110.

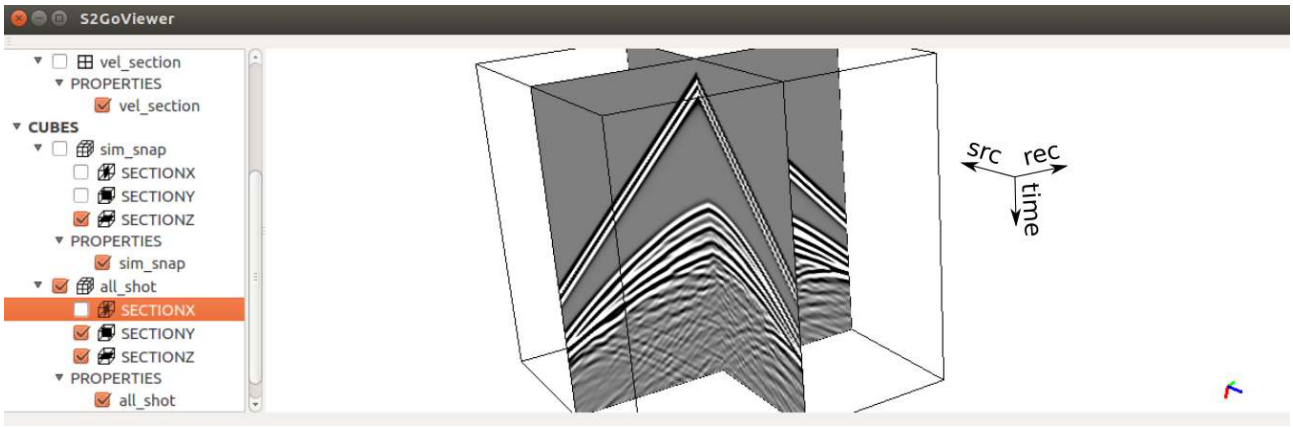

Figure 110: Shot gathers recorded at receivers. Each section in the rec-time plane is a shot gather; that is, a collection of data recorded from the same shot.

IV.2.3 Seismic imaging

The first thing we need to do is to remove the direct arrivals in a our data. We do this by making a copy of our initial data and then using the command shown in Figure 111.

```
cp all_shot.trc all_shot_muted.trc; S2Go_Mute1stArrive v0=1500 t0=1.2 traces=all_shot_muted.trc
S2GoV_LoadFile all_shot_muted.trc
```

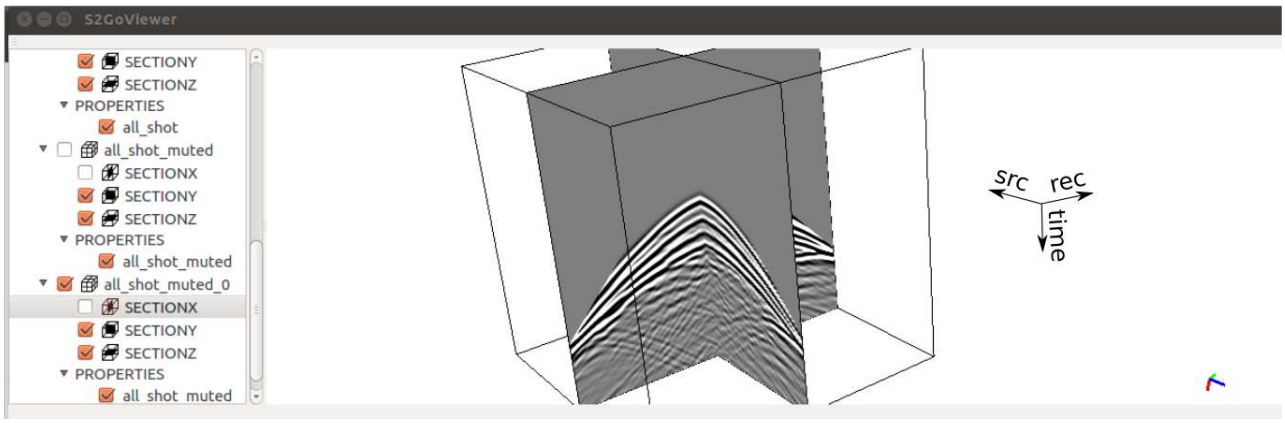

Figure 111: The same shot gather cube of Figure 110 after muting first arrivals (direct waves).

Then, we smooth our velocity section to simulate an imaging velocity model. This is done by the command shown in Figure 112. In practice, the imaging velocity model is estimated from data by migration velocity analysis (Woodward et al., 2008; Beraud et al., 2018), but this is beyond the scope of this tutorial.

```
S2Go_SmoothRSF file=vel_section.hd wln=20 > vel_smooth.hd ; S2GoV_LoadFile vel_smooth.hd
```

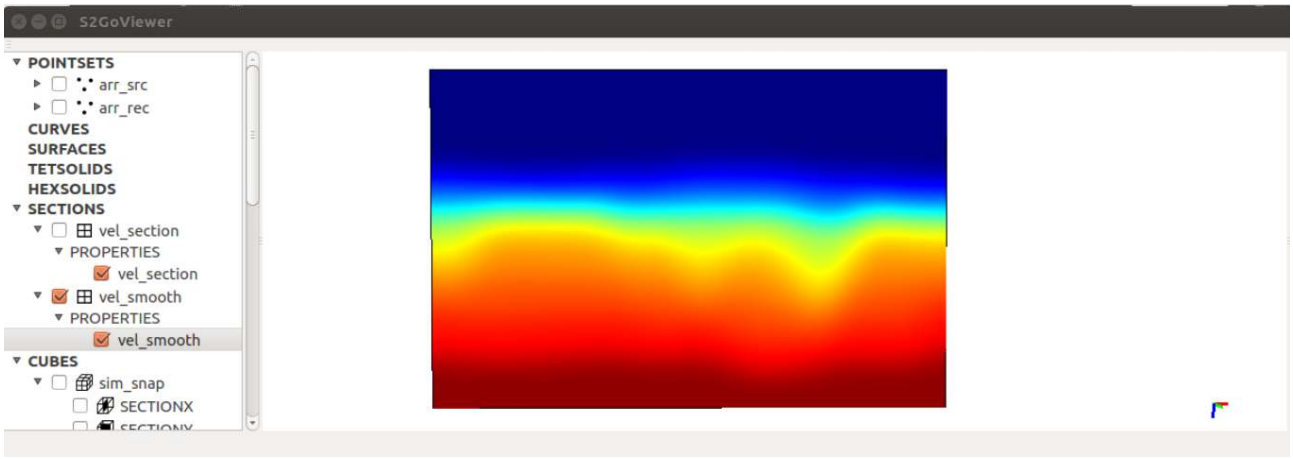

Figure 112: Smoothing the velocity section of Figure 104 to simulate an imaging velocity model.

For simplicity, in this tutorial we perform first arrival Kirchhoff imaging. The Kirchhoff imaging integral has the general form (Etgen et al., 2009)

$$\begin{aligned} I(\mathbf{x}) &= \int_{s,r,t} W(s,r,\mathbf{x})\delta\big[t-\tau(s,r,\mathbf{x})\big]\frac{\partial}{\partial t}D(s,r,t) \\ &= \int_{s,r} W(s,r,\mathbf{x})\frac{\partial}{\partial t}D\big[s,r,\tau(s,r,\mathbf{x})\big]. \end{aligned} \tag{162}$$

$I(\mathbf{x})$ is the output image; $\mathbf{x}=(x,z)$ is a point in the image domain; $\mathbf{x}_s=s$ and $\mathbf{x}_r=r$ are source and receiver coordinates respectively; $W(s,r,\mathbf{x})$ is a weight function; $\tau(s,r,\mathbf{x})$ is the traveltime from a ray from s, scattered off $\mathbf{x}$, then recorded at r; and $D(s,r,t)$ is the input data set. The traveltime maps needed to evaluate $\tau(s,r,\mathbf{x})$ are computed using the smooth velocity model by the command "S2Go_2dTTimes" as shown in Figure 113; this command solves the eikonal equation using the fast marching method (Sethian and Popovici, 1999). In particular, $\tau(s,r,\mathbf{x})=T(\mathbf{x},s)+T(\mathbf{x},r)$, where T is the traveltime cube in Figure 113.

a)

geom_ttimes.hd

```
n1=400
n2=1
n3=1
d1=20
d2=20
d3=15
o1=0
o2=0
o3=0
```

b)

```
S2Go_GeomToPointSet geom=geom_ttimes.hd > arr_ttimes.hd
S2Go_2dTTimes vel=vel_smooth.hd src=arr_ttimes.hd > ttimes.hd
S2GoV_LoadFile ttimes.hd
```

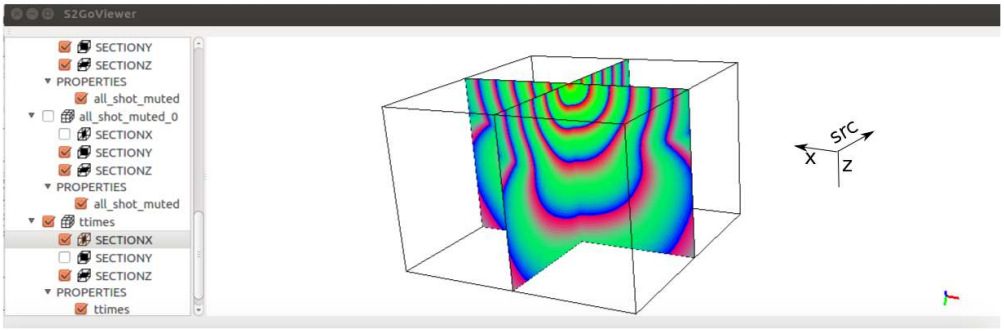

Figure 113: **a**) Geometry file describing source coordinates for computing traveltime maps; it is the same as that in Figure 109. **b**) Computing traveltime maps at sources described by the given geometry file. Each section in the x-z plane is a traveltime map from one shot. The cube has been displayed with a *stratigraphy colormap* for a better visualization of wavefronts (S2GoV_SetColormap cubes.cubename.propertyname colormap=stratigraphy).

IV.2.3.1 Common image gathers

In practice, it is useful to construct *extended* images $I(\mathbf{x}, \lambda)$ obtained by *extending* the image domain $\mathbf{x}$ to $(\mathbf{x}, \lambda)$, that is $I(\mathbf{x}) \rightarrow I(\mathbf{x}, \lambda)$ (Symes, 2008). Here, I present extended imaging as described in chapter I.3. Let us rewrite Eq.162 as

$$I(\mathbf{x}) = \int_{s,r} F(s, r, \mathbf{x}); \tag{163}$$

then we can surely write

$$I(\mathbf{x}, \lambda) = \int_{s,r} \delta\big[\lambda - \zeta(s, r, \mathbf{x})\big] F(s, r, \mathbf{x}), \tag{164}$$

for some function $\zeta(s, r, \mathbf{x})$. The physical meaning of the parameter λ will depend on our definition of $\zeta(s, r, \mathbf{x})$. Extended images are also known as common image gathers. The advantage of image extension (e.g. Eq.164) is that it outputs a collection of structural images as a function of λ. Because a structural image is expected to be invariant if imaged using an accurate velocity, any depth variation of a reflector along the λ-axis suggests an error in the imaging velocity model; this is the basis of some migration velocity analysis methods.

Shot common image gather

The shot common image gather is obtained from Eq.164 by defining

$$\zeta(s, r, \mathbf{x}) := s. \tag{165}$$

The shot common image gather of our data set (Figure 111) is shown in Figure 114 and was obtained by the command "S2Go_2dFAKDMSCIG_gpu" (short for 2d First Arrival Kirchhoff Depth Migration, Shot CIG, runs on GPU).

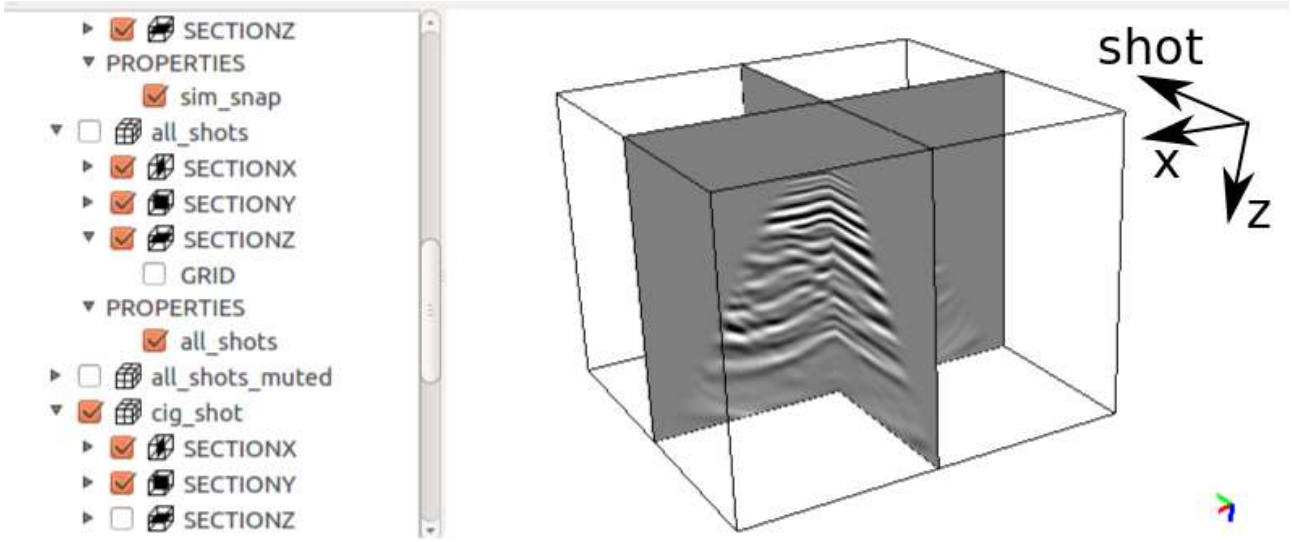

Figure 114: Shot common image gather. Each section in the x-z plane is an image obtained by migrating data from the same shot. The parameters oe, de, ne are the origin, sampling rate, and size of the "extension" axis; i.e. the shot axis in this figure.

Offset common image gathers

The offset common image gather is obtained from Eq.164 by defining

$$\zeta(s, r, \mathbf{x}) := |s - r|. \tag{166}$$

The offset common image gather of our data set (Figure 111) is shown in Figure 115 and was obtained by the command "S2Go_2dFAKDMOCIG_gpu" (short for 2d First Arrival Kirchhoff Depth Migration, Offset CIG, runs on GPU).

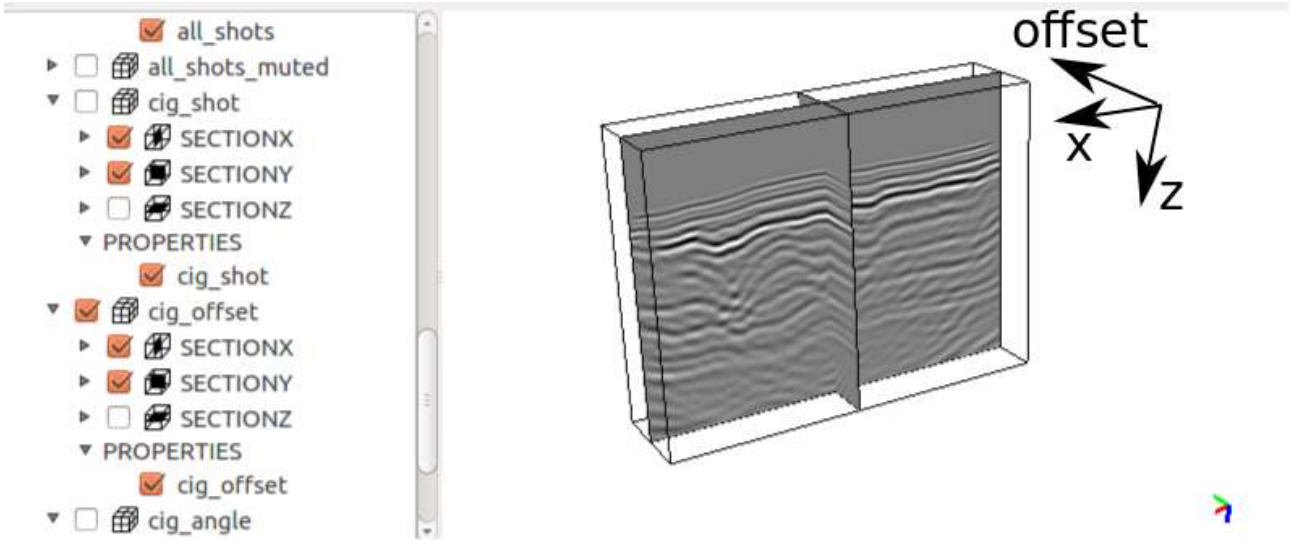

Figure 115: Offset common image gather. Each section in the x-z plane is an image obtained by migrating data that have the same offset. The parameters oe, de, ne are the origin, sampling rate, and size of the "extension" axis; i.e. the offset axis in this figure.

Angle common image gathers

The angle common image gather is obtained from Eq.164 by defining

$$\zeta(s,r,\mathbf{x}) := \text{acos}\left(\frac{\nabla T(\mathbf{x},s).\nabla T(\mathbf{x},r)}{\nabla T(\mathbf{x},s).\nabla T(\mathbf{x},s)}\right), \; T \text{ is the traveltime cube in Figure 113.} \qquad (167)$$

The angle common image gather of our data set (Figure 111) is shown in Figure 116 and was obtained by the command "S2Go_2dFAKDMACIG_gpu" (short for 2d First Arrival Kirchhoff Depth Migration, Angle CIG, runs on GPU).

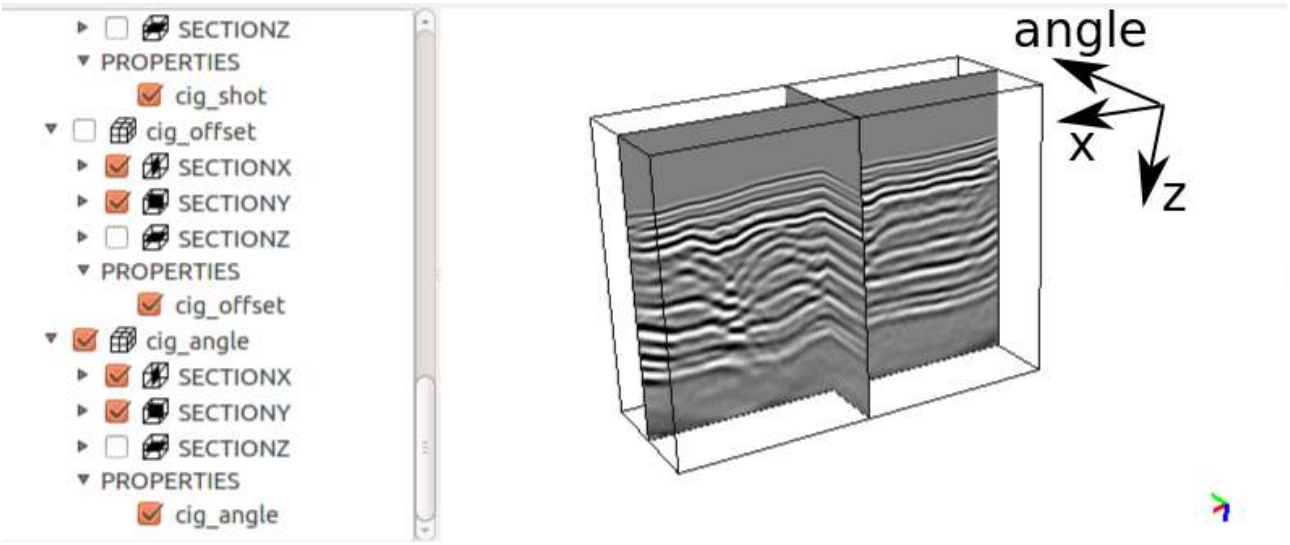

Figure 116: Angle common image gather. Each section in the x-z plane is an image obtained by migrating data that have the same reflection angle. The parameters oe , de, ne are the origin, sampling rate, and size of the "extension" axis; i.e. the angle axis in this figure.

IV.2.3.2 Stacking common image gathers

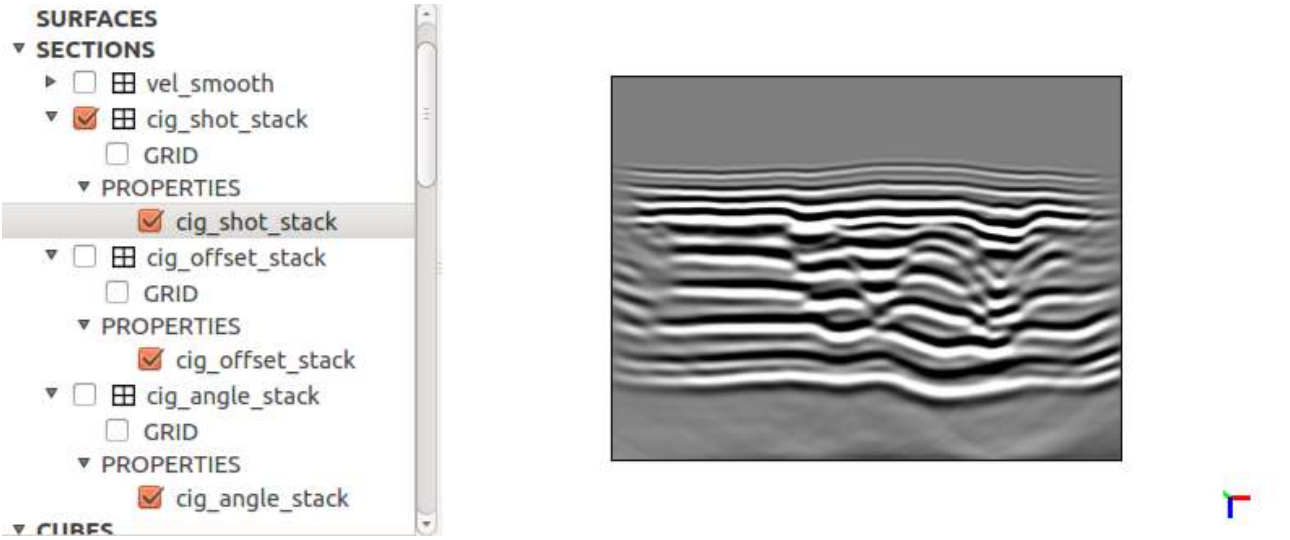

Figure 117: Common shot migration. This is the result of stacking (summing) the images (x-z planes) in Figure 114 along the shot axis.

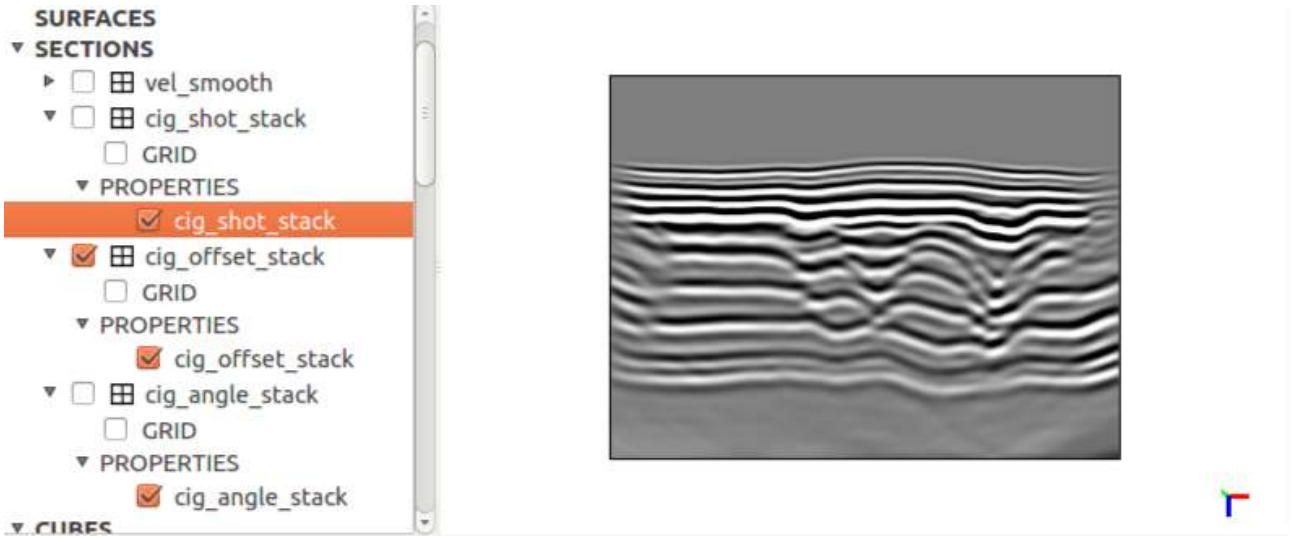

Figure 118: Common offset migration. This is the result of stacking (summing) the images (x-z planes) in Figure 115 along the offset axis.

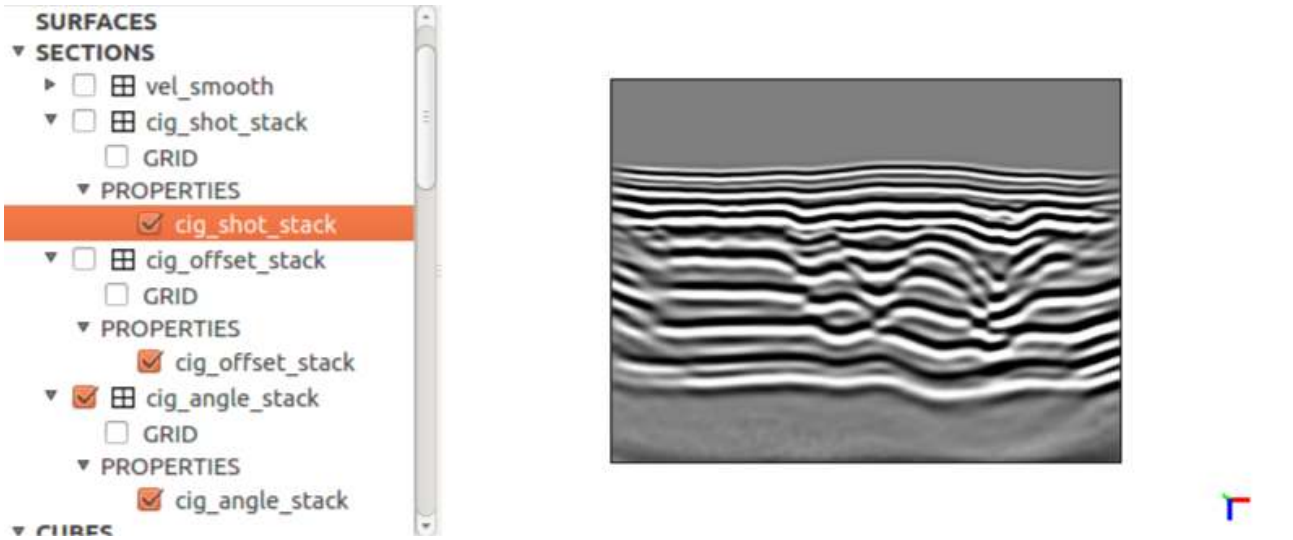

Figure 119: Common angle migration. This is the result of stacking (summing) the images (x-z planes) in Figure 116 along the angle axis.

A common-λ-migration depth image is obtained by stacking (summing) the extended image $I(\mathbf{x}, \lambda)$ along the extension axis λ, that is

$$I_\lambda(\mathbf{x}) = \sum_\lambda I(\mathbf{x}, \lambda). \tag{168}$$

The common-shot-migration image $I_s(\mathbf{x})$, common-offset-migration image $I_o(\mathbf{x})$, and common-angle-migration image $I_a(\mathbf{x})$ of our data set (Figure 111) are shown respectively in Figure 117, Figure 118 and Figure 119. The results in Figures 111-119 can be reproduced using the bash script in Figure 120. The velocity model used is shipped with the tutorial.

```
# (relative) path to vel_section header (the "in" path should have been set by now)
vel_header=../VelModel/vel_section.hd

# (relative) path to seismic data computed in the last activity
seis_data=../Activity_3_seismic_survey/all_shots.trc

# Delete earlier files (if any)
S2Go_DeleteRSFFile  cig_shot* cig_offset* cig_angle* vel_smooth.hd ttimes.hd

# Copy seismic data and mute them
cp ${seis_data} all_shots_muted.trc
S2Go_Mute1stArrive traces=all_shots_muted.trc t0=1.2 v0=1500

# Smooth velocity model to simulate a migration velocity model, and visualize
S2Go_SmoothRSF file=${vel_header} wln=20 > vel_smooth.hd

# Compute travel time maps
#(In this particuler case its OK to use the receiver coordinates)
S2Go_2dTTimes vel=vel_smooth.hd srcs=../Activity_3_seismic_survey/arr_rec.hd > ttimes.hd

#==== SHOT IMAGING ====
# Build shot CIG
S2Go_2dFAKDMSCIG_gpu ttimes=ttimes.hd geom=vel_smooth.hd obsdata=all_shots_muted.trc apert=.6 tdelay=.4 oe=200 de=100 ne=77 > cig_shot_tmp.hd
# Preprocess CIG for better visual
S2Go_RSFGainQuadratic ifile=cig_shot_tmp.hd ax=2 > cig_shot_gain.hd
S2Go_SmoothRSF file=cig_shot_gain.hd wln=3 n=4 > cig_shot.hd
# Stack CIG to obtain a shot image
S2Go_Stack file=cig_shot.hd ax=1 > cig_shot_stack.hd

#==== OFFSET IMAGING ====
# Build offset CIG
S2Go_2dFAKDMOCIG_gpu ttimes=ttimes.hd geom=vel_smooth.hd obsdata=all_shots_muted.trc apert=.6 tdelay=.4 oe=0 de=20 ne=100 > cig_offset_tmp.hd
# Preprocess CIG for better visual
S2Go_RSFGainQuadratic ifile=cig_offset_tmp.hd ax=2 > cig_offset_gain.hd
S2Go_SmoothRSF file=cig_offset_gain.hd wln=3 n=4 > cig_offset.hd
# Stack CIG to obtain an offset image
S2Go_Stack file=cig_offset.hd ax=1 > cig_offset_stack.hd

#==== ANGLE IMAGING ====
# Build angle CIG
 S2Go_2dFAKDMACIG_gpu ttimes=ttimes.hd geom=vel_smooth.hd obsdata=all_shots_muted.trc apert=.6 tdelay=.4 oe=0 de=1 ne=60 > cig_angle_tmp.hd
# Preprocess CIG for better visual
echo "d2=50" >> cig_angle_tmp.hd
S2Go_RSFGainQuadratic ifile=cig_angle_tmp.hd ax=2 > cig_angle_gain.hd
S2Go_SmoothRSF file=cig_angle_gain.hd wln=3 n=4 > cig_angle.hd
# Stack CIG to obtain an offset image
S2Go_Stack file=cig_angle.hd ax=1 > cig_angle_stack.hd
```

Figure 120: Bash script used to produce the example in Figures 111-119.

IV.2.4 Acknowledgements

This work was done within the framework of the RING project at Université de Lorraine. I would like to thank the industrial and academic sponsors of the RING-Gocad Consortium managed by ASGA for their support.

Chapter IV.3

Tutorial II: Structural Interpolation

IV.3.1 In this tutorial

In this tutorial, we will test SIGMA's finite difference structural interpolation method presented in chapter II.1 and in Irakarama et al. (2018).

IV.3.2 Structural interpolation

IV.3.2.1 Interpolation in 2D

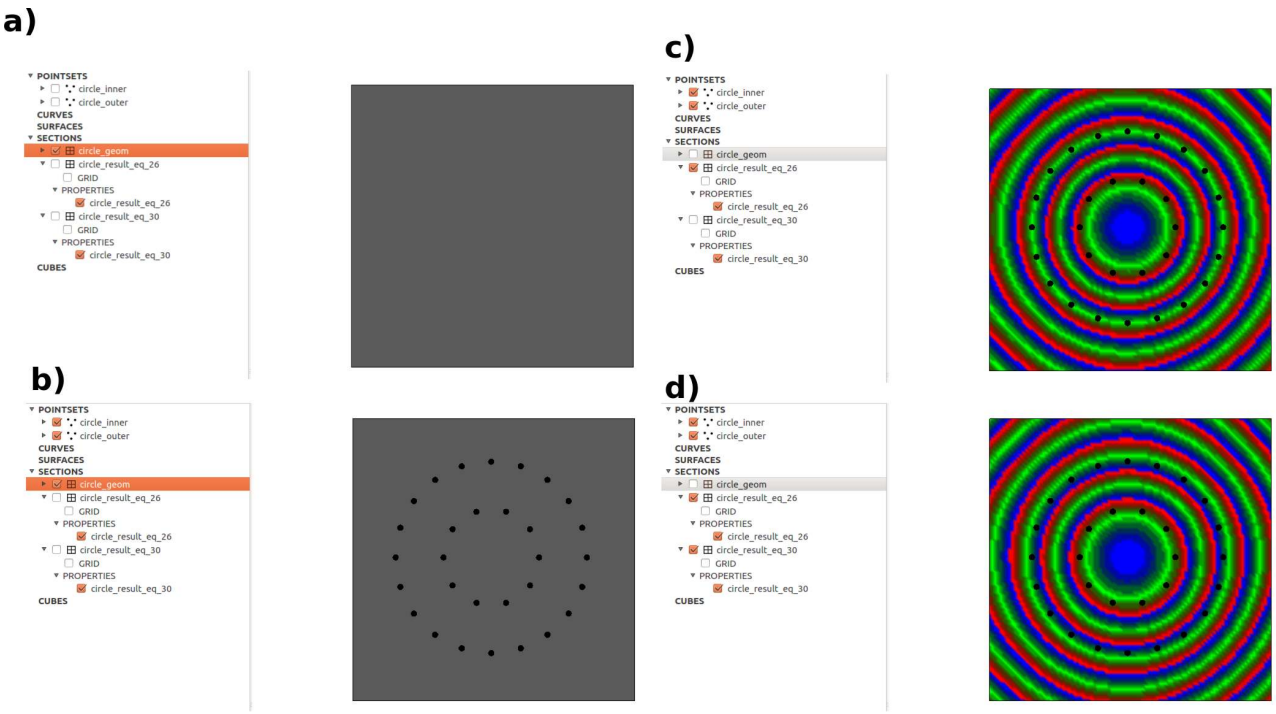

Figure 121: Interpolation of concentric circles. This experiment can be reproduced using the bash script in Figure 122. **a**) Input geometry file (region of interest). **b**) Input data consisting of an inner and an outer circle. **c**) Results obtained using equation 26 in chapter II.1 [equation 5 in Irakarama et al. (2018)]. **d**) Results obtained using equation 30 in chapter II.1 [equation 6 in Irakarama et al. (2018)].

We start with an illustrative example of interpolating two concentric circles as shown in Figure 121. The experiment can be reproduced using the bash script in Figure 122. RSF files

(.hd extension) are loaded in the viewer using the "S2GoV_LoadFile command". The default colormap is jet (rainbow); the periodic colormap in Figure 121 is obtained using the command "S2GoV_SetColormap sections.*.* colormap=structure". Alternatively, the default colormap can be changed by invoking the command "S2GoV_SetDefaultColormap colormap=structure". The commands in lines 3 and 7 in Figure 122 create an array of points along a circle; the documentation of the command , shown in Figure 123a, is obtained by invoking the command with no argument. The commands in lines 21 and 22 in Figure 122 create "2D value files" from an array of RSF points; a "2D value file" is a text file in which each line has three values: x y val. The commands in lines 27 and 31 are the actual interpolation commands; typical documentation for such commands is shown in Figure 123b. The command name "S2Go_2dFDSIS4A4System", which interpolates using the regularization operator in equation 26 in chapter II.1 [equation 5 in Irakarama et al. (2018)], stands for:

- FDSI: Finite Difference Structural Interpolation
- S4A4: The system (matrix) is constructed using a Smoothing operator with 4 equations (S4). and an Assigment-constraint (value-constraint) matrix with 4 nonzero entries in each row (A4).

The command "S2Go_2dFDSIS3A4System" interpolates using the regularization operator in equation 30 in chapter II.1 [equation 6 in Irakarama et al. (2018)].

```
1 #Create inner circle points
2 S2Go_DeleteRSFFile circle_inner.hd
3 S2Go_CirclePoints ox=30 oz=30 r=10 n=10 > circle_inner.hd
4
5 #Create outer circle points
6 S2Go_DeleteRSFFile circle_outer.hd
7 S2Go_CirclePoints ox=30 oz=30 r=20 n=20 > circle_outer.hd
8
9 #Create geometry (bounding box) info
10 echo "o1=0" > circle_geom.hd
11 echo "o2=0" >> circle_geom.hd
12 echo "o3=0" >> circle_geom.hd
13 echo "d1=1" >> circle_geom.hd
14 echo "d2=1" >> circle_geom.hd
15 echo "d3=1" >> circle_geom.hd
16 echo "n1=60" >> circle_geom.hd
17 echo "n2=60" >> circle_geom.hd
18 echo "n3=1" >> circle_geom.hd
19
20 #Create "2D value files" for FD interpolation. The format for 2D value files is x y val.
21 S2Go_2dPointSetToValsFile circle_inner.hd val=0 ofile=circle_inner_vals
22 S2Go_2dPointSetToValsFile circle_outer.hd val=1 ofile=circle_outer_vals
23 cat circle_inner_vals* circle_outer_vals* > circle_vals.txt
24
25 #Interpolate using the smoothing operator in equation 26 of the RING meeting paper
26 S2Go_DeleteRSFFile circle_result_eq_26.hd
27 S2Go_2dFDSIS4A4System_gpu geom=circle_geom.hd val=circle_vals.txt onpts=10000 > circle_result_eq_26.hd
28
29 #Interpolate using the smoothing operator in equation 30 of the RING meeting paper
30 S2Go_DeleteRSFFile circle_result_eq_30.hd
31 S2Go_2dFDSIS3A4System_gpu geom=circle_geom.hd val=circle_vals.txt onpts=10000 > circle_result_eq_30.hd
```

Figure 122: Bash script used to produce the example in Figure 121.

a)

```
Synopsis: S2Go_CirclePoints param1=val1 ... paramN=valN > ofile.hd

Returns a array points on a circle.

        [in ] ox      : origin of circle, x coordinate
        [in ] oz      : origin of circle, z coordinate
        [in ] r       : radius of the circle
        [in ] n       : number of points
        [out] ofile   : output filename
```

b)

```
Synopsis: S2Go_2dFDSIS4A4System_gpu  param1=val1 param2=val2 ... paramN=valN > ofile

Interpolation values using FD structural interpolation.

        [in ] geom    : geometry file of the region of interest
        [in ] val     : value file (txt), each line has the form x z val
        [in ] faults  : curve file (txt), each line has the form x z id, optional
        [in ] sweight : smoothing weight, optional; default is 1.0
        [in ] vweight : value (assignement) weight, optional; default is 1.0
        [in ] onpts   : output nbr of points desired, uniform resampling, optional, overrides nx,nz
        [out] ofile   : out put filename
```

Figure 123: **a**)Documentation of the command in lines 3 and 7 in Figure 122. **b**)Documentation of the command in lines 17 and 31 in Figure 122.

The second example is the interpolation of geological data from the Ribaute model (Caumon et al., 2009). Figure 124a shows the input data. Figure 124b shows results obtained using equation 26 in chapter II.1 [equation 5 in Irakarama et al. (2018)]. Figure 124c shows results obtained using equation 30 in chapter II.1 [equation 6 in Irakarama et al. (2018)]. The results can be reproduced using the bash script in Figure 125. Lines 1 to 11 of the bash script set the correct path of the RSF binary data files in the corresponding RSF headers; the input data are shipped with the tutorial. The only new command in Figure 125 is the command in line 21, which creates "2D curves file" from a collection of RSF points files. A "2D curve file" is a text file in which each line has 3 entries: x y curve_id.

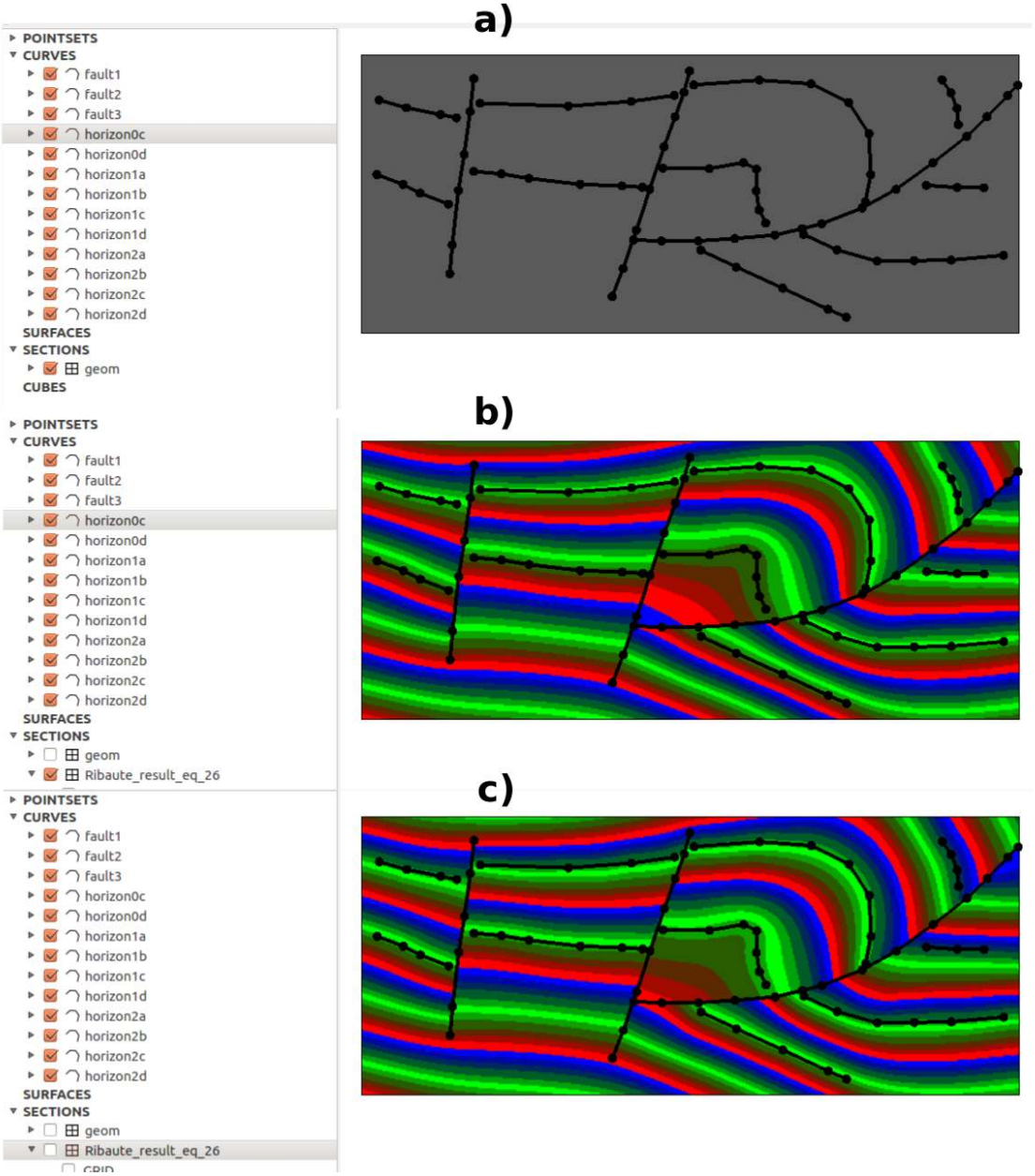

Figure 124: Interpolation of the Ribaute model (Caumon et al., 2009). This experiment can be reproduced using the bash script in Figure 125. **a**) Input geometry file (region of interest). **a**) Input data consisting of three horizons and three faults. **b**) Results obtained using equation 26 in chapter II.1 [equation 5 in Irakarama et al. (2018)]. **c**) Results obtained using equation 30 in chapter II.1 [equation 6 in Irakarama et al. (2018)].

```
#Set approriate data filepath in header files
rm -rf data
cp -rf data_orig data
for hfile in $(ls data/*.hd) #loop through header files
do
    tmp=$(echo $hfile|cut -d '/' -f 2) #remove directory from header filenames
    base=$(echo $tmp|cut -d '.' -f 1) #remove extension from header filenames
    dfile=${base}.dt #data filename
    dfile_full=$(pwd)/data/${dfile} #data filename fullpath
    echo "in=${dfile_full}" >> ${hfile} #set data filename in header file
done

#Create "2D value files" for FD interpolation. The format for 2D value files is x y val.
S2Go_2dPointSetToValsFile data/horizon0c.hd data/horizon0d.hd val=0 ofile=horizon0
S2Go_2dPointSetToValsFile data/horizon1a.hd data/horizon1b.hd data/horizon1c.hd data/horizon1d.hd val=1 ofile=horizon1
S2Go_2dPointSetToValsFile data/horizon2a.hd data/horizon2b.hd data/horizon2c.hd data/horizon2d.hd val=2 ofile=horizon2
rm horizon_vals.txt
cat horizon* > horizon_vals.txt

#Create "2D curve files" for FD interpolation. The format for 2D curve files is x y curve_id.
S2Go_2dPointSetToCurvesFile data/fault1.hd data/fault2.hd data/fault3.hd ofile=faults
rm fault_curves.txt
cp faults* fault_curves.txt

#Interpolate using the smoothing operator in equation 26 of the RING meeting paper
S2Go_DeleteRSFFile Ribaute_result_eq_26.hd
S2Go_2dFDSIS4A4System_gpu geom=geom.hd val=horizon_vals.txt faults=fault_curves.txt onpts=90000 > Ribaute_result_eq_26.hd

#Interpolate using the smoothing operator in equation 30 of the RING meeting paper
S2Go_DeleteRSFFile Ribaute_result_eq_30.hd
S2Go_2dFDSIS3A4System_gpu geom=geom.hd val=horizon_vals.txt faults=fault_curves.txt onpts=90000 > Ribaute_result_eq_30.hd
```

Figure 125: Bash script used to produce the example in Figure 124.

IV.3.2.2 Interpolation in 3D

We start with an illustrative example of interpolating two concentric spheres as shown in Figure 126; the experiment can be reproduced using the bash script in Figure 127. The second example is the interpolation of geological data from the the Balzes fold model (Ramón et al., 2015) as shown in Figure 128; the experiment can be reproduced using the bash script in Figure 129. Note that the data set in Figure 128a is loaded from the "horizons_vals.txt" file created in line 17 of Figure 129; such a "3D value file" can be loaded in the viewer using the command "S2GoV_LoadFDIVals3d horizons_vals.txt".

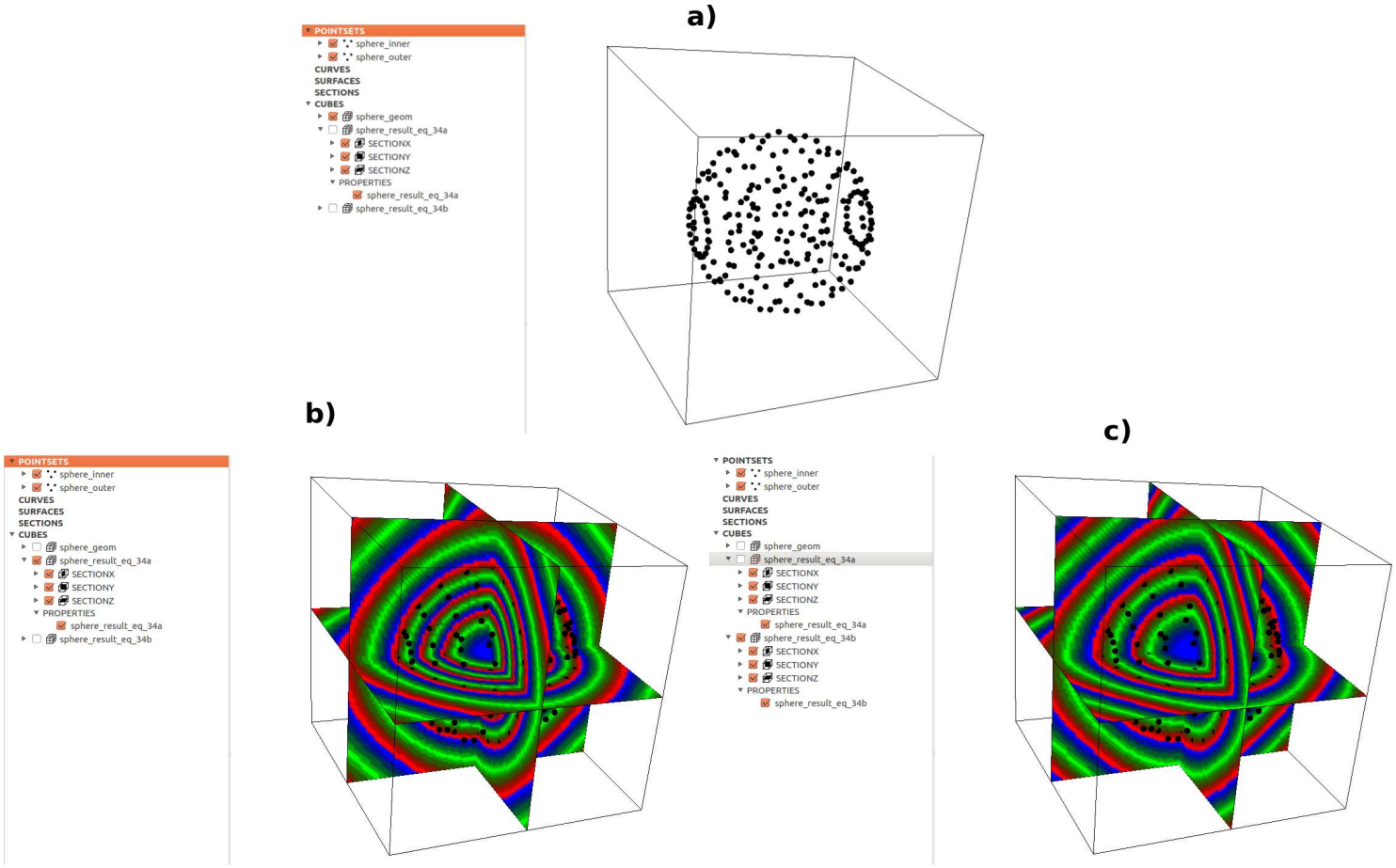

Figure 126: Interpolation of concentric spheres. This experiment can be reproduced using the bash script in Figure 127. **a**) Input data consisting of an inner and an outer sphere. **b**) Results obtained using equation 34a in chapter II.1 [equation 5 in Irakarama et al. (2018)]. **c**) Results obtained using equation 34b in chapter II.1 [equation 6 in Irakarama et al. (2018)].

```
#Create inner sphere points
S2Go_DeleteRSFFile sphere_inner.hd
S2Go_SpherePoints ox=30 oy=30 oz=30 r=10 n1=10 n2=10 > sphere_inner.hd

#Create outer
S2Go_DeleteRSFFile sphere_outer.hd
S2Go_SpherePoints ox=30 oy=30 oz=30 r=20 n1=20 n2=20 > sphere_outer.hd

#Create geometry (bounding box) info
echo "o1=0" > sphere_geom.hd
echo "o2=0" >> sphere_geom.hd
echo "o3=0" >> sphere_geom.hd
echo "d1=1" >> sphere_geom.hd
echo "d2=1" >> sphere_geom.hd
echo "d3=1" >> sphere_geom.hd
echo "n1=60" >> sphere_geom.hd
echo "n2=60" >> sphere_geom.hd
echo "n3=60" >> sphere_geom.hd

#Create "3D value files" for FD interpolation. The format for 3D value files is x y z val.
S2Go_3dPointSetToValsFile sphere_inner.hd val=0 ofile=sphere_inner_vals
S2Go_3dPointSetToValsFile sphere_outer.hd val=1 ofile=sphere_outer_vals
cat sphere_inner_vals* sphere_outer_vals* > sphere_vals.txt

#Interpolate using the smoothing operator in equation 34a of the RING meeting paper
S2Go_DeleteRSFFile sphere_result_eq_34a.hd
S2Go_3dFDSIS13A8System_gpu geom=sphere_geom.hd val=sphere_vals.txt onpts=1000000 > sphere_result_eq_34a.hd

#Interpolate using the smoothing operator in equation 34b of the RING meeting paper
S2Go_DeleteRSFFile sphere_result_eq_34b.hd
S2Go_3dFDSIS12A8System_gpu geom=sphere_geom.hd val=sphere_vals.txt onpts=1000000 > sphere_result_eq_34b.hd
```

Figure 127: Bash script used to produce the example in Figure 126.

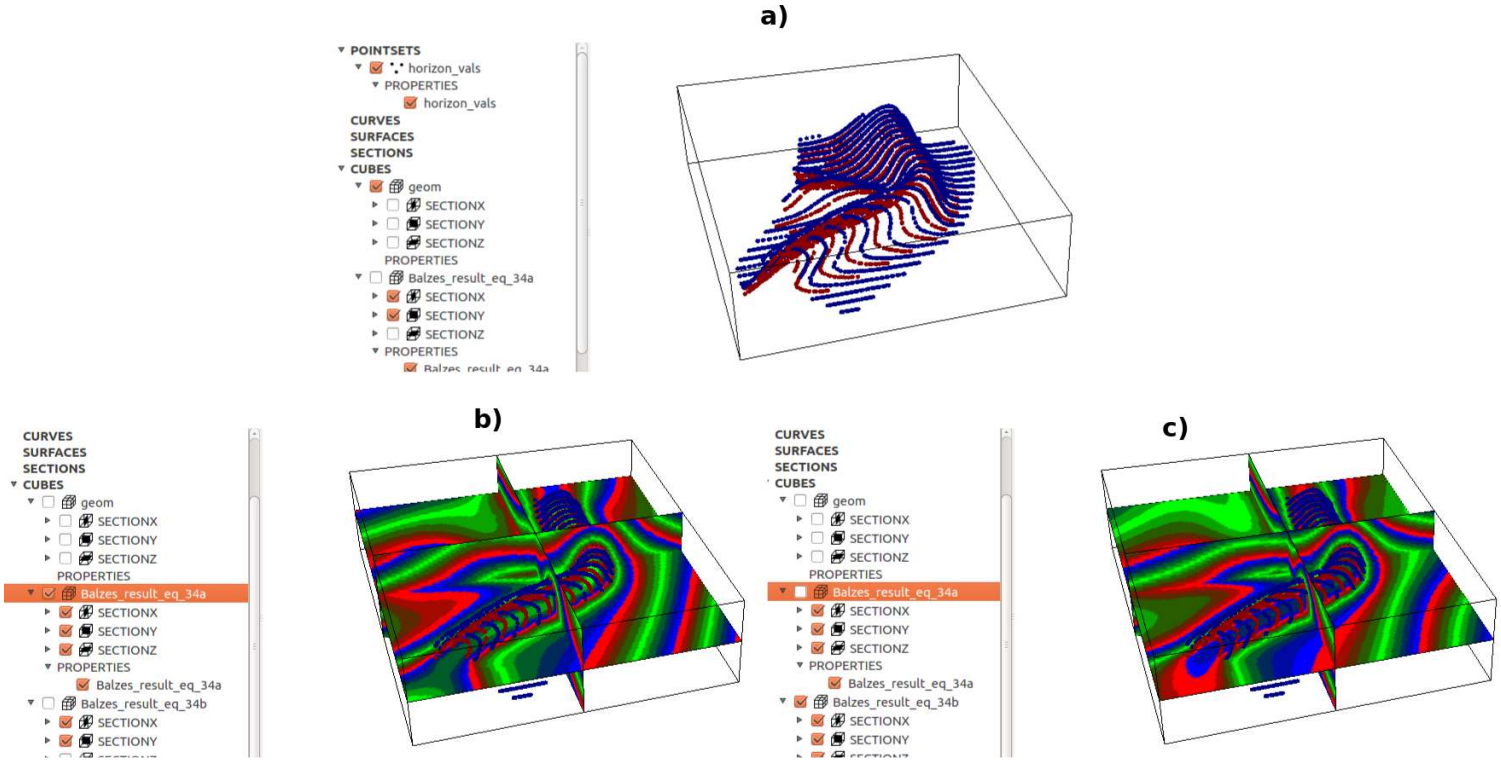

Figure 128: Interpolation of the Balzes fold model (Ramón et al., 2015). This experiment can be reproduced using the bash script in Figure 129. **a**) Input data consisting of two horizons. **b**) Results obtained using equation 34a in chapter II.1 [equation 5 in Irakarama et al. (2018)]. **c**) Results obtained using equation 34b in chapter II.1 [equation 6 in Irakarama et al. (2018)].

```
#Set approriate data filepath in header files
rm -rf data
cp -rf data_orig data
for hfile in $(ls data/*.hd) #loop through header files
do
    tmp=$(echo $hfile|cut -d '/' -f 2) #remove directory from header filenames
    base=$(echo $tmp|cut -d '.' -f 1) #remove extension from header filenames
    dfile=${base}.dt #data filename
    dfile_full=$(pwd)/data/${dfile} #data filename fullpath
    echo "in=${dfile_full}" >> ${hfile} #set data filename in header file
done

#Create "3D value files" for FD interpolation. The format for 3D value files is x y z val.
S2Go_3dPointSetToValsFile data/horizon1.hd val=0 ofile=horizon1
S2Go_3dPointSetToValsFile data/horizon2.hd val=1 ofile=horizon2
rm horizon_vals.txt
cat horizon* > horizon_vals.txt


#Interpolate using the smoothing operator in equation 34a of the RING meeting paper
S2Go_DeleteRSFFile Balzes_result_eq_34a.hd
S2Go_3dFDSIS13A8System_gpu geom=geom.hd val=horizon_vals.txt onpts=1000000 > Balzes_result_eq_34a.hd

#Interpolate using the smoothing operator in equation 34b of the RING meeting paper
S2Go_DeleteRSFFile Balzes_result_eq_34b.hd
S2Go_3dFDSIS12A8System_gpu geom=geom.hd val=horizon_vals.txt onpts=1000000 > Balzes_result_eq_34b.hd
```

Figure 129: Bash script used to produce the example in Figure 128.

IV.3.3 Acknowledgements

This work was done within the framework of the RING project at Université de Lorraine. I would like to thank the industrial and academic sponsors of the RING-Gocad Consortium managed by ASGA for their support.

Bibliography

T. Beraud, M. Irakarama, and P. Cupillard. Ray tomography: implementation, structural uncertainties and geological constraints. In *2018 RING meeting*, Université de Lorraine-ENSG, Nancy, France, September 2018. ASGA.

G. Caumon, P. Collon, C. L. C. de Veslud, S. Viseur, and J. Sausse. Surface-Based 3D Modeling of Geological Structures. *Mathematical Geosciences*, 41:927–945, 2009.

B. Colletta, J. Letouzey, R. Pinedo, J. F. Ballard, and P. Balé. Computerized X-ray tomography analysis of sandbox models: Examples of thin-skinned thrust systems. *Geology*, 19:1063–1067, 1991.

F. Collino and C. Tsogka. Application of perfectly machted absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. *GEOPHYSICS*, 66(1), 2001.

J. Etgen, S. Gray, and Y. Zhang. An overview of depth migration in exploration geophysics. *GEOPHYSICS*, 74(6), 2009.

S. Fomel, P. Sava, I. Vlad, Y. Liu, and V. Bashkardin. . *Madagascar: open-source software project for multidimensional data analysis and reproducible computational experiments*, 1, 2013.

M. Irakarama, G. Laurent, J. Renaudeau, and G. Caumon. Finite Difference Implicit Modeling of Geological Structures. In *80th EAGE Conference and Exhibition*, Copenhagen, Denmark, 2018.

P. Micikevicius. 3D finite-difference computation on GPUs using CUDA. In *GPGPU-2: Proceedings of the 2nd Workshop on General Purpose Processing on GPUs*, pages 79–84, Washington DC, USA, 2009.

M. J. Ramón, E. L. Pueyo, G. Caumon, and J. L. Briz. Parametric unfolding of flexural folds using palaeomagnetic vectors. *Geological Society of London*, 425:247–258, 2015.

J. Sethian and A. Popovici. 3d traveltime computation using the fast marching method. *GEOPHYSICS*, 64(2), 1999.

W. Symes. Migration velocity analysis and waveform inversion. *Geophysical Prospecting*, 56: 765–790, 2008.

J. Virieux. SH-wave propagation in heterogeneous media: Velocity-stress finite-difference method. *GEOPHYSICS*, 49:1933–1942, 1984.

M. Woodward, D. Nichols, O. Zdraveva, P. Whitfield, and T. Johns. A decade of tomography. *GEOPHYSICS*, 73(5), 2008.

Vers la réduction des incertitudes d'interprétation structurale à l'aide de données sismiques

Résumé :
Les modèles géologiques sont couramment utilisés pour estimer les ressources souterraines, pour faire des simulations numériques, et pour évaluer les risques naturels; il est donc important que les modèles géologiques représentent la géométrie des objets géologiques de façon précise. La première étape pour construire un modèle géologique consiste souvent à interpréter des surfaces structurales, telles que les failles et horizons, à partir d'une image sismique; les objets géologiques identifiés sont ensuite utilisés pour construire le modèle géologique par des méthodes d'interpolation. Les modèles géologiques construits de cette façon héritent donc les incertitudes d'interprétation car une image sismique peut souvent supporter plusieurs interprétations structurales. Dans ce manuscrit, j'étudie le problème de réduire les incertitudes d'interprtation à l'aide des données sismiques. Particulièrement, j'étudie le problème de déterminer, à l'aide des données sismiques, quels modèles sont plus probables que d'autres dans un ensemble des modèles géologiques cohérents. Ce problème sera connu par la suite comme *le problème d'évaluation des modèles géologiques par données sismiques*. J'introduis et formalise ce problème. Je propose de le résoudre par génération des données sismiques synthétiques pour chaque interprétation structurale dans un premier temps, ensuite d'utiliser ces données synthétiques pour calculer la fonction-objectif pour chaque interprétation; cela permet de classer les différentes interprétations structurales. La difficulté majeure d'évaluer les modèles structuraux à l'aide des données sismiques consiste à proposer des fonctions-objectifs adéquates. Je propose un ensemble de conditions qui doivent être satisfaites par la fonction-objectif pour une évaluation réussie des modèles structuraux à l'aide des données sismiques. Ces conditions imposées à la fonction-objectif peuvent, en principe, être satisfaites en utilisant les données sismiques de surface ("surface seismic data"). Cependant, en pratique il reste tout de même difficile de proposer et de calculer des fonctions-objectifs qui satisfassent ces conditions. Je termine le manuscrit en illustrant les difficultés rencontrées en pratique lorsque nous cherchons à évaluer les interprétations structurales à l'aide des donnés sismiques de surface. Je propose une fonction-objectif générale faite de deux composants principaux: (1) un opérateur de résidus qui calcule les résidus des données, et (2) un opérateur de projection qui projette les résidus de données depuis l'espace de données vers l'espace physique (le sous-sol). Cette fonction-objectif est donc localisée dans l'espace car elle génère des valeurs en fonction de l'espace. Cependant, je ne suis toujours pas en mesure de proposer une implémentation pratique de cette fonction-objectif qui satisfasse les conditions imposées pour une évaluation réussie des interprétations structurales; cela reste un sujet de recherche.

Mots-clés: incertitudes structurales, modélisation structurale, imagerie sismique, interprétation sismique, problèmes inverses

Towards Reducing Structural Interpretation Uncertainties Using Seismic Data

Abstract: Subsurface structural models are routinely used for resource estimation, numerical simulations, and risk management; it is therefore important that subsurface models represent the geometry of geological objects accurately. The first step in building a subsurface model is usually to interpret structural features, such as faults and horizons, from a seismic image; the identified structural features are then used to build a subsurface model using interpolation methods. Subsurface models built this way therefore inherit interpretation uncertainties since a single seismic image often supports multiple structural interpretations. In this manuscript, I study the problem of reducing interpretation uncertainties using seismic data. In particular, I study the problem of using seismic data to determine which structural models are more likely than others in an ensemble of geologically plausible structural models. I refer to this problem as *appraising structural models using seismic data*. I introduce and formalize the problem of appraising structural interpretations using seismic data. I propose to solve the problem by generating synthetic data for each structural interpretation and then to compute misfit values for each interpretation; this allows us to rank the different structural interpretations. The main challenge of appraising structural models using seismic data is to propose appropriate data misfit functions. I derive a set of conditions that have to be satisfied by the data misfit function for a successful appraisal of structural models. I argue that since it is not possible to satisfy these conditions using vertical seismic profile (VSP) data, it is not possible to appraise structural interpretations using VSP data in the most general case. The conditions imposed on the data misfit function can in principle be satisfied for surface seismic data. In practice, however, it remains a challenge to propose and compute data misfit functions that satisfy those conditions. I conclude the manuscript by highlighting practical issues of appraising structural interpretations using surface seismic data. I propose a general data misfit function that is made of two main components: (1) a residual operator that computes data residuals, and (2) a projection operator that projects the data residuals from the data-space into the image-domain. This misfit function is therefore localized in space, as it outputs data misfit values in the image-domain. However, I am still unable to propose a practical implementation of this misfit function that satisfies the conditions imposed for a successful appraisal of structural interpretations; this is a subject for further research.

Keywords: structural uncertainties, structural modeling, seismic imaging, seismic interpretation, inverse problems