The Dissertation Committee for Siwei Li
certifies that this is the approved version of the following dissertation:

Seismic Imaging and Velocity Model Building with the Linearized Eikonal Equation and Upwind Finite-differences

Committee:

Sergey B. Fomel, Supervisor

Bjorn Engquist

Omar Ghattas

Kyle T. Spikes

Kui Ren

Seismic Imaging and Velocity Model Building with the Linearized Eikonal Equation and Upwind Finite-differences

by

Siwei Li, B.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2014

Dedicated to my family.

Acknowledgments

It was an easy decision for me to seek a higher level degree after completing B.S. from Peking University. After four years of undergraduate education, I realized the rich mathematical, physical and programming contents in the field of geophysics required more systematic study. The best way would be to expose myself in a thought-provoking research environment, where I may not only apply what I had learned in the past fundamental courses but also refine some specific knowledge that I found interesting.

Coming to The University of Texas at Austin was my luck. For this, I would like to thank my adviser Sergey Fomel for bringing me into the world of exploration seismology. I could not find another discipline other than exploration seismology that might fit my expectation better: a study that deals with daily industrial challenges by most advanced science and technology. Sergey had been a patient and encouraging adviser. I was more than grateful to his top standard academic training, from which I was able to gradually build up a rewarding way of conducting research. I usually describe my relationship with Sergey as collaboration, instead of pure supervision. This feeling grew more intense as I became a senior student in the group. Sergey had given me freedom and trust in developing my interests. At critical stages of the research as well as when unexpected difficulties appeared, Sergey was always attentive to my analysis and provided me enlightening discussions. He was to me an experienced guide and ready-to-help friend.

Although my undergraduate degree was in geophysics, the curriculum emphasized global seismology in a large extent. Unsurprisingly, there existed a noticeable gap between global and exploration seismologies. More importantly, I was in deep need of relevant knowledge and skills so that I could work competently. Besides Sergey, I learned enormously from people I have met during the five years. Within Jackson School of Geosciences, my acknowledgments go to Paul Stoffa, Mrinal Sen, Kyle Spikes, Omar Ghattas, Stephen Grand, Bob Hardage, Hongliu Zeng, Steffen Saustrup and Thomas Hess. It turned out I worked also very closely with people from Department of Mathematics and Institute of Computational Engineering and Sciences. I thank Lexing Ying, Björn Engquist, Jack Poulson and Kui Ren in particular. I had also benefited a lot from courses of Carlos Torres-Verdin of Petroleum and Geosystem Engineering, and Lars Koesterke and Kent Milfeld of Texas Advanced Computing Center. Outside of The University of Texas at Austin, I collaborated with Alexander Vladimirsky of Cornell University and Tariq Alkhalifah of King Abdullah University of Science and Technology. Yang Liu of Jilin University was previously a post-doc in our group. During our overlap in Austin, he played an important role in my transition from an undergraduate student to a Ph.D.

Luckily I had three internships that provided me valuable technical and life experiences. When I was still a first year student, BP kindly extended me an opportunity to work in their world-leading seismic R&D team. Saudi Aramco invited me to their Dhahran office and generously shared with me their unique geophysical expertise. Chevron allowed me to contribute to one of their frontier researches, which in turn helped broaden my vision in pushing elegant theories towards practical appli-

cability. I would like to thank my mentors Petr Jilek, Timothy Keho, Yi Luo, Ross Hill and Uwe Albertin as well as my supervisors and colleagues for their instructions and supports.

After spending five years here, Austin had become my home. I already felt nostalgia when I realized that my next stop would probably be somewhere else. In the Bureau of Economical Geology, I parked in the same area (if I waked up on time), walked through the same corridor, and sat in the same office as my first day of arrival in 2009. I greatly appreciate caring from Jennifer Edwards, Cathy Brown, Joseph Yeh, Reuben Reyes, Eric Potter and Melissa Garcia. I thank my colleagues in Texas Consortium for Computational Seismology and many friends outside of our group: William Burnet, Vladimir Bashkardin, Xiaolei Song, Parvaneh Karimi, Alexander Klokov, Philip Guerrero, Yi Tao, Rui Zhang... The list of names runs much longer, and my sweet memories with you will for sure keep on growing. I was my pleasure to meet you and be part of your story here too.

Finally, I dedicate the thesis to my family. I am indebted to my parents, Yonghong Luo and Wei Li, for inculcating in me the dedication and discipline to do whatever I undertake well. Thanks to my grandparents, cousins, and all other family members, I had enjoyed a wonderful childhood and a memorable youth. I am never alone because of their love, for which I know there is a harbor right behind me no matter how far away I have sailed.

It was an easy decision to come here. It was one of my best decisions ever made.

SIWEI LI

The University of Texas at Austin

May 2014

Seismic Imaging and Velocity Model Building with the Linearized Eikonal Equation and Upwind Finite-differences

Siwei Li, Ph.D.

The University of Texas at Austin, 2014

Supervisor: Sergey B. Fomel

Ray theory plays an important role in seismic imaging and velocity model building. Although rays are the high-frequency asymptotic solutions of the wave equation and therefore do not usually capture all details of the wave physics, they provide a convenient and effective tool for a wide range of geophysical applications. Especially, ray theory gives rise to traveltimes. Even though wave-based methods for imaging and model building had attracted significant attentions in recent years, traveltime-based methods are still indispensable and should be further developed for improved accuracy and efficiency. Moreover, there are possibilities for new ray theoretical methods that might address the difficulties faced by conventional traveltime-based approaches. My thesis consists of mainly four parts. In the first part, starting from the linearized eikonal equation, I derive and implement a set of linear operators by upwind finite-differences. These operators are not only consistent with fast-marching eikonal solver that I use for traveltime computation but also computationally efficient. They are fundamental elements in the numerical implementations of my other

works. Next, I investigate feasibility of using the double-square-root eikonal equation for near surface first-break traveltime tomography. Compared with traditional eikonal-based approach, where the gradient in its adjoint-state tomography neglects information along the shot dimension, my method handles all shots together. I show that the double-square-root eikonal equation can be solved efficiently by a causal discretization scheme. The associated adjoint-state tomography is then realized by linearization and upwind finite-differences. My implementation does not need adjoint state as an intermediate parameter for the gradient and therefore the overall cost for one linearization update is relatively inexpensive. Numerical examples demonstrate stable and fast convergence of the proposed method. Then, I develop a strategy for compressing traveltime tables in Kirchhoff depth migration. The method is based on differentiating the eikonal equation in the source position, which can be easily implemented along with the fast-marching method. The resulting eikonal-based traveltime source-derivative relies on solving a version of the linearized eikonal equation, which is carried out by the upwind finite-differences operator. The source-derivative enables an accurate Hermite interpolation. I also show how the method can be straightforwardly integrated in anti-aliasing and Kirchhoff redatuming. Finally, I revisit the classical problem of time-to-depth conversion. In the presence of lateral velocity variations, the conversion requires recovering geometrical spreading of the image rays. I recast the governing ill-posed problem in an optimization framework and solve it iteratively. Several upwind finite-differences linear operators are combined to implement the algorithm. The major advantage of my optimization-based time-to-depth conversion is its numerical stability. Synthetic and field data examples demonstrate practical applicability of the new approach.

Table of Contents

Acknowledgments **v**

Abstract **ix**

List of Figures **xiii**

Chapter 1. Introduction **1**

Chapter 2. Linear operators by upwind finite-differences **25**

Chapter 3. First-break traveltime tomography with the double-square-root eikonal equation **33**

Chapter 4. Kirchhoff redatuming and migration using eikonal-based computation of traveltime source-derivatives **67**

Chapter 5. A robust approach to time-to-depth conversion and interval velocity estimation from time migration in the presence of lateral velocity variations **98**

Chapter 6. Conclusions **153**

Appendix **157**

Bibliography **165**

Vita **180**

List of Figures

1.1 Schematic drawing of a seismic survey with one source-receiver pair $s-r$. The three wavefields u_s, u_r and u_t are source, receiver (reflected) and transmitted wavefields, respectively. 2

1.2 The model (top) is a smoothed cross-line section of SEG/EAGE overthrust model. A point source wavefield is synthesized (bottom). The traveltime contours are overlaid on top of the wavefield snapshots as red dashed lines. 5

1.3 Impulse response of the Kirchhoff depth migration. For a given source-receiver pair $s-r$ and a reflection signal at time t, equation 1.5 spreads the signal in $\mathbf{x}$ along a constant traveltime contour (dashed line) that satisfies $t = T(\mathbf{x};\mathbf{s}) + T(\mathbf{x};\mathbf{r})$. For a constant velocity model v_0, this contour is an ellipse with two foci at $\mathbf{s}$ and $\mathbf{r}$ and a transverse diameter of $v_0 t$. 6

1.4 The Kirchhoff depth migration can output two types of CIGs. Both types are based on extending the image $I(\mathbf{x})$ to an additional dimension, where I along this extra axis should be flat (coherent). For a 2-D model (left), the offset h, i.e. distance between source and receiver, and reflection angle θ are popular choices. They are called surface-offset CIG $I(\mathbf{x}, h)$ (top right) and angle domain CIG $I(\mathbf{x}, \theta)$ (bottom right), respectively. 8

1.5 Traveltimes are computed in the Pluto model with both ray tracing (solid cyan and magenta lines) and FMM (dashed yellow line). The rays and FMM contour coincide at approximately $(7.3, 17.2)$ km. Due to salt complexity, multi-pathing is common in deep part of the model. 9

1.6 The eikonal equation is solved by FMM at three sources. Traveltime contours are overlaid on top of the model. A traveltime interpolation in source position might save computation and storage, if a sparse source sampling (solid cyan and magenta) can be used to approximate another source in-between (dashed yellow). 10

1.7 In a wave-based MVA, the image perturbation δI at sub-surface offset $\mathbf{h}$ is linked to scattered u_s (left) and u_r (right) under the Born approximation. In comparison, FWI measures the data-misfit from $\mathbf{s}$ to δv and then directly to $\mathbf{r}$. 14

1.8 A diving ray and zoom-in of the ray segments between two depth levels. 16

1.9 Each depth coordinate (z, x) (left) along the image ray is mapped into (right) the time coordinate (t_0, x_0) by using its corresponding traveltime t_0 and source location x_0. 18

1.10 A common synthetic model used in chapters 3-5. Overlaid white lines are horizontal reflectors for synthesizing reflection signals. The model is Cartesian discretized into 10 m grid spacings in z and x. 22

1.11 The synthetic data of model in Figure 1.10. Top and bottom plots are the zero-offset section and prestack data, respectively. 24

3.1 All four branches of DSR eikonal equation from different combination of upward or downward pointing of slowness vectors. Whether the slowness vector is pointing leftward or rightward does not matter because the partial derivatives with respect to s and r in equation 3.1 are squared. Figure 1.8 and equation 1.9 belong to the last situation. 37

3.2 An explicit discretization scheme. Compare with Figure 3.3. The arrow depicts a DSR characteristic with its root confined in the simplex $T^z T^r T^s$. 38

3.3 An implicit discretization scheme. The arrow indicates a DSR characteristic. Its root is located in the simplex $T^z T^r T^s$. 39

3.4 The gradient produced by standard tomography. The solid curve indicates a shot-indexed characteristic. 43

3.5 The gradient produced by DSR tomography. The solid curve indicates a DSR characteristic, which has one end in plane $z = 0$ and the other in plane $s = r$. Compare with Figure 3.4. 46

3.6 The synthetic model used for DSR FMM accuracy test. The overlaid curves are rays traced from a shot at $(0, 0)$ km. 50

3.7 Grid refinement experiment (implicit discretization). In both figures, the solid blue curve is the reference values and the dashed curves are computed by DSR FMM. Top: fixed $\Delta x = 10$ m and $\Delta z = 50$ m (cyan), 10 m (magenta), 5 m (black). Bottom: fixed $\Delta z = 10$ m and $\Delta x = 50$ m (cyan), 10 m (magenta), 5 m (black). 51

3.8 Grid refinement experiment (explicit discretization). The experiment set-ups are the same as in Figure 3.7. 52

3.9 A smoothed Marmousi model overlaid with rays traced from a shot at $(0, 0)$ km. Because of velocity variations, multi-pathing is common in this model, especially at large offsets. 53

3.10 DSR FMM with non-causal branches. The solid black lines are reference values. There are two groups of dashed lines, both from DSR FMM but one with the optional search process turned-on and the other without. The differences between them are negligible and hardly visible. 54

3.11 (Top) model overlaid with traveltime contours of a source at $(0, 0)$ km and sensitivity kernels of (middle) the standard tomography and (bottom) the DSR tomography. 55

3.12 DSR first-break traveltimes in the Marmousi model. The original model is decimated by 2 in both vertical and lateral directions, such that $nz = 376$, $nx = 1151$ and $\Delta z = \Delta x = 8$ m. 57

3.13 (Top) a zoom-in of Marmousi model and (bottom) the initial model for tomography. 58

3.14 Convergence history of DSR tomography (solid) and standard tomography (dashed). There is no noticeable improvement on misfit after the fourth update. 59

3.15 Inverted model of (top) standard tomography and (bottom) DSR tomography. Compare with Figure 3.13. 60

3.16 Inversion with noisy data. Convergence history of DSR tomography (solid) and standard tomography (dashed). No significant decrease in misfit appears after the fourth update. 61

3.17 Inversion with noisy data. Inverted model of (top) standard tomography and (bottom) DSR tomography. Compare with Figure 3.15. . . . 62

3.18 (Top) a shot gather at $(0, 6)$ km. First-breaks are overlaid on top of the seismograms. (Bottom) input first-breaks for DSR tomography. . 63

3.19 The inverted near-surface model (top) and its difference with exact model in Figure 1.10 (bottom). 64

4.1 (Left) a constant-velocity-gradient model $v(z, x) = 2 + 0.5x$ km/s and (right) its analytical traveltime source-derivative for a source at origin $\mathbf{x_s} = (0, 0)$ km. 78

4.2 Comparison of error in computed source-derivative by (left) the proposed method and (right) a centered second-order finite-difference estimation based on traveltime tables. The maximum absolute errors are 0.15 s/km and 0.56 s/km, respectively. 78

4.3 Traveltime interpolation error of three different schemes: (top left) the analytical traveltime of a source at location $(0, 0.25)$ km; (top right) error of the cubic Hermite interpolation; (bottom left) error of the linear interpolation; (bottom right) error of the shift interpolation. Using derivatives in interpolation enables a significantly higher accuracy. The $l2$ norm of the error are 1.5 s, 9.2 s and 6.0 s respectively. 79

4.4 Source-sampling refinement experiment. The plot shows, at a fixed model grid sampling of 0.01 km and increasing source sampling, the error in source-derivative estimated by a first-order finite-difference (solid) and a centered second-order finite-difference scheme (dotted) decrease. The horizontal axis is the number of sources and the source sampling is uniform. The vertical axis is the natural logarithm of the absolute error. The flat line (dash) is from the proposed eikonal-based method and is source-sampling independent. 80

4.5 Gird-spacing refinement experiment. The plot shows, at a fixed source sampling of 1 km and increasing model grid sampling, the error in source-derivative estimated by the proposed eikonal-based method decreases. Meanwhile, the errors of both first- and second-order finite-difference estimations do not improve noticeably. The horizontal axis is the number of grid points in both directions and the grid sampling is uniform. See Figure 4.4 for descriptions of the vertical axis and the lines. 81

4.6 Constant-velocity-gradient background model $v(z, x) = 1.5 + 0.25z + 0.25x$ km/s with dome shaped reflectors. 83

4.7 Zero-offset Kirchhoff migration image with (top) the cubic Hermite interpolation and (bottom) the shift interpolation. 84

4.8 Zero-offset Kirchhoff migration image with (top) the linear interpolation and (bottom) the cubic Hermite interpolation without anti-aliasing. 85

4.9 (Top) the smoothed Marmousi model. The model has a 4 m fine grid. (Bottom) the traveltime error by the cubic Hermite interpolation. . . 86

4.10 The traveltime error by (top) the linear interpolation and (bottom) the shift interpolation. 87

4.11 Traveltime interpolation for a fixed subsurface location. Compare between the result from a dense source sampling (solid blue), cubic Hermite interpolation (dotted magenta), linear interpolation (dashed cyan) and shift interpolation (dashed black). The $l2$ norm of the error (against the dense source sampling results) of 49 evenly interpolated sources between interval $(0, 3)$ km and $(0, 3.2)$ km for all locations but the top 100 m source singularity region are 3.9 s, 9.2 s and 11.6 s respectively. 88

4.12 Image of Kirchhoff migration with first-arrivals (no interpolation). . . 90

4.13 Image of Kirchhoff migration with first-arrivals and a sparse source/receiver sampling. 91

4.14 Image of semi-recursive Kirchhoff migration with a three-step redatuming from top surface to 1.5 km depth and a 0.5 km interval each time. The sparse source/receiver sampling is the same as in Figure 4.13. 92

4.15 (Top) the result of sparse source interpolation. Values are traveltimes from sources at surface $z = 0$ km to the datum at depth $z = 0.4$ km. (Bottom) relative absolute errors of the traveltime interpolation against a dense source sampling of 10 m. 94

4.16 A common shot gather at $(0, 8)$ km before redatuming (top) and at $(0.4, 8)$ km after redatuming (bottom). 95

5.1 A modified FMM for forward modeling. Black dots represent region that is *IN*, gray dots are *FRONT* and white dots are region of *OUT*. See section on FMM in Chapter 2 for explanations on these labels. . . 111

5.2 (Top) a constant velocity gradient model and (bottom) the analytical Dix velocity v_d. A curved image ray is mapped to the time domain as a straight line. 117

5.3 Analytical values of (top) t_0 and (bottom) x_0 of the model in Figure 5.2. Both figures are overlaid with contour lines that, according to equation 5.7, are perpendicular to each other. Each contour line of x_0 is an image ray, while the contours of t_0 illustrate the propagation of a plane-wave. 118

5.4 The cost defined by equation 5.20 (top) before and (bottom) after inversion. The least-squares norm of cost E is decreased from 9.855 to 0.057. 119

5.5 The difference between exact model and (top) initial model and (bottom) inverted model. The least-squares norm of model misfit is decreased from 15.6 km^2/s^2 to 2.7 km^2/s^2. 120

5.6 Comparison of the common-image-gathers at $x = 1.5$ km of (left) exact model, (middle) prior model and (right) inverted model. 121

5.7 Analytical values of (top) t_0 and (bottom) x_0, overlaid with contour lines. 125

5.8 The (top) geometrical spreading and (bottom) Dix velocity associated with the model used in Figure 5.7. 126

5.9 The cost (top) before and (bottom) after inversion. The least-squares norm of cost E is decreased from 10.431 to 0.047. 127

5.10 The difference between exact model and (top) initial model and (bottom) inverted model. The least-squares norm of model misfit is decreased from 5.0 km^2/s^2 to 0.5 km^2/s^2. 128

5.11 (Top) a synthetic model and (middle) Dix velocity converted to depth. Both overlaid with image rays. (Bottom) the model perturbation for testing linearization. 129

5.12 (Top) exact dt_0 and (bottom) linearly predicted dt_0 by equation 5.27. 131

5.13 (Top) exact dx_0 and (bottom) linearly predicted dx_0 by equation 5.26. 132

5.14 (Top) exact df and (bottom) linearly predicted df by equation 5.28. . 133

5.15 (Top) the exact δw and (middle) the computed δw of the first linearization step. (Bottom) the inverted interval velocity model. Compare with Figure 5.11. 134

5.16 (Top) the estimated time-migration velocity of a section of Gulf of Mexico dataset and (bottom) the corresponding time-migrated image. 135

5.17 (Top) the Dix velocity converted from v_m in Figure 5.16 and (bottom) the Dix-inverted prior model for inversion, overlaid with image rays. . 136

5.18 The costs of (top) prior model ($E = 140.25$) and (bottom) inverted model ($E = 1.81$). 137

5.19 Convergence history of the proposed optimization-based time-to-depth conversion. 138

5.20 (Top) the inverted model, overlaid with image rays, and (bottom) its difference from the prior model in Figure 5.17. 139

5.21 (Top) the time-migrated image in Figure 5.16 is mapped to depth using products of the time-to-depth conversion. (Bottom) PSDM image using inverted model in Figure 5.20. 140

5.22 PSDM images of (left) the prior model and (right) the inverted model. Both images are plotted for the same central deep part. 141

5.23 The surface offset common-image gathers at $x = 11$ km of (left) prior model and (right) inverted model. 142

5.24 Zoom-in of deep parts of Figure 5.23. The events of prior model curve downward. 143

5.25 The surface offset common-image gathers at $x = 12$ km of (left) prior model and (right) inverted model. 144

5.26 Zoom-in of deep parts of Figure 5.25. The events of prior model curve upward. 145

5.27 The time-migrated image (top) of redatumed data that is free of near-surface effects and the Dix-inverted prior model (bottom). 147

5.28 The inverted model by proposed iterative time-to-depth conversion (top) and the exact model (bottom). Compare with Figures 1.10 and 5.27. 148

5.29 The costs of (top) prior model and (bottom) inverted model. 149

5.30 Time-migrated image in Figure 5.27 mapped to depth using (top) Dix-inverted model and (bottom) inverted model. 150

Chapter 1

Introduction

RAY THEORY AND TRAVELTIMES IN SEISMIC IMAGING

In exploration geophysics, the major task of imaging, or equivalently, migration, is to place recorded reflection events in their correct geological positions. The concept of seismic imaging can be explained by a simple example. Consider in Figure 1.1 an Earth model of two layers. In a seismic survey, multiple sources are shot at the surface in order to achieve illumination of a target sub-surface area. The source wavefield propagates mainly downward until it hits the layer boundaries, where it is reflected and transmitted. The reflected wave then propagates upward and is finally recorded by the receivers. Given the Earth model of top layer, seismic imaging locates the layer boundary, i.e. reflector, by first finding source wavefield u_s and receiver wavefield u_r in the sub-surface, and then zero-lag cross-correlating these two wavefields in time:

$$I(\mathbf{x}) = \int \int_{t=0}^{\infty} u_s(\mathbf{x}, t; \mathbf{s}) \cdot u_r(\mathbf{x}, t; \mathbf{s})\, dt\, d\mathbf{s} \; . \tag{1.1}$$

Equation 1.1 is commonly referred to as the *imaging condition* (Claerbout, 1985). Here $\mathbf{x}$ is a vector containing depth, in-line and cross-line coordinates. Ideally, u_s and u_r should coincide with each other at the reflector, where the cross-correlation is maximum in magnitude. An integration (summation) over shots $\mathbf{s}$ helps reconstruct the part of reflector that is under illumination. Note that the image I obtained by equation 1.1 is only a *map* of the reflectors and the magnitude of I should not be

interpreted as the Earth reflectivity.

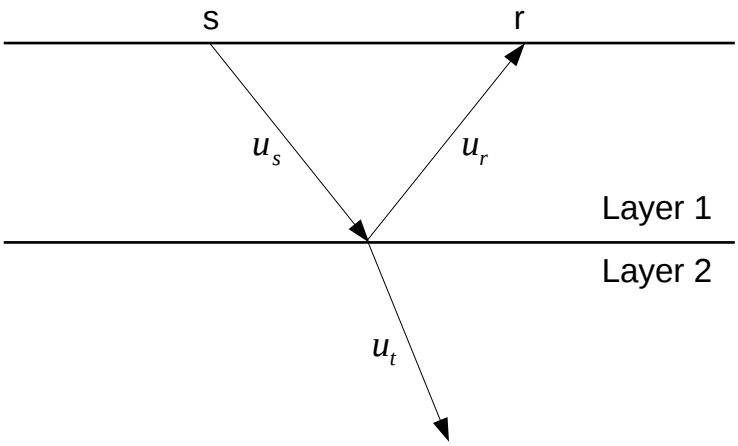


Figure 1.1: Schematic drawing of a seismic survey with one source-receiver pair $s-r$. The three wavefields u_s, u_r and u_t are source, receiver (reflected) and transmitted wavefields, respectively. chapter-introduction/figs layer

Seismic imaging can be divided into two broad categories according to the methods used for wave simulation (Etgen et al., 2009), which is critical for obtaining u_s and u_r in equation 1.1. Intuitively, we may consider solving the wave equation. This gives rise to the wave-based algorithms (Jones et al., 2008). There are many techniques that fall into this category, ranging from traditional one-way wave equation migration (Claerbout and Doherty, 1970; Loewenthal et al., 1976; Berkhout, 1979; Gazdag, 1981; Stoffa et al., 1990; Ristow and Ruhl, 1994; Zhang et al., 2005) to modern reverse-time (two-way wave equation) migration (Baysal et al., 1983; Etgen, 1986; Symes, 2007; Fletcher et al., 2009; Zhang and Zhang, 2009; Fowler et al., 2010; Fomel et al., 2013). Thanks to increasing computation power, the evolution follows a trend of applying less approximation in mathematics and physics. For example, the one-

way wave equation considers waves propagating either upward or downward. From one-way to two-way migration, we are able to handle the dip limitation and prismatic wave problems but at a higher cost. Meanwhile, going from acoustic wave equation to orthorhombic or even full elastic wave equation enables us to better honor the realistic wave propagation. All these advancements are important for seismic imaging in complex geological areas such as sub-salt and deep water environments (Farmer et al., 2006).

Traveltime-based migration

The other category of seismic imaging algorithms is the traveltime ray-based methods. Ray theory was introduced into seismic imaging at its earliest age and has since then been an indispensable tool (Červený, 2001; Yilmaz, 2001; Popovici and Sethian, 2002). Consider a wavefield in the frequency domain $u = u(\mathbf{x}, \omega)$, ray theory expands u into

$$u(\mathbf{x}, \omega) \approx \exp\{i\omega T(\mathbf{x})\} \sum_{j=0}^{\infty} \frac{A_j(\mathbf{x})}{(i\omega)^j} \ . \tag{1.2}$$

The asymptotic series 1.2 include a *traveltime* term T and an amplitude term A. Expression 1.2 is then inserted into the wave equation. For simplicity, I use the constant-density acoustic wave equation

$$\nabla^2 u(\mathbf{x}, \omega) + \frac{\omega^2}{v^2(\mathbf{x})} u(\mathbf{x}, \omega) = 0 \ , \tag{1.3}$$

where $v(\mathbf{x})$ is the velocity. After collecting terms in (angular) frequency ω, the leading order term of $(i\omega)^2$ leads to the eikonal equation (Chapman, 2002):

$$\nabla T(\mathbf{x}) \cdot \nabla T(\mathbf{x}) = \frac{1}{v^2(\mathbf{x})} \equiv w(\mathbf{x}) \ . \tag{1.4}$$

Here I define w as slowness-squared. Equation 1.4 means traveltimes T can be computed by solving a nonlinear partial differential equation. The next term of order $i\omega$ results in so-called transport equation that gives amplitude A. In case of anisotropy, $v(x)$ is replaced with $v(x, \nabla T/|\nabla T|)$ because it depends on the angle of wave propagation (Červený, 2001).

A classical method for solving the eikonal equation 1.4 is ray tracing. As characteristics of the eikonal equation, rays intuitively connect the velocity model with traveltimes. To illustrate the effectiveness of traveltimes in explaining kinematics of the wavefield, I compare in Figure 1.2 several wavefield snapshots and their corresponding traveltime contours. The source is placed in the middle of the model and traveltimes are computed for the earliest possible values, i.e. *first-arrivals*, from the source according to equation 1.4. A remarkably good agreement between the wave-front and traveltime verifies the applicability of traveltimes in seismic imaging. Also note that the first-arrival is not necessarily the most energetic arrival. However, their separation does not take place immediately but after a certain duration of wave propagation.

There are several traveltime-based imaging methods, among which Kirchhoff migration is the most familiar one. The Kirchhoff migration can be implemented in either depth domain or time domain. While the Kirchhoff depth migration is performed in the physical space $\mathbf{x}$, the Kirchhoff time migration adopts a different coordinate, i.e. time coordinate. The depth-to-time coordinate transformation is beneficial especially when it comes to velocity estimation. The Kirchhoff time migration, as well as time-to-depth conversion, will be introduced later in this chapter. On the

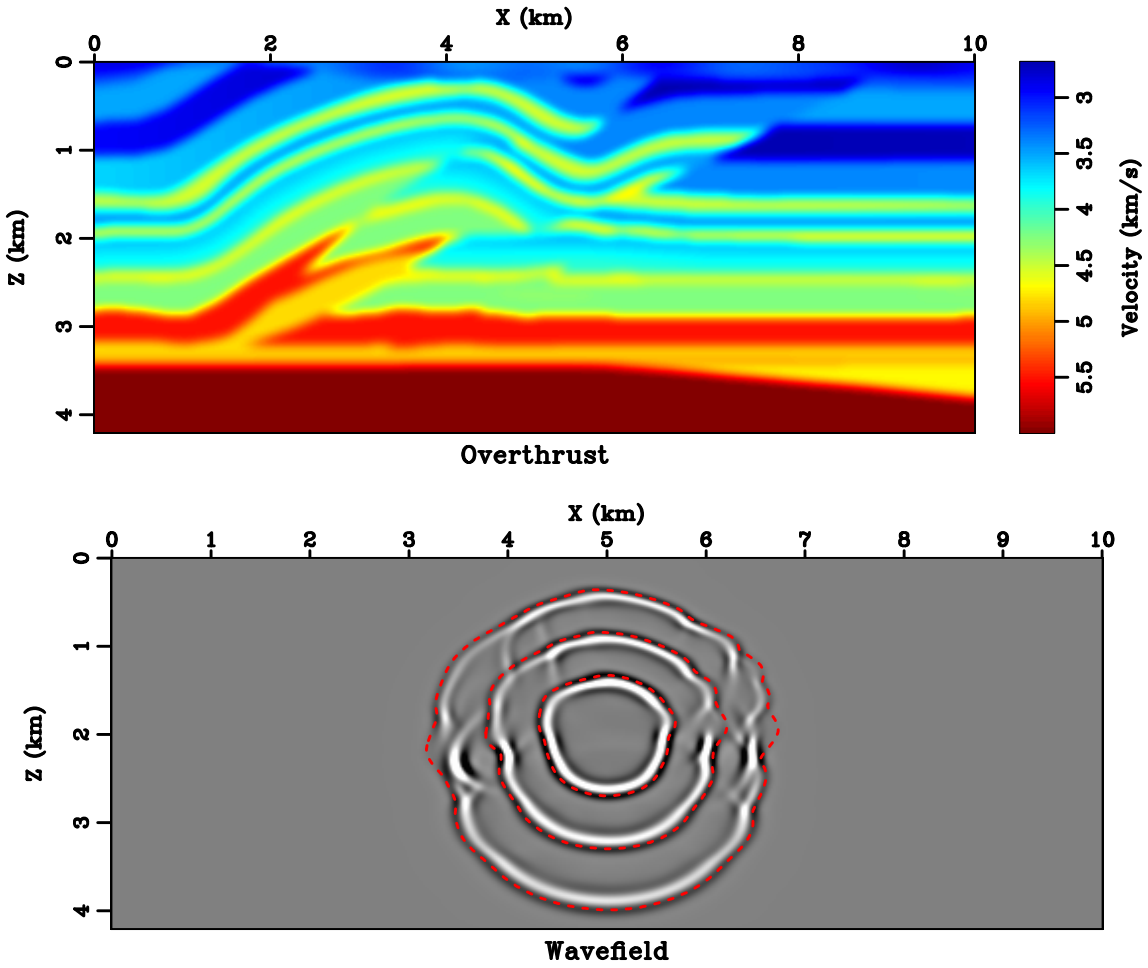


Figure 1.2: The model (top) is a smoothed cross-line section of SEG/EAGE overthrust model. A point source wavefield is synthesized (bottom). The traveltime contours are overlaid on top of the wavefield snapshots as red dashed lines. chapter-introduction/overthrust wave0

other hand, a Kirchhoff depth migration links the kinematic operation of diffraction stack to an asymptotically correct solution of downward-continued seismic wavefield (French, 1975; Schneider, 1978). Its general form is

$$I(\mathbf{x}) = \int\int M(\mathbf{x};\mathbf{s},\mathbf{r})\,\frac{\partial}{\partial t} D(T(\mathbf{x};\mathbf{s}) + T(\mathbf{x};\mathbf{r});\mathbf{s},\mathbf{r})\,d\mathbf{s}\,d\mathbf{r}\ , \tag{1.5}$$

where D is the recorded data at receivers and M is an optional data weighting operator. Figure 1.3 shows the impulse response of the Kirchhoff depth migration operator for one source-receiver pair. Comparing equations 1.1 and 1.5, the imaging condition is replaced by an integration. At the same time, wavefields in the sub-surface are approximated by the traveltimes between sources/receivers and the image locations. The summation $T(\mathbf{x};\mathbf{s}) + T(\mathbf{x};\mathbf{r})$ implies zero-lag cross-correlation. Thanks to these asymptotic approximations, the Kirchhoff depth migration is significantly cheaper than the wave-based imaging methods. Over the years, it has been the most widely used method in the industry.

In order to perform the integration in equation 1.5, traveltimes between image location $\mathbf{x}$ and both source $\mathbf{s}$ and receiver $\mathbf{r}$ are required. Neglecting M, formula 1.5 accounts for only traveltime but still provides a kinematically correct I. Developments in true-amplitude Kirchhoff migration (Beylkin, 1985; Bleistein, 1987; Bleistein et al., 2001; Schleicher et al., 2007) further combine eikonal equation with transport equation that prescribes M by taking into account A in the ray series 1.2, resulting in good fidelity in both kinematics and dynamics. Moreover, Kirchhoff migration offers efficiency and flexibility. It can be easily implemented in a target-oriented fashion (selective $\mathbf{x}$) for cost-effective 3-D migrations (Audebert, 2001). Common-image-gathers (CIGs) are its convenient by-products that are important for tomography purposes (Xu et al., 2001). The idea is that for a (kinematically) correct model, the image I

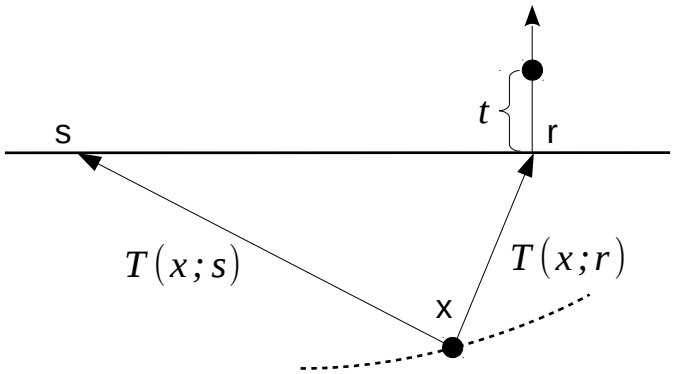


Figure 1.3: Impulse response of the Kirchhoff depth migration. For a given source-receiver pair $s-r$ and a reflection signal at time t, equation 1.5 spreads the signal in $\mathbf{x}$ along a constant traveltime contour (dashed line) that satisfies $t = T(\mathbf{x};\mathbf{s}) + T(\mathbf{x};\mathbf{r})$. For a constant velocity model v_0, this contour is an ellipse with two foci at $\mathbf{s}$ and $\mathbf{r}$ and a transverse diameter of $v_0 t$. chapter-introduction/figs depkirdraw

should be coherent between various source-receiver pairs, because the same reflector should stay stationary regardless of illumination. As illustrated in Figure 1.4, a velocity update can be obtained by examining the flatness of CIGs along either offset or angle axis. This concept will be revisited in the next section when discussing velocity model building.

In the family of Kirchhoff-type migrations, a significant extension is beam migration (Hill, 1990, 2001; Gray, 2005). Instead of real-valued traveltimes in classical ray theory, Gaussian beams add a complex correction term to the traveltimes so that the beam allows certain width but the energy quickly attenuates away from the central ray (Li et al., 2011). It has been shown that summation of Gaussian beams will approach a solution of the wave equation (Liu et al., 2013). Consequently, compared

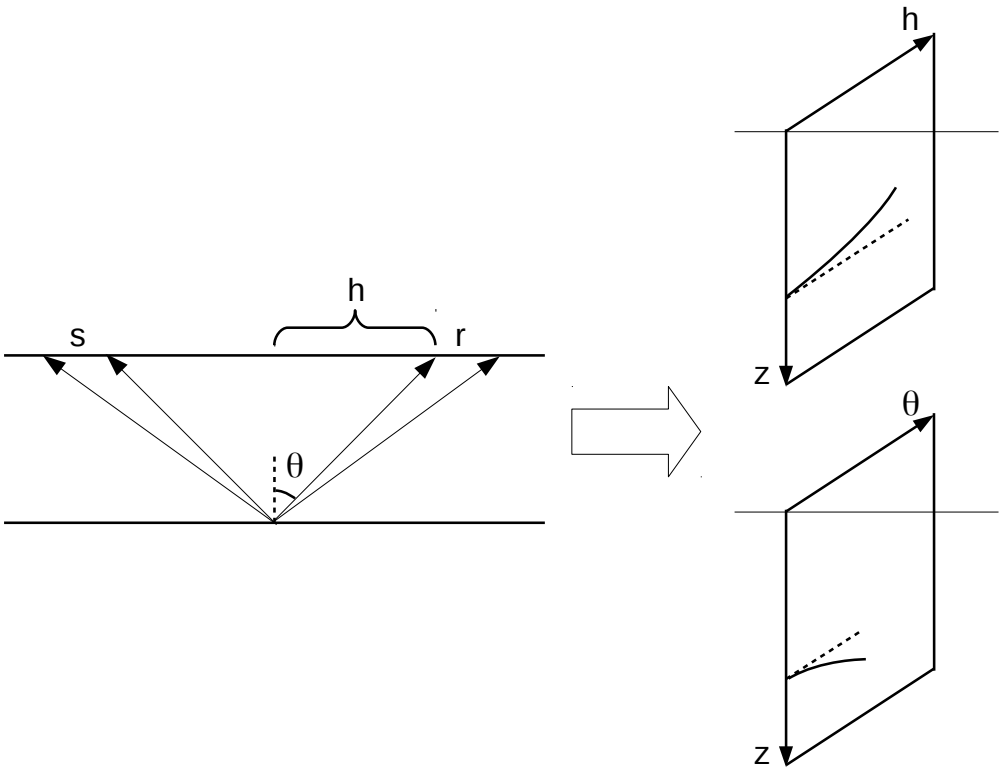


Figure 1.4: The Kirchhoff depth migration can output two types of CIGs. Both types are based on extending the image $I(\mathbf{x})$ to an additional dimension, where I along this extra axis should be flat (coherent). For a 2-D model (left), the offset h, i.e. distance between source and receiver, and reflection angle θ are popular choices. They are called surface-offset CIG $I(\mathbf{x}, h)$ (top right) and angle domain CIG $I(\mathbf{x}, \theta)$ (bottom right), respectively. chapter-introduction/figs cigintro

with standard Kirchhoff migration, Gaussian beam migration is more consistent with the wave equation migration 1.1.

Traveltime computation

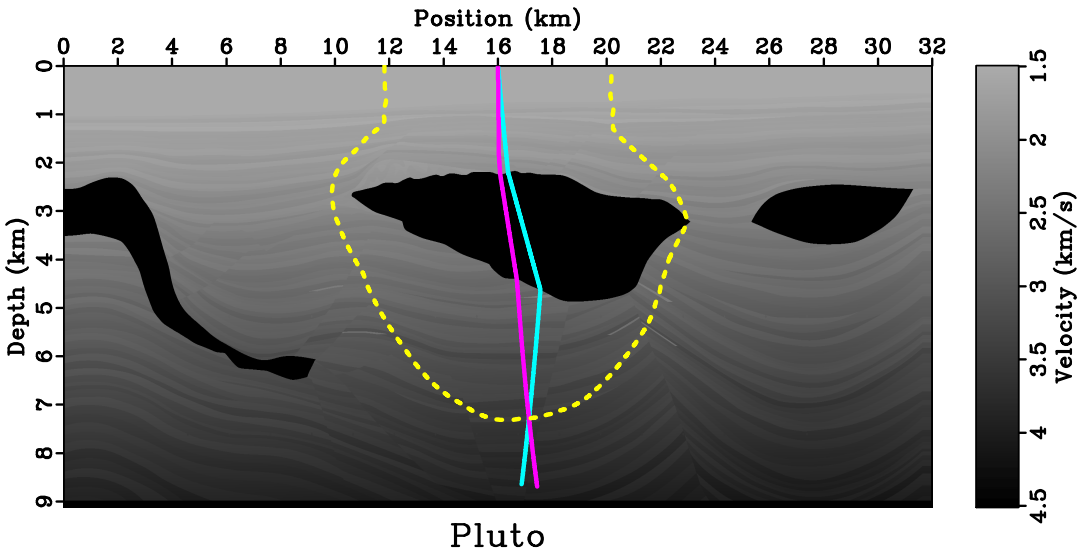


Figure 1.5: Traveltimes are computed in the Pluto model with both ray tracing (solid cyan and magenta lines) and FMM (dashed yellow line). The rays and FMM contour coincide at approximately $(7.3, 17.2)$ km. Due to salt complexity, multi-pathing is common in deep part of the model. chapter-introduction/pluto pluto

In Kirchhoff migrations, traveltime computation and storage are usually the first steps. Aside from ray tracing, the eikonal equation can also be solved by various finite-difference methods (Reshef and Kosloff, 1986; Vidale, 1988, 1990). Two popular finite-difference eikonal solvers are fast-marching method (FMM) (Sethian, 1999; Sethian and Popovici, 1999) and fast-sweeping method (FSM) (Zhao, 2005). In Figure 1.5 I solve the eikonal equation in Pluto model (Stoughton et al., 2001) by both ray tracing and FMM for the same source at $(0, 16)$ km, where traveltimes

develop along the rays and are plotted in a contour in FMM. FMM, as well as FSM, are limited to computing only the first-arrival branch because they consider the viscosity solution of eikonal equation. Figure 1.5 demonstrates that ray tracing, on the other hand, can directly handle multi-pathing but requires extra irregular mapping from ray coordinate to migration coordinate. However, recent developments in finite-difference eikonal solvers show not only possible strategies to accommodate all arrivals (Fomel and Sethian, 2002; Symes and Qian, 2003; Bashkardin et al., 2012) but also improved efficiency (Kim, 2001; Jeong and Whitaker, 2008; Chacon and Vladimirsky, 2012a), accuracy (Popovici and Sethian, 2002), and extensions to anisotropic wavefront propagation (Sethian and Vladimirsky, 2001; Fomel, 2004).

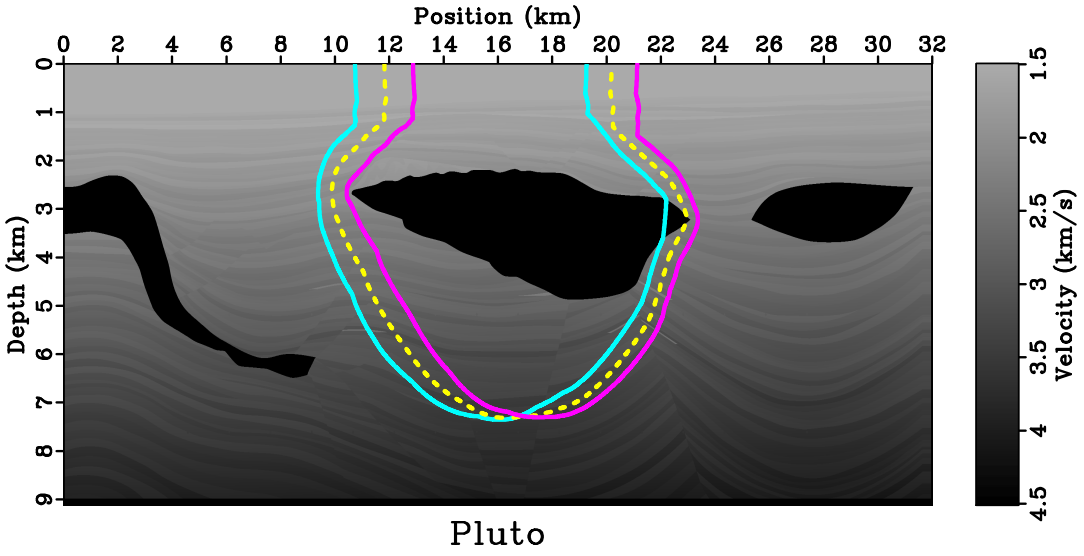


Figure 1.6: The eikonal equation is solved by FMM at three sources. Traveltime contours are overlaid on top of the model. A traveltime interpolation in source position might save computation and storage, if a sparse source sampling (solid cyan and magenta) can be used to approximate another source in-between (dashed yellow). chapter-introduction/pluto plutos

In practice, due to the dense sampling of shots and a large number of discretized grid points, especially in 3-D, traveltime computation still takes considerable cost (Mendes, 2000). Many works have been proposed to reduce the cost. One intuitive idea would be to compute traveltime tables on a coarse source sampling and then interpolate the results to a finer one. Figure 1.6 shows in the same model as Figure 1.5 three traveltime contours by FMM. These contours are for sources at $(0, 15)$ km, $(0, 16)$ km and $(0, 17)$ km, respectively, and they intersect with each other at an image location around $(7.2, 16.9)$ km. In this case, the motivation is to find the traveltime between this image location and the middle source from those of other two sources. An interpolation in source might achieve higher accuracy than the simplest linear interpolation if the derivatives are taken into account. Vanelle and Gajewski (2002) advocate estimating the derivative of traveltime with respect to source position by explicit finite differences and then using these derivatives for traveltime interpolation. Their method relies on a fine source sampling for accurate traveltime derivatives. Alternatively, it is more desirable to find a traveltime source-derivative that is independent of source sampling and build a high-order traveltime interpolation scheme.

RAY THEORY AND TRAVELTIMES IN VELOCITY MODEL BUILDING

The velocity model is of crucial importance for a successful seismic imaging. In general, the Earth model may contain not only the seismic velocity but also other earth medium parameters that should be accounted for, such as density (Aki and Richards, 1980), anisotropy (Tsvankin, 2012) and attenuation (Liu et al., 1976; Carcione et al.,

1988). Usually the velocity model building and the imaging are interleaved. Several iterations of model refinements based on optimizing the image qualities are necessary (Jones, 2010). Commonly this iterative process is referred to as tomography, which is a type of inversion based on linearization (Backus and Gilbert, 1968). Due to the fact that the tomographic inversion is mostly ill-conditioned, a good prior model and proper constraints must be supplied in order to obtain a satisfying final result.

Similarly to the classification of methods in seismic imaging, tools for velocity model building are either traveltime-based or wave-based. The wave-based tomography has attracted noticeable attentions following the recent progress in wave-based imaging. Full-waveform inversion (FWI) was first proposed by Tarantola (1984) as a data-domain nonlinear optimization. Applying partial derivative in velocity to both sides of the wave equation 1.3 results in

$$\left(\nabla^2 + \frac{\omega^2}{v^2}\right) \delta u = \frac{2\omega^2 u}{v^3} \delta v \ , \tag{1.6}$$

where for conciseness I omit $\mathbf{x}$. Equation 1.6 means the wavefield perturbation δu in response to a velocity perturbation δv can be found by solving the wave equation again, where the source is replaced with a scaled background u located at δv. Equation 1.6 is a linearization of equation 1.3 and retains only the first-order term. This linearization is referred to as the Born approximation.

At the limit, FWI tries to predict and fit observed refraction and reflection data in both kinematics and dynamics (Zhou et al., 2012). It can be superior to ray-based approach for its finite frequency wave-path kernel (Woodward and Rocca, 1988; Tromp et al., 2005; Xie and Yang, 2008). However, in practice the effective part

of FWI is limited to refraction energies (Luo and Schuster, 1991; Sheng et al., 2006; Brenders and Pratt, 2007). More crucially, FWI requires an accurate low wavenumber prior model to keep the inversion away from local minima (Virieux and Operto, 2009). A straightforward least-squares FWI cost function is vulnerable because it might compare two seismograms that are out of phase by more than one signal cycle. A number of alternative cost functions and regularizations have been proposed to overcome this difficulty (Causse et al., 1999; Shin and Min, 2006; Guitton et al., 2012; Ma et al., 2012; Zhang et al., 2012; Baek et al., 2013; Tang et al., 2013).

Another branch of wave-based tomography addresses so-called migration velocity analysis (MVA). The basic methodology of MVA is that, after migration, the CIGs should be flat (Ehinger and Lailly, 1995). If not, the residual moveout (RMO) indicates errors in the model (Liu and Bleistein, 1995; Biondi and Symes, 2004). In Figure 1.4, wave-based MVA replaces Kirchhoff integral 1.5 with wave-based imaging 1.1, and uses sub-surface offset instead of surface offset. The sub-surface offset is created by cross-correlating u_s and u_r at shifted image locations during imaging condition, for example $u_s(\mathbf{x}-\mathbf{h})$ and $u_r(\mathbf{x}+\mathbf{h})$. In consistent with Figure 1.4, MVA assumes that a correct model should focus energies at zero sub-surface offset (Sava and Fomel, 2003). If not, a model update can be computed following the relationship between an image perturbation with respect to a velocity perturbation:

$$\delta I = \int \int_{t=0}^{\infty} [\delta u_s \cdot u_r + u_s \cdot \delta u_r] \, dt \, d\mathbf{s} \; . \tag{1.7}$$

Equation 1.7 is derived by linearizing equation 1.1 and I omit t, and $\mathbf{x}$ in u_s and u_r. Note δu_s and δu_r satisfy equation 1.6. Therefore, a velocity perturbation δv influences both wave-paths between $\mathbf{s}$ to $\mathbf{x}$ and $\mathbf{r}$ to $\mathbf{x}$, as illustrated in Figure 1.7.

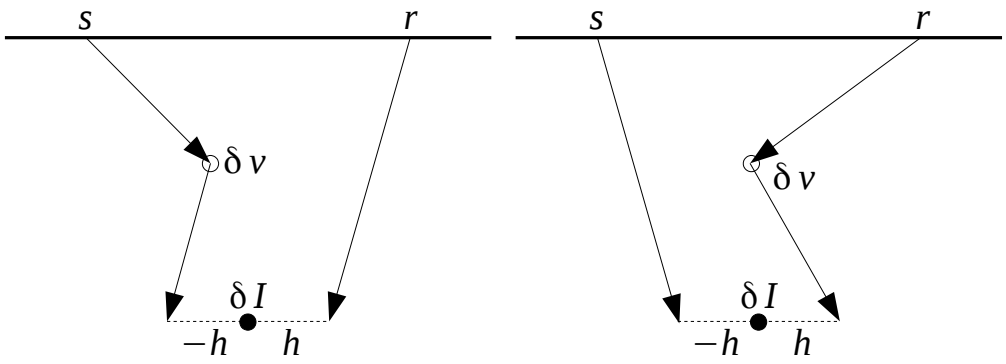


Figure 1.7: In a wave-based MVA, the image perturbation δI at sub-surface offset $\mathbf{h}$ is linked to scattered u_s (left) and u_r (right) under the Born approximation. In comparison, FWI measures the data-misfit from $\mathbf{s}$ to δv and then directly to $\mathbf{r}$. chapter-introduction/figs mvadraw

Wave-based MVA can be carefully implemented as an automatic procedure (Symes and Carazzone, 1991; Mulder and ten Kroode, 2002; Sava and Biondi, 2004; Shen and Symes, 2008; Tang et al., 2013; Weibull and Arntsen, 2013; Zhang and Biondi, 2013; Perrone et al., 2014). But the underlying Born approximation demands preprocessing of the data to minimize non-primary reflection events, or else the inversion might be misled to nonphysical solution. Meanwhile, the computational costs of both FWI and wave-based MVA are exceptionally high. Ray-based tomographies are frequently revisited because they are commonly used to construct the prior model for FWI and wave-based MVA.

The major difference between wave-based and traveltime-based approaches is the amount of information involved in model construction. Ray-based methods focus

mainly on the kinematic information carried by traveltimes. They have been successfully applied in many disciplines, ranging from exploration geophysics to global seismology and medical imaging (Nolet, 1987). Traveltime-based tomography can be further categorized by refraction and reflection tomography, depending on the domain for inversion. The traveltime-based reflection tomography is connected with MVA, both targeting deep structures and operating in the image domain. Instead of wave-paths, in this case the velocity updates follow rays. For near-surface model construction, however, refraction tomography is more favorable than reflection tomography.

First-break traveltime tomography

In distinction to tomographies in the image domain, first-break traveltime tomography, or turning-ray tomography (Zhu et al., 1992), is a data-domain approach that seeks to minimize the misfits between forward modeled and observed refraction-arrival traveltimes. It is capable of resolving near-surface medium-to-long wave-length velocity structures. Conventionally for each iteration, one needs to first solve eikonal equation 1.4 for traveltimes, and then calculate a gradient (model update) from the data residual according to the linearized eikonal equation (Aldridge, 1994):

$$2\left(\nabla T \cdot \nabla\right)\delta T = \delta w \ . \tag{1.8}$$

Equation 1.8 can be derived by differentiating equation 1.4. In first-break traveltime tomography, δT is the data residual limited by acquisition and δw is the model perturbation. The linear operator $\nabla T \cdot \nabla$ corresponds to the Fréchet derivative matrix.

Great potentials remain in improving the reliability and resolution of first-break traveltime tomography (Bube and Washbourne, 2008; Hardy, 2013). In case of 3-D as well as large 2-D models, an eikonal solver combined with the adjoint-state formulation (Plessix, 2006) is advantageous to an explicit evaluation of the Fréchet derivative matrix for solving equation 1.8 (Leung and Qian, 2006; Taillandier et al., 2009). However, such an implementation assembles the gradient in a summation-over-shots fashion and therefore forbids communication between individual shots. Possible improvements in convergence and robustness may be gained by switching from the eikonal equation 1.4 and its linearization 1.8 to the so-called double-square-root (DSR) eikonal equation (Belonosova and Alekseev, 1974). Unlike eikonal equation, the DSR eikonal equation allows both source and receiver positions to move along the refraction ray-path and, consequently, considers the prestack data as a whole. The DSR eikonal equation can be derived by considering a ray-path and its segments between two depth levels. Figure 1.8 illustrates a diving ray (Zhu et al., 1992) in 2-D with velocity $v = v(z, x)$. I use $T(z, r, s)$ to denote the total traveltime of the ray-path beneath depth z, where r and s are sub-surface receiver and source lateral positions, respectively. At both source and receiver sides traveltimes satisfy the eikonal equation 1.4, therefore

$$\frac{\partial T}{\partial z} = -\sqrt{\frac{1}{v^2(z,s)} - \left(\frac{\partial T}{\partial s}\right)^2} - \sqrt{\frac{1}{v^2(z,r)} - \left(\frac{\partial T}{\partial r}\right)^2}\,. \tag{1.9}$$

Obviously, from equation 1.4 to equation 1.9, both the forward modeling and tomographic inversion of the DSR tomography are fundamentally different from the traditional eikonal-based tomography. This means detailed investigations must be done in order to verify the feasibility of DSR tomography.

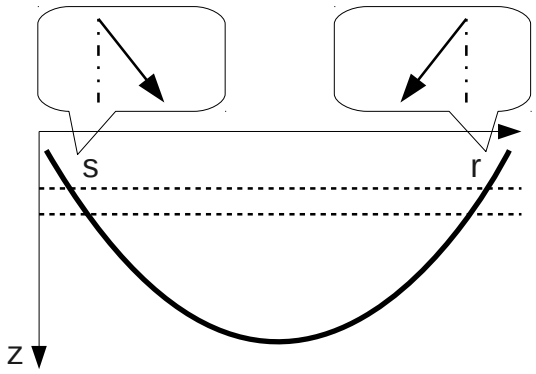


Figure 1.8: A diving ray and zoom-in of the ray segments between two depth levels. chapter-introduction/../dsrtomo/figs raypath

Time-domain model building and time-to-depth conversion

Historically, the first velocity model building technique was to perform normal moveout (NMO) correction to the prestack data (Sheriff and Geldart, 1995). Considering the tomographic methods introduced above, this traditional approach could be interpreted as MVA in the data-domain. By NMO correction, an effective velocity model is defined in time (commonly two-way traveltime) instead of depth, where the optimum NMO velocity flattens the primary reflection events in a common midpoint (CMP) gather. In case of dipping reflectors, a dip moveout (DMO) (Hale, 1984; Anderson and Tscankin, 1997) after NMO attempts to correct the effect of non-vertical zero-offset rays. Afterwards, if there is no lateral velocity variations, Dix (1955) proved that the NMO velocity equals the root-mean-square (RMS) velocity and a Dix inversion can find the interval velocity. No tomographic update is needed.

When lateral velocity variations are present, the above work-flow is no longer

valid. Instead, proper time migration should be performed. Time migration takes a different route from depth migration by reducing the problem of velocity estimation to a parameter scan. This is done with the help of a coordinate transformation, i.e. from depth coordinate $\mathbf{x}$ to time coordinate. Following Hubral (1977), I denote the time coordinate in 2-D as $\hat{\mathbf{x}} = (t_0, x_0)$. An important concept in understanding time migration and time coordinate is the *image ray*. Here x_0 is the location of the image ray at the earth surface. t_0 is the traveltime along the image ray, which is traced with a source at location x_0 and a direction normal to the earth surface. Figure 1.9 illustrates image rays in 2-D and a mapping from depth coordinate to time coordinate.

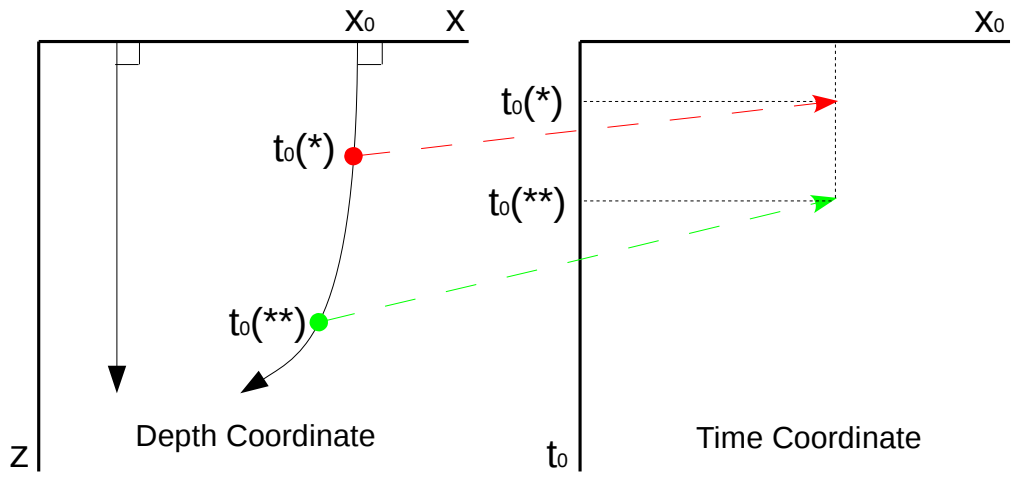


Figure 1.9: Each depth coordinate (z, x) (left) along the image ray is mapped into (right) the time coordinate (t_0, x_0) by using its corresponding traveltime t_0 and source location x_0. chapter-introduction/../time2dep/figs imageray

If the image rays do not intersect each other, i.e. there are no caustics, then

a unique inverse transformation $z(t_0, x_0)$ and $x(t_0, x_0)$ exists. The forward mapping $t_0(z, x)$ and $x_0(z, x)$ can be done with a knowledge of seismic velocity in depth $v(z, x)$. The counterpart for v in the time-domain is the time-migration velocity $v_m = v_m(t_0, x_0)$, where v_m is defined from the traveltime approximation commonly used in prestack Kirchhoff time migration (Yilmaz, 2001):

$$\hat{T}(t_0, x_0; s, r) \approx T(z, x; s) + T(z, x; r) \ , \tag{1.10}$$

where s and r are source and receiver positions in 2-D, respectively, and

$$\hat{T}(t_0, x_0; s, r) = \sqrt{t_0^2 + \frac{|x_0 - s|^2}{v_m^2(t_0, x_0)}} + \sqrt{t_0^2 + \frac{|x_0 - r|^2}{v_m^2(t_0, x_0)}} \ . \tag{1.11}$$

Based on equations 1.10 and 1.11, a general form of the Kirchhoff time migration reads

$$I(\hat{\mathbf{x}}) = \int \int M(\hat{\mathbf{x}}; \mathbf{s}, \mathbf{r}) \, D(\hat{T}(\hat{\mathbf{x}}; \mathbf{s}, \mathbf{r})) \, d\mathbf{s} \, d\mathbf{r} \ . \tag{1.12}$$

Equation 1.11 is the result of a truncated Taylor expansion for traveltimes around location x_0. According to equations 1.10 and 1.11, the total traveltime from source s to an image point (z, x) and then to receiver r is estimated by using the traveltime t_0 of an image ray passing through (z, x) and the distances between s, r and x_0. An important feature of v_m is that it is independent of each other for different (t_0, x_0). Finding v_m amounts to a parameter search for each point in the (t_0, x_0) space (Fomel, 2003). As a result, the estimation of time-migration velocity is rather automatic and velocity model building in the time domain is more convenient than in the depth domain. However, note the time coordinate $\hat{\mathbf{x}}$ is distorted and time migration velocity v_m does not trivially equal to interval velocity v.

Given the advantages of time-domain processing, it is of great interest to convert time migration velocity and time-migrated image to the depth domain. Unfortunately, the explanations of Hubral (1977) did not result in a constructive generalized Dix inversion that deals with lateral velocity variations. This last step in time-to-depth conversion is studied by Cameron et al. (2007), who proved by applying paraxial ray theory to image rays that the problem is governed by a system of ill-posed nonlinear partial differential equations (PDEs). Cameron et al. (2009) also explained that, due to ill-posedness of the problem, algorithms based on extrapolation are subject to numerical instability. In this regard, a robust time-to-depth conversion algorithm is worth studying, which may serve as a bridge between the convenient time processing and a reasonable prior for further depth model refinement.

THESIS OUTLINE

My thesis consists of six chapters:

- Chapter 1: Introduction
- Chapter 2: Linear Operators by Upwind Finite-differences

  I introduce in this chapter four linear operators by upwind finite-differences. The first operator is inspired by the linearized eikonal equation, while the other three operators are its adjoint, inverse, and adjoint of the inverse. The methods I propose in the following chapters 3-5 rely on these operators in their numerical implementations.
- Chapter 3: First-break Traveltime Tomography with the Double-square-root Eikonal Equation

The first-break traveltime tomography is capable of near-surface velocity model construction. In this chapter, I investigate the feasibility of DSR-based first-break traveltime tomography. I provide an efficient DSR eikonal solver with causal implicit discretization and demonstrate that the DSR gradient does not suffer from conflicting information among shots as in standard eikonal-based framework.

- Chapter 4: Kirchhoff Redatuming and Migration Using Eikonal-based Computation of Traveltime Source-derivatives

  Once the near-surface velocity model is at hand, the prestack data can be redatumed. I examine in this chapter the response of eikonal equation in source location perturbation and obtain an eikonal-based traveltime source-derivative that is independent of source sampling. A Kirchhoff depth migration and redatuming can make use of these derivatives for accurate traveltime interpolation and anti-aliasing.

- Chapter 5: A Robust Approach to Time-to-depth Conversion and Interval Velocity Estimation from Time Migration in the Presence of Lateral Velocity Variations

  For deeper parts of the velocity model that are not resolved by first-break traveltime tomography, I turn to reflection energy and employ time domain processing. I revisit in this chapter the problem of time-to-depth conversion and propose a novel optimization formulation. A key advantage of my method is that it addresses the stability issue faced by previous works.

- Chapter 6: Conclusions

I organize chapters 3-5 in a sequence that resembles a work-flow in seismic data processing. To make connections between these chapters, I create a synthetic dataset that will be introduced in the following section. Because the contents of each chapter are relatively independent of each other, I start each chapter by first giving a more specific introduction. Next, I provide details on theoretical derivations and numerical implementations. I then demonstrate applications of the proposed methods by several examples, including the common synthetic dataset. Each chapter is finally concluded with some discussion on limitations and possible future extensions.

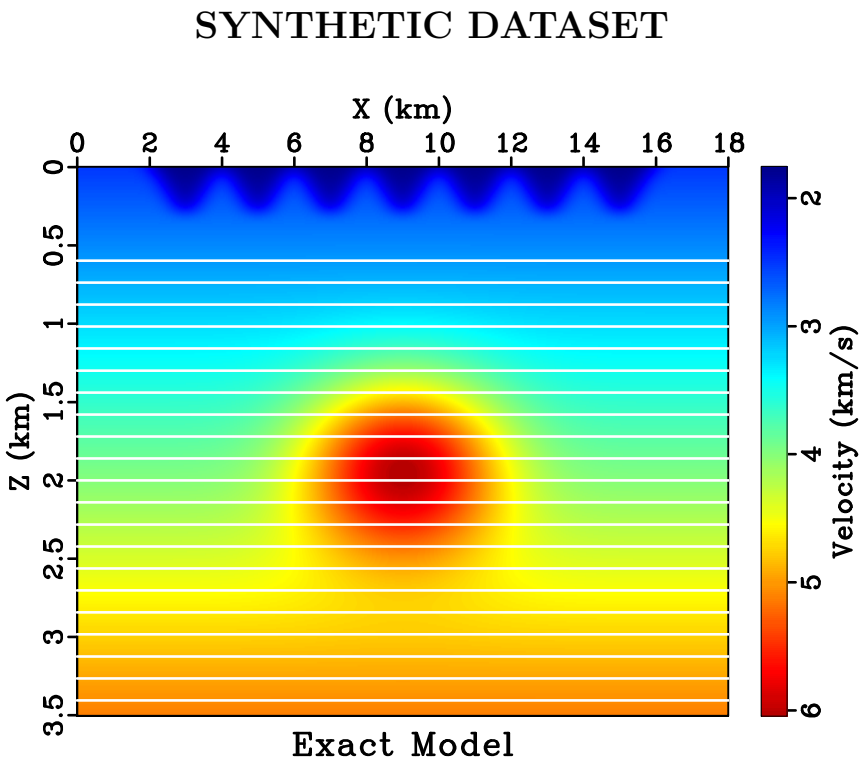


Figure 1.10: A common synthetic model used in chapters 3-5. Overlaid white lines are horizontal reflectors for synthesizing reflection signals. The model is Cartesian discretized into 10 m grid spacings in z and x. chapter-introduction/thesis exact

To create a synthetic model and dataset that consist typical problems targeted in chapters 3-5, I first make a $v(z)$ model where the velocity increases linearly in depth z. Several sinuous inclusions with low velocity are added to the model to represent near-surface geologies. Moreover, a strong positive Gaussian anomaly in deeper part of the model results in significant lateral velocity variations. The final model is illustrated in Figure 1.10.

The synthetic dataset includes both refraction and reflection signals. I use a Ricker wavelet of peak frequency 10 Hz. There are in total 401 shots uniformly distributed between $(0, 3)$ km and $(0, 8)$ km with an absolute maximum offset of 5 km. The refraction part is obtained by a two-way wave equation modeling (Fomel et al., 2013). For the reflection part, I use a one-way wavefield extrapolation with the multi-reference split-step Fourier method (Stoffa et al., 1990; Popovici, 1996). Figure 1.11 shows the synthetic data at zero-offset as well as in prestack. From the zero-offset data, it is easy to observe the influence of near surface, which creates the short-period oscillations along shot axis. The positive anomaly, on the other hand, is responsible for the asymmetry in offset that is prominent in shot gathers.

Due to the presence of both near-surface and deep velocity complexities, appropriate time-domain processings should be performed after correcting the influences of near surface. Chapter 3 will reconstruct the sinuous near-surface structures from first-breaks of the refractions. Next, based on this reconstructed near-surface model, Chapter 4 redatums the reflection signals. Finally, Chapter 5 will use the redatumed reflections that are free of near-surface effects for velocity estimation and migration.

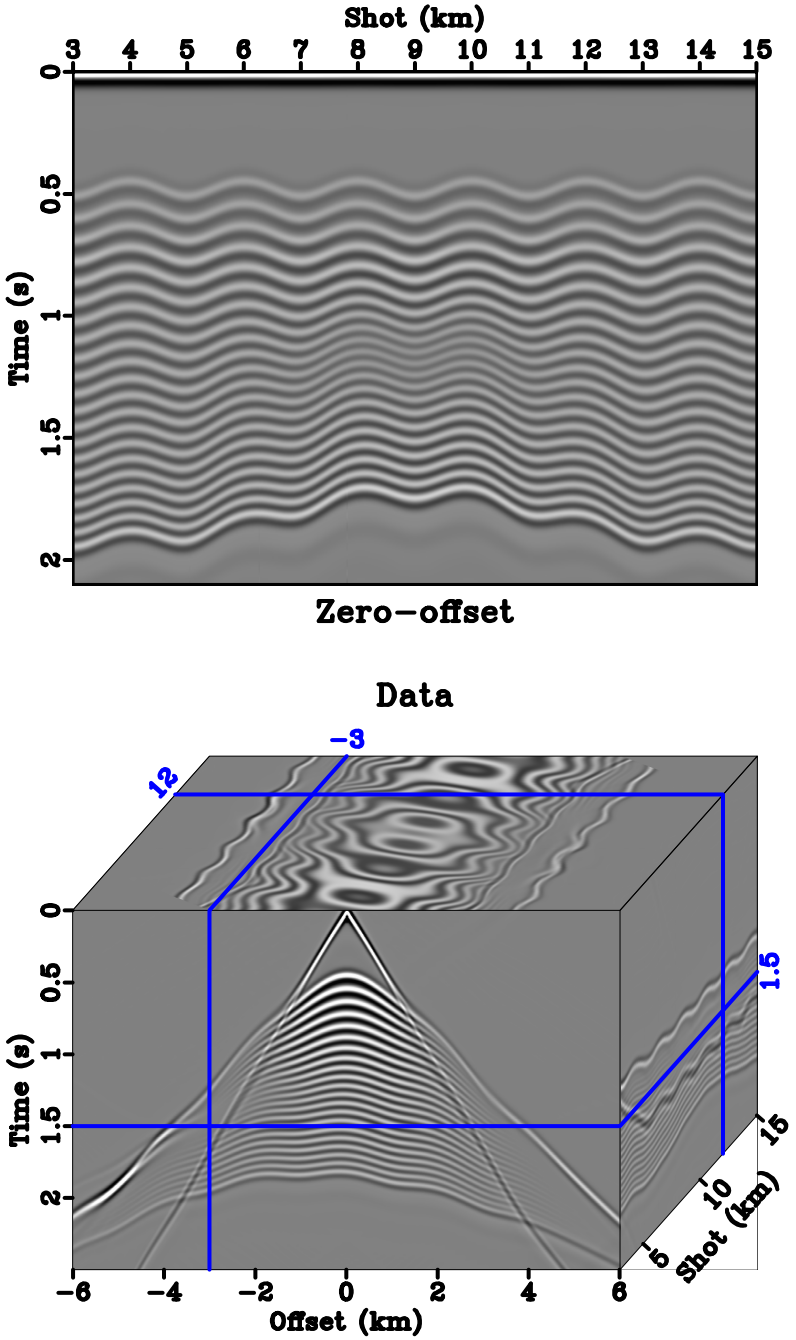


Figure 1.11: The synthetic data of model in Figure 1.10. Top and bottom plots are the zero-offset section and prestack data, respectively. chapter-introduction/thesis rcmp

Chapter 2

Linear operators by upwind finite-differences

INTRODUCTION

Traveltimes play a key role throughout this thesis. I adopt the fast-marching eikonal solver (FMM) for computing traveltimes. Sethian (1999) provides a detailed discussion on FMM. FMM is a non-recursive algorithm with a global update scheme that ensures monotonic wave-front propagation and a local update scheme that considers only causal solution of the viscous eikonal equation. The local scheme employs upwind finite-differences in discretizing the eikonal equation. It uses traveltimes of neighboring grid points to advance the wave-front to the current grid point. The sequence of visiting all the grid points is actively maintained by a priority queue according to the global scheme.

Franklin and Harris (2001) studies upwind finite-difference scheme for a family of first-order partial differential equations, of which the linearized eikonal equation 1.8 is a member. Essentially, the linearized eikonal equation is first discretized such that at each grid point the finite-difference stencil is chosen towards the upwind direction of a background traveltime table. Next, the linearized eikonal equation is solved, one grid point at a time, in the same order which FMM would follow in computing the background traveltime. Franklin and Harris (2001) showes that the upwind finite-difference scheme is stable, accurate and efficient.

There are in total four linear operators based on upwind finite-differences that are necessary for the methods I propose in chapters 3-5. In this chapter, I show their derivations as well as implementations.

FAST-MARCHING METHOD

The global scheme of FMM defines three labels: *IN*, *FRONT* and *OUT*. As indicated by their names, these flags stand for the status of grid points during traveltime computation:

- *IN*: the traveltime of this grid point is already computed and should not be modified;
- *FRONT*: the traveltime of this grid point has a temporary value and can be updated;
- *OUT*: this grid point has never been visited.

FMM starts with labeling all grid points as *OUT* (and assigning a large traveltime value). Next, the source is initiated. At the core of FMM is a priority queue that stores the index of grid points and arranges them in an decreasing order in their traveltimes. The source wave-front is pushed into priority queue with *IN* attached. The rest of the algorithm is a dynamic process that keeps looping over until the priority queue is empty (Sethian, 1996):

- Extract the minimum traveltime grid point from queue:

1. Label it as *IN* if it is not;
2. For each of its neighbors that is not *IN*, compute a temporary traveltime by the local update scheme;
3. Replace the neighbor's current traveltime with the temporary one if the latter is smaller;
4. If the temporary traveltime is accepted and the neighbor is outside of queue, label the neighbor as *FRONT* and insert it into queue.

- Restore the priority queue.

Conceptually, *IN*, *FRONT* and *OUT* denote regions where the wave-front has swept through, is current at, and has not yet reached. Clearly, the version FMM I outline here is limited to first-arrival traveltimes. If there are in total N discretized grid points, FMM has a computational complexity of $O(N \log N)$, where the $\log N$ factor comes from the priority queue.

UPWIND FINITE-DIFFERENCES

Consider in 3-D a domain discretized into Cartesian grids with uniform grid size of $(\Delta x, \Delta y, \Delta z)$. Let $T^k_{i,j}$ be the given background traveltime at vertices $\mathbf{x}^k_{i,j} = (x_i, y_j, z_k)$ and define difference operator $D^{\pm}_x$ for x direction as

$$D^{\pm}_x T^k_{i,j} = \pm \frac{T^k_{i\pm1,j} - T^k_{i,j}}{\Delta x} \ , \tag{2.1}$$

where the $\pm$ sign corresponds to the two neighbors of $T^k_{i,j}$ in x direction. An upwind scheme (Franklin and Harris, 2001) picks the sign by

$$D_x T^k_{i,j} = \max\left(D^-_x T^k_{i,j}, -D^+_x T^k_{i,j}, 0\right) \ . \tag{2.2}$$

After similar definitions for D_y and D_z, the linearized eikonal equation 1.8 can be discretized as

$$(D_x T^k_{i,j} \cdot D_x)\,\delta T^k_{i,j} + (D_y T^k_{i,j} \cdot D_y)\,\delta T^k_{i,j} + (D_z T^k_{i,j} \cdot D_z)\,\delta T^k_{i,j} = \frac{1}{2}\delta w^k_{i,j}\ . \tag{2.3}$$

Here the subscript and superscript i, j and k of δT and δw have the same meaning as of T. Note in equation 2.3 $\nabla\delta T^k_{i,j}$ is also discretized by upwind finite-differences. However, its choice of upwind neighbors follows $T^k_{i,j}$ instead of itself. $D_m T \cdot D_m$ with $m = x, y, z$ are linear operators, from which $\nabla T \cdot \nabla$ can be assembled. In fact, $\nabla T \cdot \nabla$ is one of the four linear operators that build numerical implementations of the methods in chapters 3-5. The other three operators are namely the adjoint $(\nabla T \cdot \nabla)^T$, the inverse $(\nabla T \cdot \nabla)^{-1}$, and finally adjoint of the inverse $(\nabla T \cdot \nabla)^{-T}$.

The upwind stencil 2.1 is first-order accurate. It is possible to construct high-order upwind stencils in the same fashion as 2.1, as long as the stencil is strictly one-sided (Franklin and Harris, 2001). For instance, a third-order upwind stencil reads:

$$D^+_x T^k_{i,j} = \frac{2T^k_{i+3,j} - 9T^k_{i+2,j} + 18T^k_{i+1,j} - 11T^k_{i,j}}{6\Delta x}\ , \tag{2.4}$$

where the coefficients are determined by Taylor series. In my thesis I adopt only the first-order stencil.

For a Cartesian ordering of $T^k_{i,j}$, the discretized operators $D_m T \cdot D_m$ are matrices. Thanks to upwind finite-differences, these matrices are sparse and contain only two non-zero entries per row. For instance, suppose $T^k_{i,j}$ has its upwind neighbor in z

at $T_{i,j}^{k-1}$, then

$$D_z T \cdot D_z = \begin{bmatrix} \ddots & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & -\kappa_z & & \kappa_z & \\ & & & & \ddots \end{bmatrix}, \tag{2.5}$$

where

$$\kappa_z \equiv \frac{D_z T_{i,j}^k}{\Delta z} = \frac{T_{i,j}^k - T_{i,j}^{k-1}}{(\Delta z)^2} \ . \tag{2.6}$$

Definitions of κ_x and κ_y follow their upwind approximations, respectively. In equation 2.5, $\pm\kappa_z$ are located in the same row as that of $T_{i,j}^k$. While κ_z sits on the diagonal, $-\kappa_z$ has a column index equals to the row of $T_{i,j}^{k-1}$. At $T = 0$, there is no upwind neighbor and the corresponding row contains all zeros. It is evident that the cost of applying $\nabla T \cdot \nabla$ to a vector is $O(N)$.

DERIVATION AND IMPLEMENTATION OF THE LINEAR OPERATORS

According to equation 2.3, computing δw for given T and δT amounts to simple sparse matrix-vector multiplication. The ordering of $T_{i,j}^k$, $\delta T_{i,j}^k$ and $\delta w_{i,j}^k$ is not important. However, if we change $T_{i,j}^k$ in equation 2.5 from ordinary Cartesian ordering to *upwind ordering*, we effectively perform column-wise permutations to $D_m T \cdot D_m$. The resulting matrices are not only sparse but also lower triangular. A pre-sort of T here is not necessary if it is computed by FMM, because such an upwind ordering is maintained and updated by the priority queue during eikonal solving and thus can be conveniently imported for usage. From another point of view, this means that the linear operator $\nabla T \cdot \nabla$ is consistent with FMM. In implementing $\nabla T \cdot \nabla$, both

the sparsity and triangularization should be fully exploited for optimum performance.

The next operator is the inverse $(\nabla T \cdot \nabla)^{-1}$ appearing in traveltime tomography:

$$\frac{1}{2}\left(\nabla T \cdot \nabla\right)^{-1} \delta w = \delta T \ . \tag{2.7}$$

In order to find the expression for inverse, let us further assume the upwind neighbors of $T^k_{i,j}$ in x and y are $T^k_{i-1,j}$ and $T^k_{i,j-1}$, respectively. Then with notations 2.6, I re-write equation 2.3 as

$$2\kappa_x \left(\delta T^k_{i,j} - \delta T^k_{i-1,j}\right) + 2\kappa_y \left(\delta T^k_{i,j} - \delta T^k_{i,j-1}\right) + 2\kappa_z \left(\delta T^k_{i,j} - \delta T^{k-1}_{i,j}\right) = \delta w^k_{i,j} \ , \tag{2.8}$$

which in a matrix-vector form reads

$$\begin{bmatrix} \ddots & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ -2\,\kappa_x & -2\,\kappa_y & -2\,\kappa_z & 2\left(\kappa_x+\kappa_y+\kappa_z\right) & & \\ & & & & \ddots & \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \delta T \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \delta w \\ \vdots \end{bmatrix} \ . \tag{2.9}$$

After regrouping the terms, equation 2.8 gives

$$\delta T^k_{i,j} = \frac{2\kappa_x\, \delta T^k_{i-1,j} + 2\kappa_y\, \delta T^k_{i,j-1} + 2\kappa_z\, \delta T^{k-1}_{i,j} + \delta w^k_{i,j}}{2\left(\kappa_x+\kappa_y+\kappa_z\right)} \ . \tag{2.10}$$

Equation 2.10 means the linear operator $(\nabla T \cdot \nabla)^{-1}$ does not need to be computed by an explicit matrix inversion as shown in equation 2.9. Instead, its application to a vector can be performed by a single sweep based on upwind ordering of T. Note that unlike the implementations of $\nabla T \cdot \nabla$ where upwind ordering is optional, $(\nabla T \cdot \nabla)^{-1}$ must be carried out sequentially so that $\delta T^k_{i,j}$ could access its own neighbors at exactly the positions prescribed by the upwind stencils 2.2. For the singularity at $T = 0$, we

can set entries of the corresponding row to zero.

The third operator to study is the adjoint $(\nabla T \cdot \nabla)^T$. Let us consider

$$\frac{1}{2}\left(\nabla T \cdot \nabla\right)^T \delta w = \delta T \,. \tag{2.11}$$

A direct matrix transpose of equation 2.5 gives

$$\left(D_z T \cdot D_z\right)^T = \begin{bmatrix} \ddots & & & & \\ & \ddots & & -\kappa_z & \\ & & \ddots & & \\ & & & \kappa_z & \\ & & & & \ddots \end{bmatrix} . \tag{2.12}$$

Now $-\kappa_z$ has a row index equals to that of $T_{i,j}^{k-1}$. It means in equation 2.11 δT relies on its *downwind* neighbors. The same analysis applied to $(D_x T \cdot D_x)^T$, $(D_y T \cdot D_y)^T$ and consequently $(\nabla T \cdot \nabla)^T$. It might be more straightforward to draw this conclusion by noticing the fact that the lower triangularized $\nabla T \cdot \nabla$ after upwind sorting becomes upper triangularized by transposition.

The last operator of interest is adjoint of the inverse $(\nabla T \cdot \nabla)^{-T}$. Based on equation 2.9, the matrix-vector form of

$$\frac{1}{2}\left(\nabla T \cdot \nabla\right)^{-T} \delta T = \delta w \tag{2.13}$$

is

$$\begin{bmatrix} \ddots & & & & \\ & \ddots & & -2\,\kappa_x & \\ & & \ddots & -2\,\kappa_y & \\ & & & -2\,\kappa_z & \\ & & & 2\left(\kappa_x+\kappa_y+\kappa_z\right) & \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \delta w \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \delta T \\ \vdots \end{bmatrix} \tag{2.14}$$

As an upper triangular matrix-vector multiplication, the value of $\delta w_{i,j}^k$ depends on the input $\delta T_{i,j}^k$ as well as its neighbors in T's downwind directions. Similarly to $(\nabla T \cdot \nabla)^T$, I solve equation 2.14 by following the downwind ordering of T.

CONCLUSION

I show derivations and implementations of four linear operators by upwind finite-differences. These operators are consistent with FMM in the sense that their applications to a vector are performed sequentially by either upwind ordering or downwind ordering. It is possible, however, to supply a background traveltime table from another eikonal solver other than FMM. If so, the extra cost of background traveltime sorting can not be avoided. The cost of sorting with the quicksort algorithm is $O(N \log N)$ (Knuth, 2011).

My implementation of these linear operators consists of two major components. One is the upwind (downwind) ordering that is critical in $(\nabla T \cdot \nabla)^{-1}$ and $(\nabla T \cdot \nabla)^{-T}$. The other is upwind stencil 2.2, for which I store only the sparse coefficients. All operators have a low computational complexity of $O(N)$, and can be further combined to construct more involved operators.

Chapter 3

First-break traveltime tomography with the double-square-root eikonal equation

INTRODUCTION

The first-break traveltime tomography (Zhu et al., 1992; Osypov, 2000; Leung and Qian, 2006; Taillandier et al., 2009; Noble et al., 2010) is an established tool for estimating near-surface macro-feature seismic velocities. Starting from a prior model, tomographic inversion gradually modifies the velocities such that the misfits between predicted and observed first-breaks decrease. Since the problem is nonlinear, several linearization iterations may be required until convergence. Moreover, inversion must be carried out with careful choice of regularization in order to avoid local minima (Stefani, 1993; Simmons and Bernitsas, 1994). The estimated model has a direct influence on subsequent applications, such as static corrections (Marsden, 1993; Cox, 1999; Bergman et al., 2004) where it provides a medium-to-long wavelength near-surface model, and waveform tomography (Sheng et al., 2006; Brenders and Pratt, 2007; Virieux and Operto, 2009) where it serves as a low-frequency prior.

The traditional first-break traveltime tomography is based on the eikonal equa-

Parts of this chapter were published in Li, S., A. Vladimirsky, and S. Fomel, 2013, First-break traveltime tomography with the double-square-root eikonal equation: Geophysics, **78**, no. 6, U89-U101. This work started during Vladimirsky's visit to Austin during his sabbatical in 2011 and was then done under the supervision of Fomel.

tion that arises from high-frequency approximation of the wave equation (Chapman, 2002). During forward modeling, the first-breaks computed through the eikonal equation are naturally shot-indexed because only receiver coordinates move while the source is fixed. At tomography stage, one may formulate the minimization of cost function as a sequence of explicitly linearized problems or directly as a nonlinear optimization problem. The first choice (Zelt and Barton, 1998; Zhu et al., 2000; Dessa et al., 2004; Pei, 2009) requires computation of Fréchet derivatives, which is usually carried out by combining an eikonal solver with posterior ray tracing. Then an algorithm such as conjugate gradients (Hestenes and Stiefel, 1952) or LSQR (Paige and Saunders, 1982) is applied to solve the linearized tomographic system iteratively. While this approach accounts for information from both source and receiver dimensions, it faces computational limitations when the Fréchet derivative matrix becomes difficult to handle because of a large number of model parameters. The nonlinear optimization approach, on the other hand, can be combined with the adjoint-state method (Plessix, 2006) and avoids an explicit computation of Fréchet derivatives (Taillandier et al., 2009). The cost of computing the gradient is equivalent to twice the solution of the forward modeling problem, regardless of the size of input data. However, one major drawback of this approach, as I will show later, is that the resulting gradient disregards information available along the shot dimension.

The drawback of eikonal-based adjoint-state tomographies is that they always face conflicting information that propagates across different shots. Such conflicts must be resolved during inversion, or else an erroneous model update may appear. In practice, the inversion may be less robust and may take more iterations to converge, compared with the situation where we replace the eikonal equation with another

governing equation that allows both source and receiver positions to change along ray-paths. The double-square-root (DSR) eikonal equation is a promising candidate in this regard, because it describes the prestack data as a whole by linking the evolution of traveltimes to both sub-surface source and receiver positions. In this chapter, I investigate the feasibility of using the DSR eikonal equation for first-break traveltime tomography with the adjoint-state method.

The DSR eikonal was analyzed previously by Belonosova and Alekseev (1974), Duchkov and de Hoop (2010) and Alkhalifah (2011b). Ray-tracing methods applied to DSR are capable of providing multi-arrivals by extrapolating isochron rays (Iversen, 2004) or using perturbation theory, but their extra cost in computing non-first-breaks is not necessary for first-break tomography purposes. Li et al. (2013) prove that an implicit discretization of the DSR eikonal equation is causal and thus can be solved by a Dijkstra-like non-iterative method (Dijkstra, 1959). The DSR singularity and two DSR branches that are non-causal need special treatment.

My current implementation employs a modified fast-marching method (FMM) (Sethian, 1999) DSR eikonal solver. I first test its accuracy by DSR forward modeling. Next, I linearize the DSR eikonal equation and use the resulting operators in adjoint-state tomography. For comparison, I apply an analogous linearization and adjoint-state formulation to the traditional tomography based on shot-indexed eikonal equation. Then I demonstrate the differences between the proposed and traditional approaches and justify advantages of the new method using several synthetic model examples. I conclude by discussing possible further improvements and extensions of the proposed method.

THEORY

DSR eikonal equation

I continue here from the discussion on DSR eikonal equation and Figure 1.8 in Chapter 1. The negative signs before the two square-roots in equation 1.9 correspond to a decrease of traveltime with increasing depth, or, geometrically, a *downward* pointing of slowness vectors on both s and r sides. Since the slowness vectors could also be pointing *upward* and the directions may be different at r and s, the DSR eikonal equation (Belonosova and Alekseev, 1974) should account for all the possibilities (Figure 3.1):

$$\frac{\partial T}{\partial z} = \pm\sqrt{\frac{1}{v^2(z,s)} - \left(\frac{\partial T}{\partial s}\right)^2} \pm \sqrt{\frac{1}{v^2(z,r)} - \left(\frac{\partial T}{\partial r}\right)^2} \ . \tag{3.1}$$

The boundary condition for DSR eikonal equation is that traveltimes at the subsurface zero-offset plane, i.e. $r = s$, are zero: $T(z, r = s) = 0$.

Equation 3.1 has a *singularity* when $\partial T/\partial z = 0$, in which case the slowness vectors at s and r sides are both horizontal and equation 3.1 reduces to

$$\left(\frac{\partial T}{\partial s}\right)^2 = \frac{1}{v^2(z,s)} \ ; \ \ \left(\frac{\partial T}{\partial r}\right)^2 = \frac{1}{v^2(z,r)} \ . \tag{3.2}$$

The two independent equations in 3.2 are not in conflict according to the source-receiver reciprocity, because they are the same with an exchange of s and r.

Note that equations 3.1 and 3.2 describe T in full prestack domain (z, r, s) by allowing not only receivers but also sources to change positions. In contrast, the

eikonal equation 1.4 with $\mathbf{x} = (z, x)$ and boundary condition $T(z = 0, x = s) = 0$ can be used only for one fixed source position at a time and thus traveltimes of different shots are independent of each other. Here s is *surface* source lateral position. For 3-D model, prestack traveltime is in a 5-D space. The analysis in this chapter is restricted to 2-D for simplicity. I consider the 3-D extension in the Discussion section.

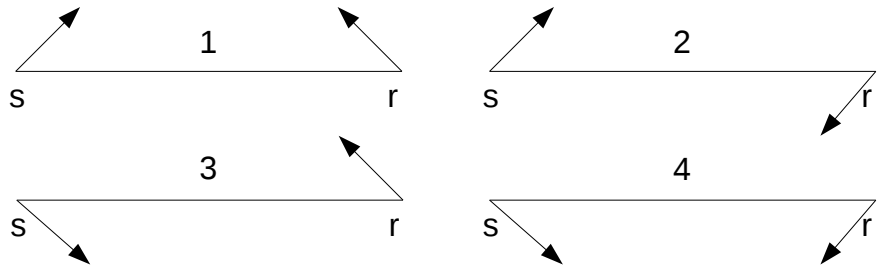


Figure 3.1: All four branches of DSR eikonal equation from different combination of upward or downward pointing of slowness vectors. Whether the slowness vector is pointing leftward or rightward does not matter because the partial derivatives with respect to s and r in equation 3.1 are squared. Figure 1.8 and equation 1.9 belong to the last situation. dsrtomo/figs root

Similarly to the eikonal equation, the DSR eikonal equation is a nonlinear first-order partial differential equation. Its solutions include in general not only first-breaks but all arrivals, and can be computed by solving separate eikonal equations for each sub-surface source-receiver pair followed by extracting the traveltime and putting its values into prestack volume (Iversen, 2004; Duchkov and de Hoop, 2010; Serdyukov and Duchkov, 2013). However, such an implementation is impractical due to the large amount of computations. Meanwhile, for first-break tomography purposes, we are only interested in the first-arrival solutions but require an efficient and

accurate algorithm. In this regard, a finite-difference DSR eikonal solver analogous to the fast-marching (Sethian, 1999) or fast-sweeping (Zhao, 2005) eikonal solvers is preferable.

Causal discretization

In upwind discretizations of the DSR eikonal equation on the grid in (z, r, s) domain, one has to make a decision about the z-slice, in which the finite differences are taken to approximate $\partial T/\partial s$ and $\partial T/\partial r$. In Figure 1.8, it appears natural to approximate these partial derivatives in the z-slice below $T(z, r, s)$. I refer to the corresponding scheme as *explicit*, since it allows to directly compute the grid value $T(z, r, s)$ based on the already known T values from the next-lower z. Assuming that T^i is the upwind neighbor of T in the i's direction for $i = z, r, s$, Figure 3.3 illustrates the explicit scheme. An alternative *implicit* scheme is obtained by approximating $\partial T/\partial s$ and $\partial T/\partial r$ in the same z-slice as $T(z, r, s)$ as shown in Figure 3.2, which results in a coupled system of nonlinear discretized equations.

It is necessary to study the properties of both explicit and implicit discretizations and decide which one is more appropriate for DSR forward modeling. Li et al. (2013) proved the following:

1. The explicit scheme is very efficient to use on a fixed grid, but only conditionally convergent. This property is also confirmed numerically in Synthetic Model Examples section.

2. The implicit scheme is *monotone causal*, meaning $T(z, r, s)$ depends on the

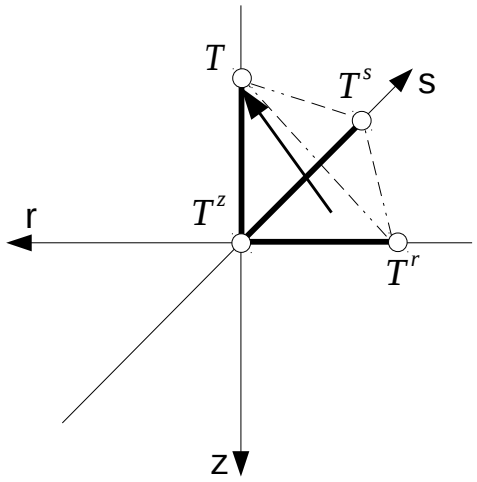


Figure 3.2: An explicit discretization scheme. Compare with Figure 3.3. The arrow depicts a DSR characteristic with its root confined in the simplex $T^zT^rT^s$. dsrtomo/figs update2t

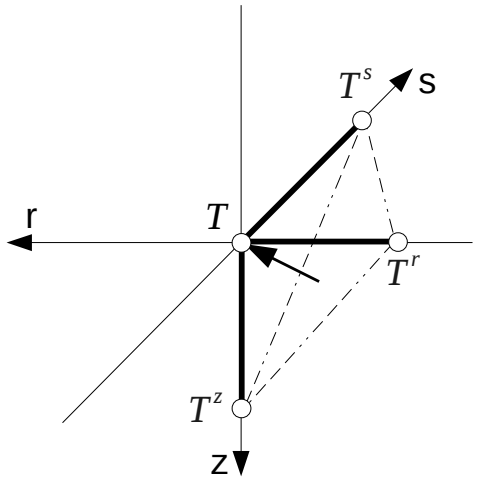


Figure 3.3: An implicit discretization scheme. The arrow indicates a DSR characteristic. Its root is located in the simplex $T^zT^rT^s$. dsrtomo/figs update1

smaller neighboring grid values only. This enables me to apply a Dijkstra-like method (Dijkstra, 1959) to solve the discretized system efficiently. Importantly, the DSR singularity requires a special ordering in the selection of upwind neighbors, which switches between equations 1.9 and 3.2 when necessary. I provide a modified fast-marching (Sethian, 1999) DSR eikonal solver along with such an ordering strategy in Implementation section.

3. The causality analysis in Li et al. (2013) applies only to the first and last *causal branches* out of all four shown in Figure 3.1. Additional post-processings, albeit expensive, can be used to recover the rest two non-causal branches as they may be decomposed into summations of the causal ones.

In practice, I find that, for moderate velocity variations, the first-breaks correspond only to causal branches. An example in Synthetic Model Examples section serves to illustrate this observation. Therefore, for efficiency, I turn off the non-causal branch post-processings in forward modeling and base the tomography solely on equations 1.9 and 3.2.

DSR tomography

The first-break traveltime tomography with DSR eikonal equation (DSR tomography) can be established by following a procedure analogous to the traditional one with the shot-indexed eikonal equation (standard tomography). To further reveal their differences, in this section I will derive both approaches.

For convenience, I use slowness-squared w defined in equation 1.4 instead of

velocity v in equations 1.9 and 3.2. Based on the causal analysis, the velocity model $w(z,x)$ and prestack cube $T(z,r,s)$ are Eulerian discretized and arranged as column vectors $\mathbf{w}$ of size $nz \times nx$ and $\mathbf{t}$ of size $nz \times nx \times nx$. I denote the observed first-breaks by $\mathbf{t}^{obs}$, and use $\mathbf{t}^{std}$ and $\mathbf{t}^{dsr}$ whenever necessary to discriminate between $\mathbf{t}$ computed from shot-indexed eikonal equation and DSR eikonal equation.

The tomographic inversion seeks to minimize the l_2 (least-squares) norm of the data residuals. I define an objective function as follows:

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{t} - \mathbf{t}^{obs})^T(\mathbf{t} - \mathbf{t}^{obs}) \ , \tag{3.3}$$

where the superscript T represents transpose. A Newton method of inversion can be established by considering an expansion of the misfit function 3.3 in a Taylor series and retaining terms up to the quadratic order (Bertsekas, 1982):

$$E(\mathbf{w} + \delta\mathbf{w}) = E(\mathbf{w}) + \delta\mathbf{w}^T\nabla_w E(\mathbf{w}) + \frac{1}{2}\delta\mathbf{w}^T\mathbf{H}(\mathbf{w})\delta\mathbf{w} + O(|\delta\mathbf{w}|^3) \ . \tag{3.4}$$

Here $\nabla_w E$ and $\mathbf{H}$ are gradient vector and Hessian matrix, respectively. I may evaluate the gradient by taking partial derivatives of equation 3.3 with respect to $\mathbf{w}$, yielding

$$\nabla_w E \equiv \frac{\partial E}{\partial \mathbf{w}} = \mathbf{J}^T(\mathbf{t} - \mathbf{t}^{obs}) \ , \tag{3.5}$$

where $\mathbf{J}$ is the Fréchet derivative matrix and can be found by further differentiating $\mathbf{t}$ with respect to $\mathbf{w}$.

I start by deriving the Fréchet derivative matrix of standard tomography, which is based on the linearized eikonal equation 1.8. Denoting

$$\hat{D}_m \equiv \frac{\partial}{\partial m} \ ; \ \ m = z, x, r, s \tag{3.6}$$

as the partial derivative operator in the m's direction, equation 1.4 can be re-written as

$$\hat{D}_z t_k^{std} \cdot \hat{D}_z t_k^{std} + \hat{D}_x t_k^{std} \cdot \hat{D}_x t_k^{std} = w \ ; \ \ k = 1, 2, 3, ..., nx \ . \tag{3.7}$$

Here I assume that there are in total nx shots and use t_k^{std} for first-breaks of the kth shot. Applying $\partial/\partial w$ to both sides of equation 3.7, I find

$$J_k^{std} \equiv \frac{\partial t_k^{std}}{\partial w} = \frac{1}{2}(\hat{D}_z t_k^{std} \cdot \hat{D}_z + \hat{D}_x t_k^{std} \cdot \hat{D}_x)^{-1} \ ; \ \ k = 1, 2, 3, ..., nx \ . \tag{3.8}$$

Kinematically, each J_k^{std} contains characteristics of the kth shot. Because shots are independent of each other, the full Fréchet derivative is a concatenation of individual J_k^{std}, as follows:

$$J^{std} = \begin{bmatrix} J_1^{std} & J_2^{std} & \cdots & J_{nx}^{std} \end{bmatrix}^T \ . \tag{3.9}$$

Inserting equation 3.9 into equation 3.5, I obtain

$$\nabla_w E = \sum_{k=1}^{nx} \left(\mathbf{J}_k^{std} \right)^T (\mathbf{t}_k^{std} - \mathbf{t}_k^{obs}) \ , \tag{3.10}$$

where, similar to t_k^{std}, t_k^{obs} stands for the observed first-breaks of the kth shot.

Figure 3.4 illustrates equation 3.10 schematically, i.e. the gradient produced by standard tomography. The first step on the left depicts the transpose of the kth Fréchet derivative acting on the corresponding kth data residual. It implies a back-projection that takes place in the $z - r$ plane of a fixed s position. The second step on the right is simply the summation operation in equation 3.10.

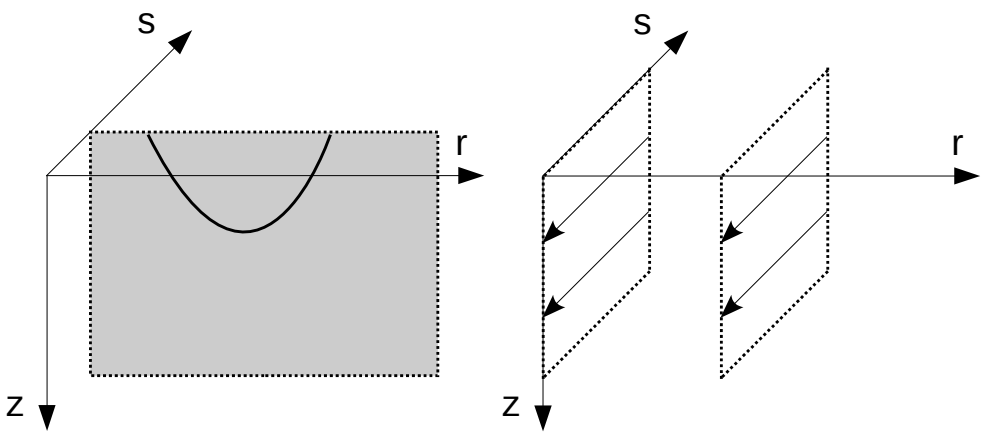


Figure 3.4: The gradient produced by standard tomography. The solid curve indicates a shot-indexed characteristic. dsrtomo/figs cartonstd

To derive the Fréchet derivative matrix associated with DSR tomography, I first re-write equation 1.9 with definition 3.6

$$\hat{D}_z t^{dsr} = -\sqrt{w_s - \hat{D}_s t^{dsr} \cdot \hat{D}_s t^{dsr}} - \sqrt{w_r - \hat{D}_r t^{dsr} \cdot \hat{D}_r t^{dsr}} \,, \tag{3.11}$$

where w_s and w_r are w at sub-surface source and receiver locations, respectively. Note that in equation 3.11 w appears twice. Thus a differentiation of t^{dsr} with respect to w must be carried out through the chain-rule:

$$J^{dsr} \equiv \frac{\partial t^{dsr}}{\partial w} = \left.\frac{\partial t^{dsr}}{\partial w_s}\right|_{w_r} \frac{\partial w_s}{\partial w} + \left.\frac{\partial t^{dsr}}{\partial w_r}\right|_{w_s} \frac{\partial w_r}{\partial w} \,. \tag{3.12}$$

I recall that w and t^{dsr} are of different lengths. Meanwhile in equation 3.11, both w_s and w_r have the size of t^{dsr}. Clearly in equation 3.12 $\partial w_s/\partial w$ and $\partial w_r/\partial w$

must achieve dimensionality enlargement. In fact, according to Figure 1.8, w_s and w_r can be obtained by spraying w such that $w_s(z,r,s) = w(z,s)$ and $w_r(z,r,s) = w(z,r)$. Therefore, $\partial w_s/\partial w$ and $\partial w_r/\partial w$ are essentially spraying operators and their adjoints perform stackings along s and r dimensions, respectively.

To derive the Fréchet derivative, I start from equations 3.11 and 3.12. Applying $\partial/\partial w_s$ to both sides of equation 3.11 results in

$$\begin{aligned}\hat{D}_z \frac{\partial t^{dsr}}{\partial w_s} &= -\frac{1}{2\sqrt{w_s - \hat{D}_s t^{dsr} \cdot \hat{D}_s t^{dsr}}} \\ &+ \left(\frac{\hat{D}_s t^{dsr} \cdot \hat{D}_s}{\sqrt{w_s - \hat{D}_s t^{dsr} \cdot \hat{D}_s t^{dsr}}} + \frac{\hat{D}_r t^{dsr} \cdot \hat{D}_r}{\sqrt{w_r - \hat{D}_r t^{dsr} \cdot \hat{D}_r t^{dsr}}} \right) \frac{\partial t^{dsr}}{\partial w_s} \ . \quad (3.13)\end{aligned}$$

Analogously

$$\begin{aligned}\hat{D}_z \frac{\partial t^{dsr}}{\partial w_r} &= -\frac{1}{2\sqrt{w_r - \hat{D}_r t^{dsr} \cdot \hat{D}_r t^{dsr}}} \\ &+ \left(\frac{\hat{D}_s t^{dsr} \cdot \hat{D}_s}{\sqrt{w_s - \hat{D}_s t^{dsr} \cdot \hat{D}_s t^{dsr}}} + \frac{\hat{D}_r t^{dsr} \cdot \hat{D}_r}{\sqrt{w_r - \hat{D}_r t^{dsr} \cdot \hat{D}_r t^{dsr}}} \right) \frac{\partial t^{dsr}}{\partial w_r} \ . \quad (3.14)\end{aligned}$$

Inserting equations 3.13 and 3.14 into 3.12 and regrouping the terms, I prove

$$J^{dsr} = B^{-1}(C_s + C_r) \ , \quad (3.15)$$

where

$$B = \hat{D}_z - \left(\frac{\hat{D}_s t^{dsr} \cdot \hat{D}_s}{\sqrt{w_s - \hat{D}_s t^{dsr} \cdot \hat{D}_s t^{dsr}}} \right) - \left(\frac{\hat{D}_r t^{dsr} \cdot \hat{D}_r}{\sqrt{w_r - \hat{D}_r t^{dsr} \cdot \hat{D}_r t^{dsr}}} \right) \ , \quad (3.16)$$

and

$$C_s = -\frac{1}{2\sqrt{w_s - \hat{D}_s t^{dsr} \cdot \hat{D}_s t^{dsr}}} \frac{\partial w_s}{\partial w} \ ; \quad (3.17)$$

$$C_r = -\frac{1}{2\sqrt{w_r - \hat{D}_r t^{dsr} \cdot \hat{D}_r t^{dsr}}} \frac{\partial w_r}{\partial w} \ . \tag{3.18}$$

At the singularity of DSR eikonal equation, the operators B, C_s and C_r take simpler forms and can be derived directly from equation 3.2. Combining equations 3.5 and 3.15 results in

$$\nabla_w E = \left(\mathbf{C}_s^T + \mathbf{C}_r^T\right) \mathbf{B}^{-T} (\mathbf{t}^{dsr} - \mathbf{t}^{obs}) \ . \tag{3.19}$$

Note that unlike equation 3.10, equation 3.19 can not be expressed as an explicit summation over shots.

Figure 3.5 shows the gradient of DSR tomography. Similarly to the standard tomography, the gradient produced by equation 3.19 is a result of two steps. The first step on the left is a back-projection of prestack data residuals according to the adjoint of operator B^{-1}. Because B contains DSR characteristics that travel in prestack domain, this back-projection takes place in (z, r, s) and is different from that in standard tomography, although the data residuals are the same for both cases. The second step on the right follows the adjoint of operators C_s and C_r and reduces the dimensionality from (z, r, s) to (z, x). However, compared to standard tomography this step involves summations in not only s but also r.

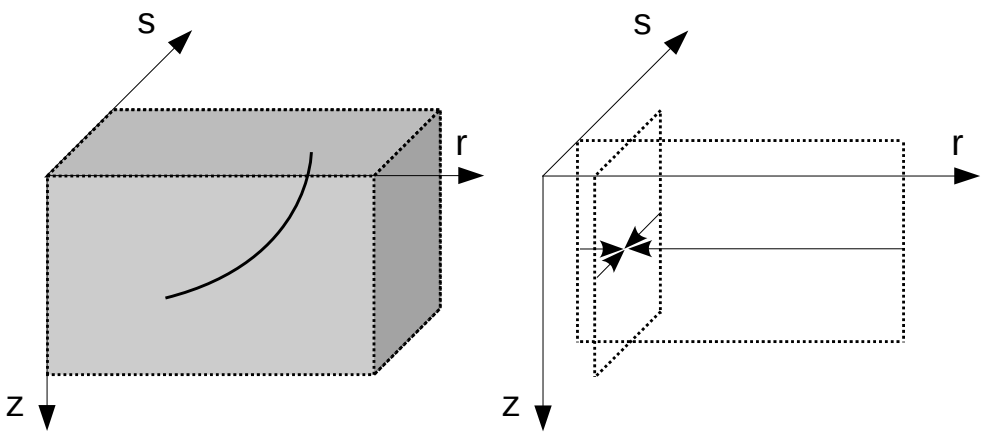


Figure 3.5: The gradient produced by DSR tomography. The solid curve indicates a DSR characteristic, which has one end in plane $z = 0$ and the other in plane $s = r$. Compare with Figure 3.4. dsrtomo/figs cartondsr

IMPLEMENTATION

Forward modeling

Following the analysis by Li et al. (2013), I consider an implicit Eulerian discretization. The uniform grid spacing of $v(z, x)$ is Δz and Δx. s and r are discretized in the same way as x, i.e. $\Delta s = \Delta r = \Delta x$. For forward modeling, I solve the DSR eikonal equation by a version of FMM. The source is a plane-wave at subsurface zero-offset $r = s$. To properly handle the DSR singularity, I design an ordering of the combination of upwind neighbors in the FMM local update scheme. Following the notations in Figure 3.3, I summarize the ordering as follows:

1. First try a three-sided update:

Solve the following equation

$$\frac{T-T^z}{\Delta z} = \sqrt{\frac{1}{v_r^2} - \left(\frac{T-T^r}{\Delta r}\right)^2} + \sqrt{\frac{1}{v_s^2} - \left(\frac{T-T^s}{\Delta s}\right)^2} \tag{3.20}$$

and return T if $T \geq \max(T^z, T^r, T^s)$;

2. Next try a two-sided update: solve equations

$$T = \min\left(T^r + \frac{\Delta r}{v_r}, T^s + \frac{\Delta s}{v_s}\right) , \tag{3.21}$$

$$\frac{T-T^z}{\Delta z} = \sqrt{\frac{1}{v_r^2} - \left(\frac{T-T^r}{\Delta r}\right)^2} + \frac{1}{v_s} , \tag{3.22}$$

$$\frac{T-T^z}{\Delta z} = \frac{1}{v_r} + \sqrt{\frac{1}{v_s^2} - \left(\frac{T-T^s}{\Delta s}\right)^2} , \tag{3.23}$$

and keep the results as T_{rs}, T_{zr} and T_{zs}, respectively.

If $T_{zr} \geq \max(T^z, T^r)$ and $T_{zs} \geq \max(T^z, T^s)$, return $\min(T_{zr}, T_{zs}, T_{rs})$;

If $T_{zr} < \max(T^z, T^r)$ and $T_{zs} \geq \max(T^z, T^s)$, return $\min(T_{zs}, T_{rs})$;

If $T_{zr} \geq \max(T^z, T^r)$ and $T_{zs} < \max(T^z, T^s)$, return $\min(T_{zr}, T_{rs})$;

3. Finally try a one-sided update:

Solve equation

$$\frac{T-T^z}{\Delta z} = \frac{1}{v_r} + \frac{1}{v_s} . \tag{3.24}$$

and return $\min(T, T_{rs})$.

An optional search routine may be added after the update to recover all branches of the DSR eikonal equation (Li et al., 2013):

$$T(z,r,s) = \min_{q \in (s,r)} \{T(z,q,s) + T(z,r,q)\} . \tag{3.25}$$

The overall cost can be reduced roughly by half by acknowledging the source-receiver reciprocity and thus computing only the positive (or negative) subsurface offset region.

On the other hand, a DSR forward modeling with the explicit scheme in Figure 3.2 does not require solving a polynomial equation such as the three-sided implicit scheme 3.20. Instead, it can be solved in a single sweep in the $-z$ direction:

$$\frac{T-T^z}{\Delta z} = \sqrt{\frac{1}{v_r^2} - \left(\frac{T^z-T^r}{\Delta r}\right)^2} + \sqrt{\frac{1}{v_s^2} - \left(\frac{T^z-T^s}{\Delta s}\right)^2}\ , \tag{3.26}$$

Tomographic inversion

For an implementation of linearized tomographic operators 3.10 and 3.19, I choose upwind approximations 2.2 for the difference operators in equation 3.6. Operator 3.8 after upwind finite-differences includes exactly $(\nabla T \cdot \nabla)^{-1}$ discussed in Chapter 2. Because the upwind finite-differences result in triangularization of matrices 3.9 and 3.15, the costs of applying $\mathbf{J}^{std}$ and $\mathbf{J}^{dsr}$ and their transposes are inexpensive. Moreover, although our implementation belongs to the family of adjoint-state tomographies, I do not need to compute the adjoint-state variable as an intermediate product for the gradient.

Additionally, the Gauss-Newton approach approximates the Hessian in equation 3.4 by $\mathbf{H} \approx \mathbf{J}^T\mathbf{J}$. An update $\delta\mathbf{w}$ at current $\mathbf{w}$ is found by taking derivative of equation 3.4 with respect to $\delta\mathbf{w}$, which results in the following normal equation:

$$\delta\mathbf{w} = \left[\mathbf{J}^T\mathbf{J}\right]^{-1}\mathbf{J}^T(\mathbf{t}^{obs}-\mathbf{t})\ . \tag{3.27}$$

To add model constraints, I combine equation 3.27 with Tikhonov regularization (Tikhonov, 1963) with the gradient operator and use the method of conjugate gradients (Hestenes and Stiefel, 1952) to solve for the model update $\delta\mathbf{w}$.

EXAMPLES

The numerical examples in this section serve several different purposes. The first example will test the accuracy of modified FMM DSR eikonal equation solver (DSR FMM) and show the drawbacks of the alternative explicit discretization. The second example will demonstrate effect of considering non-causal branches of DSR eikonal equation in forward modeling. The third example will compare the sensitivity kernels of DSR tomography and standard tomography in a simple model. The fourth example will present a tomographic inversion and demonstrate advantages of DSR method over the traditional method. Finally the proposed method is applied to the synthetic dataset introduced in Chapter 1.

Comparison between implicit and explicit discretizations

Figure 4.6 shows a 2-D velocity model with a constant-velocity-gradient background plus a Gaussian anomaly in the middle. The traveltimes on the surface $z = 0$ km of a shot at $(0, 0)$ km are computed by DSR FMM at a gradually refined Δz or Δx while fixing the other one. For reference, I also calculate first-breaks by a second-order FMM (Rickett and Fomel, 1999; Popovici and Sethian, 2002) for the same shot at a very fine grid spacing of $\Delta z = \Delta x = 1$ m. In Figure 3.7, a grid refinement in both Δz and Δx helps reducing errors of the implicit discretization, although improvements in

the Δz refinement case are less significant because the majority of the ray-paths are non-horizontal. The results are consistent with the causal analysis in Li et al. (2013), which shows that the implicit discretization is unconditionally convergent. On the other hand, as shown in Figure 3.8, the explicit discretization is only conditionally convergent when $\Delta z/\Delta x \to 0$ under grid refinement in order to resolve the flatter parts of the ray-paths. This explains why its accuracy deteriorates when refining Δx and fixing Δz.

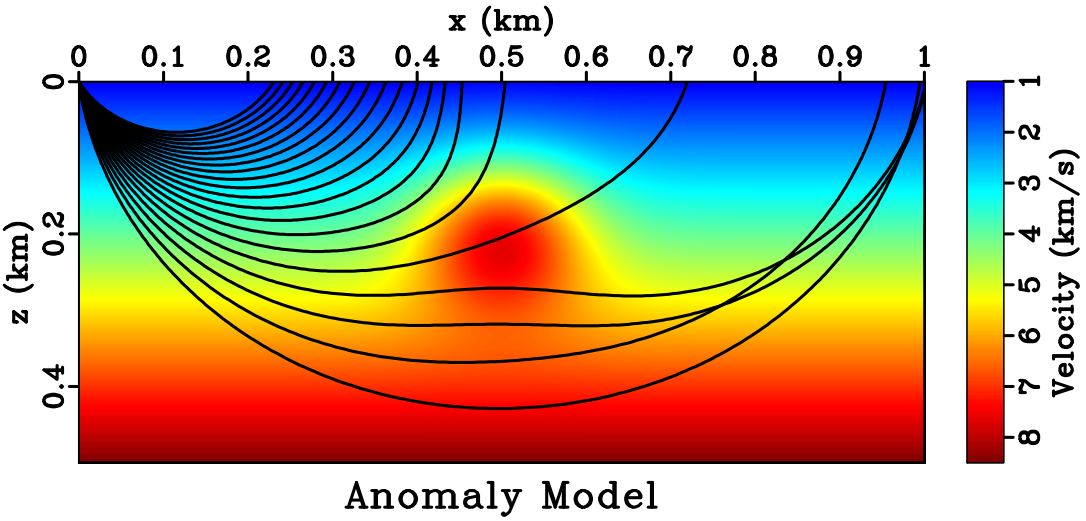


Figure 3.6: The synthetic model used for DSR FMM accuracy test. The overlaid curves are rays traced from a shot at $(0, 0)$ km. dsrtomo/accuracy modl

Effects of non-causal DSR branches

Next, I use a smoothed Marmousi model (Figure 3.9) and run two DSR FMMs, one with the search process for non-causal DSR branches turned-on and the other turned-off. In Figure 3.10, again I compute reference values by a second-order FMM. The three groups of curves are traveltimes of shots at $(0, 0)$ km, $(0.75, 0)$ km and

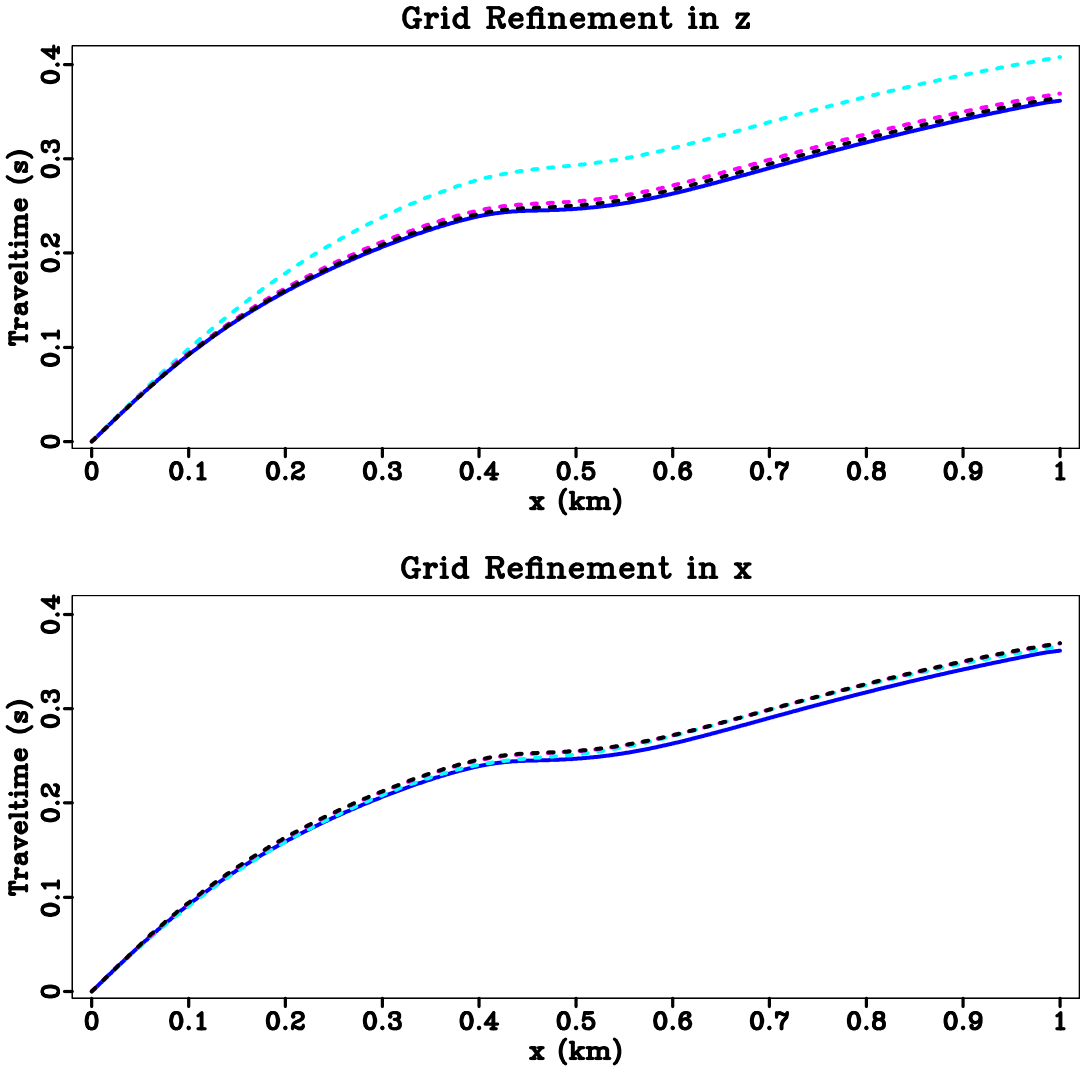


Figure 3.7: Grid refinement experiment (implicit discretization). In both figures, the solid blue curve is the reference values and the dashed curves are computed by DSR FMM. Top: fixed $\Delta x = 10$ m and $\Delta z = 50$ m (cyan), 10 m (magenta), 5 m (black). Bottom: fixed $\Delta z = 10$ m and $\Delta x = 50$ m (cyan), 10 m (magenta), 5 m (black). dsrtomo/accuracy imp

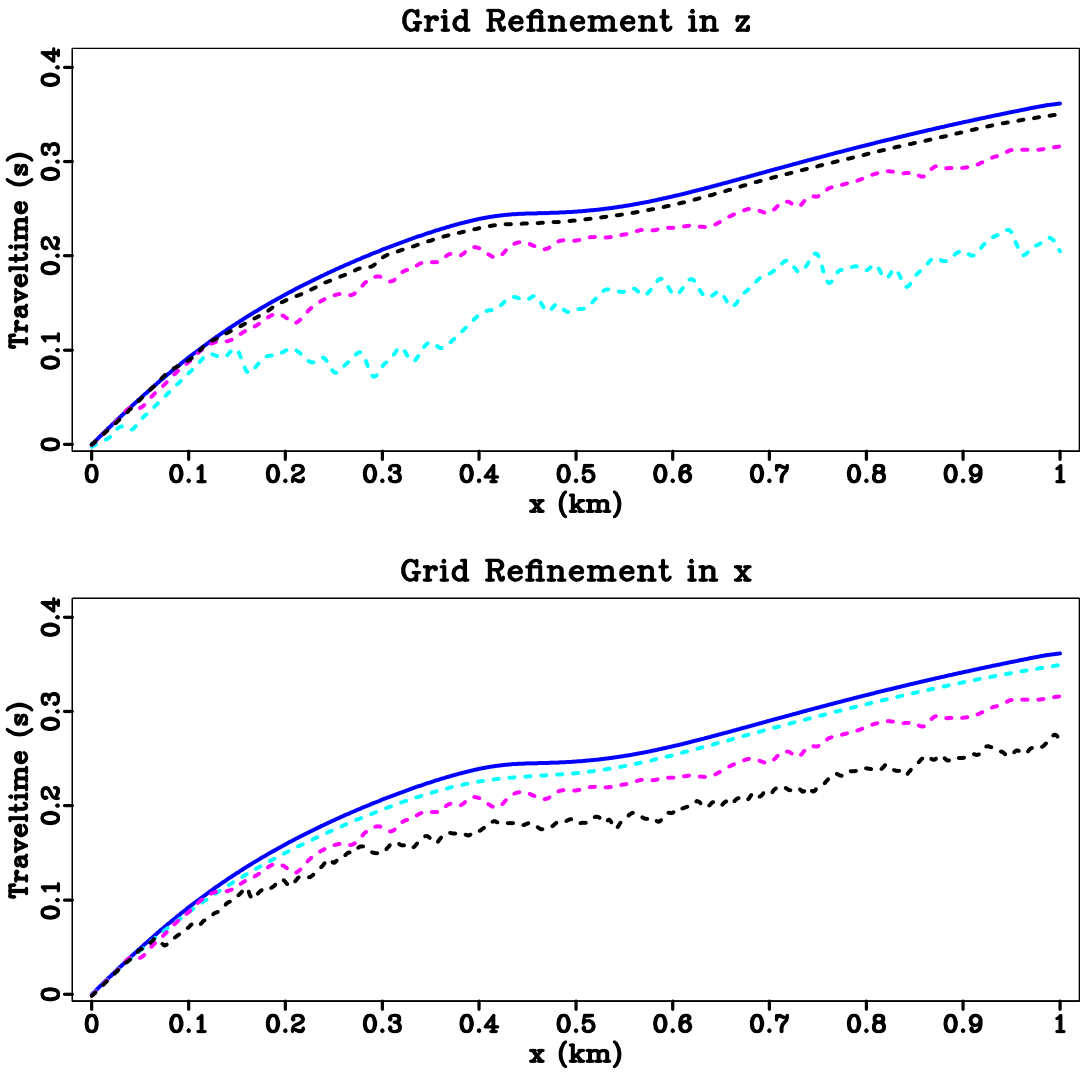


Figure 3.8: Grid refinement experiment (explicit discretization). The experiment set-ups are the same as in Figure 3.7. dsrtomo/accuracy exp

$(1.5, 0)$ km, respectively. The maximum absolute differences between the two DSR FMMs, for all three shots, are approximately 5 ms at the largest offset. This shows that, if the near-surface model is moderately complex, then the first-breaks are of causal types described by equations 1.9 and 3.2, and I therefore can use their linearizations 3.15 for tomography.

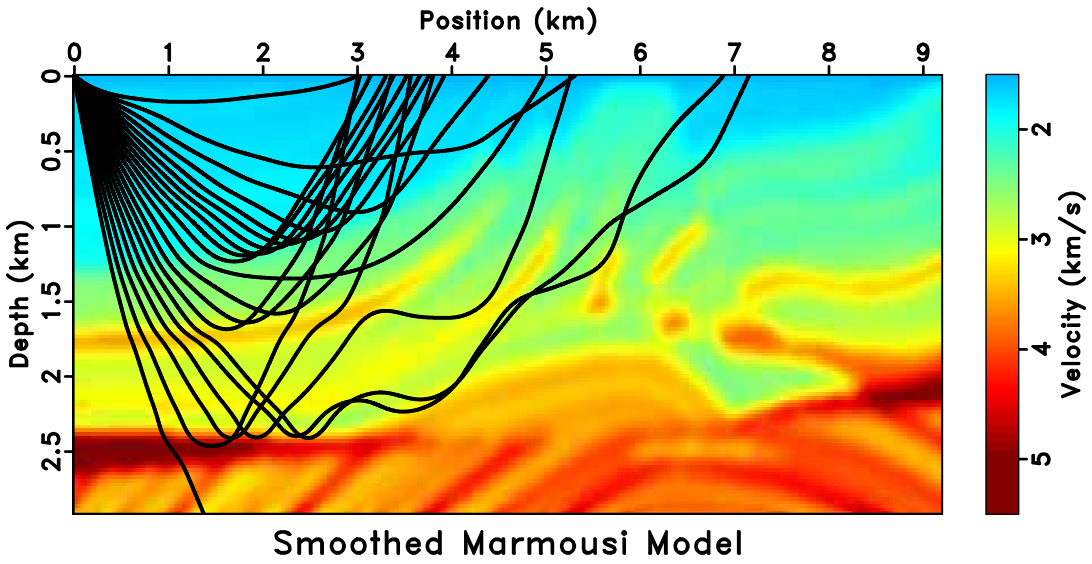


Figure 3.9: A smoothed Marmousi model overlaid with rays traced from a shot at $(0, 0)$ km. Because of velocity variations, multi-pathing is common in this model, especially at large offsets. dsrtomo/causal marmsmooth

Sensitivity kernels for tomography

According to equations 3.9 and 3.15, the sensitivity kernels (a row of Fréchet derivative matrix) of standard tomography and DSR tomography are different. Figure 3.11 compares sensitivity kernels for the same source-receiver pair in a constant velocity-gradient model. I use a fine model sampling of $\Delta z = \Delta x = 2.5$ m. The stan-

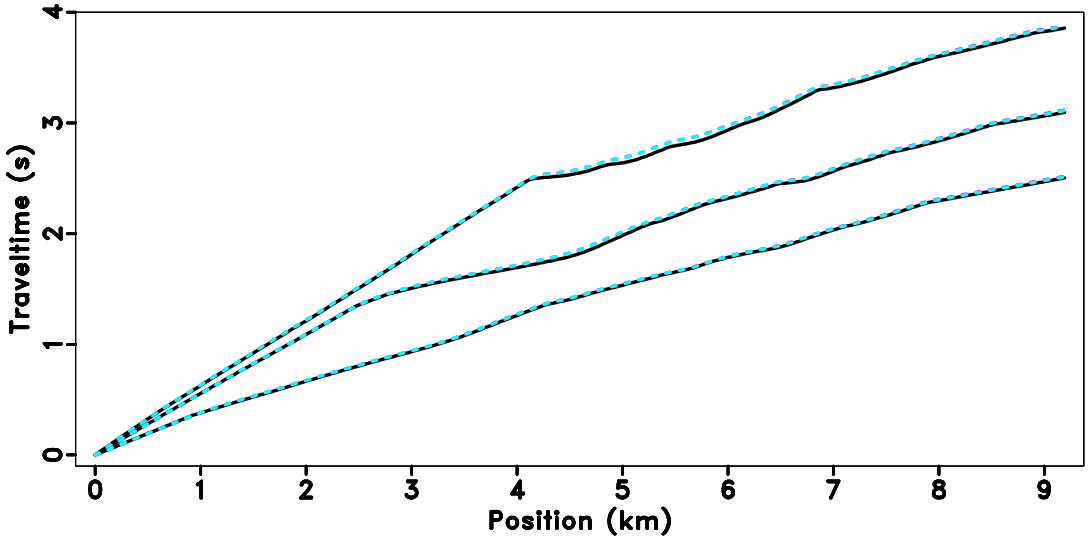


Figure 3.10: DSR FMM with non-causal branches. The solid black lines are reference values. There are two groups of dashed lines, both from DSR FMM but one with the optional search process turned-on and the other without. The differences between them are negligible and hardly visible. dsrtomo/causal causal

dard tomography kernel appears to be asymmetric. Its amplitude has a bias towards the source side, while the width is broader on the receiver side. These phenomena are related to my upwind finite-differences implementation. Note in the top plot of Figure 3.11, the curvature of first-break wave-front changes during propagation. Upwind finite-differences take the curvature variation into consideration and, as a result, back-project data-misfit with different weights along the ray-path. Meanwhile, the DSR tomography kernel is symmetric in both amplitude and width, even though it uses the same discretization and upwind approximation as in standard tomography. The source-receiver reciprocity may suggest averaging the standard tomography kernel with its own mirroring around $x = 1$ km, however the result will still be different from the DSR tomography kernel as the latter takes into consideration all sources at the same time.

Marmousi model

Figure 3.12 illustrates a prestack first-break traveltime modeling of the Marmousi model by DSR FMM. I use a constant-velocity-gradient model as the prior for inversion. There are 287 shots evenly distributed on the surface, each shot has a maximum absolute receiver offset of 6 km. Figure 3.13 shows a zoom-in of the exact model that is within the tomographic aperture. The DSR tomography and standard tomography are performed with the same parameters: 10 conjugate gradient iterations per linearization update and 4 linearization updates in total. Figure 3.14 shows the convergence histories. While both inversions converge, the relative l_2 data misfits of DSR tomography decreases faster than that of standard tomography. Figure 3.15 compares the recovered models. Although both results resemble the exact model in Figure 3.13 at the large scale, the standard tomography model exhibits several undesired structures. For example, a near-horizontal structure with a velocity of around 2.75 km/s at location $(0.85, 4.8)$ km is false. It indicates the presence of a local minimum that has trapped the standard tomography. In practice, it is helpful to tune the inversion parameters so that the standard tomography takes more iterations with a gradually reducing regularization. The inversion parameters are usually empirical and hard to control. My analysis in preceding sections suggests that part of the role of regularization is to deal with conflicting information between shots. In contrast, I find DSR tomography less dependent on regularization and hence more robust.

The advantage of DSR tomography becomes more significant in the presence of noise in the input data. I generate random noise of normal distribution with zero

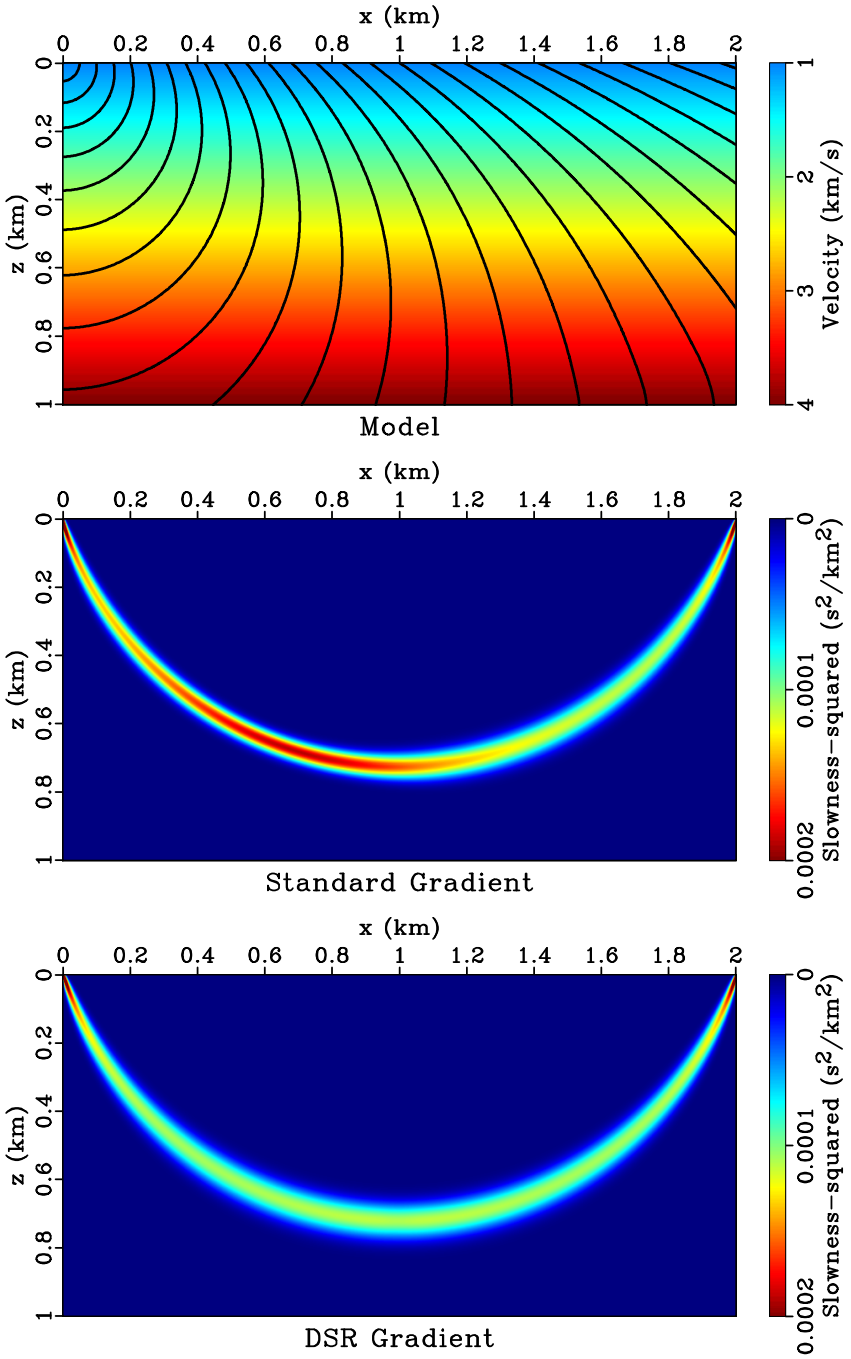


Figure 3.11: (Top) model overlaid with traveltime contours of a source at $(0, 0)$ km and sensitivity kernels of (middle) the standard tomography and (bottom) the DSR tomography. dsrtomo/hessian grad

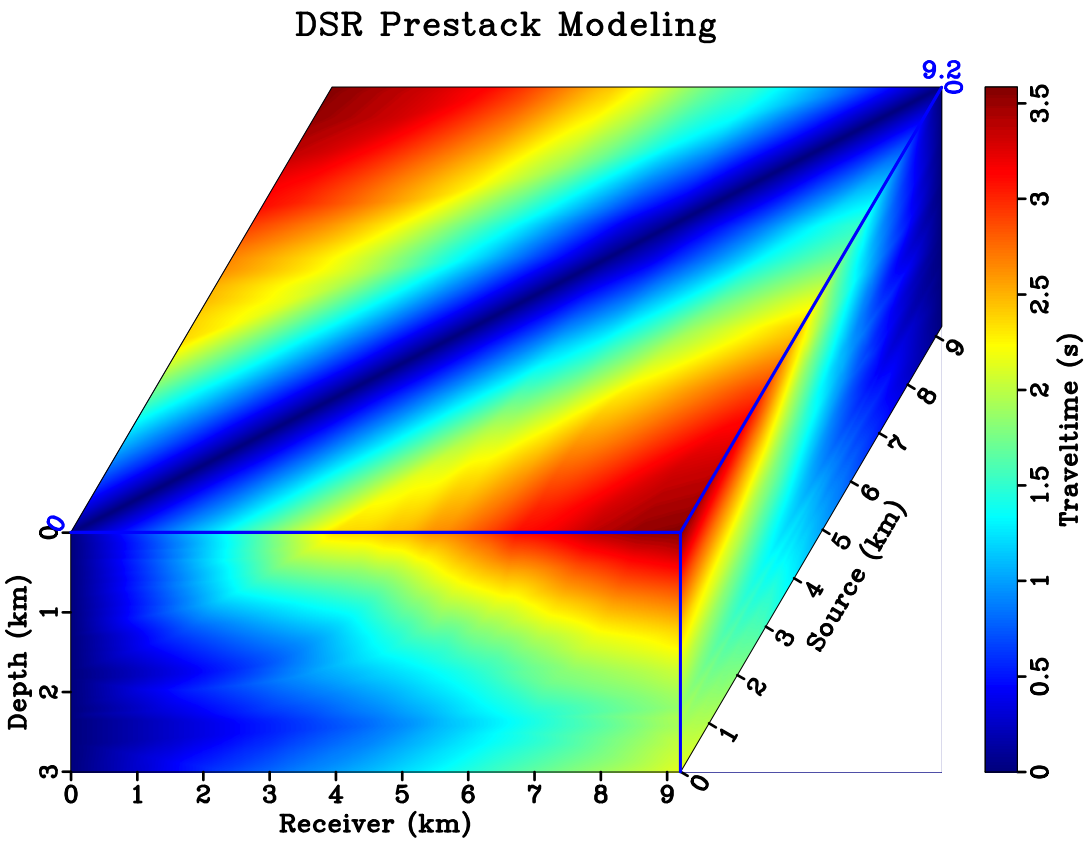


Figure 3.12: DSR first-break traveltimes in the Marmousi model. The original model is decimated by 2 in both vertical and lateral directions, such that $nz = 376$, $nx = 1151$ and $\Delta z = \Delta x = 8$ m. dsrtomo/marm data

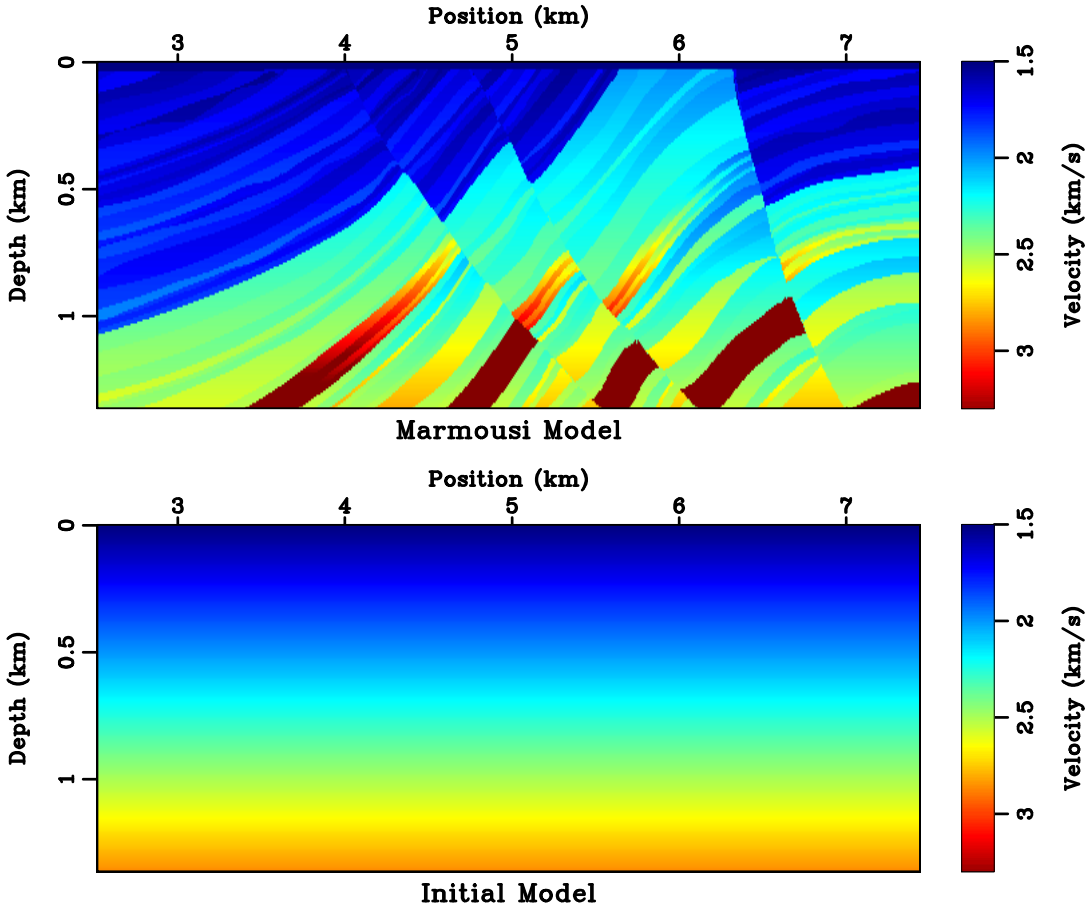


Figure 3.13: (Top) a zoom-in of Marmousi model and (bottom) the initial model for tomography. dsrtomo/marm marm

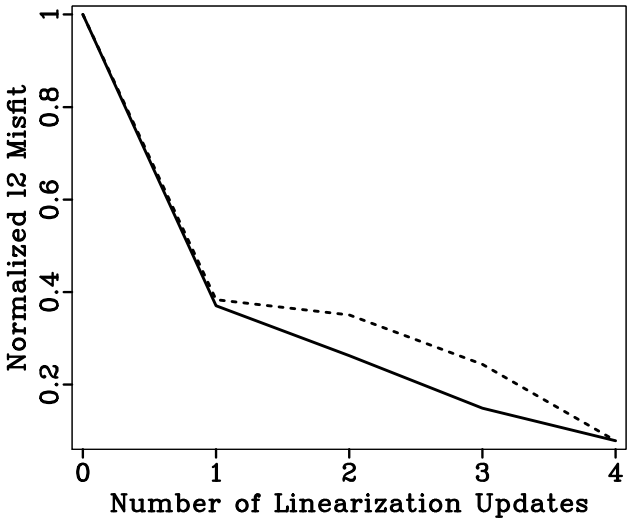


Figure 3.14: Convergence history of DSR tomography (solid) and standard tomography (dashed). There is no noticeable improvement on misfit after the fourth update. dsrtomo/marm conv

mean and a range between ±600 ms, then threshold the result with a minimum absolute value of 250 ms. This is to mimic the spiky errors in first-breaks estimated from an automatic picker. After adding noise to the data, I run inversions with the same parameters as in Figures 3.14 and 3.15. Figures 3.16 and 3.17 show the convergence history and inverted models. Again, the standard tomography seems to provide a model with higher resolution, but a close examination reveals that many small scale details are in fact non-physical. On the other hand, DSR tomography suffers much less from the added noise. Adopting a l_1 norm in objective function 3.3 can improve the inversion, especially for standard tomography. However, it would also raise the difficulty in selecting appropriate inversion parameters.

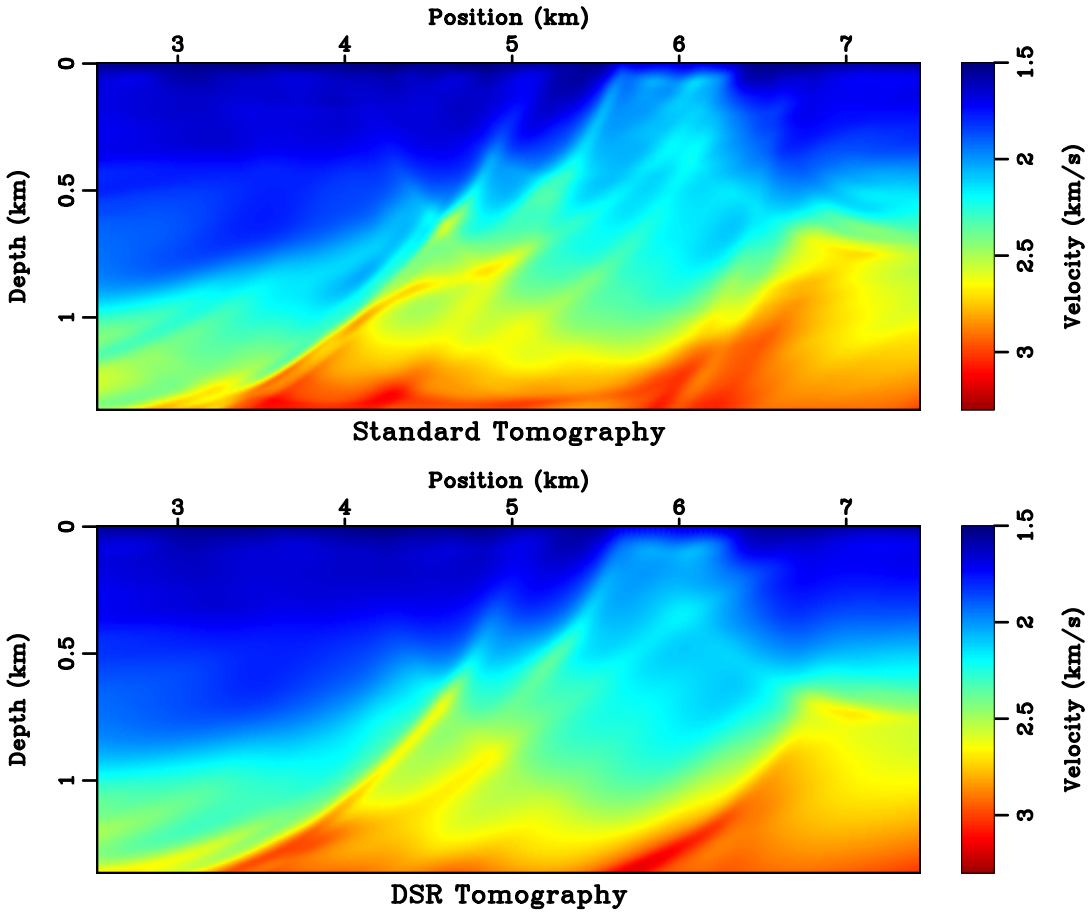


Figure 3.15: Inverted model of (top) standard tomography and (bottom) DSR tomography. Compare with Figure 3.13. dsrtomo/marm tomo

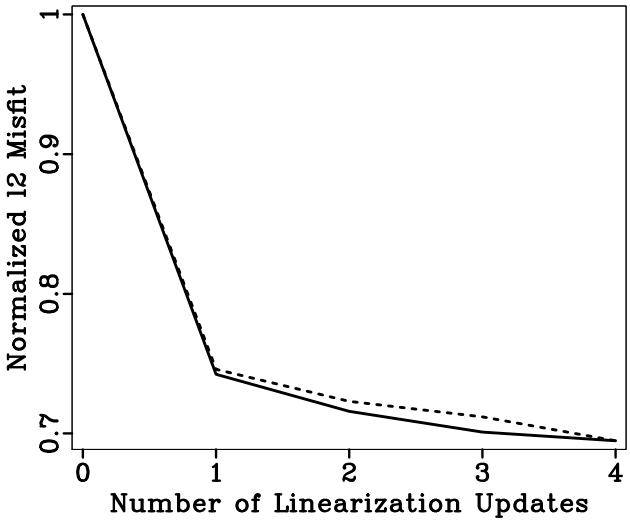


Figure 3.16: Inversion with noisy data. Convergence history of DSR tomography (solid) and standard tomography (dashed). No significant decrease in misfit appears after the fourth update. dsrtomo/marm nconv

Synthetic dataset

Finally, I apply DSR tomography to the synthetic dataset described in Chapter 1 Figure 1.10. Figure 3.18 shows a shot gather and the corresponding first-breaks. Note that the near-surface inclusions have effects on both refractions and reflections. However, the reflections are also influenced by velocity variations in deeper part of the model. Refractions on the other hand carry mostly near-surface information, including topography that might be a prominent factor in land datasets. I use the background $v(z)$ model in Figure 1.10 as the prior for DSR tomography. After 7 linearization updates (10 conjugate gradient iterations per update), the l_2 data misfit drops to relative 1.5%. The near surface is well recovered as illustrated in Figure 3.19.

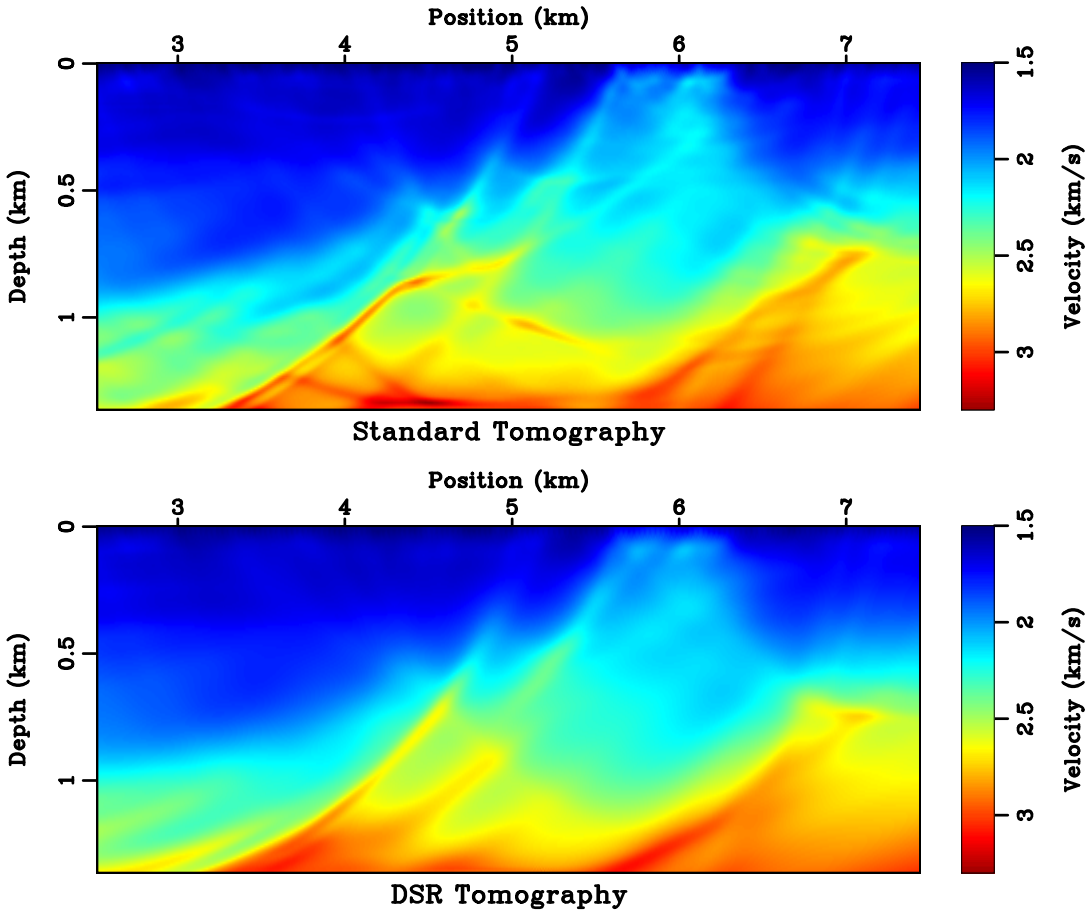


Figure 3.17: Inversion with noisy data. Inverted model of (top) standard tomography and (bottom) DSR tomography. Compare with Figure 3.15. dsrtomo/marm ntomo

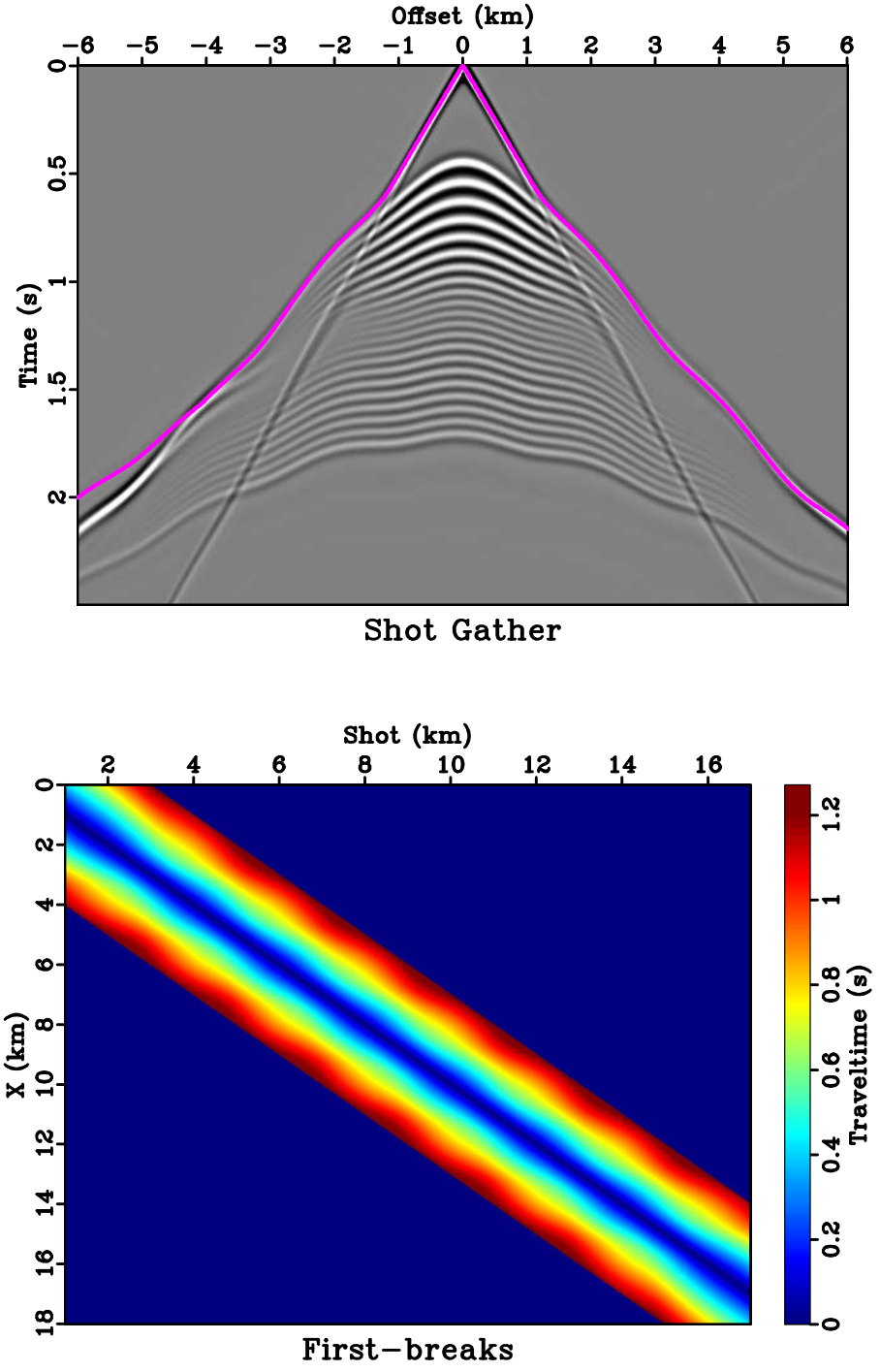


Figure 3.18: (Top) a shot gather at $(0, 6)$ km. First-breaks are overlaid on top of the seismograms. (Bottom) input first-breaks for DSR tomography. dsrtomo/thesis cmp

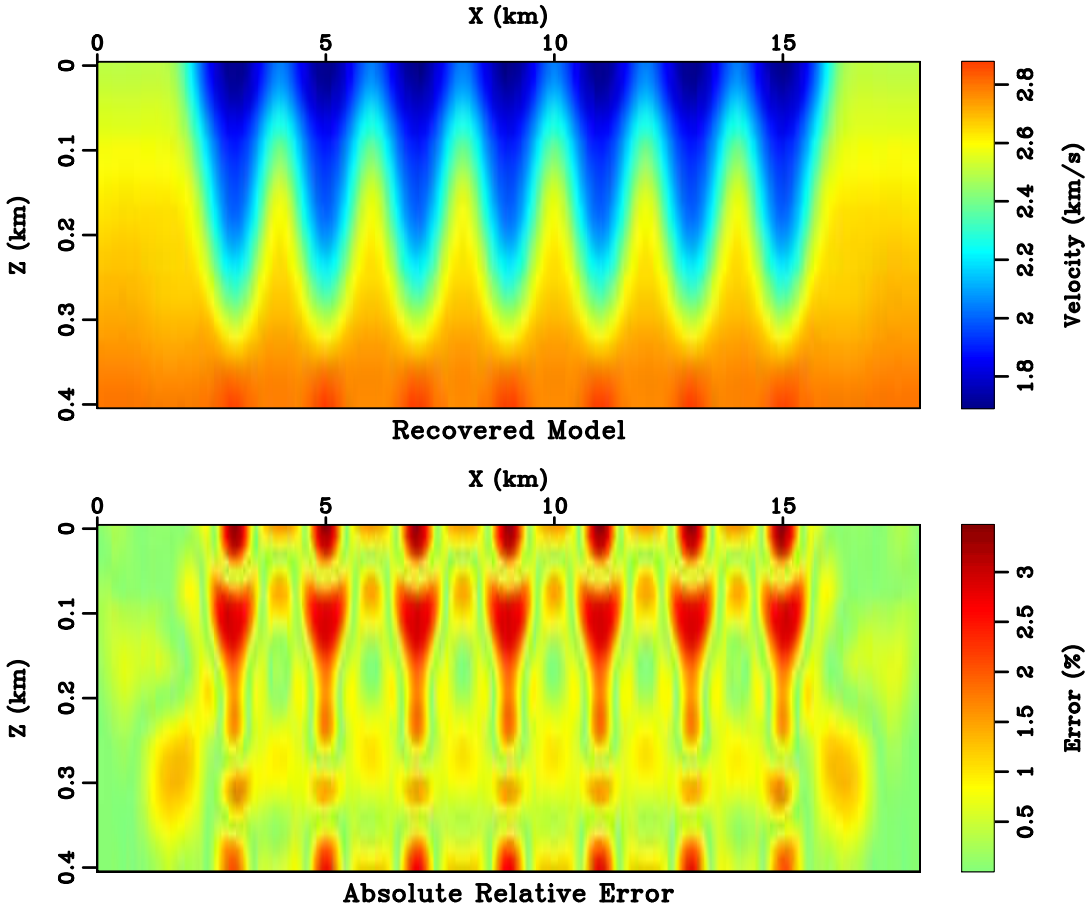


Figure 3.19: The inverted near-surface model (top) and its difference with exact model in Figure 1.10 (bottom). dsrtomo/thesis dsrt

DISCUSSION

There are three main issues in the DSR tomography. The first issue comes from a large dimensionality of the prestack space, which results in a considerable computational domain size after discretization. The memory consumption becomes an immediate problem for 3-D models, where the prestack traveltime belongs to a 5-D space and may require distributed storage.

The second issue is related to the computational cost. The FMM DSR I have introduced in this chapter has a computational complexity of $O(N \log N)$, where N is the total number of grid points after discretization, $N = nz \times nx^2$. Same as FMM for the eikonal equation 1.4, the $\log N$ factor in FMM DSR arises in the priority queue used for keeping track of expanding wave-fronts. Some known extensions could accelerate FMM to an $O(N)$ complexity and may be applicable to the DSR eikonal equation (Kim, 2001; Yatziv et al., 2006). A number of alternative fast methods developed for the eikonal equation might be similarly applicable to the DSR eikonal equation. These include fast sweeping (Zhao, 2005), hybrid two-scale marching-sweeping methods and label-correcting algorithms (Chacon and Vladimirsky, 2012a).

The last issue is possible parallelization of the proposed method. My current implementation of the FMM DSR tomography algorithm is sequential, while the traditional tomography could be parallelized among different shots. However, I notice that the DSR eikonal equation has a plane-wave source, therefore a distributed wave-front propagating at the beginning followed by a subdomain merging is possible. A number of parallelizable algorithms for the eikonal equation have been developed

(Zhao, 2007; Jeong and Whitaker, 2008; Weber et al., 2008; Chacon and Vladimirsky, 2012b; Detrixhe et al., 2013). Extending these methods to the DSR eikonal equation would be the first step in parallelizing DSR tomography.

CONCLUSIONS

I propose to use the DSR eikonal equation for the first-break traveltime tomography. The proposed method relies on an efficient DSR solver, which is realized by a version of the fast-marching method based on an implicitly causal discretization. Since the DSR eikonal equation allows changing of source position along the ray-path, its linearization results in a tomographic inversion that naturally handles possible conflicting information between different shots. My numerical tests in this chapter show that, compared with the traditional tomography with a shot-indexed eikonal equation, the DSR tomography is more robust and provides more accurate solutions. Its benefits may be particularly significant in the presence of noise in the data.

Chapter 4

Kirchhoff redatuming and migration using eikonal-based computation of traveltime source-derivatives

INTRODUCTION

The quality of traveltime computation has a direct influence on Kirchhoff-type migrations since it determines the kinematic behavior of the imaged wavefields. In practice, traveltime tables can be pre-computed on coarse grids and saved on disk, then serve as a dictionary when read by Kirchhoff migration algorithms. It is common to carry out certain interpolations in this process in order to satisfy the needs of depth migration for fine-gridded traveltime tables at a large number of source locations (Mendes, 2000; Vanelle and Gajewski, 2002; Alkhalifah, 2011a). Kirchhoff migrations with traveltime tables computed on the fly face the same issue.

During the traveltime computation stage, accuracy requirements from eikonal solvers may lead to a fine model sampling. Combined with a large survey, traveltime computation for each shot can be costly. Because all traveltime computations handle one shot at a time, the overall cost increases linearly with the number of sources. Moreover, we need to store a large amount of traveltimes out of a dense source sampling. For saving storage, a sparse source sampling is preferred. In this chapter, I

Parts of this chapter were published in Li, S., and S. Fomel, 2013, Kirchhoff migration using eikonal-based computation of traveltime source-derivatives: Geophysics, **78**, no. 4, S211-S219. This work is done under the supervision of Fomel.

address the problem of traveltime table interpolation between sparse source samples.

The traveltime table estimated with simple nearest-neighbor or linear interpolation may not provide satisfying accuracy unless the velocity model has small variations. One possible improvement is to include derivatives in interpolation. During ray tracing, traveltime source-derivatives are directly connected to the slowness vector at the source and stay constant along individual rays, thus could be outputted as a by-product of traveltimes. For finite-difference eikonal solvers, such a convenience is not easily available. In these cases, we would like to avoid an extra differentiation on traveltime tables along the source dimension to compute such derivatives (Vanelle and Gajewski, 2002), because its accuracy in turn relies on a dense source sampling and induces additional computations.

Alkhalifah and Fomel (2010) derived an equation for the traveltime perturbation with respect to the source location changes. The governing equation is a first-order partial differential equation (PDE) that describes traveltime source-derivatives in a relative coordinate moving along with the source. I will show that the traveltime source-derivative desired by interpolation is related to this relative-coordinate quantity by a simple subtraction of the slowness vector. Unlike a finite-difference approach, traveltime source-derivatives computed by the proposed PDE method are source-sampling independent. The extra costs are relatively inexpensive. I then apply this method to Kirchhoff redatuming and migration with first-arrival traveltimes computed by the fast-marching (FMM) eikonal solver.

This chapter is organized as follows. In the first section, I present the theory and implementation of eikonal-based traveltime source-derivatives. Next, I use both simple and complex synthetic models to demonstrate the accuracy of a cubic Hermite traveltime table interpolation using the source-derivatives, and show effects of incorporating such an interpolation into Kirchhoff migration. I will focus mainly on the kinematics in these experiments by neglecting possible true-amplitude weights in Kirchhoff migration (Schleicher et al., 2007). Finally, I discuss limitations and possible extensions of the proposed method.

THEORY

In this chapter, I focus on point-source solutions of the eikonal equation 1.4, i.e. those with the initial condition $T(\mathbf{x_s}) = 0$ where $\mathbf{x_s}$ denotes the source location. The point-source traveltime $T(\mathbf{x})$ clearly depends on the source location $\mathbf{x_s}$. To explicitly show such a dependency in the eikonal equation, I define a relative coordinate $\mathbf{q}$ as

$$\mathbf{q} = \mathbf{x} - \mathbf{x_s} \ , \tag{4.1}$$

and use $\tilde{T}(\mathbf{q}; \mathbf{x_s})$ to denote traveltime expressed in the relative coordinates. After inserting this definition into equation 1.4, I obtain

$$\nabla_{\mathbf{q}} \tilde{T} \cdot \nabla_{\mathbf{q}} \tilde{T} = w(\mathbf{q} + \mathbf{x_s}) \ . \tag{4.2}$$

Here the differentiation $\nabla_{\mathbf{q}}$ stands for gradient operator in the relative coordinate $\mathbf{q}$ and is taken with a fixed source location $\mathbf{x_s}$. In 3-D, if $\mathbf{q} = (q_1, q_2, q_3)$ and denoting $\mathbf{e}_i$ with $i = \{1, 2, 3\}$ to be the unit vector in depth, inline and crossline directions, respectively, then

$$\nabla_{\mathbf{q}} \equiv \frac{\partial}{\partial q_1}\mathbf{e}_1 + \frac{\partial}{\partial q_2}\mathbf{e}_2 + \frac{\partial}{\partial q_3}\mathbf{e}_3 \ . \tag{4.3}$$

Because I am interested in the traveltime derivative with respect to the source, i.e. $\partial T/\partial \mathbf{x_s}$, I apply the directional derivative $\partial/\partial \mathbf{x_s}$ to $\tilde{T}(\mathbf{q};\mathbf{x_s})$ and apply the chain-rule according to equation 4.1:

$$\frac{\partial T}{\partial \mathbf{x_s}} \equiv \frac{\partial \tilde{T}}{\partial \mathbf{x_s}} = \frac{\partial \tilde{T}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{x_s}} + \frac{\partial \tilde{T}}{\partial \mathbf{q}}\frac{\partial \mathbf{q}}{\partial \mathbf{x_s}} = \frac{\partial \tilde{T}}{\partial \mathbf{x}} - \frac{\partial \tilde{T}}{\partial \mathbf{q}} \; . \tag{4.4}$$

Equation 4.4 describes a gradient vector that contains the traveltime source-derivatives in depth, inline and crossline directions. In accordance with $\partial/\partial \mathbf{x_s}$, $\partial/\partial \mathbf{x}$ and $\partial/\partial \mathbf{q}$ are also directional derivatives. All numerical examples in this chapter are based on a typical 2-D acquisition, where I assume a constant source depth and thus only the inline traveltime source-derivative is of interest. The quantity $\partial \tilde{T}/\partial \mathbf{q}$ coincides with the slowness vector of the ray that originates from $\mathbf{x_s}$. For a finite-difference eikonal solver such as FMM and FSM, it is usually estimated by an upwind scheme during traveltime computations at each grid point and thus can be easily extracted. Applying $\partial/\partial \mathbf{x}$ to both sides of equation 4.2, I find

$$\nabla_{\mathbf{q}} \tilde{T} \cdot \nabla_{\mathbf{q}} \frac{\partial \tilde{T}}{\partial \mathbf{x}} = \frac{1}{2}\frac{\partial w}{\partial \mathbf{x}} \; . \tag{4.5}$$

Equation 4.5 has the form of the linearized eikonal equation 1.8 and was previously derived, in a slightly different notation, by Alkhalifah and Fomel (2010). It implies that $\partial \tilde{T}/\partial \mathbf{x}$, as needed by equation 4.4, can be determined along the characteristics of $\tilde{T}$. Since the right-hand side contains a slowness-squared derivative, the velocity model must be differentiable, as is usually required by traveltime computations. The derivation also indicates that the accuracy of an eikonal-based traveltime source-derivative is source-sampling independent but model-sampling dependent, as from equations

4.4 and 4.5 $\partial/\partial\mathbf{x_s}$ relies on $\tilde{T}$, $\partial/\partial\mathbf{q}$ and $\partial/\partial\mathbf{x}$. The accuracy from a direct finite-difference estimation on $\partial/\partial\mathbf{x_s}$, in comparison, is both source- and model-sampling dependent.

Continuing applying differentiation and the chain-rule to equation 4.4 will result in higher-order traveltime source-derivatives. For example, the second-order derivative reads:

$$\begin{aligned}\frac{\partial^2 T}{\partial \mathbf{x_s^2}} \equiv \frac{\partial^2 \tilde{T}}{\partial \mathbf{x_s^2}} &= \frac{\partial}{\partial \mathbf{x}}\frac{\partial \tilde{T}}{\partial \mathbf{x}}\cdot\frac{\partial \mathbf{x}}{\partial \mathbf{x_s}} + \frac{\partial}{\partial \mathbf{q}}\frac{\partial \tilde{T}}{\partial \mathbf{x}}\cdot\frac{\partial \mathbf{q}}{\partial \mathbf{x_s}} - \frac{\partial}{\partial \mathbf{x}}\frac{\partial \tilde{T}}{\partial \mathbf{q}}\cdot\frac{\partial \mathbf{x}}{\partial \mathbf{x_s}} - \frac{\partial}{\partial \mathbf{q}}\frac{\partial \tilde{T}}{\partial \mathbf{q}}\cdot\frac{\partial \mathbf{q}}{\partial \mathbf{x_s}} \\ &= \frac{\partial^2 \tilde{T}}{\partial \mathbf{x}^2} - 2\frac{\partial^2 \tilde{T}}{\partial \mathbf{x}\partial \mathbf{q}} + \frac{\partial^2 \tilde{T}}{\partial \mathbf{q}^2}\ . \end{aligned} \tag{4.6}$$

Further, differentiating equation 4.5 once more by $\mathbf{x}$ provides

$$\nabla_{\mathbf{q}}\frac{\partial \tilde{T}}{\partial \mathbf{x}}\cdot\nabla_{\mathbf{q}}\frac{\partial \tilde{T}}{\partial \mathbf{x}} + \nabla_{\mathbf{q}}\tilde{T}\cdot\nabla_{\mathbf{q}}\frac{\partial^2 \tilde{T}}{\partial \mathbf{x}^2} = \frac{1}{2}\frac{\partial^2 w}{\partial \mathbf{x}^2}\ . \tag{4.7}$$

It is easy to verify that any order of the traveltime source-derivative will require the corresponding order of the slowness-squared derivative. An approximation based on Taylor expansions of the traveltime around a fixed source location can make use of these derivatives. Previously, Ursin (1982) and Bortfeld (1989) introduced parabolic and hyperbolic traveltime approximations with the first- and second-order derivatives. Notice that the need for slowness-squared derivatives may cause instability unless the velocity model is sufficiently smooth. Alkhalifah and Fomel (2010) proved the following relationship between $\partial w/\partial\mathbf{x}$ and $\partial\tilde{T}/\partial\mathbf{q}$:

$$\nabla_{\mathbf{q}}\tilde{T}\cdot\nabla_{\mathbf{q}}\frac{\partial \tilde{T}}{\partial(\mathbf{q}+\mathbf{x_s})} = \frac{1}{2}\frac{\partial w}{\partial \mathbf{x}}\ , \tag{4.8}$$

which implies that the traveltime source-derivative can be computed from the given traveltime tables only. However, the velocity smoothness is still implicitly assumed as the second-order spatial derivatives of traveltimes appear in the equation. For this reason, I restrict my current implementation to first-order derivatives only.

In a ray-tracing eikonal solver, $\partial T/\partial \mathbf{x_s}$ is the slowness vector of a particular ray at $\mathbf{x_s}$ and holds constant along the trajectory. As it may also require irregular coordinate mappings, one may use the same strategy as for the traveltime tables. In this way, there is no necessity for any additional effort. Equations 4.4 and 4.5 and their second-order extensions can also provide important attributes for use in Gaussian beams, which are commonly calculated by the dynamic ray tracing (Červený, 2001). They might be alternatively estimated by the eikonal-based source-derivative formulas but with the traveltime tables from a finite-difference eikonal solver. However, this application is beyond the scope of current work. In the following sections, I consider only the source-derivative estimation from traveltimes computed by the finite-difference eikonal solver FMM.

IMPLEMENTATION

Because equation 4.5 does not change the nonlinear nature of the eikonal equation, the resulting traveltime source-derivative can be related to any branch of multi-arrivals, if one supplies the corresponding traveltime in $\tilde{T}$. The source-derivatives can be computed either along with traveltimes or separately. Below I describe a first-arrival implementation based on a modification of FMM.

First I set up upwind finite-difference stencils 2.2 based on $\tilde{T}$ and obtain linear operators $\nabla\tilde{T}\cdot\nabla$ and its inverse $(\nabla\tilde{T}\cdot\nabla)^{-1}$ as described in Chapter 2. Then for $\partial\tilde{T}/\partial\mathbf{x}$ in equation 4.5, I find

$$D_x\tilde{T}^k_{i,j}\cdot D_x\left(\frac{\partial\tilde{T}}{\partial\mathbf{x}}\right)^k_{i,j}+D_y\tilde{T}^k_{i,j}\cdot D_y\left(\frac{\partial\tilde{T}}{\partial\mathbf{x}}\right)^k_{i,j}+D_z\tilde{T}^k_{i,j}\cdot D_z\left(\frac{\partial\tilde{T}}{\partial\mathbf{x}}\right)^k_{i,j}=\frac{1}{2}\left(\frac{\partial w}{\partial\mathbf{x}}\right)^k_{i,j}\ , \quad (4.9)$$

or equivalently,

$$\frac{\partial\tilde{T}}{\partial\mathbf{x}}=\frac{1}{2}\left(\nabla\tilde{T}\cdot\nabla\right)^{-1}\left(\frac{\partial w}{\partial\mathbf{x}}\right)\ . \quad (4.10)$$

As explained in Chapter 2, computation of equation 4.10 can be done efficiently in the upwind ordering of $\tilde{T}$. Meanwhile, $\partial\tilde{T}/\partial\mathbf{q}$ in equation 4.4 turns out to be simply the upwind stencils themselves:

$$\left(\frac{\partial\tilde{T}}{\partial\mathbf{q}}\right)^k_{i,j}=D_{\mathbf{q}}\tilde{T}^k_{i,j},\ \mathbf{q}=(x,y,z)\ . \quad (4.11)$$

Finally,

$$\left(\frac{\partial T}{\partial\mathbf{x_s}}\right)^k_{i,j}=\left(\frac{\partial\tilde{T}}{\partial\mathbf{x}}\right)^k_{i,j}-\left(\frac{\partial\tilde{T}}{\partial\mathbf{q}}\right)^k_{i,j}\ . \quad (4.12)$$

To incorporate the computation of traveltime source-derivatives into eikonal solving, one only needs to add equations 4.9, 4.11 and 4.12 after the FMM local update stage. An extra upwind sorting and solving after pre-computing $\tilde{T}$ is not necessary. For a total N grid points, the computational complexity of FMM with auxiliary traveltime source-derivative output remains $O(N\log N)$.

APPLICATION

Traveltime interpolation

The first-order traveltime source-derivative enables a cubic Hermite interpolation (Press et al., 2007). Geometrically, such an interpolation is valid only when the selected wave-front in the interpolation interval is smooth and continuous. For a 2-D model and a source interpolation along the inline direction only, the Hermite interpolation reads:

$$\begin{aligned} T(z,x;z_s,x_s+\alpha\Delta x_s) &= (2\alpha^3-3\alpha^2+1)\,T(z,x;z_s,x_s) \\ &+ (\alpha^3-2\alpha^2+\alpha)\,\frac{\partial T}{\partial x_s}(z,x;z_s,x_s) \\ &+ (-2\alpha^3+3\alpha^2)\,T(z,x;z_s,x_s+\Delta x_s) \\ &+ (\alpha^3-\alpha^2)\,\frac{\partial T}{\partial x_s}(z,x;z_s,x_s+\Delta x_s)\;, \end{aligned} \tag{4.13}$$

where $\alpha \in [0,1]$ controls the source position to be interpolated between known values at (z_s, x_s) and $(z_s, x_s + \Delta x_s)$. For comparison, the linear interpolation can be represented by:

$$T(z,x;z_s,x_s+\alpha\Delta x_s) = (1-\alpha)\,T(z,x;z_s,x_s) + \alpha\,T(z,x;z_s,x_s+\Delta x_s)\;. \tag{4.14}$$

The linear interpolation fixes the subsurface image point (z, x). A possible improvement is to instead fix the vector that links the source with the image, such that on the right-hand side the traveltimes are taken at shifted image locations:

$$\begin{aligned} T(z,x;z_s,x_s+\alpha\Delta x_s) &= (1-\alpha)\,T(z,x-\alpha\Delta x_s;z_s,x_s) \\ &+ \alpha\,T(z,x+(1-\alpha)\Delta x_s;z_s,x_s+\Delta x_s)\;. \end{aligned} \tag{4.15}$$

I will refer to scheme 4.15 as shift interpolation. According to my definition of the relative coordinate $\mathbf{q}$ in equation 4.1, shift interpolation amounts to a linear interpolation in $\tilde{T}(\mathbf{q};\mathbf{x_s})$. It is easy to verify that, for a constant-velocity medium, both Hermite and shift interpolations are accurate, while the linear interpolation is not. However, the accuracy of shift interpolation deteriorates with increasing velocity variations, as it assumes that the wave-front remains invariant in the relative coordinate. Equations 4.13-4.15 can be generalized to 3-D by cascading the inline and crossline interpolations (for example equation 4.14 in 3-D case becomes bilinear interpolation). The interpolated source does not need to lie collinear with source samples.

Kirchhoff anti-aliasing

The derivatives themselves can also be directly applied for Kirchhoff anti-aliasing (Lumley et al., 1994; Abma et al., 1999; Biondi, 2001; Fomel, 2002). Equations 4.13, 4.14 and 4.15 give rise to their corresponding source-derivative interpolations after applying the following chain-rule to both sides:

$$\frac{\partial}{\partial(x_s+\alpha\Delta x_s)} = \frac{\partial}{\partial\alpha}\frac{\partial\alpha}{\partial(x_s+\alpha\Delta x_s)} = \frac{1}{\Delta x_s}\frac{\partial}{\partial\alpha}\ . \tag{4.16}$$

Applying the chain-rule 4.16 to equation 4.13, I arrive at the interpolation equation for source-derivatives in the cubic Hermite scheme:

$$\begin{aligned}\Delta x_s \frac{\partial T(z,x;z_s,x_s+\alpha\Delta x_s)}{\partial(x_s+\alpha\Delta x_s)} &= (6\alpha^2-6\alpha)\,T(z,x;z_s,x_s)\\ &+ (3\alpha^2-4\alpha+1)\,\frac{\partial T}{\partial x_s}(z,x;z_s,x_s)\\ &+ (-6\alpha^2+6\alpha)\,T(z,x;z_s,x_s+\Delta x_s)\end{aligned}$$

$$+ \quad (3\alpha^2 - 2\alpha)\,\frac{\partial T}{\partial x_s}(z, x; z_s, x_s + \Delta x_s)\ . \quad (4.17)$$

Analogously, the interpolation of source-derivatives in the linear scheme 4.14 reads:

$$\Delta x_s\,\frac{\partial T(z, x; z_s, x_s + \alpha\Delta x_s)}{\partial(x_s + \alpha\Delta x_s)} = -T(z, x; z_s, x_s) + T(z, x; z_s, x_s + \Delta x_s)\ , \qquad (4.18)$$

which is a simple first-order finite-difference estimation. Finally, in the case of shift scheme 4.15, the partial derivative $\partial/\partial\alpha$ must be applied to the shifted traveltime terms at the same time:

$$\begin{aligned}
\Delta x_s\,\frac{\partial T(z, x; z_s, x_s + \alpha\Delta x_s)}{\partial(x_s + \alpha\Delta x_s)} \quad &= \quad -T(z, x - \alpha\Delta x_s; z_s, x_s) \\
&- \quad (1-\alpha)\Delta x_s\,\frac{\partial T(z, x - \alpha\Delta x_s; z_s, x_s)}{\partial(x - \alpha\Delta x_s)} \\
&+ \quad T(z, x + (1-\alpha)\Delta x_s; z_s, x_s + \Delta x_s) \\
&- \quad \alpha\Delta x_s\,\frac{\partial T(z, x + (1-\alpha)\Delta x_s; z_s, x_s + \Delta x_s)}{\partial(x + (1-\alpha)\Delta x_s)} \qquad (4.19)
\end{aligned}$$

The required spatial derivatives can be estimated from the traveltime table by means of finite-differences, for example by using the upwind approximation 2.2.

EXAMPLES

Constant-velocity-gradient model

In a 2-D medium of linearly changing velocities, $v(z, x) = v_0 + ax + bz$ where x is the lateral position and z is the depth, the traveltimes and source-derivatives have analytical solutions (Slotnick, 1959). Figure 4.1 shows the model used in the first numerical test and the analytical source-derivative for a source located at $(0, 0)$ km. The domain is of size 4km $\times$ 4km with grid spacing 0.01 km in both directions. I solve for the traveltime tables at five sources of uniform spacing 1 km along the top

domain boundary by FMM, as well as their associated source-derivatives using the method described in previous section. Figure 4.2 compares the errors in computed source-derivative between the proposed approach and a centered second-order finite-difference estimation for the same source shown in Figure 4.1. The proposed method is sufficiently accurate except for the small region around the source. This is due to the source singularity of the eikonal equation and can be improved by adaptive or high-order upwind finite-difference methods (Qian and Symes, 2002) or by factoring the singularity (Fomel et al., 2009). Since I am aiming at using the interpolated traveltime tables for migration purposes and the reflection energy around the sources is usually low, these errors in current implementation can be neglected. In Figure 4.3, I interpolate the traveltime table for a source at location $(0, 0.25)$ km from the nearby source samples at $(0, 0)$ km and $(0, 1)$ km by the cubic Hermite, linear and shift interpolations. I use the eikonal-based source-derivative in the cubic Hermite interpolation. The shift interpolation is not applicable for some $\mathbf{q}$ and $\mathbf{x_s}$ if $\mathbf{x} = \mathbf{q} + \mathbf{x_s}$ is beyond the computational domain. In these regions, I use a linear interpolation to fill the traveltime table. As expected, the cubic Hermite interpolation achieves the best result, while its misfits near the source are related to the errors in source-derivatives. The shift interpolation performs generally better than the linear interpolation, especially in the regions close to the source where the wave-fronts are simple.

The difference between a cubic Hermite interpolation and a linear or shift one is in the usage of source-derivatives. In this regard, one may think of supplying the finite-difference estimated derivatives to the interpolation. Indeed, a refined source sampling and higher-order differentiation may lead to more accurate derivatives. However the additional computation is considerable. For the same model in Figure 4.1, I carry out

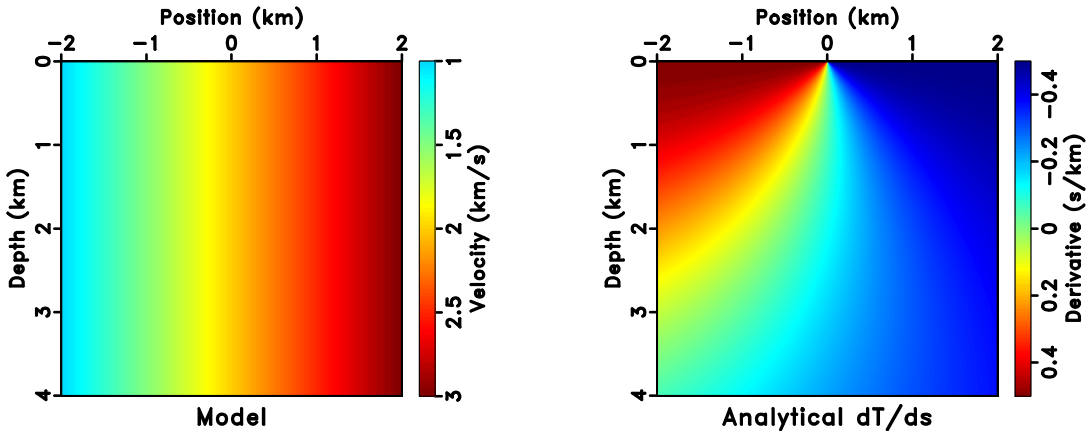


Figure 4.1: (Left) a constant-velocity-gradient model $v(z, x) = 2 + 0.5x$ km/s and (right) its analytical traveltime source-derivative for a source at origin $\mathbf{x_s} = (0, 0)$ km. eikods/check model

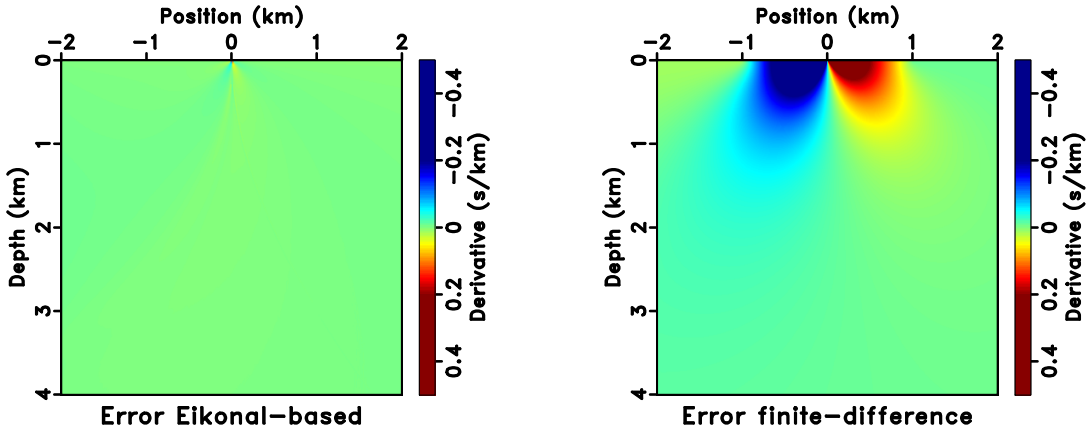


Figure 4.2: Comparison of error in computed source-derivative by (left) the proposed method and (right) a centered second-order finite-difference estimation based on traveltime tables. The maximum absolute errors are 0.15 s/km and 0.56 s/km, respectively. eikods/check diff

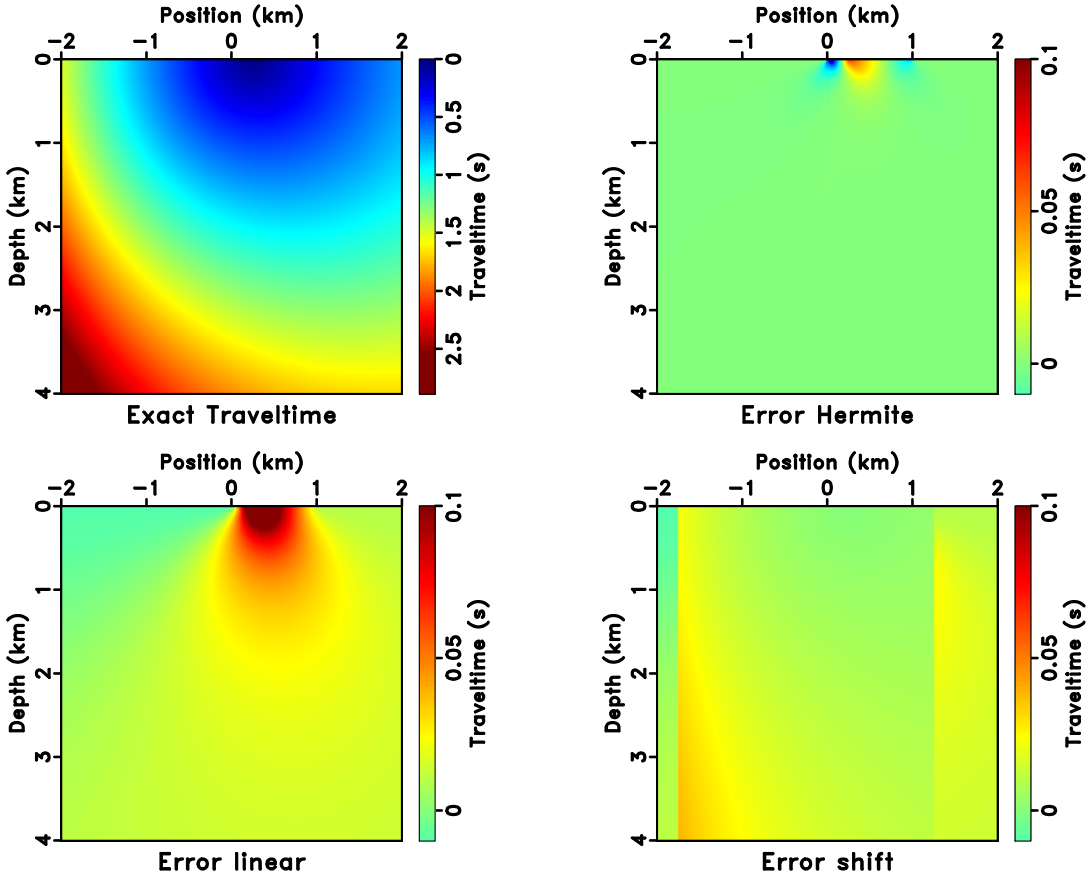


Figure 4.3: Traveltime interpolation error of three different schemes: (top left) the analytical traveltime of a source at location $(0, 0.25)$ km; (top right) error of the cubic Hermite interpolation; (bottom left) error of the linear interpolation; (bottom right) error of the shift interpolation. Using derivatives in interpolation enables a significantly higher accuracy. The $l2$ norm of the error are 1.5 s, 9.2 s and 6.0 s respectively. eikods/check ierror

both a source sampling refinement experiment and a model grid spacing refinement experiment. The results are shown in Figures 4.4 and 4.5. Both figures are plotted for the traveltime at subsurface location $(1.5, -0.5)$ km for the source at location $(0, 0)$ km. Although the curves vary for different locations, the source sampling refinement experiment suggests the general need for approximately three times finer source-sampling than that of Figure 4.2 to achieve the same level of accuracy.

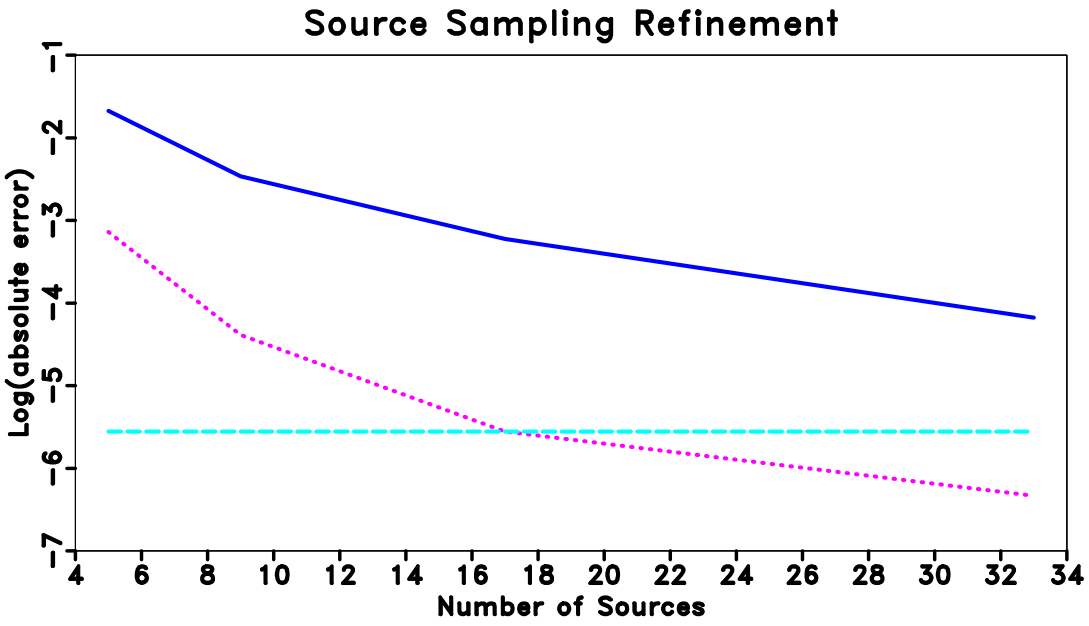


Figure 4.4: Source-sampling refinement experiment. The plot shows, at a fixed model grid sampling of 0.01 km and increasing source sampling, the error in source-derivative estimated by a first-order finite-difference (solid) and a centered second-order finite-difference scheme (dotted) decrease. The horizontal axis is the number of sources and the source sampling is uniform. The vertical axis is the natural logarithm of the absolute error. The flat line (dash) is from the proposed eikonal-based method and is source-sampling independent. eikods/check sfddiff

Kirchhoff migration can use traveltime source-derivatives in two ways: for traveltime interpolation when the source and receiver of a trace does not lie on the

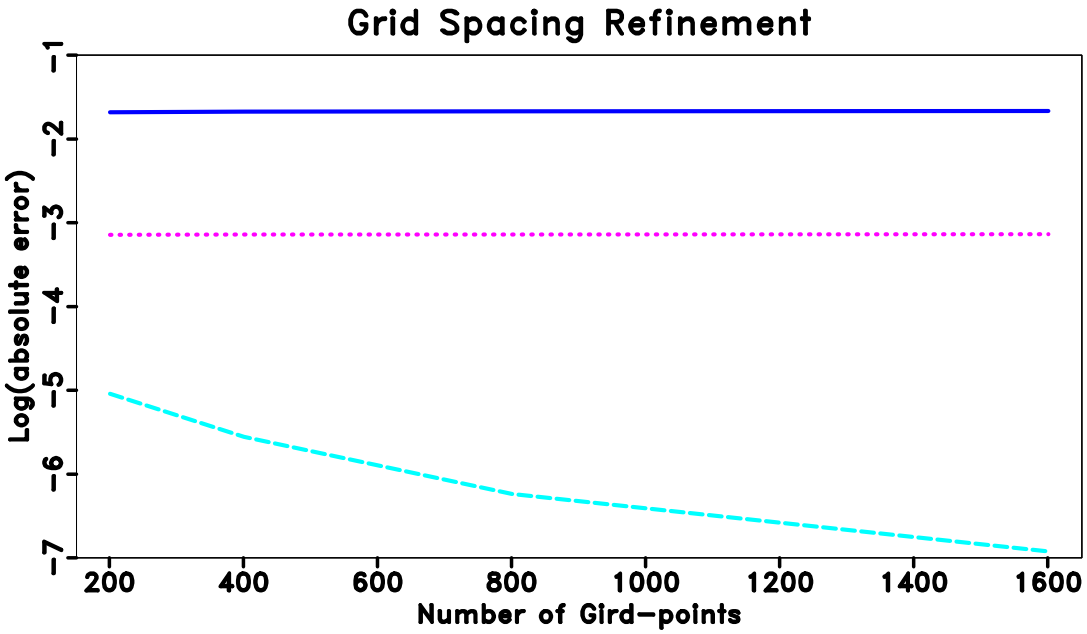


Figure 4.5: Gird-spacing refinement experiment. The plot shows, at a fixed source sampling of 1 km and increasing model grid sampling, the error in source-derivative estimated by the proposed eikonal-based method decreases. Meanwhile, the errors of both first- and second-order finite-difference estimations do not improve noticeably. The horizontal axis is the number of grid points in both directions and the grid sampling is uniform. See Figure 4.4 for descriptions of the vertical axis and the lines. eikods/check gfddiff

source grid of pre-computed traveltime tables, and for anti-aliasing. Figure 4.6 shows a synthetic model of constant-velocity-gradient with five dome-shaped reflectors. The model has a 0.01 km grid spacing in both directions. I solve for traveltimes and source-derivatives by the modified FMM introduced in at 21 sparse shots of uniform spacing 0.5 km, and migrate synthetic zero-offset data. The interpolation of source-derivative for the anti-aliasing purpose follows the method described in Application section. 48 interpolations are carried out within each sparse source sampling interval. Figures 4.7 and 4.8 compare the images obtained by three different interpolations and the effect of anti-aliasing. All images are plotted at the same scale. I do not limit migration aperture for all cases and adopt the anti-aliasing criteria suggested by Abma et al. (1999) to filter the input trace before mapping a sample to the image, where the source-derivative and receiver-derivative (in the zero-offset case they coincide) determine the filter coefficients. As expected, the cubic Hermite interpolation with anti-aliasing leads to the most desirable image. The image could be further improved by considering not only the kinematics predicted by the traveltimes but also the amplitude factors (Dellinger et al., 2000; Vanelle et al., 2006).

Marmousi model

The Marmousi model (Versteeg, 1994) has large velocity variations and is challenging for Kirchhoff migration with first-arrivals (Geoltrain and Brac, 1993). I apply a single-fold 2-D triangular smoothing of radius 20 m to the original model (see Figure 4.9) to remove sharp velocity discontinuities while retaining complex velocity structures. Because wave-fronts change shapes rapidly, the traveltime interpolation may be subject to inaccurate source-derivatives and provide less satisfying accuracy

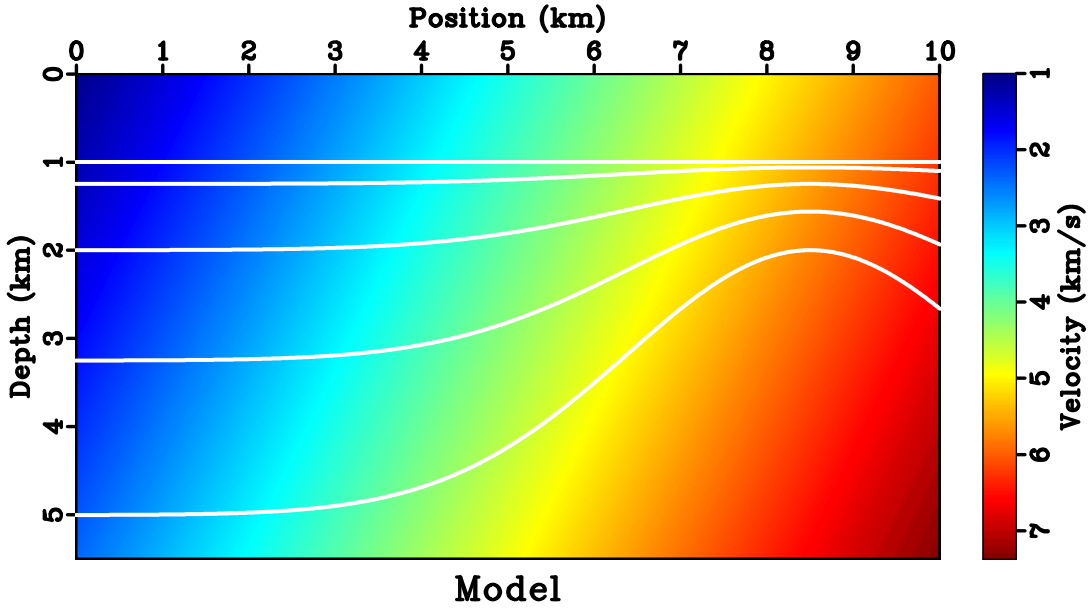


Figure 4.6: Constant-velocity-gradient background model $v(z, x) = 1.5+0.25z+0.25x$ km/s with dome shaped reflectors. eikods/migration modl

compared to that in a simple model. Although the derivative computation in the proposed eikonal-based method is source-sampling independent, in practice we should limit the interpolation interval to be sufficiently small, so that the traveltime curve could be well represented by a cubic spline. For the smoothed Marmousi model, I use a sparse source sampling of 0.2 km based on observations of the horizontal width of major velocity structures.

Figures 4.9 and 4.10 compare the traveltime interpolation errors of three methods as in Figure 4.3 for a source located at $(0, 3.1)$ km from nearby source samples at $(0, 3)$ km and $(0, 3.2)$ km. Figure 4.11 plots a reference traveltime curve for the fixed subsurface location $(2, 3.3)$ km computed by a dense eikonal solving of 4 m source spacing against curves produced by the interpolations. While these comparisons vary

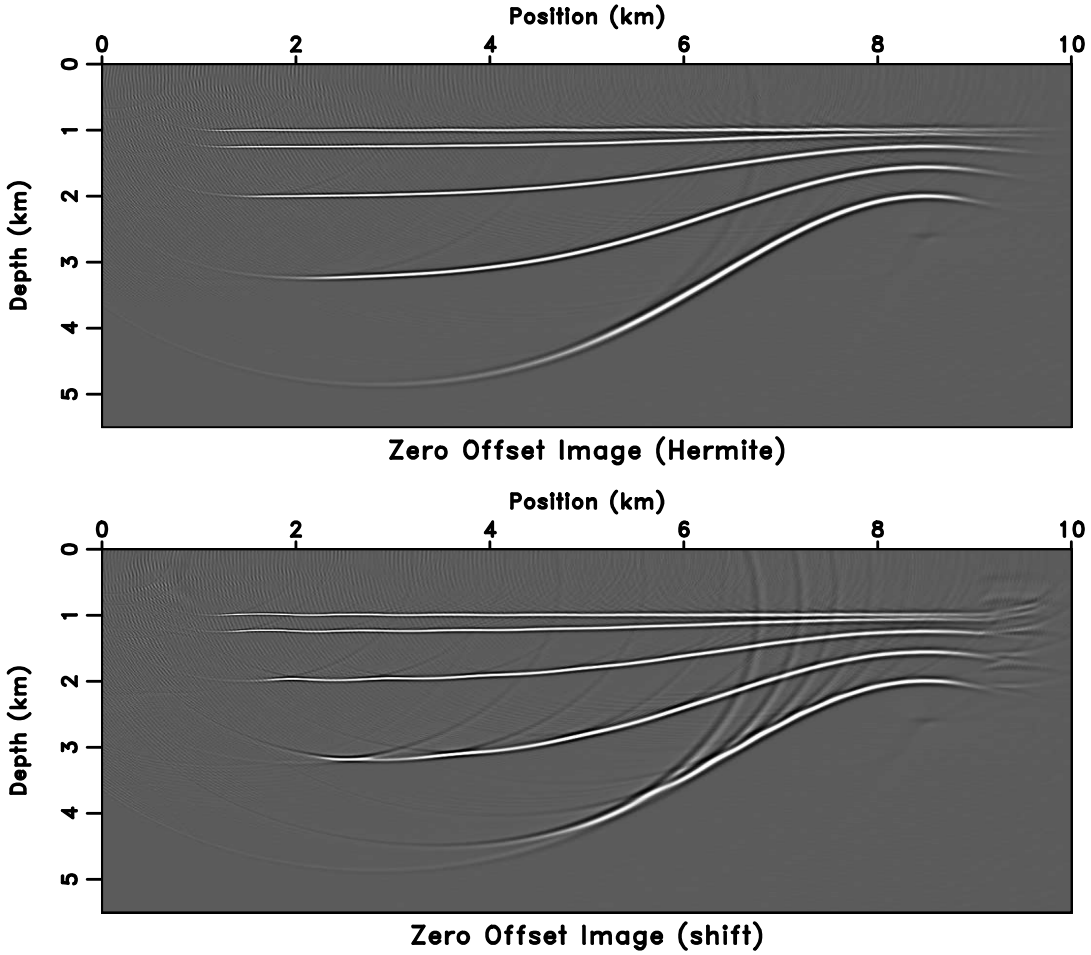


Figure 4.7: Zero-offset Kirchhoff migration image with (top) the cubic Hermite interpolation and (bottom) the shift interpolation. eikods/migration hzodmig

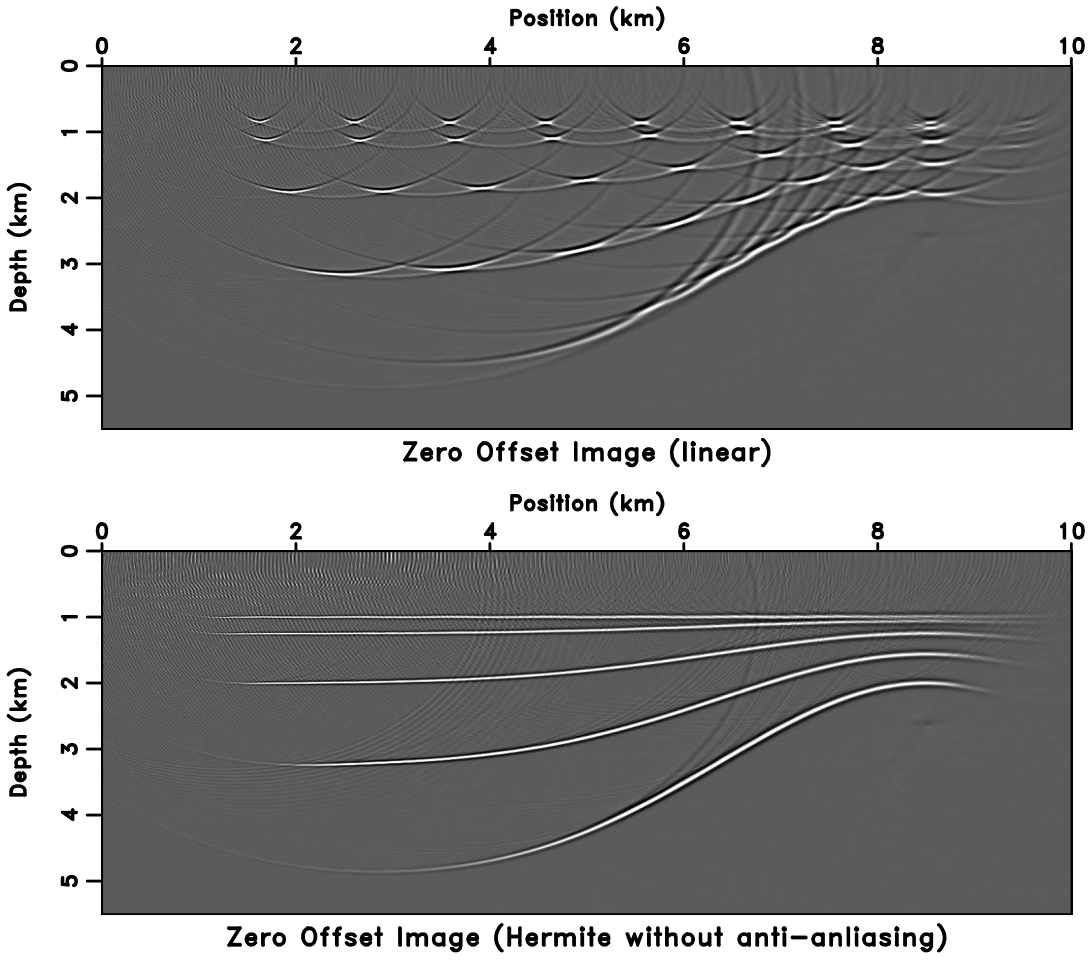


Figure 4.8: Zero-offset Kirchhoff migration image with (top) the linear interpolation and (bottom) the cubic Hermite interpolation without anti-aliasing. eikods/migration lzodmig

between different source intervals and subsurface locations, the cubic Hermite interpolation out-performs the linear and the shift interpolations except for the source singularity region. However in Figure 4.9 the errors are relatively large in the upper-left region. These errors occur due to the collapse of overlapping branches of the traveltime field (Xu et al., 2001) that causes wave-front discontinuities and undermines the assumptions of the proposed method.

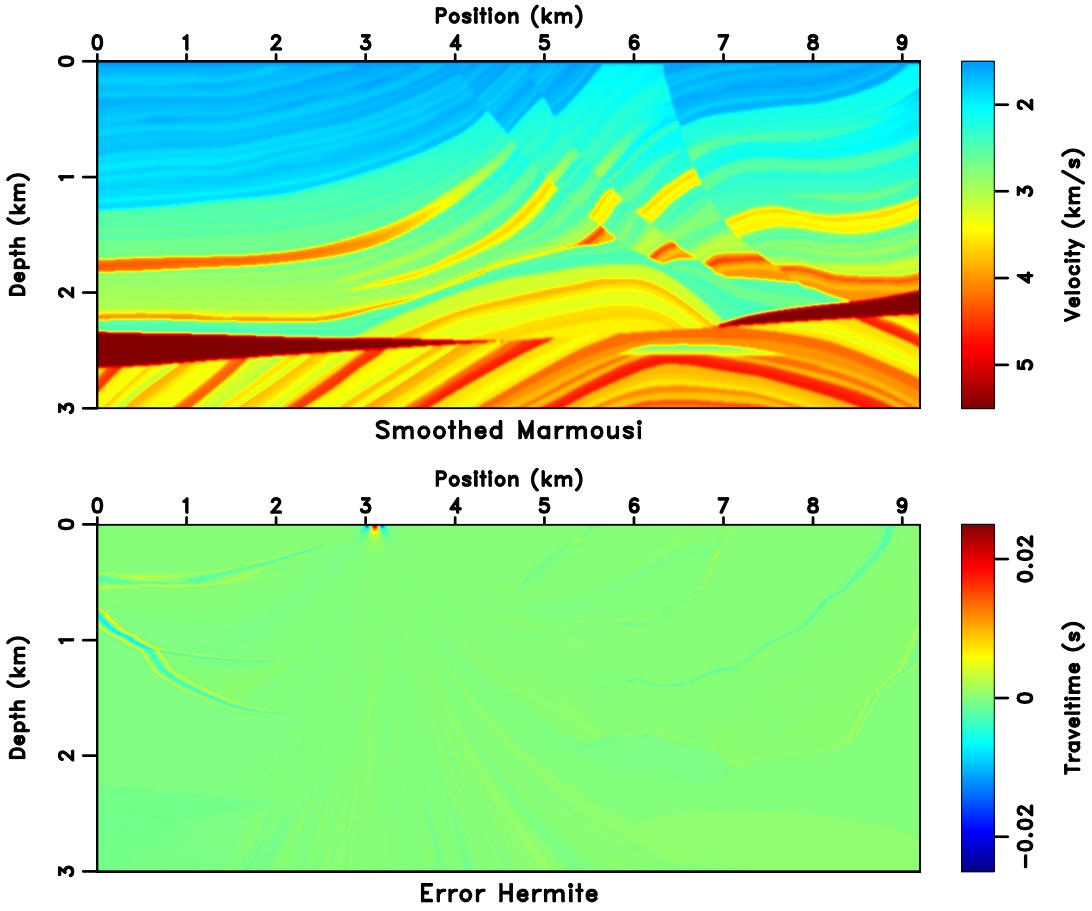


Figure 4.9: (Top) the smoothed Marmousi model. The model has a 4 m fine grid. (Bottom) the traveltime error by the cubic Hermite interpolation. eikods/marm vel

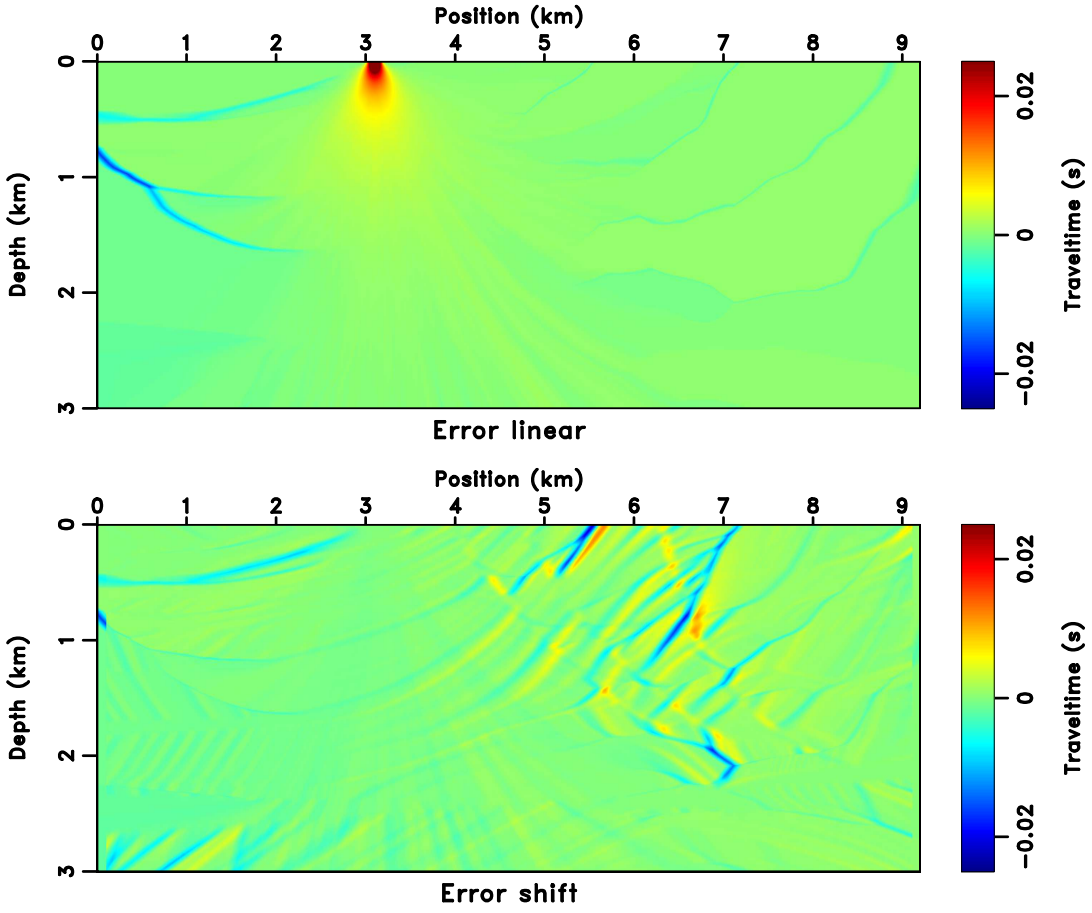


Figure 4.10: The traveltime error by (top) the linear interpolation and (bottom) the shift interpolation. eikods/marm slice

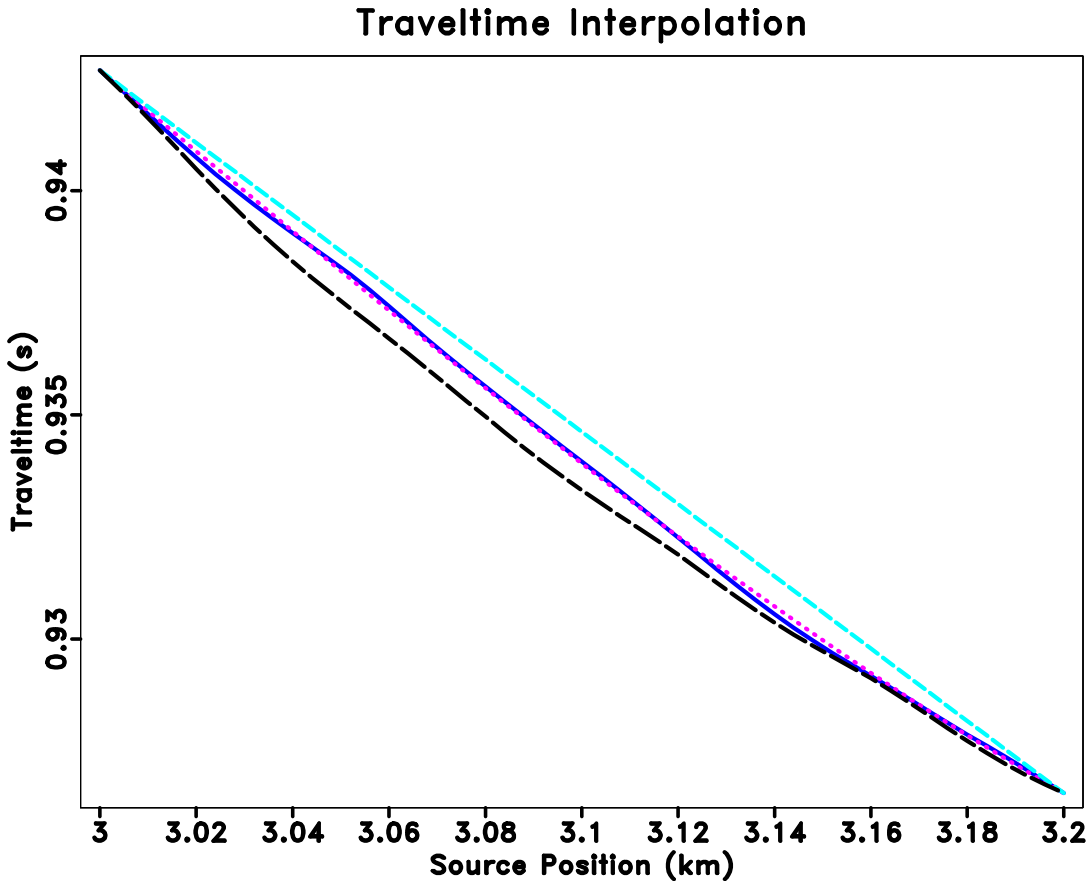


Figure 4.11: Traveltime interpolation for a fixed subsurface location. Compare between the result from a dense source sampling (solid blue), cubic Hermite interpolation (dotted magenta), linear interpolation (dashed cyan) and shift interpolation (dashed black). The $l2$ norm of the error (against the dense source sampling results) of 49 evenly interpolated sources between interval $(0, 3)$ km and $(0, 3.2)$ km for all locations but the top 100 m source singularity region are 3.9 s, 9.2 s and 11.6 s respectively. eikods/marm curve

One strategy for imaging multi-arrival wavefields with first-arrival traveltimes is the semi-recursive Kirchhoff migration proposed by Bevc (1997). It breaks the image into several depth intervals, applies Kirchhoff redatuming to the next interval, performs Kirchhoff migration from there, and so on. The small redatuming depth effectively limits the maximum traveltime and the evolving of complex waveforms before the most energetic arrivals separate from first-arrivals (see Figure 1.2 for an example). Since Kirchhoff redatuming also relies on traveltimes between datum levels, my method can be fully incorporated into the whole process. Again, for simplicity, I do not consider amplitude factors during migration. I use the Marmousi dataset with a source/receiver sampling of 25 m. Due to the source and receiver reciprocity, the receiver side interpolations are equivalent to those on the source side.

Figure 4.12 is the result of a Kirchhoff migration with eikonal solvings at each source/receiver location, i.e. no interpolation performed. Only the upper portion is well imaged. Figure 4.13 shows the image after employing the cubic Hermite interpolation with a 0.2 km sparse source/receiver sampling, which means 7 source interpolations within each interval. Even though a 7 times speed-up is not attainable in practice due to the extra computations in source-derivative and interpolation, I are still able to gain an approximately 5-fold cost reduction in traveltime computations, while keeping the image quality comparable between Figures 4.12 and 4.13. Next, following Bevc (1997), I downward continue the data to a depth of 1.5 km in three datuming steps. The downward continued data are then Kirchhoff migrated and combined with the upper portion of Figure 4.13. I keep the same 0.2 km sparse source/receiver sampling whenever eikonal solvings are required in this process. Figure 4.14 shows the image obtained by the semi-recursive Kirchhoff migration. The

target zone at approximately $(2.5, 6.5)$ km appears better imaged.

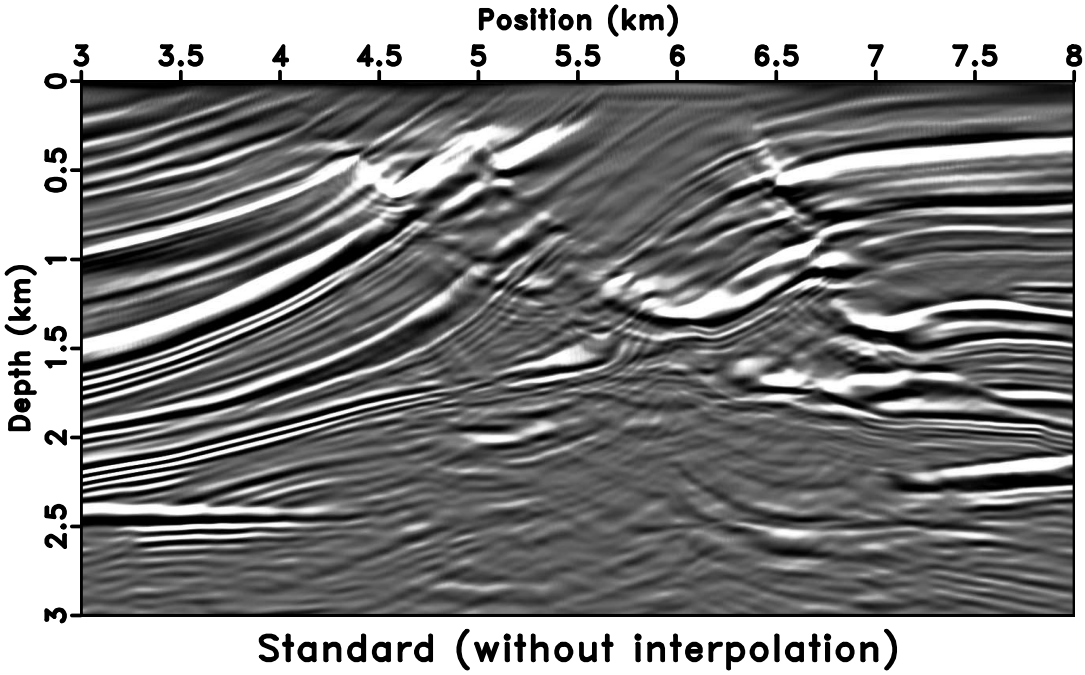


Figure 4.12: Image of Kirchhoff migration with first-arrivals (no interpolation). eikods/marm dmig0d

Synthetic dataset

Lastly I return to the synthetic dataset of Figure 1.10. Different from the Marmousi example above, a (recursive) Kirchhoff migration is not doable because the velocity model, especially in deep part where the reflections are from, is still unavailable. Following Chapter 3, it is advantageous to redatum the reflections to a datum beneath the near-surface inclusions and then continue with reflection-oriented processings such as time migration.

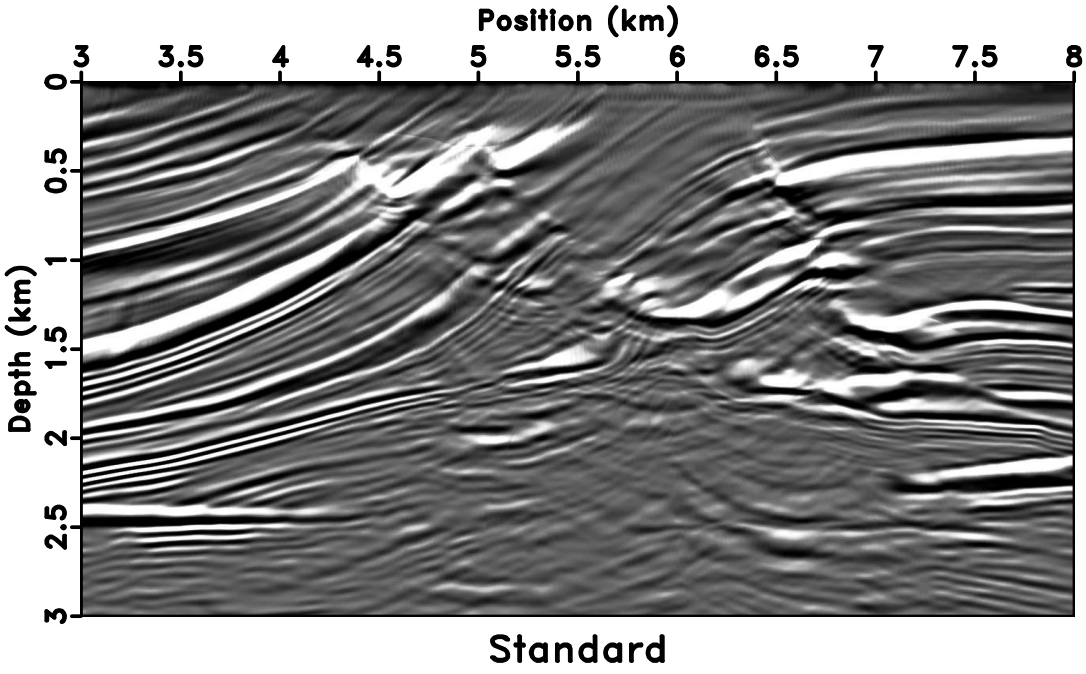


Figure 4.13: Image of Kirchhoff migration with first-arrivals and a sparse source/receiver sampling. eikods/marm dmig0

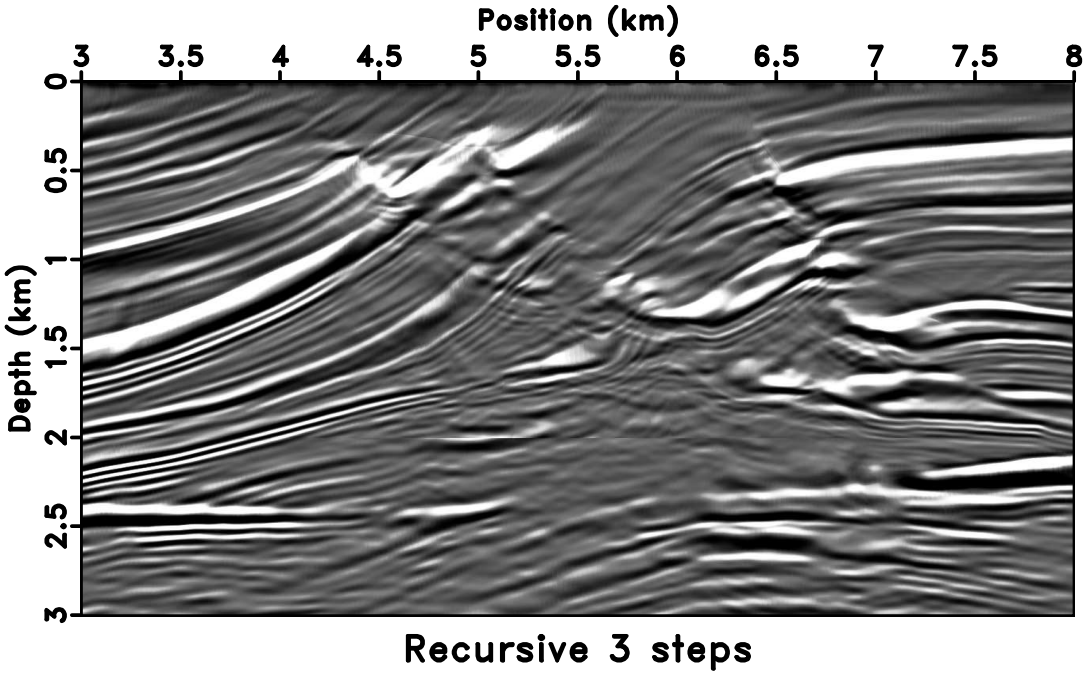


Figure 4.14: Image of semi-recursive Kirchhoff migration with a three-step redatuming from top surface to 1.5 km depth and a 0.5 km interval each time. The sparse source/receiver sampling is the same as in Figure 4.13. eikods/marm dmig2

I choose $z = 0.4$ km as the datum depth and use the near-surface model of Figure 3.19 for traveltime computations. Direct-arrivals are subtracted from the shot gathers in Figure 3.18. Then sparse sources are sampled at a 100 m interval between $(0, 0)$ km and $(0, 14)$ km, followed by traveltime interpolation with the traveltime source-derivatives to a fine source spacing of 10 m. Figure 4.15 compares the result against that of a dense source sampling. The interpolated traveltimes of Figure 4.15 are then input to Kirchhoff redatuming, as shown by a common shot gather in Figure 4.16. Note that after redatuming, the kinematic effects of near surface have been successfully removed from the prestack data. The redatumed data is ready for subsequent time-domain processings.

DISCUSSION

The proposed approach could be implemented either along with a finite-difference eikonal solver or separately. My current implementation outputs both traveltime and source-derivative, with a roughly 30% extra cost per eikonal solve compared to a FMM solver without the source-derivative functionality. An interpolation with these source-derivatives is superior to the one without them and thus enables an effective traveltime-table compression. For 3-D datasets, as both inline and crossline directions may benefit from the source-derivative and interpolation, the overall data compression savings could be significant. For instance, interpolating 10 shots within each sparse source sampling interval in both inline and crossline directions leads to an approximately 100-fold savings in traveltime storage. The method could be further combined with an interpolation within each source, for example from a coarse grid to a fine grid, for a greater traveltime compression.

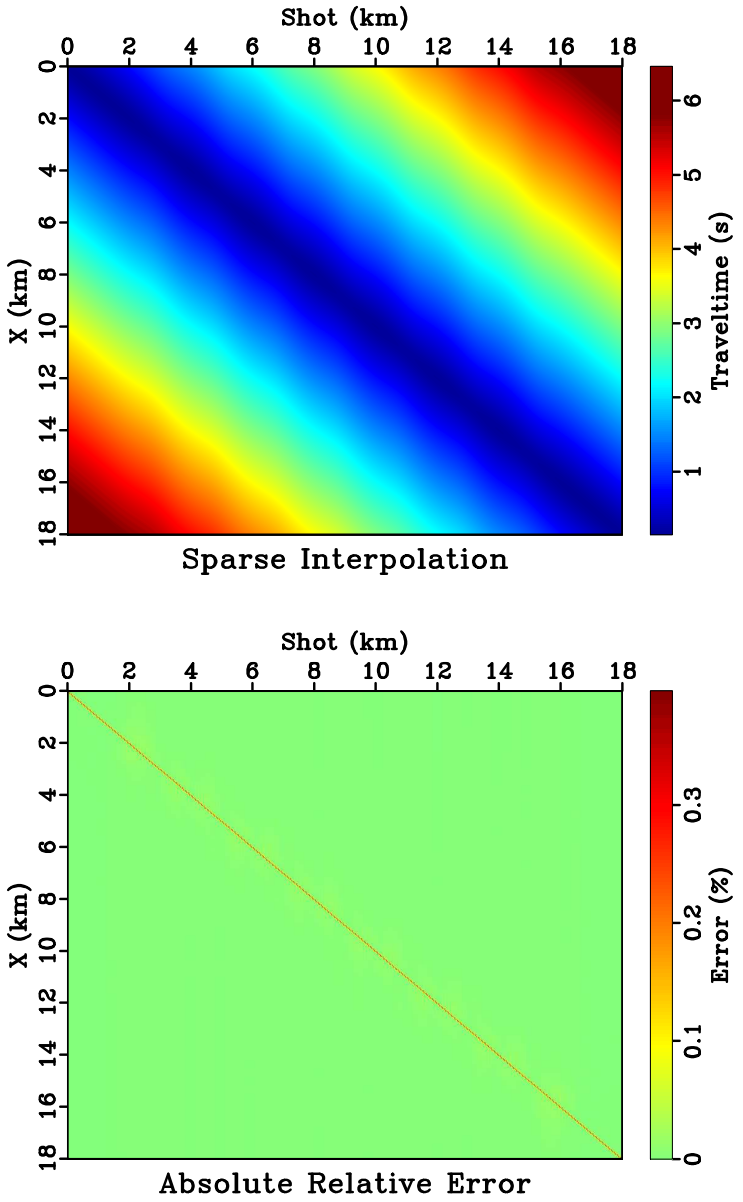


Figure 4.15: (Top) the result of sparse source interpolation. Values are traveltimes from sources at surface $z = 0$ km to the datum at depth $z = 0.4$ km. (Bottom) relative absolute errors of the traveltime interpolation against a dense source sampling of 10 m. eikods/thesis green

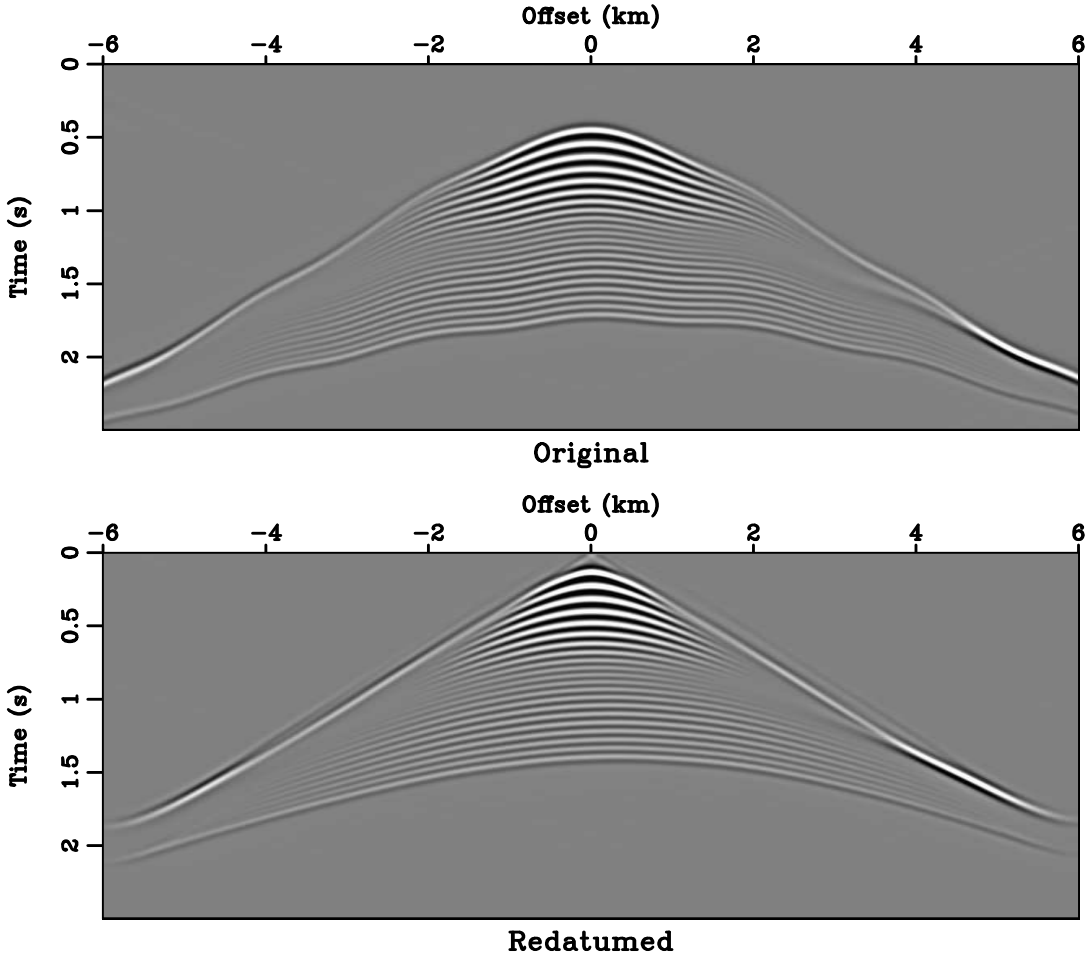


Figure 4.16: A common shot gather at $(0, 8)$ km before redatuming (top) and at $(0.4, 8)$ km after redatuming (bottom). eikods/thesis redat

While my implementation is for first-arrivals only, the governing equations are valid also for other characteristic branches, for example the most energetic arrivals. However, an underlying assumption of the proposed method is a continuous change in the wave-front of selected arrivals within individual sources. For first-arrivals, this condition always holds valid. However, the most energetic wave-front can be more complicated than that of first-arrival, for example only piece-wise continuous, which may lead to a potential degradation in accuracy. Nichols (1994) showed the most energetic discontinuous wave-fronts in the Marmousi model. Another assumption is that the traveltime source-derivatives are continuous between nearby sources. This condition breaks down when multi-pathing takes place. Vanelle and Gajewski (2002) suggested to smooth traveltimes around the discontinuities in order to overcome this limitation. In theory, one can try to identify the discontinuities and only perform interpolation within individual continuous pieces by using the eikonal-based source-derivatives. By doing so, one should be able to recover branch jumping in interpolated traveltimes, but only for those locations within the identified continuous pieces. For the discontinuities themselves as well as the gaps between them, additional eikonal solving may be required. An efficient implementation of this strategy remains open for future research.

CONCLUSION

I have shown an application of computing traveltime source-derivatives in Kirchhoff migration. For first-arrivals, a cubic Hermite traveltime interpolation using the

first-order source-derivatives speeds up computation and reduces storage without noticeably sacrificing accuracy. Anti-aliasing is another direct application of traveltime source-derivatives which can be easily incorporated into Kirchhoff migration.

Generalization of the method to 3-D is straightforward. The computed derivative attributes may find applications in other areas besides the kinematic-only Kirchhoff migration shown in this chapter. However, an extension to multi-arrival traveltimes needs further investigation.

Chapter 5

A robust approach to time-to-depth conversion and interval velocity estimation from time migration in the presence of lateral velocity variations

INTRODUCTION

Time-domain seismic processing has been a popular and effective tool in areas with mild lateral velocity variations (Yilmaz, 2001; Robein, 2003; Bartel et al., 2006). Time migration produces images in the time coordinate as opposed to the depth coordinate. The time coordinate, along with time-migration velocities, is determined during time migration by optimizing image qualities. However, it is highly sensitive to lateral velocity changes (Black and Brzostowski, 1994). Therefore, the time-migrated image is often distorted (Hubral, 1977; Lynn and Claerbout, 1982; Iversen, 2004; Cameron et al., 2007, 2008, 2009). For this reason, in many cases time migrations are inadequate for accurate geological interpretation of subsurface structures. On the other hand, time-migration velocities can be conveniently and efficiently estimated either by repeated migrations (Yilmaz et al., 2001) or by velocity continuation (Fomel, 2003). Depth migration can handle general media and output images in regular Cartesian depth coordinates. But it requires an accurate interval velocity model construction, which is in practice both challenging and time-consuming. An iterative process of

Parts of this chapter will appear in Li, S., and S. Fomel, 2014, A robust approach to time-to-depth conversion and interval velocity estimation from time migration in the presence of lateral velocity variations: Geophysical Prospecting, accepted. This work is done under the supervision of Fomel.

tomographic updates is often employed, where a good initial interval velocity model is crucial for achieving the global minimum. It is thus of great interest to convert the time-migration velocity to the depth domain in order to unravel inherent distortions in time-domain images and to provide a reasonable starting model for building depth-imaging velocities.

The relationship between time and depth coordinates was first explained by Hubral (1977) through the concept of *image rays*. An interval velocity model can be converted to one in the time domain by tracing image rays that dive into earth with slowness vector normal to the surface. The time coordinate is then defined by the traveltime along image rays and its surface location (Larner et al., 1981; Robein, 2003). However this process is not trivially revertible and it does not reveal directly the connection the between time-migration velocity and the interval velocity. According to Dix (1955), in a layered medium where $v = v(z)$, image rays run straightly downward and the time-migration velocities are the root-mean-square (RMS) velocities appearing in a truncated Taylor approximation for traveltimes. The Dix inversion formula is exact in $v(z)$ medium and the conversion between time- and depth-domain attributes in this case is theoretically straightforward. However, this is not the case in the presence of lateral velocity variations.

Even a mild lateral velocity variation can cause image rays to bend and invalidate the $v(z)$ assumption in Dix inversion. Cameron et al. (2007, 2008, 2009) studied this problem and established theoretical relations between the time-migration velocity and the seismic velocity in depth for general media using the paraxial ray-tracing theory. In 2-D, the conventional Dix velocity is equal to the ratio of the interval

velocity and the geometrical spreading of image rays. This is consistent with the Dix formula because when $v = v(z)$ the geometrical spreading equals to one everywhere. In order to carry out the time-to-depth conversion in the presence of lateral velocity variations, one can solve a nonlinear partial differential equation (PDE) of elliptic type with boundary conditions on the surface. The problem is mathematically ill-posed. Cameron et al. (2007, 2008) adopted a two-step numerical procedure for the time-to-depth conversion. First, they used a Lax-Friedrichs-like finite-difference method or a spectral Chebyshev method to solve for geometrical spreading in the time coordinate. Next, they performed a Dijkstra-like solver motivated by fast-marching method (FMM) for velocity conversion and coordinate mapping. In this approach, it is crucial to preserve the development of low harmonics and damp the high harmonics during the first stage. Iversen and Tygel (2008) discussed an extension of the time-to-depth conversion problem along 2-D profiles in 3-D. Similarly to Cameron et al. (2007, 2008), an essential part of their algorithm is the time-stepping (integration) along image-rays. The robustness of these methods may not be satisfactory in practice because of stability issues that arise from the ill-posed nature of the problem (Cameron et al., 2009).

In this chapter, I start with the theoretical relations derived by Cameron et al. (2007) but cast the original problem in a nonlinear iterative optimization framework. This idea is motivated by the observation that, for arbitrary depth velocity model, two of the PDEs can be always satisfied, while the remaining one associated with image ray geometrical spreading can be rewritten as a cost functional that indicates errors in interval velocity. A key benefit in the new formulation is that each linearization update on the interval velocity will contain information from the whole computational

domain. In conparison, the previous approach can only consider information locally at each time step. Another advantage of the optimization approach is the ability to adopt regularization in order to deal with the ill-posedness issue. I first show theoretical derivations of all necessary components involved in the new approach. Next, I develop a numerical implementation for 2-D time-to-depth conversion. Finally, I test the algorithm on both synthetic and field data examples. Some discussions on the proposed method are given in concluding this chapter.

THEORY

For completeness, I will first review some basic concepts related to the time-to-depth conversion problem. Then I show the theoretical derivation for an optimization formulation of the problem and a corresponding inversion procedure. For simplicity, I restrict the analysis to the 2-D case.

Connection between time- and depth-domain attributes

As discussed in Chapter 1, Kirchhoff migration can be performed in both depth domain 1.5 and time domain 1.12. In Figure 1.9, I illustrate image rays in 2-D and a *forward* mapping from depth coordinate $\mathbf{x} = (z, x)$ to time coordinate $\hat{\mathbf{x}} = (t_0, x_0)$ (Hubral, 1977). Under the assumption of no caustics, for each subsurface location (z, x) I consider the image ray from (z, x) to the surface, where it emerges at point $(0, x_0)$, with slowness vector normal to the surface. Here x_0 is the location of the image ray at the earth surface and is a scalar. t_0 is the traveltime along this image ray between (z, x) and $(0, x_0)$. The forward mapping $t_0(z, x)$ and $x_0(z, x)$ can be done

with a knowledge of interval velocity $v(z, x)$. A unique *inverse* mapping $z(t_0, x_0)$ and $x(t_0, x_0)$ also exists that enables us to directly map the time-migrated image to depth.

In a $v(z)$ medium, v_m corresponds to the root-mean-square (RMS) velocity:

$$v_m(t_0) = \sqrt{\frac{1}{t_0}\int_0^{t_0} v^2(z(t))\,dt}\ . \tag{5.1}$$

This RMS v_m is optimum in a sense that it provides the best traveltime approximation in equation 1.11. A time-to-depth velocity conversion can be done by first applying the Dix inversion formula (Dix, 1955), which is theoretically exact in this case:

$$v_d(t) = \sqrt{\frac{d}{dt_0}(t_0 v_m^2(t_0))}\ , \tag{5.2}$$

followed by performing a simple conversion from $v_d(t)$ to $v(z)$ according to $z = \frac{1}{2}\int_0^t v_d(t)dt$.

In equations 5.1 and 5.2, there is no dependency on x_0 or x because image rays are identical in the lateral direction. For an arbitrary medium, image rays will bend as they travel. Therefore, in general, v_m is a function of both t_0 and x_0 and no longer satisfies the simple expression 5.1, which limits the applicability of Dix formula. Cameron et al. (2007) proved that the seismic velocity and the Dix velocity in this case are connected through geometrical spreading Q of image rays:

$$v_d(t_0, x_0) \equiv \sqrt{\frac{\partial}{\partial t_0}(t_0 v_m^2(t_0, x_0))} = \frac{v(z(t_0, x_0), x(t_0, x_0))}{Q(t_0, x_0)}\ . \tag{5.3}$$

In equation 5.3, the generalized Dix velocity is defined by Cameron et al. (2007) in a way similar to equation 5.2 with a change from d/dt_0 to a partial differentiation with

respect to t_0. The quantity Q is related to $x_0(z,x)$ using its definition (Popov, 2002), as follows:

$$|\nabla x_0|^2 = \frac{1}{Q^2} \; . \tag{5.4}$$

Combining equations 5.3 and 5.4 results in

$$|\nabla x_0(z,x)|^2 = \frac{v_d^2(t_0(z,x), x_0(z,x))}{v^2(z,x)} \; . \tag{5.5}$$

The traveltimes along image rays obey the eikonal equation (Hubral, 1977; Chapman, 2002), thus

$$|\nabla t_0(z,x)|^2 = \frac{1}{v^2(z,x)} \; . \tag{5.6}$$

Finally, since x_0 remains constant along each image ray, there is an orthogonality condition between gradients of t_0 and x_0 (Cameron et al., 2007):

$$\nabla t_0(z,x) \cdot \nabla x_0(z,x) = 0 \; . \tag{5.7}$$

Equations 5.5, 5.6 and 5.7 form a system of nonlinear PDEs for $t_0(z,x)$ and $x_0(z,x)$. The input is $v_d(t_0,x_0)$, estimated from $v_m(t_0,x_0)$ by equation 5.3. Solving a boundary-value problem for the PDEs should provide $v(z,x)$, as well as $t_0(z,x)$ and $x_0(z,x)$. Because seismic acquisitions are limited to the earth surface, I can only use boundary conditions at the surface. For a rectangular Cartesian domain with $z = 0$ being the surface, the boundary conditions are

$$\left\{ \begin{array}{rcl} t_0(0,x) & = & 0 \; , \\ x_0(0,x) & = & x \; . \end{array} \right. \tag{5.8}$$

Ill-posedness of the time-to-depth conversion problem

Let us consider the problem of solving for $z(t_0, x_0)$ and $x(t_0, x_0)$ instead of $t_0(z, x)$ and $x_0(z, x)$ by recasting equations 5.5, 5.6 and 5.7 in the time coordinate:

$$\left(\frac{\partial x}{\partial x_0}\right)^2 + \left(\frac{\partial z}{\partial x_0}\right)^2 = \frac{v^2(z(t_0, x_0), x(t_0, x_0))}{v_d^2(t_0, x_0)} \ , \tag{5.9}$$

$$\left(\frac{\partial x}{\partial t_0}\right)^2 + \left(\frac{\partial z}{\partial t_0}\right)^2 = v^2(z(t_0, x_0), x(t_0, x_0)) \ , \tag{5.10}$$

$$\frac{\partial x}{\partial x_0}\frac{\partial x}{\partial t_0} + \frac{\partial z}{\partial x_0}\frac{\partial z}{\partial t_0} = 0 \ . \tag{5.11}$$

The corresponding boundary conditions are

$$\begin{cases} z(0, x_0) &= 0 \ , \\ x(0, x_0) &= x_0 \ . \end{cases} \tag{5.12}$$

From equation 5.11

$$\frac{\partial x}{\partial t_0} = -\frac{\partial z}{\partial x_0}\frac{\partial z}{\partial t_0} \Big/ \frac{\partial x}{\partial x_0} \ . \tag{5.13}$$

Substituting equation 5.13 into equation 5.10 and combining with equation 5.9 produces

$$\frac{\partial z}{\partial t_0} = v_d \frac{\partial x}{\partial x_0} \ ; \tag{5.14}$$

$$\frac{\partial x}{\partial t_0} = -v_d \frac{\partial z}{\partial x_0} \ , \tag{5.15}$$

where I assume that both $\partial z/\partial t_0$ and $\partial x/\partial x_0$ remain positive and there is no caustics.

On the first glance, equations 5.14 and 5.15 seem suitable for numerically extrapolating $x(t_0, x_0)$ and $z(t_0, x_0)$ in t_0 direction using the boundary conditions 5.12. After such an extrapolation, one would be able to reconstruct $v(t_0, x_0)$ from equation 5.10 and thus solve the original problem. However, by further decoupling the system using the equivalence of the second-order mixed derivatives, I discover

that the underlying PDEs are elliptic. For instance, applying $\partial/\partial x_0$ to both sides of equation 5.15 results in

$$\frac{\partial^2 x}{\partial t_0 \partial x_0} = -\frac{\partial}{\partial x_0}\left(v_d \frac{\partial z}{\partial x_0}\right) . \tag{5.16}$$

Meanwhile, dividing by v_d and applying $\partial/\partial t_0$ to both sides of equation 5.14 leads to

$$\frac{\partial}{\partial t_0}\left(\frac{1}{v_d}\frac{\partial z}{\partial t_0}\right) = \frac{\partial^2 x}{\partial x_0 \partial t_0} . \tag{5.17}$$

Comparing equations 5.16 and 5.17, I find

$$\frac{\partial}{\partial x_0}\left(v_d \frac{\partial z}{\partial x_0}\right) + \frac{\partial}{\partial t_0}\left(\frac{1}{v_d}\frac{\partial z}{\partial t_0}\right) = 0 . \tag{5.18}$$

Analogously,

$$\frac{\partial}{\partial x_0}\left(v_d \frac{\partial x}{\partial x_0}\right) + \frac{\partial}{\partial t_0}\left(\frac{1}{v_d}\frac{\partial x}{\partial t_0}\right) = 0 . \tag{5.19}$$

Solving equations 5.18 and 5.19 with the Cauchy-type boundary conditions 5.12 is an ill-posed problem. The missing boundary conditions on sides of the computational domain other than 5.8 can induce numerical instability when extrapolating in t_0 (equivalently z). Instead, I consider an alternative formulation of the problem in the following section.

The optimization formulation

Given boundary conditions 5.8, equation 5.6 stands for the traveltime t_0 of a velocity model with a plane-wave source at the surface. For a given t_0, equation 5.7 is a first-order PDE on x_0 and thus computation of x_0 is straightforward. The idea

of the optimization formulation is inspired by a natural logic: if the resulting x_0 does not satisfy equation 5.5, we need to modify v in a way such that the misfit decreases, and repeat the process until convergence.

Mathematically, I define a cost function $f(z,x)$ based on equation 5.5:

$$f(z,x) = \nabla x_0 \cdot \nabla x_0 - v_d^2\, w \ , \tag{5.20}$$

where for convenience I use slowness-squared $w(z,x)$ instead of v. Note f is dimensionless. The discretized form of equation 5.20 reads

$$\mathbf{f} = (\nabla \mathbf{x_0} \cdot \nabla)\, \mathbf{x_0} - \mathbf{v_d} \star \mathbf{v_d} \star \mathbf{w} \equiv \mathbf{L}_{x_0}\, \mathbf{x_0} - \mathbf{v_d} \star \mathbf{v_d} \star \mathbf{w} \ . \tag{5.21}$$

In equation 5.21, $\mathbf{f}, \mathbf{x_0}, \mathbf{v_d}$ and $\mathbf{w}$ are all column vectors after discretizing the computational domain (z,x). For example, $\mathbf{x_0}$ is the discretized column vector of $x_0(z,x)$. The vector $\mathbf{v_d}$ may require interpolation because it is in (t_0, x_0) while the discretization is in (z,x). The interpolation can be done after forward mapping from (z,x) to (t_0,x_0) at current velocity model. The symbol $\star$ stands for an element-wise vector-vector multiplication. Finally, I denote an operator which is a matrix $\mathbf{L}_{x_0} = \nabla \mathbf{x_0} \cdot \nabla$. This operator has the same form as $\nabla T \cdot \nabla$ introduced in Chapter 2, and I will detail its implementation in the next section.

As is common in many optimization problems, I seek to minimize the least-squares norm of $\mathbf{f}$:

$$E[\mathbf{w}] = \frac{1}{2}\mathbf{f}^T\mathbf{f} \ , \tag{5.22}$$

where the superscript T stands for transpose. The Newton's method in optimization

can be derived by linearizing the cost function in equation 5.21:

$$\frac{\partial f}{\partial w} = 2\,(\nabla x_0 \cdot \nabla)\,\frac{\partial x_0}{\partial w} - 2\,v_d\,w\,\frac{\partial v_d}{\partial w} - v_d^2\;. \tag{5.23}$$

The Fréchet derivative matrix $\mathbf{J}$ required by inversion is the discretized form of equation 5.23, i.e., $\mathbf{J} = \partial\mathbf{f}/\partial\mathbf{w}$.

Using the same notations as introduced in equation 5.21, after discretization equation 5.23 becomes

$$\mathbf{J} \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{w}} = 2\,\mathbf{L}_{x_0}\,\frac{\partial \mathbf{x_0}}{\partial \mathbf{w}} - 2\,\mathrm{diag}(\mathbf{v_d} \star \mathbf{w})\,\frac{\partial \mathbf{v_d}}{\partial \mathbf{w}} - \mathrm{diag}(\mathbf{v_d} \star \mathbf{v_d})\;, \tag{5.24}$$

where the operator diag() expands a vector into a diagonal matrix. Because v_d is in time-domain (t_0, x_0), I need to apply the chain-rule for its derivative with respect to w, i.e.,

$$\frac{\partial \mathbf{v_d}}{\partial \mathbf{w}} = \frac{\partial \mathbf{v_d}}{\partial \mathbf{t_0}}\,\frac{\partial \mathbf{t_0}}{\partial \mathbf{w}} + \frac{\partial \mathbf{v_d}}{\partial \mathbf{x_0}}\,\frac{\partial \mathbf{x_0}}{\partial \mathbf{w}}\;, \tag{5.25}$$

where $\mathbf{t_0}$ is the discretized column vector of $t_0(z, x)$. According to equation 5.7 and after denoting another matrix operator $\mathbf{L}_{t_0} = \nabla \mathbf{t_0} \cdot \nabla$, I find

$$\frac{\partial \mathbf{x_0}}{\partial \mathbf{w}} = -(\nabla \mathbf{t_0} \cdot \nabla)^{-1}(\nabla \mathbf{x_0} \cdot \nabla)\,\frac{\partial \mathbf{t_0}}{\partial \mathbf{w}} \equiv -\mathbf{L}_{t_0}^{-1}\mathbf{L}_{x_0}\,\frac{\partial \mathbf{t_0}}{\partial \mathbf{w}}\;. \tag{5.26}$$

Another differentiation of equation 5.6 leads to the linearized eikonal equation 1.8

$$\frac{\partial \mathbf{t_0}}{\partial \mathbf{w}} = \frac{1}{2}(\nabla \mathbf{t_0} \cdot \nabla)^{-1} = \frac{1}{2}\mathbf{L}_{t_0}^{-1}\;. \tag{5.27}$$

The operator $\mathbf{L}_{t_0}^{-1}$ in equations 5.26 and 5.27 has been developed in Chapter 2 and is extended in Chapter 3 for double-square-root eikonal-based tomography. Finally, by inserting equations 5.25 through 5.27 into 5.24, I complete the derivation of the

Fréchet derivative matrix:

$$\begin{aligned}\mathbf{J} &= -\mathbf{L}_{x_0}\mathbf{L}_{t_0}^{-1}\mathbf{L}_{x_0}\mathbf{L}_{t_0}^{-1} - \mathrm{diag}(\mathbf{v_d} \star \mathbf{w})\frac{\partial \mathbf{v_d}}{\partial \mathbf{t_0}}\mathbf{L}_{t_0}^{-1} \\ &+ \mathrm{diag}(\mathbf{v_d} \star \mathbf{w})\frac{\partial \mathbf{v_d}}{\partial \mathbf{x_0}}\mathbf{L}_{t_0}^{-1}\mathbf{L}_{x_0}\mathbf{L}_{t_0}^{-1} - \mathrm{diag}(\mathbf{v_d} \star \mathbf{v_d}) \ . \end{aligned} \tag{5.28}$$

Equation 5.28 means that $\mathbf{J}$ is a cascade and summation of several parts. An update $\delta\mathbf{w}$ at current $\mathbf{w}$ is found by solving the following normal equation arising from Gauss-Newton approach (Björck, 1996):

$$\delta\mathbf{w} = \left[\mathbf{J}^T\mathbf{J}\right]^{-1}\mathbf{J}^T(-\mathbf{f}) \ . \tag{5.29}$$

Equations 5.22 and 5.29 together indicate a nonlinear inversion procedure for solving the original system of PDEs 5.5, 5.6 and 5.7. The inversion is similar to the traveltime tomography but with more complexity. The cost 5.20 can be interpreted as difference between modeled and observed geometrical spreadings. However, both of them depend on the model v, while in traveltime tomography the observed arrival times are independent of v. The forward modeling in my case involves two steps, which construct a curvilinear coordinate that is sensitive to lateral velocity variations. On the other hand, the forward modeling in traveltime tomography consists only one step, i.e., solving equation 5.6 with a point-source boundary condition. Last but not least, unlike traveltime tomography, we have receivers everywhere, except for some null spaces created during numerical implementation, as I will discuss later.

Note that the inversion is applied directly for seismic slowness-squared $\mathbf{w}$ (and thus velocity), and the update $\delta\mathbf{w}$ will incorporate mutual dependency of velocities

throughout the domain, especially in the depth direction, as physically honored by image rays. Previous methods (Iversen, 2004; Cameron et al., 2007, 2008) that rely on time-stepping in t_0 may fail to sufficiently account for such dependencies.

Before introducing a numerical implementation, I need to point out several important facts and assumptions that make a successful time-to-depth conversion possible by the proposed method:

- Caustics must be excluded from the computational domain. In regions where caustics develop, the gradient ∇x_0 goes to infinity and the cost function is not well-defined. For all numerical examples in this chapter, I do not encounter this issue. In the Discussion section, I provide a possible strategy to cope with this limitation;

- According to derivations in equation 5.25, the calculation of $\delta\mathbf{w}$ depends on values of $\partial\mathbf{v_d}/\partial\mathbf{t_0}$ and $\partial\mathbf{v_d}/\partial\mathbf{x_0}$. Thus the input v_d should be differentiable. This requirement can be satisfied during v_m estimation;

- Similarly to all nonlinear inversions, the proposed method asks for a prior model that is sufficiently close to desired model at global minimum $E = 0$. Meanwhile, to guarantee stability and a smooth output, some form of regularization should be imposed during inversion (Engl et al., 1996);

- My formulation does not change the ill-posed nature of the original problem. One assumption is that condition 5.8 describes all *in-flow* domain boundaries of t_0 and x_0. In other words, the image rays are only allowed to be either parallel to or exiting (*out-flow*) all other boundaries except the surface.

For the prior model, I adopt the Dix-inverted model. In other words, I seek to refine the interval velocity given by equation 5.2 by taking the geometrical spreading of image rays into consideration.

IMPLEMENTATION

Below I outline sequentially the steps involved in computing one linearization update:

1. Given current $v(z,x)$, solve equation 5.6 for $t_0(z,x)$ with $t_0(0,x)=0$;
2. Given $t_0(z,x)$ and $x_0(0,x)=x$, solve equation 5.7 for $x_0(z,x)$;
3. Given $t_0(z,x)$ and $x_0(z,x)$, interpolate $v_d(z,x)$ from $v_d(t_0,x_0)$ and compute $f(z,x)$ based on equation 5.20;
4. Assemble linear operator 5.28 and solve equation 5.29 for $\delta w(z,x)$.

First, I use FMM to solve the eikonal equation 5.6 by initializing a plane-wave source at $z=0$. Computation for x_0 can be incorporated into t_0 by adopting the upwind finite-differences of t_0 for equation 5.7. In Figure 5.1, consider a currently updated grid point C during forward modeling of t_0. If it has only one upwind neighbor A that is inside the wave-front, $t_0(C) > t_0(A)$, then the image ray must be aligned with grid segment AC and

$$x_0(C) = x_0(A)\ . \tag{5.30}$$

If C has two upwind neighbors A and B, $t_0(C) > \max\{t_0(A), t_0(B)\}$, and they are both inside the wave-front, then the image ray must intersect the simplex ABC from

an angle. In this case, I compute $x_0(C)$ from

$$\frac{(t_0(C)-t_0(A))(x_0(C)-x_0(A))}{(\Delta z)^2}+\frac{(t_0(C)-t_0(B))(x_0(C)-x_0(B))}{(\Delta x)^2}=0\ , \qquad (5.31)$$

where Δz and Δx are the uniform grid spacings. Because $\mathbf{x_0}$ at certain grid points are calculated by one-sided scenario 5.30, $\mathbf{L}_{x_0}$ there contains all zeros. Consequently, an evaluation of the cost $\mathbf{f}$ at these locations with $\mathbf{L}_{x_0}\,\mathbf{x_0}$ becomes inaccurate. I exclude these regions from $\mathbf{f}$ and expect inversion to fill the spaces.

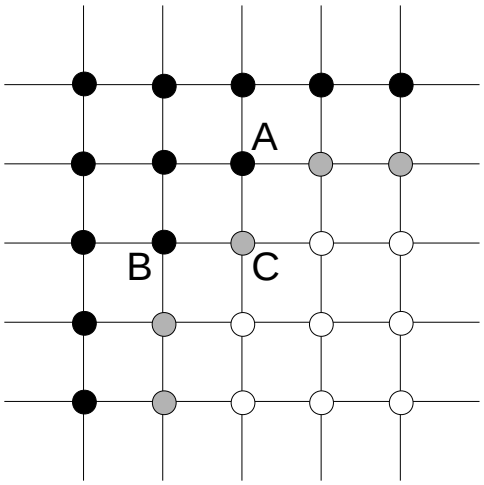


Figure 5.1: A modified FMM for forward modeling. Black dots represent region that is *IN*, gray dots are *FRONT* and white dots are region of *OUT*. See section on FMM in Chapter 2 for explanations on these labels. time2dep/figs fmm

Next, I apply simple bilinear interpolation for $v_d(z,x)$ and estimate $\delta\mathbf{w}$ by solving equation 5.29 using shaping regularization (Fomel, 2007). I use a triangular

smoother with adjustable size as the shaping operator. I find in numerical tests that shaping significantly improves convergence speed compared to that of the traditional Tikhonov regularization (Tikhonov, 1963) with gradient operators. I also observe that without regularization the model update is undesirably oscillatory.

Finally, I choose the upwind finite-differences 2.2 based on $\mathbf{t_0}$ for both $\mathbf{L}_{t_0}$ and $\mathbf{L}_{x_0}$. It is important that $\mathbf{L}_{x_0}$ must be triangularized by the upwind ordering of t_0 instead of x_0 because of equations 5.30 and 5.31. The computational cost in equation 5.29 is further reduced by adopting the method of conjugate gradients (Hestenes and Stiefel, 1952), which requires the adjoint $\mathbf{J}^T$ according to equation 5.28. From Chapter 2, applying $\mathbf{J}$ and $\mathbf{J}^T$ involves only triangularized matrix-vector multiplications and is therefore inexpensive.

EXAMPLES

Constant velocity gradient model

In a constant velocity gradient medium

$$v(z,x) = v_0 + g_z z + g_x x \ , \tag{5.32}$$

all attributes involved in time-to-depth conversion have analytical solutions. I first trace image rays in the depth coordinate for $z(t_0, x_0)$ and $x(t_0, x_0)$. Then I carry out an inversion to find $t_0(z,x)$ and $x_0(z,x)$. The Dix velocity can be obtained at last following equations 5.3 and 5.4.

Continuing from equation 5.32, I write the velocity in a coordinate relative to the image ray

$$v(z,x) = v_0 + g_z z + g_x x = \tilde{v}_0 + \mathbf{g}\cdot(\mathbf{x}-\mathbf{x_0}) \ , \tag{5.33}$$

where $\mathbf{g} = [g_z, g_x]^T$ and $\tilde{v}_0 = v_0 + g_x x_0$. The image ray satisfies

$$\begin{cases} \mathbf{x_0} &= [0, x_0]^T \ , \\ \mathbf{p_0} &= [\tilde{v}_0^{-1}, 0]^T \ , \\ t_0 &= t \ . \end{cases} \tag{5.34}$$

Here I denote ray parameter $\mathbf{p} = \nabla t$ and $\mathbf{p_0}$ is the ray parameter at source. The Hamiltonian for ray tracing reads $H(\mathbf{x},\mathbf{p}) = \mathbf{p}\cdot\mathbf{p} - v^{-2} \equiv 0$. H can be differentiated with respect to an auxiliary parameter ξ defined along the ray

$$\frac{dH}{d\xi} = 2\frac{d\mathbf{p}}{d\xi}\cdot\mathbf{p} + \frac{2}{v^3}\frac{dv}{d\xi} = 2\frac{d\mathbf{p}}{d\xi}\cdot\mathbf{p} + \frac{2}{v^3}\nabla v\frac{d\mathbf{x}}{d\xi} \equiv 0 \ . \tag{5.35}$$

From equation 5.35 I establish the ray tracing system, as follows:

$$\begin{cases} d\mathbf{x}/d\xi &= \mathbf{p}v^3 \ , \\ d\mathbf{p}/d\xi &= -\nabla v \ , \\ dt/d\xi &= \nabla t\cdot d\mathbf{x}/d\xi = \mathbf{p}\cdot\mathbf{p}v^3 = v \ . \end{cases} \tag{5.36}$$

Equation 5.33 indicates $\nabla v = \mathbf{g}$, which means $d\mathbf{p}/d\xi$ can be integrated analytically and provides

$$\mathbf{p} = \mathbf{p_0} - \mathbf{g}\xi \ . \tag{5.37}$$

From the eikonal equation and considering $\mathbf{p_0}\cdot\mathbf{p_0} = \tilde{v}_0^{-2}$ and $g = |\mathbf{g}| = \sqrt{g_z^2 + g_x^2}$, I have

$$v = \frac{1}{\sqrt{\mathbf{p}\cdot\mathbf{p}}} = \left[\tilde{v}_0^{-2} - 2\mathbf{p_0}\cdot\mathbf{g}\xi + g^2\xi^2\right]^{-\frac{1}{2}} \ . \tag{5.38}$$

Integrating equation 5.38 over ξ gives

$$t = \frac{1}{g}\text{arccosh}\left(1 + \frac{g^2\xi^2}{\tilde{v}_0^{-2} + v^{-1}\tilde{v}_0^{-1} - \mathbf{p_0}\cdot\mathbf{g}\xi}\right) \ . \tag{5.39}$$

Meanwhile, combining equations 5.33 and 5.36, I find $d\mathbf{p}/d\xi \cdot (\mathbf{x}-\mathbf{x_0}) + d\mathbf{x}/d\xi \cdot \mathbf{p} = \tilde{v}_0$, i.e., $\mathbf{p} \cdot (\mathbf{x} - \mathbf{x_0}) = \tilde{v}_0 \xi$. Suppose

$$\mathbf{x} - \mathbf{x_0} = \alpha \mathbf{p_0} + \beta \mathbf{g} \tag{5.40}$$

then

$$\begin{cases} \mathbf{g} \cdot (\mathbf{x} - \mathbf{x_0}) &= \alpha \mathbf{p_0} \cdot \mathbf{g} + \beta g^2 = v - \tilde{v}_0 \ , \\ \mathbf{p_0} \cdot (\mathbf{x} - \mathbf{x_0}) &= \alpha \tilde{v}_0^{-2} + \beta \mathbf{p_0} \cdot \mathbf{g} = \tilde{v}_0 \xi + (v - \tilde{v}_0)\xi = v\xi \ . \end{cases} \tag{5.41}$$

Solving equation 5.41 provides $\alpha(\xi, v)$ and $\beta(\xi, v)$, which after substituting into equation 5.40 leads to

$$\mathbf{x} = \mathbf{x_0} + \frac{(v - \tilde{v}_0)\left[\mathbf{g} - (\mathbf{p_0} \cdot \mathbf{g})\tilde{v}_0^2 \mathbf{p_0}\right] + v\tilde{v}_0^2 \left[g^2 \mathbf{p_0} - (\mathbf{p_0} \cdot \mathbf{g})\mathbf{g}\right] \xi}{g^2 - (\mathbf{p_0} \cdot \mathbf{g})^2 \tilde{v}_0^2} \ . \tag{5.42}$$

Note equation 5.34 states $\mathbf{p_0} \cdot \mathbf{g} = g_z \tilde{v}_0^{-1}$ and thus equations 5.37, 5.39 and 5.42 can be further simplified.

To connect depth- and time-domain attributes, I first invert equation 5.39 such that ξ is expressed by t_0 and x_0

$$\xi(t_0, x_0) = \frac{g_z(1 - \cosh(|gt_0|)) + g \sinh(gt_0)}{g^2 \tilde{v}_0^2} \ . \tag{5.43}$$

Next, I insert equations 5.38 and 5.43 into 5.42 in order to change its parameterization from (ξ, v) to (t_0, x_0). The result is written for the z and x components of $\mathbf{x}$ separately, as follows:

$$x(t_0, x_0) = x_0 + \frac{\tilde{v}_0 g_x (1 - \cosh(gt_0))}{g(g \cosh(gt_0) - g_z \sinh(gt_0))} \ , \tag{5.44}$$

$$z(t_0, x_0) = \frac{\tilde{v}_0 \left[g_z(1 - \cosh(gt_0)) + g \sinh(gt_0)\right]}{g(g \cosh(gt_0) - g_z \sinh(gt_0))} \ . \tag{5.45}$$

Inverting equations 5.44 and 5.45 results in

$$x_0(z, x) = x + \frac{\sqrt{(v_0 + g_x x)^2 + g_x^2 z^2} - (v_0 + g_x x)}{g_x} \ , \tag{5.46}$$

$$t_0(z,x) = \frac{1}{g}\text{arccosh}\left\{\frac{g^2\left[\sqrt{(v_0+g_x x)^2+g_x^2 z^2}+g_z z\right]-v g_z^2}{v g_x^2}\right\} . \tag{5.47}$$

In the last step, I derive the analytical formula for the Dix velocity. Note that from equation 5.46 $|\nabla x_0|^2 = 1$, it is easy to verify that $|\nabla x_0| = 1$, $|\nabla t_0| = 1/v$ and $\nabla x_0 \cdot \nabla t_0 = 0$. Because there is no geometrical spreading in this case, the Dix velocity will be equal to the interval velocity according to equation 5.3. However, a Dix-inverted model will still be distorted if $g_x \neq 0$ because of the lateral shift of image rays.

According to equation 5.3 $v_d = v$ and is found by combining equations 5.38 and 5.43

$$v_d(t_0,x_0) = \frac{(v_0+g_x x_0)g}{g\cosh(gt_0)-g_z\sinh(gt_0)} . \tag{5.48}$$

The migration velocity v_m, on the other hand, reads

$$v_m(t_0,x_0) = \frac{(v_0+g_x x_0)^2}{t_0(g\coth(gt_0)-g_z)} . \tag{5.49}$$

Figures 5.2 and 5.3 show a velocity model with $v(z,x) = 1.5 + 0.75z + 0.5x$ km/s and the corresponding analytical $v_d(t_0,x_0)$, $x_0(z,x)$ and $t_0(z,x)$. Clearly, the right domain boundary is of in-flow type that violates my assumption. To address this challenge, I include Dix velocity in regions beyond the original left and right boundaries during inversion, but mask out the cost in these regions. It means the

time-to-depth conversion is performed in a sub-domain of time-domain attributes, such that information on the in-flow boundary is available. Afterwards, I apply Dix inversion to the expanded model and use the result as the prior model. I use the exact Dix velocity in equation 5.48 for evaluating 5.5. Then, in total three linearization updates are carried out. At last, I cut the computational domain back to its original size. Figures 5.4 and 5.5 compare the cost and model misfit before and after inversion.

I also synthesize data using Kirchhoff modeling (Haddon and Buchen, 1981) for several horizontal reflectors using the exact model, and examine the subsurface scattering-angle common-image-gathers from Kirchhoff prestack depth migrations (Xu et al., 2001; Bashkardin et al., 2012) as an evidence of interval velocity improvements. In Figure 5.6, the shallower events do not improve significantly because the image rays have not yet bend considerably. As for those deeper events, the gather becomes noticeably flatter after applying the proposed method.

Constant horizontal slowness-squared gradient model

Another medium that provides analytical time-to-depth conversion formulas is

$$w(z,x) = w_0 - 2q_x x = w_0 - 2\mathbf{q}\cdot\mathbf{x}\ , \tag{5.50}$$

where $\mathbf{q} = [0, q_x]^T$, i.e., the slowness-squared changes linearly in the horizontal direction.

Similarly to the constant velocity gradient medium, the eikonal equation can

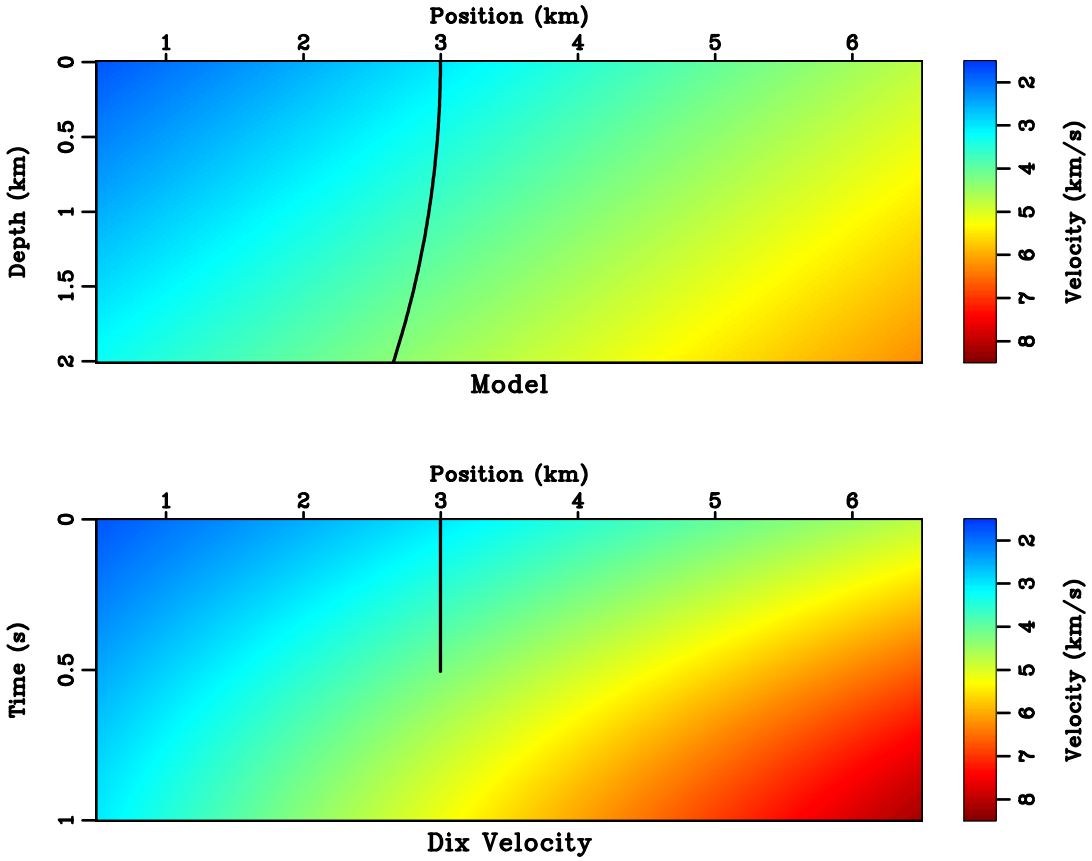


Figure 5.2: (Top) a constant velocity gradient model and (bottom) the analytical Dix velocity v_d. A curved image ray is mapped to the time domain as a straight line. time2dep/vgrad vgrad

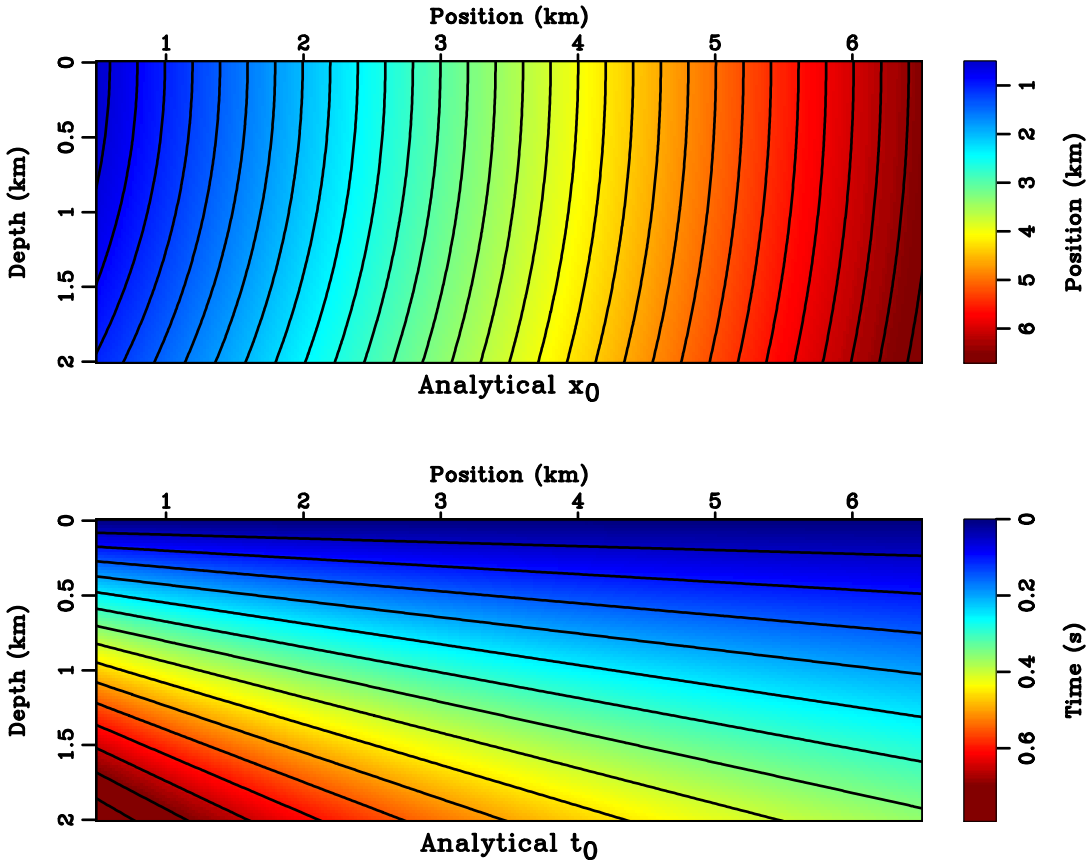


Figure 5.3: Analytical values of (top) t_0 and (bottom) x_0 of the model in Figure 5.2. Both figures are overlaid with contour lines that, according to equation 5.7, are perpendicular to each other. Each contour line of x_0 is an image ray, while the contours of t_0 illustrate the propagation of a plane-wave. time2dep/vgrad analy

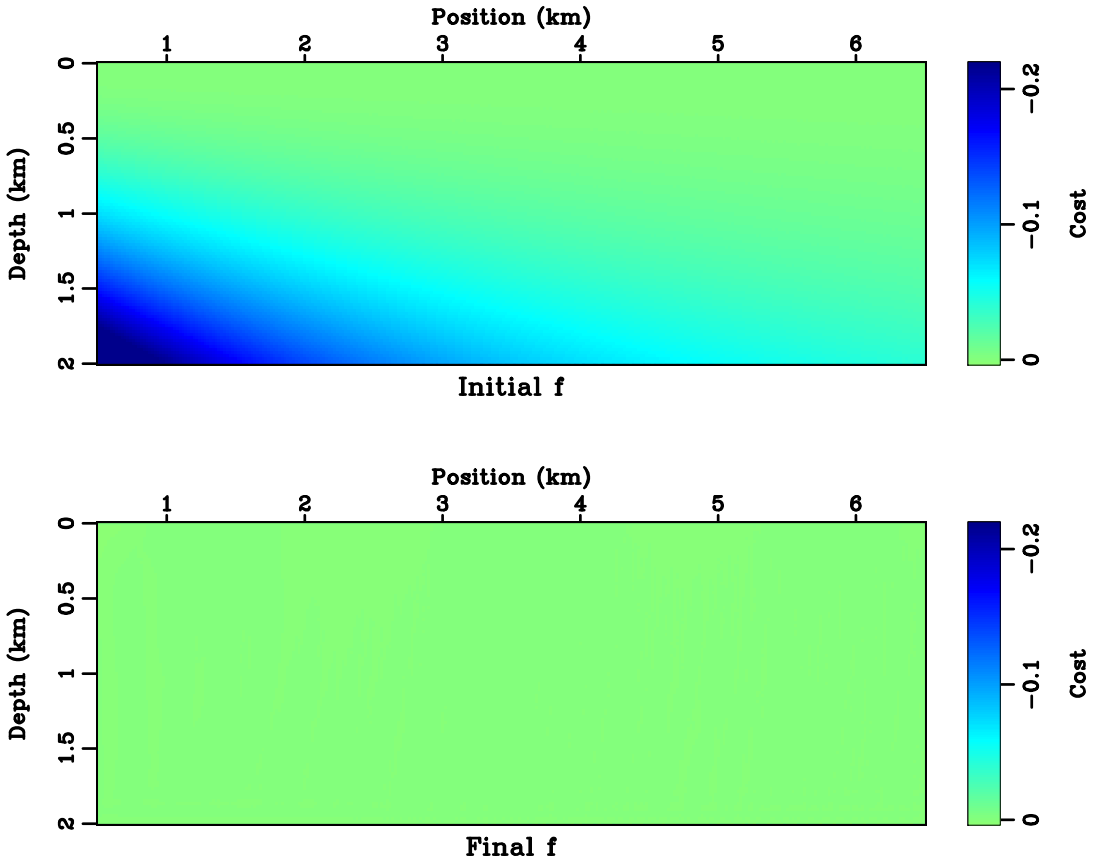


Figure 5.4: The cost defined by equation 5.20 (top) before and (bottom) after inversion. The least-squares norm of cost E is decreased from 9.855 to 0.057. time2dep/vgrad cost

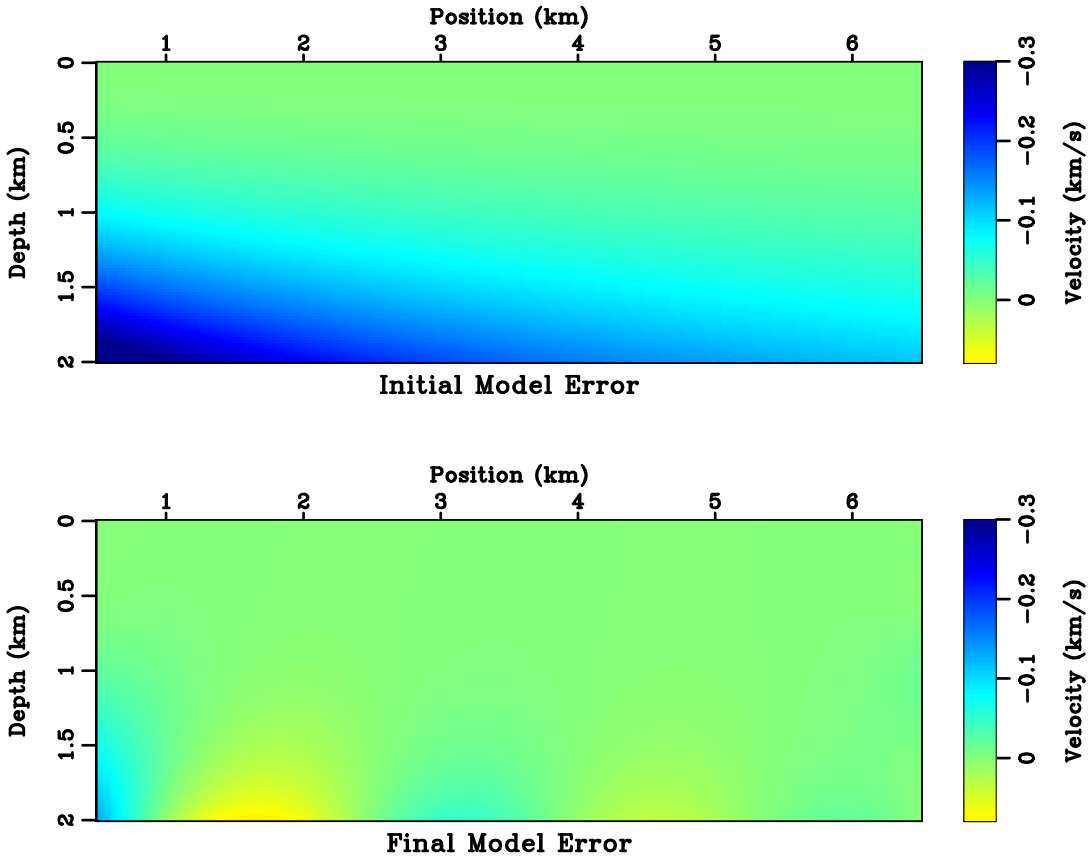


Figure 5.5: The difference between exact model and (top) initial model and (bottom) inverted model. The least-squares norm of model misfit is decreased from 15.6 km^2/s^2 to 2.7 km^2/s^2. time2dep/vgrad error

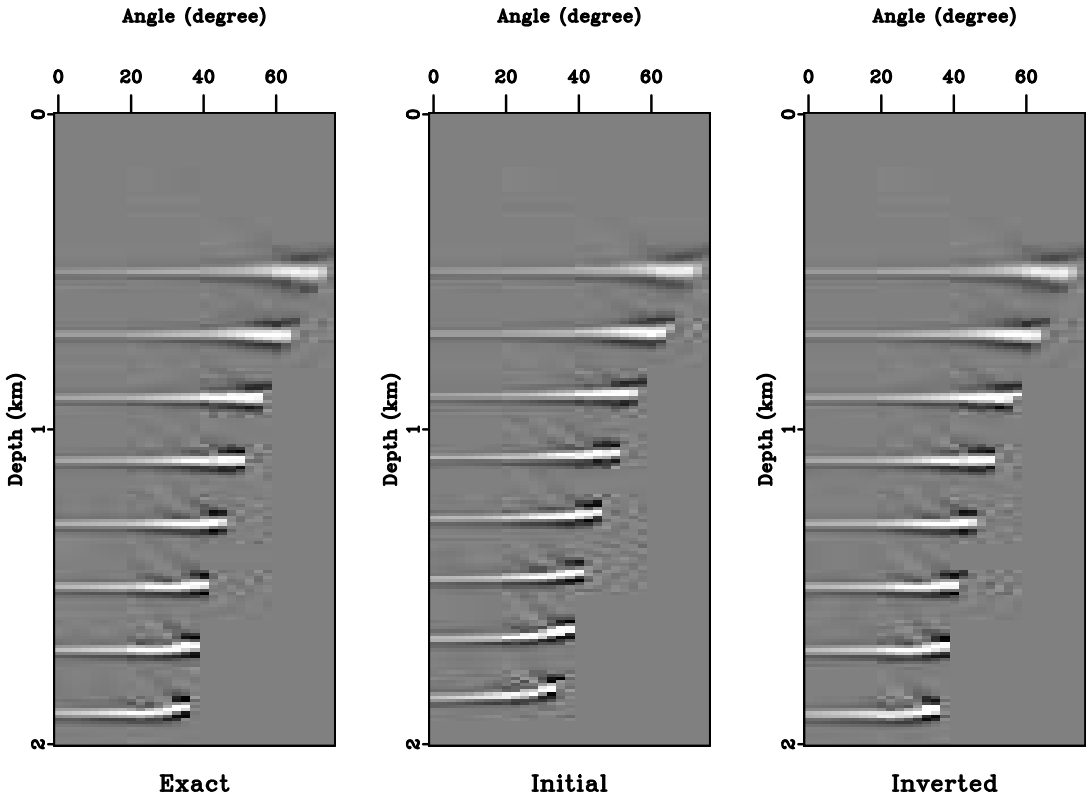


Figure 5.6: Comparison of the common-image-gathers at $x = 1.5$ km of (left) exact model, (middle) prior model and (right) inverted model. time2dep/vgrad cigv

be rewritten as Hamiltonian $H(\mathbf{x},\mathbf{p}) = \mathbf{p}\cdot\mathbf{p} - w \equiv 0$. Differentiating H with respect to an incremental parameter σ leads to

$$\frac{dH}{d\sigma} = 2\mathbf{p}\cdot\frac{d\mathbf{p}}{d\sigma} - \frac{dw}{d\sigma} = 2\mathbf{p}\cdot\frac{d\mathbf{p}}{d\sigma} - \nabla w\cdot\frac{d\mathbf{x}}{d\sigma} \equiv 0\ . \tag{5.51}$$

From equation 5.51, I write down the ray-tracing system

$$\left\{\begin{array}{lcl} d\mathbf{x}/d\sigma &=& \mathbf{p}\ , \\ d\mathbf{p}/d\sigma &=& \nabla w/2\ , \\ dt/d\sigma &=& \nabla t\cdot d\mathbf{x}/d\sigma = \mathbf{p}\cdot\mathbf{p}\ . \end{array}\right. \tag{5.52}$$

Given 5.50, $\nabla w = -2\mathbf{q}$ and thus $d\mathbf{p}/d\sigma = -\mathbf{q}$. After integration over σ, equation 5.52 becomes

$$\left\{\begin{array}{lcl} \mathbf{x} &=& \mathbf{x_0} + \mathbf{p_0}\sigma - \mathbf{q}\sigma^2/2\ , \\ \mathbf{p} &=& \mathbf{p_0} - \mathbf{q}\sigma\ , \\ t &=& |\mathbf{p_0}|^2\sigma - \mathbf{p_0}\cdot\mathbf{q}\sigma^2 + |\mathbf{q}|^2\sigma^3/3\ . \end{array}\right. \tag{5.53}$$

For a particular image ray

$$\left\{\begin{array}{lcl} \mathbf{x_0} &=& [0, x_0]^T\ , \\ \mathbf{p_0} &=& [\sqrt{w_0 - 2q_x x_0}, 0]^T\ , \\ t_0 &=& t\ , \end{array}\right. \tag{5.54}$$

the equation for $\mathbf{x}$ in 5.53 simplifies to

$$\left\{\begin{array}{l} x = x_0 - q_x\sigma^2/2\ , \\ z = \sigma\sqrt{w_0 - 2q_x x_0}\ . \end{array}\right. \tag{5.55}$$

Solving equation 5.55 for σ as a function of z and x

$$\sigma(z,x) = \left[\frac{(w_0 - 2q_x x) - \sqrt{(w_0 - 2q_x x)^2 - 4q_x^2 z^2}}{2q_x^2}\right]^{\frac{1}{2}}\ . \tag{5.56}$$

Combining equations 5.53 through 5.56, I find

$$\begin{array}{rcl} x_0(z,x) &=& x + \dfrac{1}{2}q_x\sigma^2 \\ &=& \dfrac{2w_0 x + q_x z^2}{w_0 + 2q_x x + \sqrt{(w_0 - 2q_x x)^2 - 4q_x^2 z^2}}\ , \end{array} \tag{5.57}$$

$$
\begin{aligned}
t_0(z,x) &= (w_0 - 2q_x x_0)\sigma + \frac{1}{3}q_x^2\sigma^3 \\
&= \frac{\sqrt{2}z\left[2w_0 - 4q_x x + \sqrt{(w_0-2q_x x)^2 - 4q_x^2 z^2}\right]}{3\left[w_0 - 2q_x x + \sqrt{(w_0-2q_x x)^2 - 4q_x^2 z^2}\right]^{\frac{1}{2}}} \; .
\end{aligned} \tag{5.58}
$$

According to equation 5.4, 5.57 can give rise to the geometrical spreading:

$$
Q^2(z,x) = \frac{2[(w_0-2q_x x)^2 - 4q_x^2 z^2]}{(w_0 - 2q_x x)\left[w_0 - 2q_x x + \sqrt{(w_0-2q_x x)^2 - 4q_x^2 z^2}\right]} \; . \tag{5.59}
$$

It is more convenient to express equations 5.50 and 5.59 in σ and x_0 instead of directly in t_0 and x_0:

$$
w(\sigma, x_0) = w_0 - 2q_x x_0 + q_x^2\sigma^2 \; , \tag{5.60}
$$

$$
Q^2(\sigma, x_0) = 1 + q_x^2\sigma^2\left(\frac{1}{w_0 - 2q_x x_0} - \frac{4}{w_0 - 2q_x x_0 + q_x^2\sigma^2}\right) \; , \tag{5.61}
$$

where I must resolve $\sigma = \sigma(t_0, x_0)$. This is done by revisiting equation 5.58. For given t_0 and x_0, σ is the root of a depressed cubic function of the following form:

$$
\sigma^3 + \frac{3(w_0 - 2q_x x_0)}{q_x^2}\sigma - \frac{3}{q_x^2}t_0 = 0 \; . \tag{5.62}
$$

After some algebraic manipulations, I find

$$
\begin{aligned}
\sigma(t_0, x_0) &= \left[\frac{3t_0 + \sqrt{9t_0^2 + 4(w_0 - 2q_x x_0)^3/q_x^2}}{2q_x^2}\right]^{\frac{1}{3}} \\
&- \frac{w_0 - 2q_x x_0}{q_x}\left[\frac{2}{3q_x t_0 + \sqrt{9q_x^2 t_0^2 + 4(w_0 - 2q_x x_0)^3}}\right]^{\frac{1}{3}} \; .
\end{aligned} \tag{5.63}
$$

Finally, inserting equations 5.60 and 5.61 into equation 5.3 results in the Dix velocity:

$$v_d(t_0, x_0) = \frac{\sqrt{w_0 - 2q_x x_0}}{w_0 - 2q_x x_0 - q_x^2 \sigma^2} \ . \tag{5.64}$$

In Figure 5.7 I illustrate $x_0(z, x)$ and $t_0(z, x)$ in the model $w(z, x) = 1 - 0.052x$ $\mathrm{s}^2/\mathrm{km}^2$. To deal with the in-flow boundary issue, I apply the same method as in the constant velocity gradient example. Unlike equation 5.46, 5.57 indicates varying geometrical spreadings in the domain. Figure 5.8 shows the corresponding analytical $Q^2(z, x)$ and $v_d(t_0, x_0)$. Note the geometrical spreading is significant at the lower-right corner of the domain, which translates to the cost at approximately the same location in Figure 5.9. I again use analytical Dix velocity in the inversion. Starting from the Dix-inverted model and after three linearization updates, I decrease E to relative 0.0045%. The model misfit, as demonstrated in Figure 5.10, is also improved.

Spiral model

Figure 5.11 shows a synthetic model borrowed from Cameron et al. (2008). The Dix inversion recovers the shallow part of the model but deteriorates quickly as geometrical spreading of image rays grows in the deeper section.

As a simple verification for the linearization process, I add a small positive velocity perturbation at location $(1, 3)$ km to the synthetic model. Comparisons between the exact and linearly predicted attributes are illustrated in Figures 5.12, 5.13

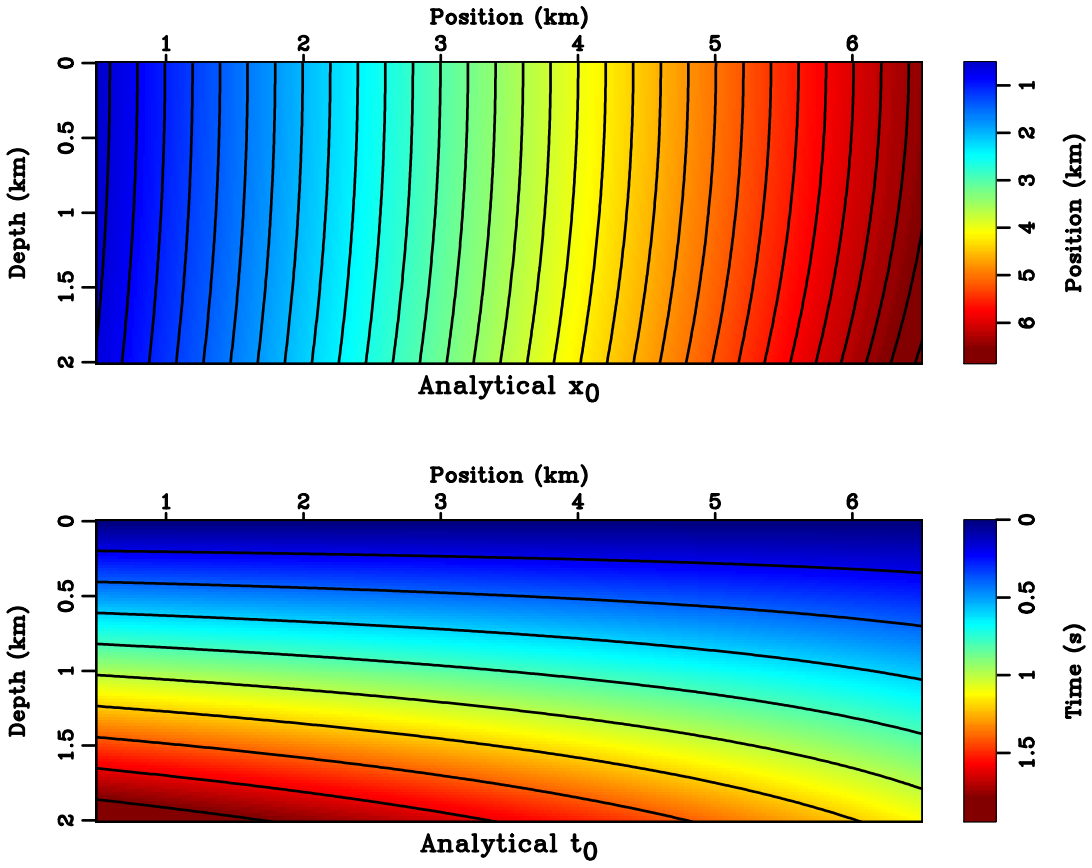


Figure 5.7: Analytical values of (top) t_0 and (bottom) x_0, overlaid with contour lines.
time2dep/hs2grad hs2analy

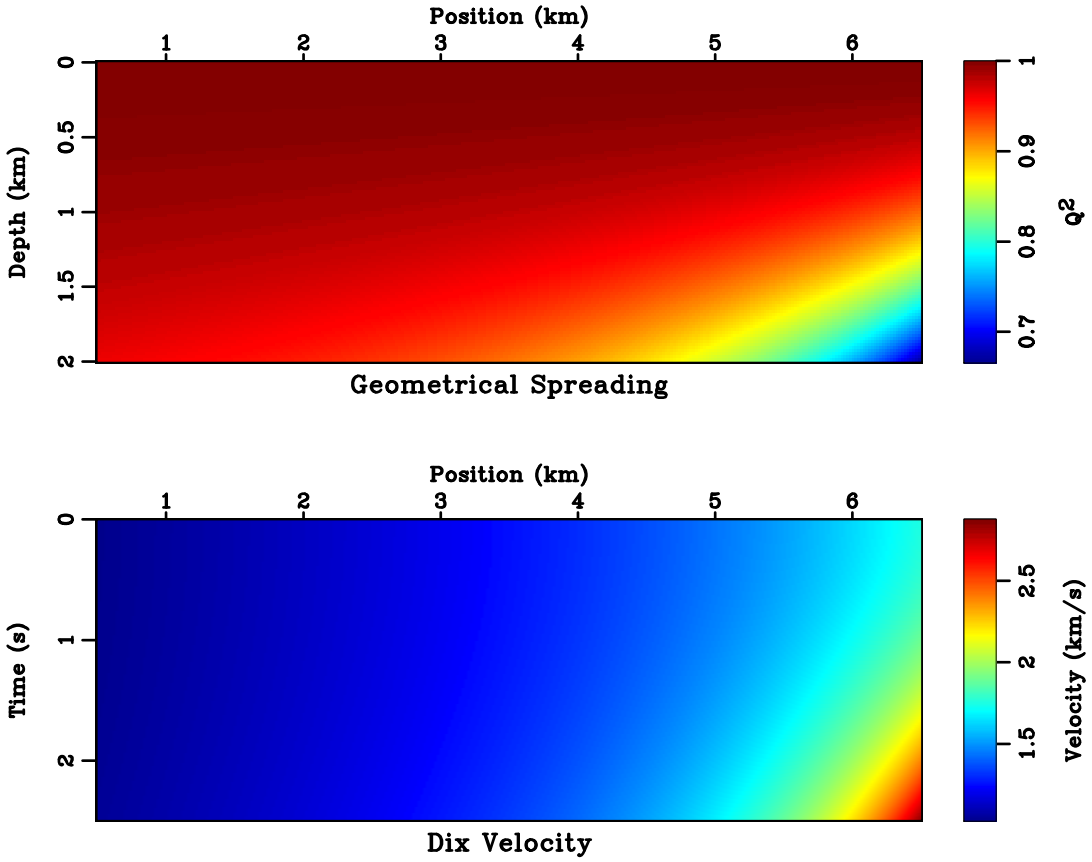


Figure 5.8: The (top) geometrical spreading and (bottom) Dix velocity associated with the model used in Figure 5.7. time2dep/hs2grad hs2grad

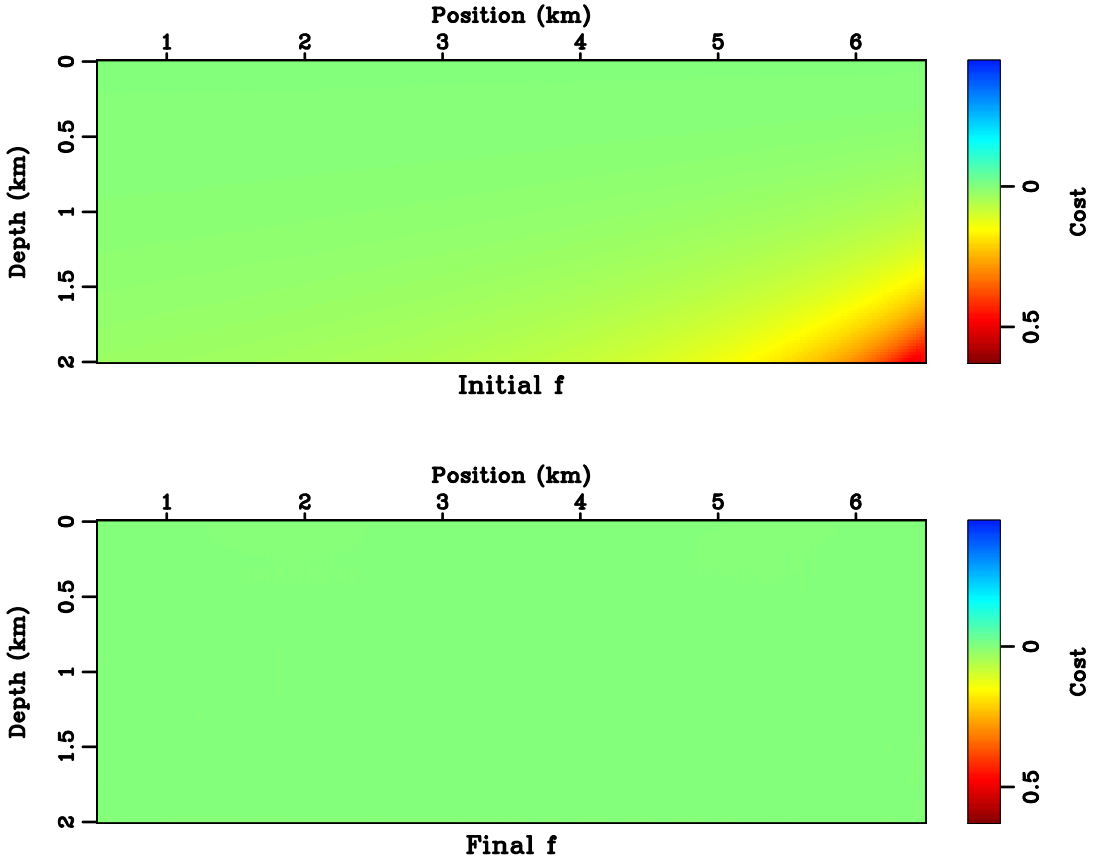


Figure 5.9: The cost (top) before and (bottom) after inversion. The least-squares norm of cost E is decreased from 10.431 to 0.047. time2dep/hs2grad hs2cost

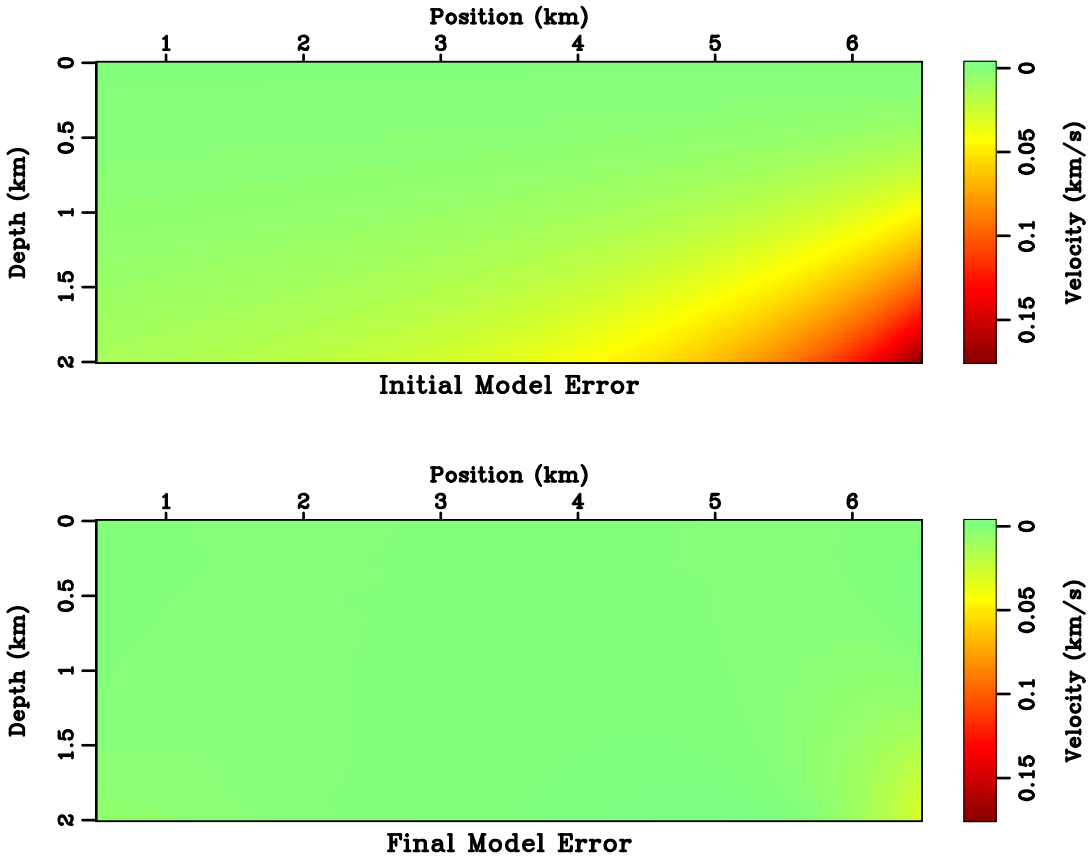


Figure 5.10: The difference between exact model and (top) initial model and (bottom) inverted model. The least-squares norm of model misfit is decreased from 5.0 km^2/s^2 to 0.5 km^2/s^2. time2dep/hs2grad hs2error

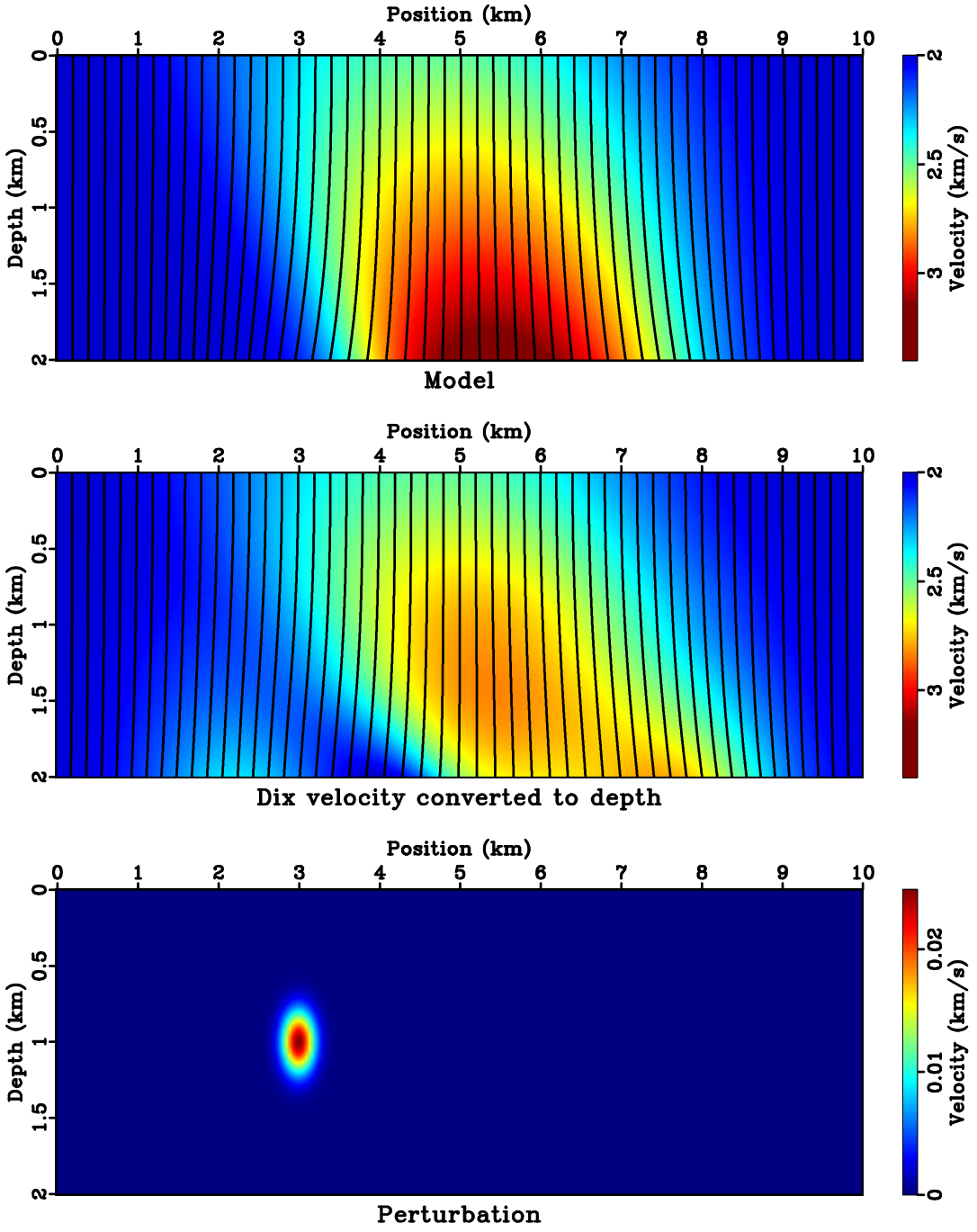


Figure 5.11: (Top) a synthetic model and (middle) Dix velocity converted to depth. Both overlaid with image rays. (Bottom) the model perturbation for testing linearization. time2dep/synth vz0

and 5.14. In accordance with forward modeling, where I solve firstly t_0, then x_0, and finally f, the linearization in Appendix B is carried out following the same sequence. First, Figure 5.12 justifies my upwind finite-differences implementation of the linearized eikonal equation. The positive perturbation in v in Figure 5.11 causes t_0 to decrease in a narrow downwind region. Next, the area affected by the perturbation in Figure 5.13 is wider than that in Figure 5.12. It also has both positive and negative amplitudes. These phenomenon are physical because image rays should bend in opposite directions in response to the perturbation. Finally, effects in cost f in Figure 5.14 shows alternating polarities and are broader in width compared to that of dt_0 and dx_0. They indicate a complicated dependency of f on w. Note the good agreements in both shapes and magnitudes between exact and linearly predicted quantities in all cases.

Because there is no analytical formula for Dix velocity in this model, I compute v_d by tracing image rays numerically in the exact model v. Also, based on Figure 5.11, there is no in-flow boundary other than $z = 0$. Therefore, I do not extend the domain as in the preceding examples. I use again the Dix-inverted model as the prior model and run the inversion. It turns out that the first linearization update as shown in Figure 5.15 is sufficient for achieving the desired global minimum.

Field data example

The field data shown in Figure 5.16 is from a section of Gulf of Mexico dataset (Claerbout, 1996). I estimate v_m using the method of velocity continuation (Fomel, 2003) and convert it to v_d. Similar to the spiral model, no domain extension is needed.

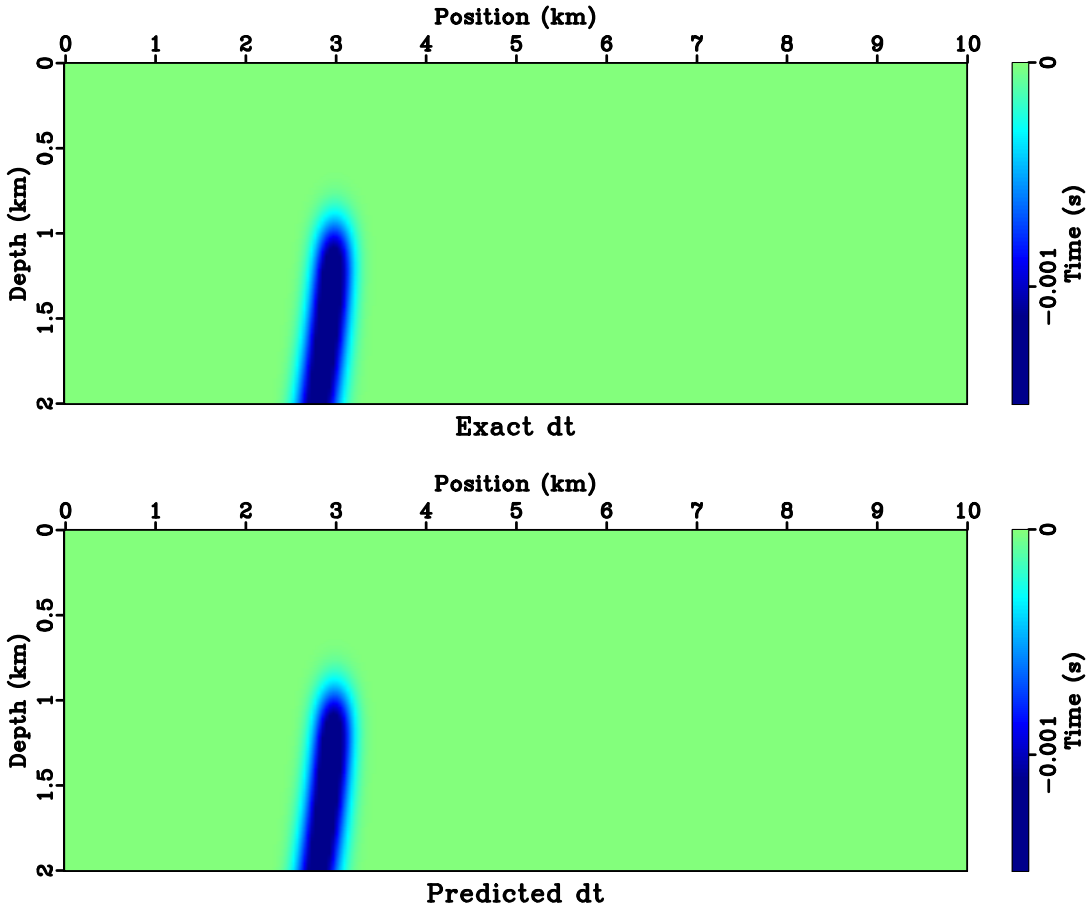


Figure 5.12: (Top) exact dt_0 and (bottom) linearly predicted dt_0 by equation 5.27. time2dep/synth pdt

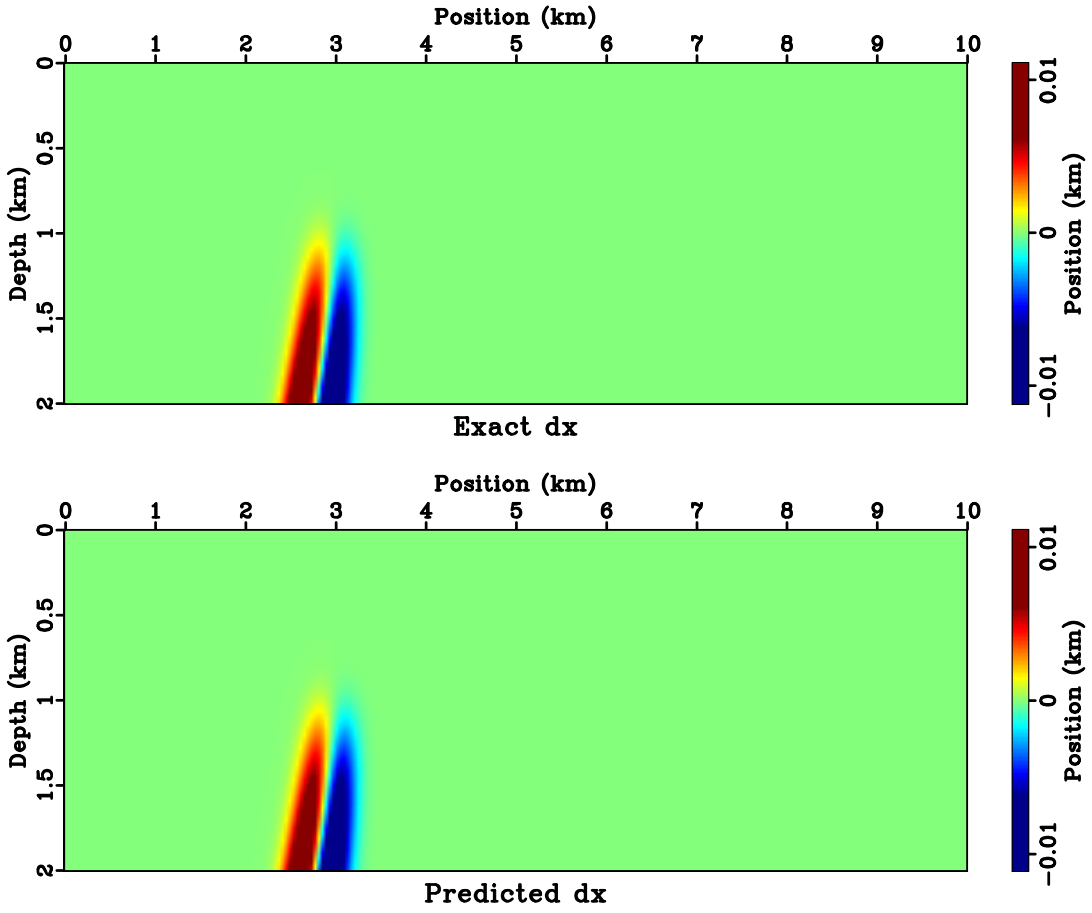


Figure 5.13: (Top) exact dx_0 and (bottom) linearly predicted dx_0 by equation 5.26. time2dep/synth pdx

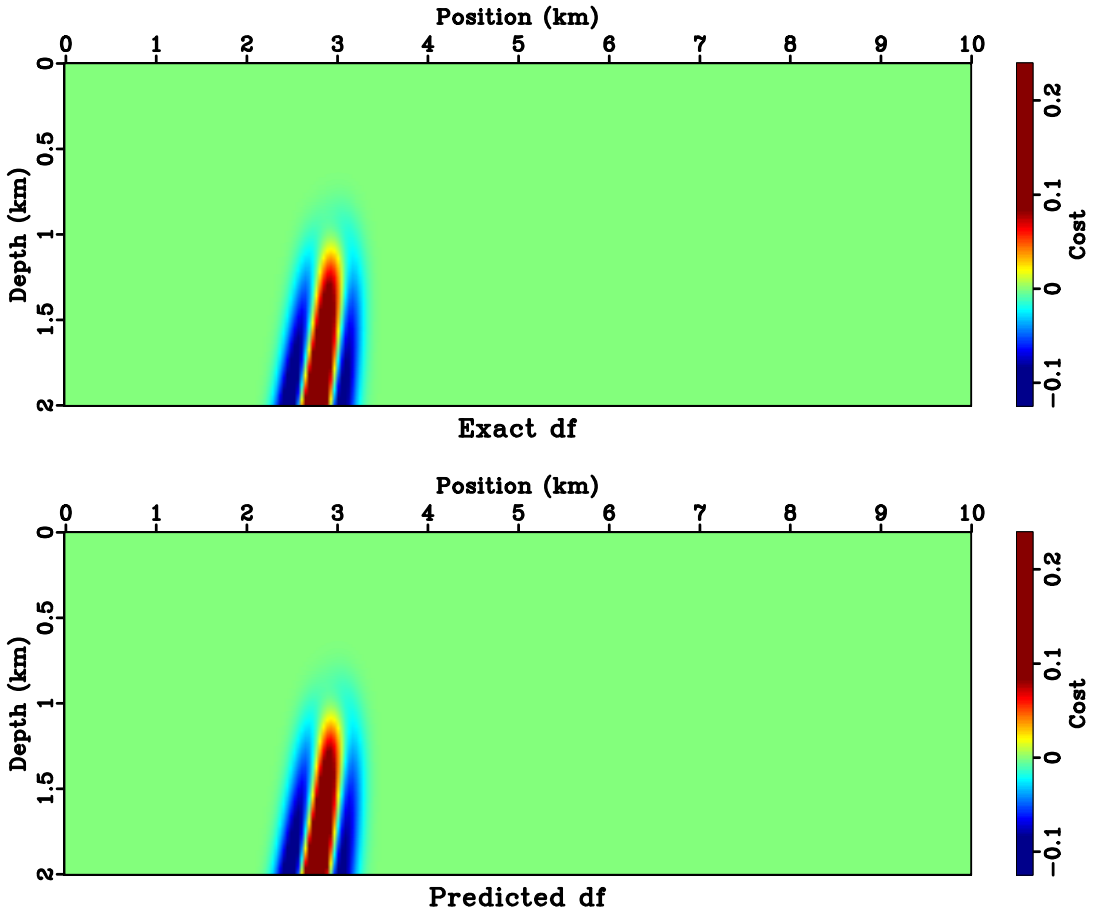


Figure 5.14: (Top) exact *df* and (bottom) linearly predicted *df* by equation 5.28. time2dep/synth diffcost

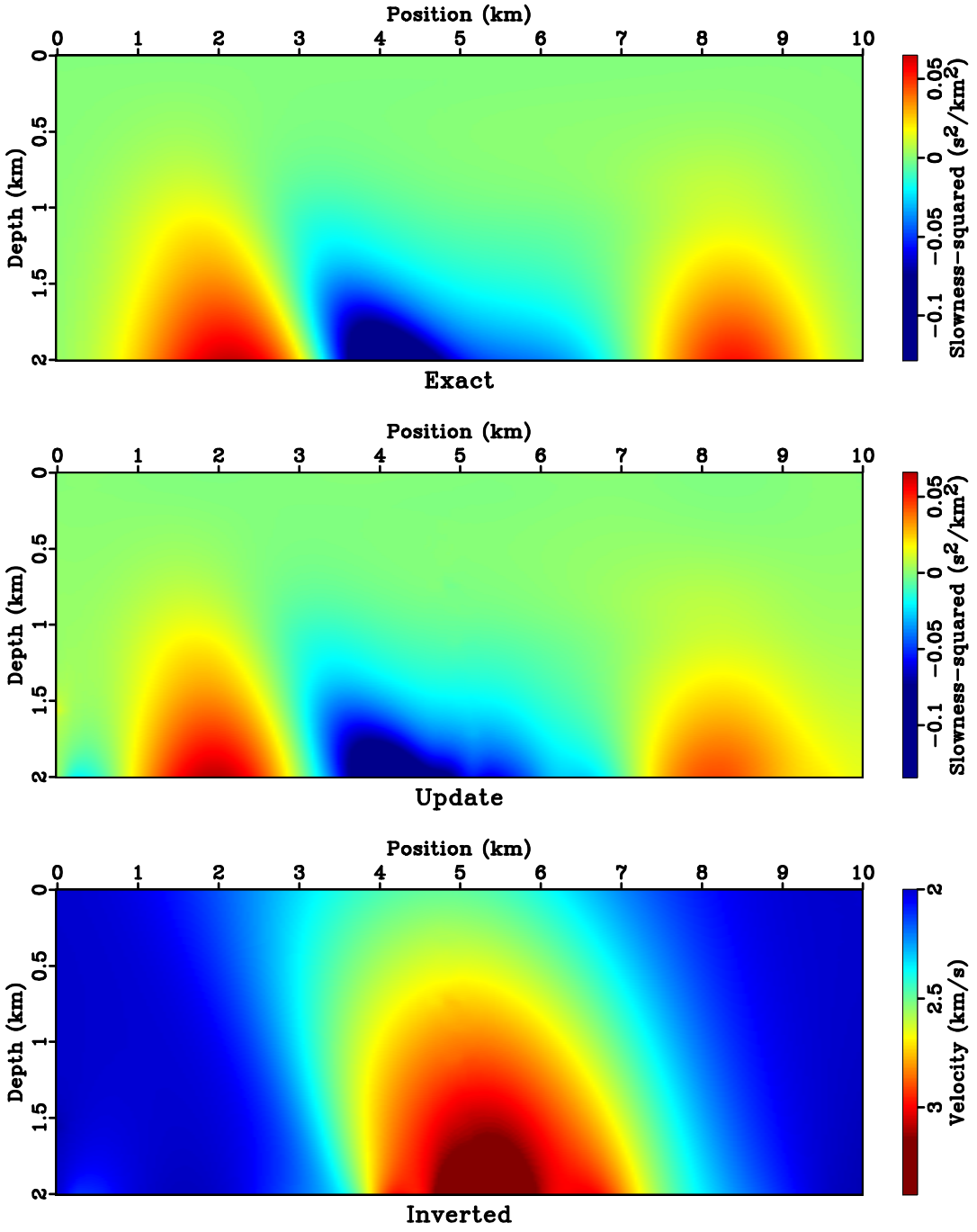


Figure 5.15: (Top) the exact δw and (middle) the computed δw of the first linearization step. (Bottom) the inverted interval velocity model. Compare with Figure 5.11.
time2dep/synth bgrad

In Figure 5.17, the Dix-inverted prior model highly resembles the Dix velocity, because the Dix formula only scales the vertical axis from time to depth regardless of horizontal v_d variations. Figure 5.18 compares the cost before and after five linearization updates. In Figure 5.19, the l_2 norm of the cost, E, has a rapid decrease to relative 1.3%. Figure 5.20 illustrates the inverted model and interval velocity update.

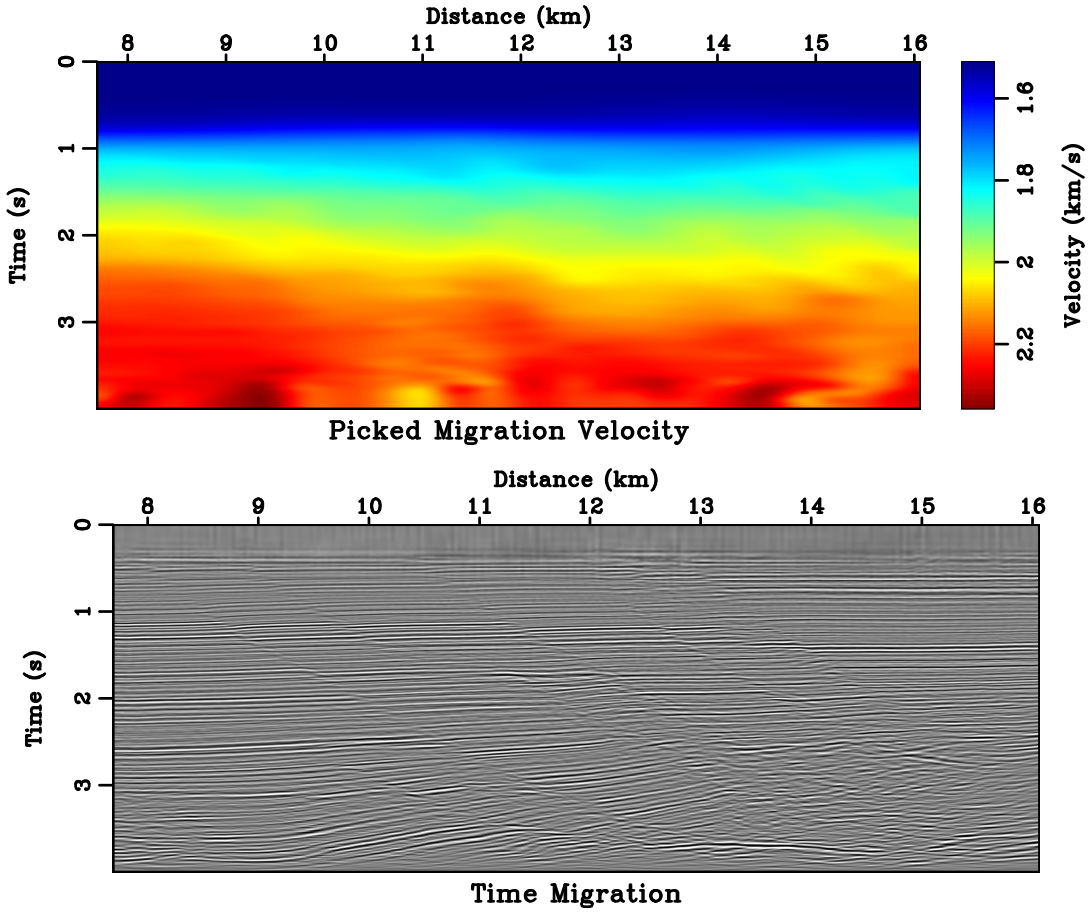


Figure 5.16: (Top) the estimated time-migration velocity of a section of Gulf of Mexico dataset and (bottom) the corresponding time-migrated image. time2dep/beinew vdix

Next, I map the time-migrated image to depth using t_0 and x_0 generated during

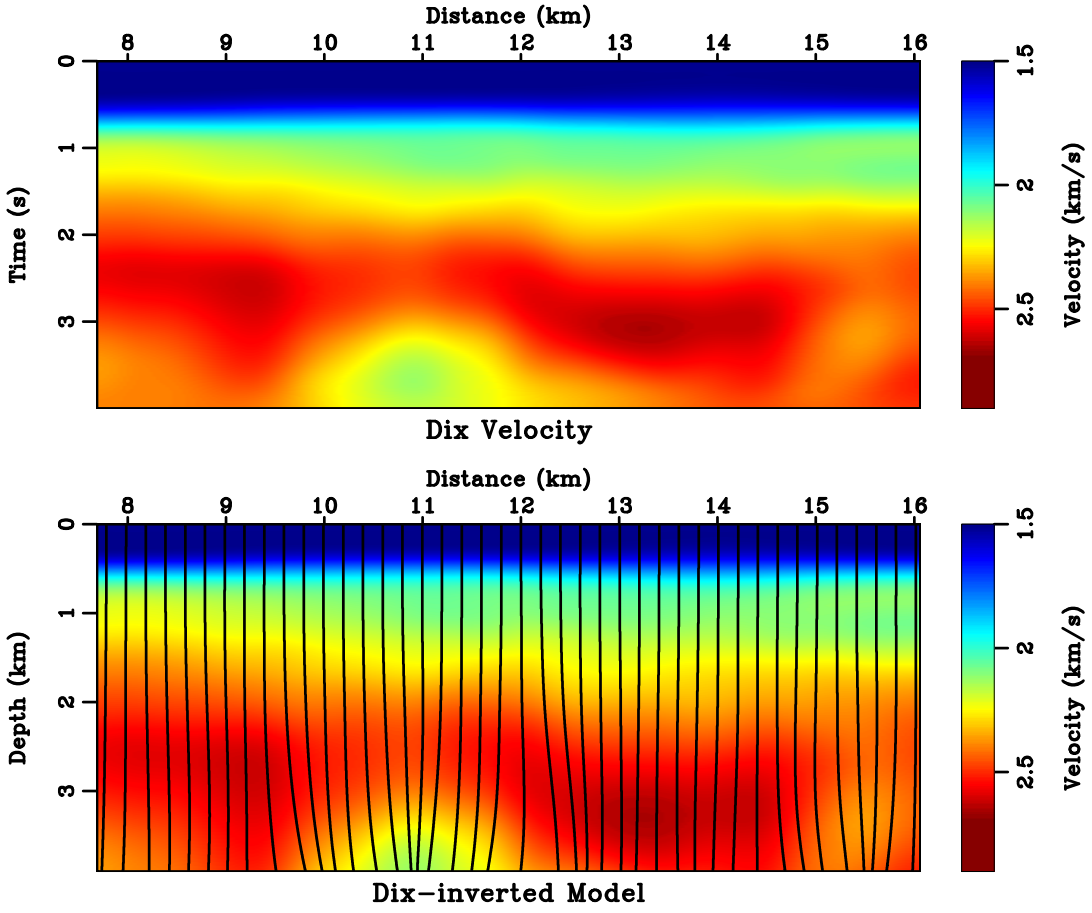


Figure 5.17: (Top) the Dix velocity converted from v_m in Figure 5.16 and (bottom) the Dix-inverted prior model for inversion, overlaid with image rays. time2dep/beinew init

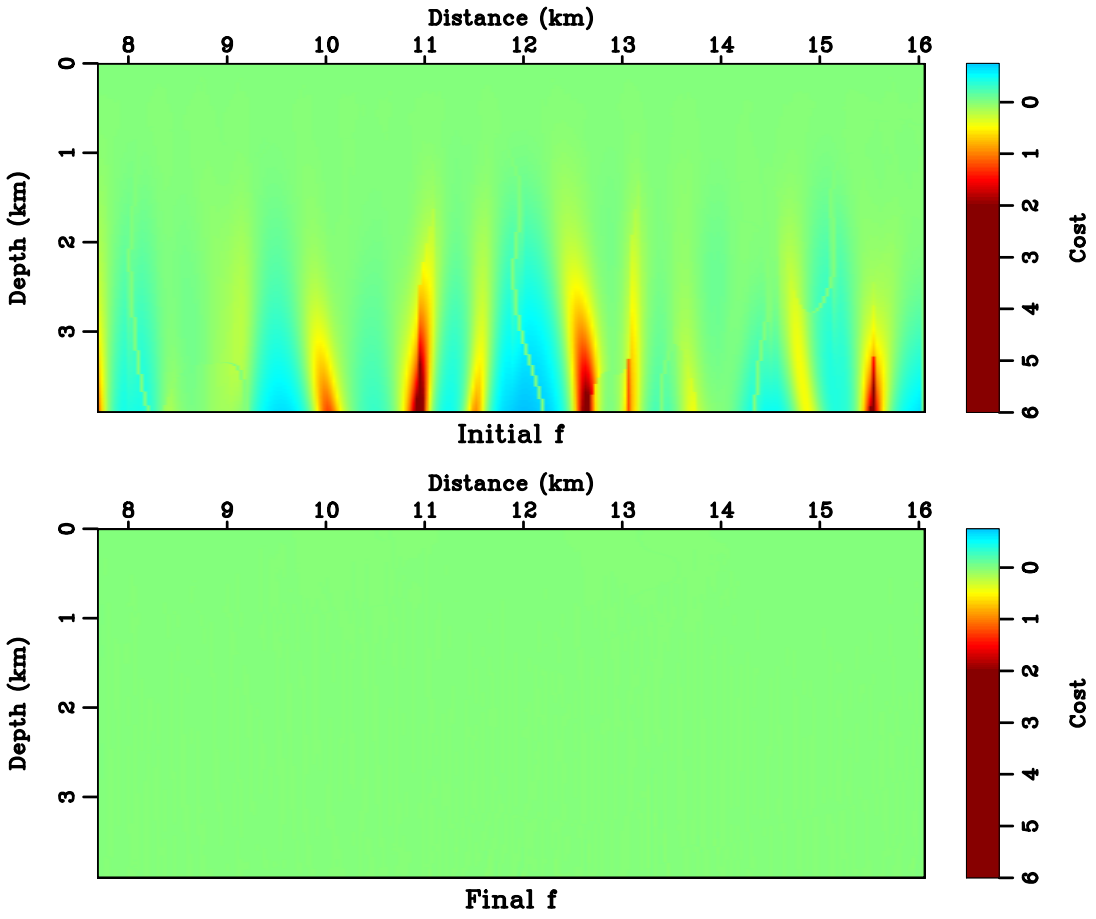


Figure 5.18: The costs of (top) prior model ($E = 140.25$) and (bottom) inverted model ($E = 1.81$). time2dep/beinew inv

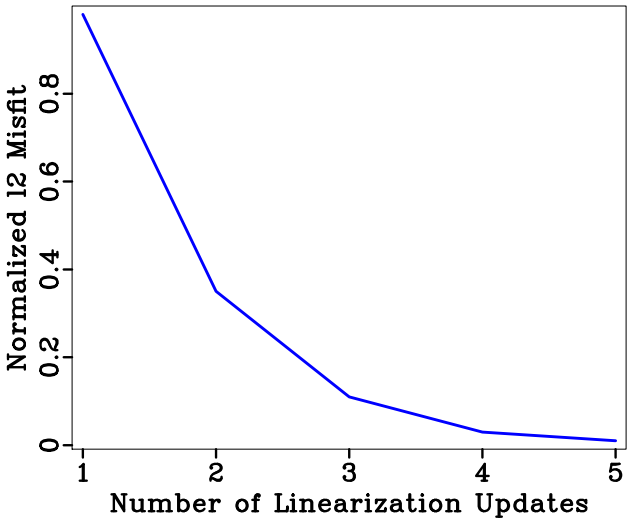


Figure 5.19: Convergence history of the proposed optimization-based time-to-depth conversion. time2dep/beinew hist

inversion. Spline interpolation (Press et al., 2007) is used during the coordinate mapping. I also migrate the prestack data by Kirchhoff depth migration (Li and Fomel, 2013) (PSDM) that is developed in Chapter 4. Figure 5.21 compares the time-mapped image and PSDM image of the inverted model. A good agreement between these two images justifies that time-to-depth conversion has effectively unraveled the tilted time coordinate. Figure 5.22 compares PSDM images of the prior and inverted models. The velocity update in Figure 5.20 results in not only changes in structural dips (for example at $(3, 12)$ km) but also improved reflector continuity (for example at $(3.7, 11)$ km). Moreover, the Kirchhoff migration outputs surface offset common-image gathers. I choose two midpoint locations, $x = 11$ km and $x = 12$ km, and show their common-image gathers in Figures 5.23 and 5.25. In deeper sections, I further zoom in the gathers in Figures 5.24 and 5.26. Flat dashed lines are overlaid as references for the flatness of gathers. The two common-image gathers of prior model appear curved in opposite directions. After time-to-depth conversion, both gathers

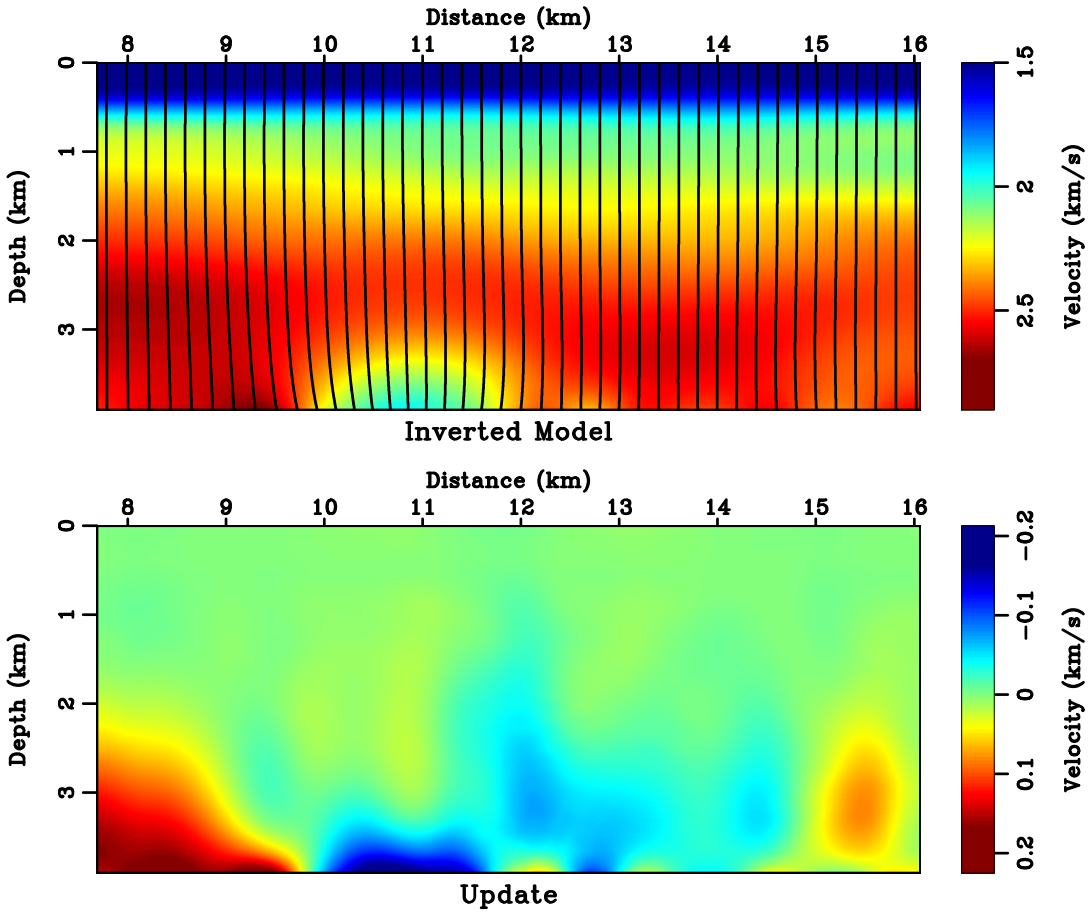


Figure 5.20: (Top) the inverted model, overlaid with image rays, and (bottom) its difference from the prior model in Figure 5.17. time2dep/beinew dinv

get flattened across the whole offset range, verifying a correct velocity update.

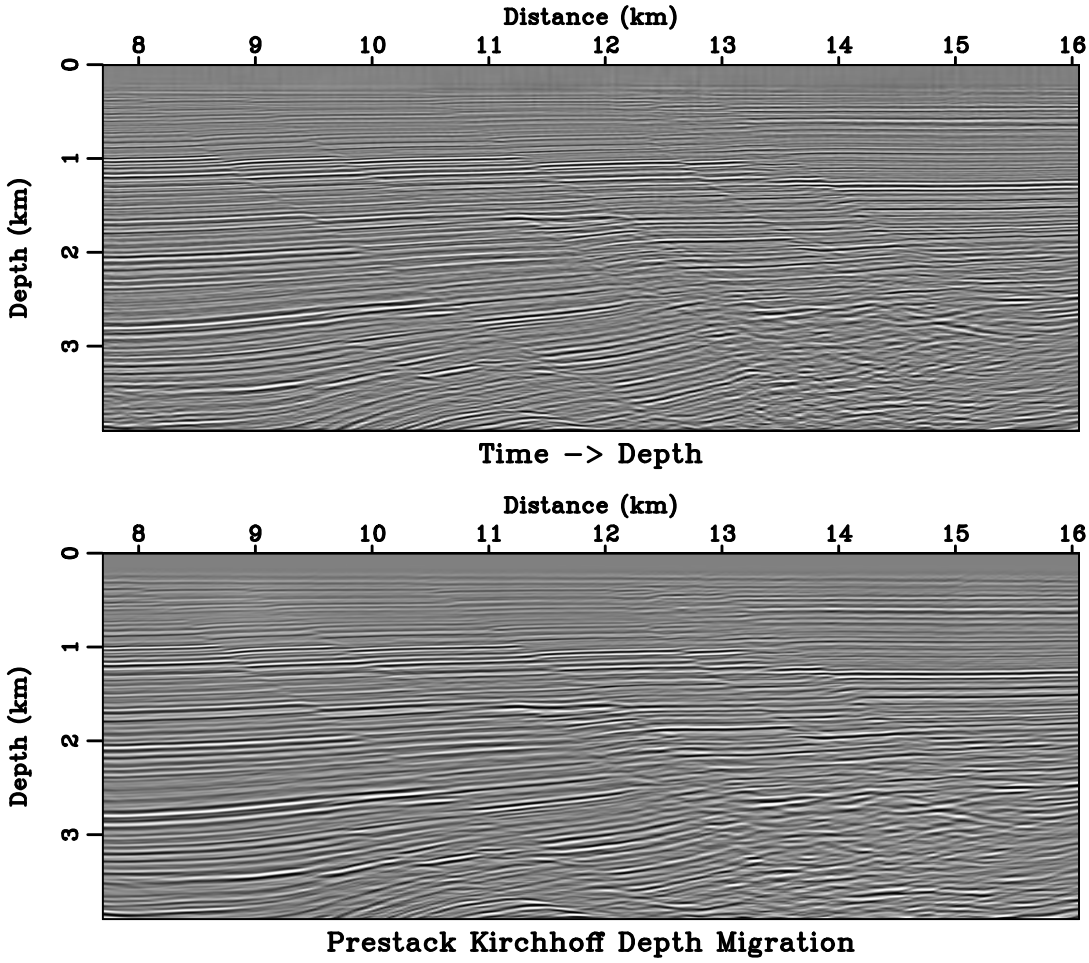
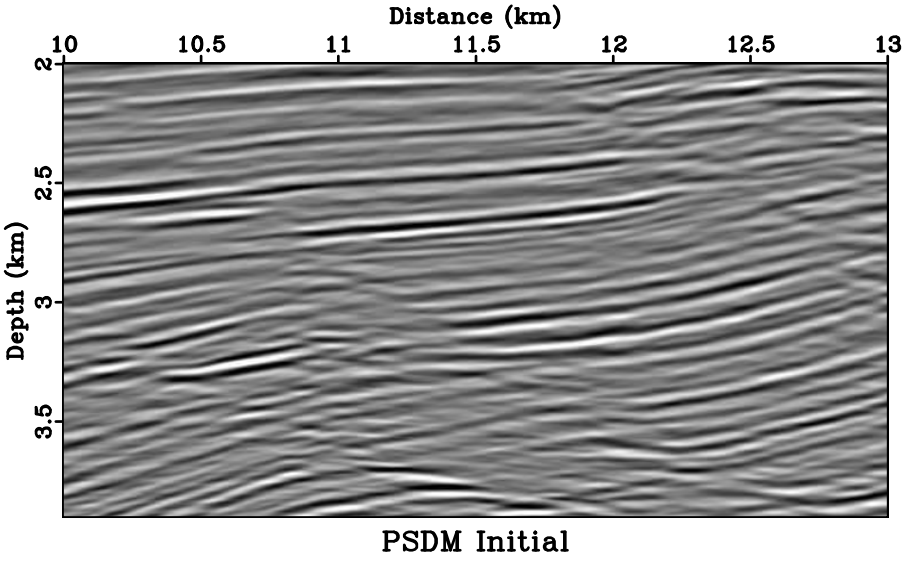


Figure 5.21: (Top) the time-migrated image in Figure 5.16 is mapped to depth using products of the time-to-depth conversion. (Bottom) PSDM image using inverted model in Figure 5.20. time2dep/beinew dmig

Synthetic dataset

The last step in processing the synthetic dataset of Chapter 1 is time migration and time-to-depth conversion using the redatumed data from Chapter 4. Similarly



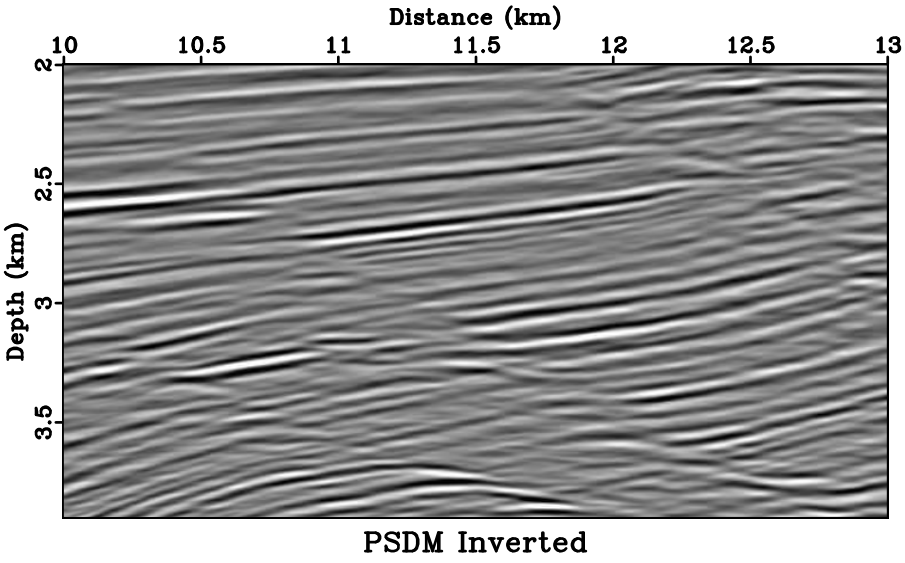


Figure 5.22: PSDM images of (left) the prior model and (right) the inverted model. Both images are plotted for the same central deep part. time2dep/beinew ddmig0

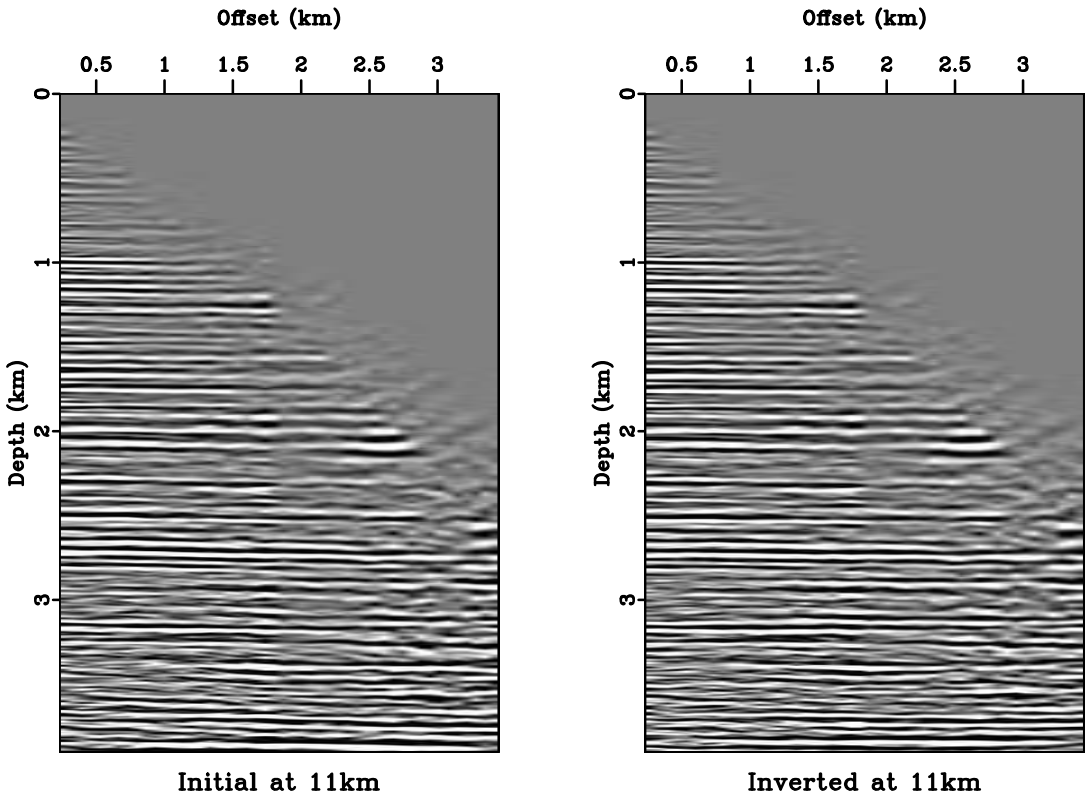


Figure 5.23: The surface offset common-image gathers at $x = 11$ km of (left) prior model and (right) inverted model. time2dep/beinew cig1a

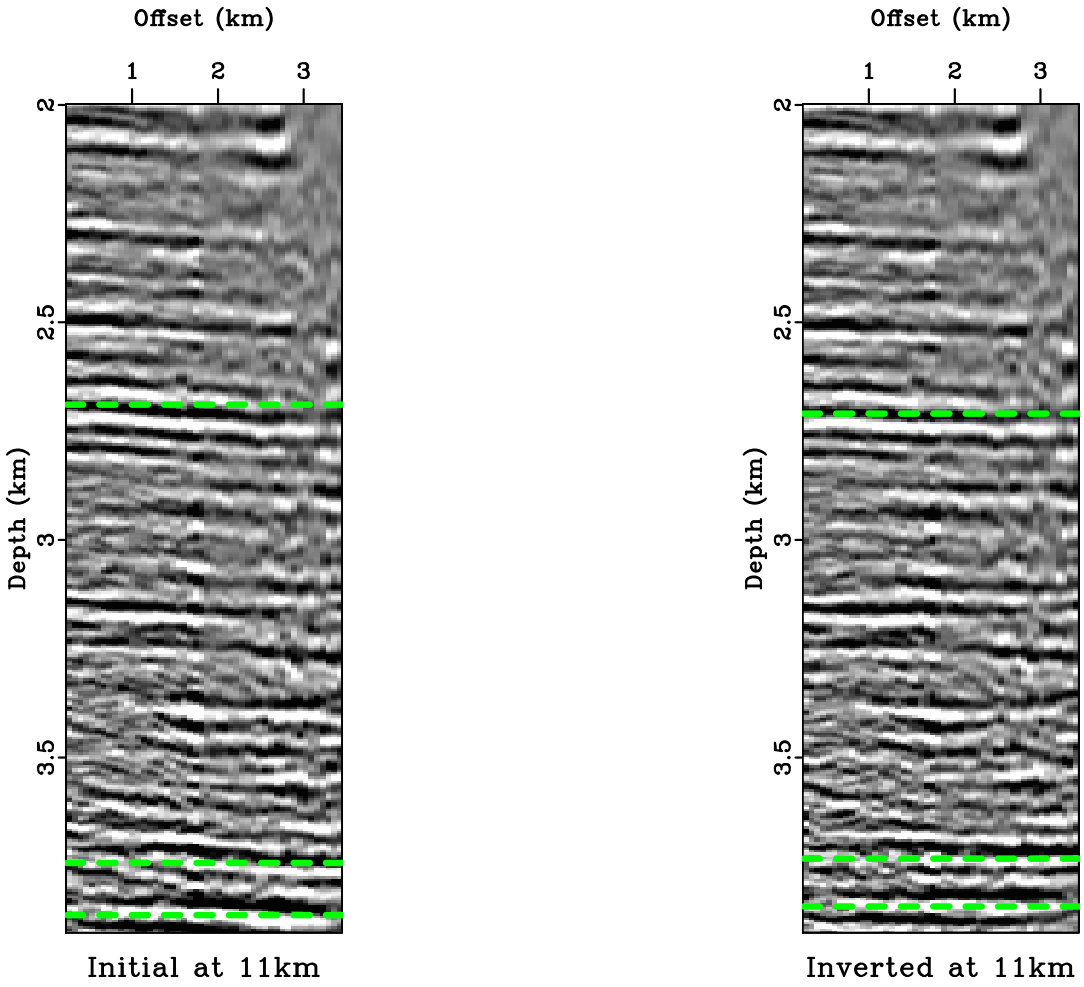


Figure 5.24: Zoom-in of deep parts of Figure 5.23. The events of prior model curve downward. time2dep/beinew cig1

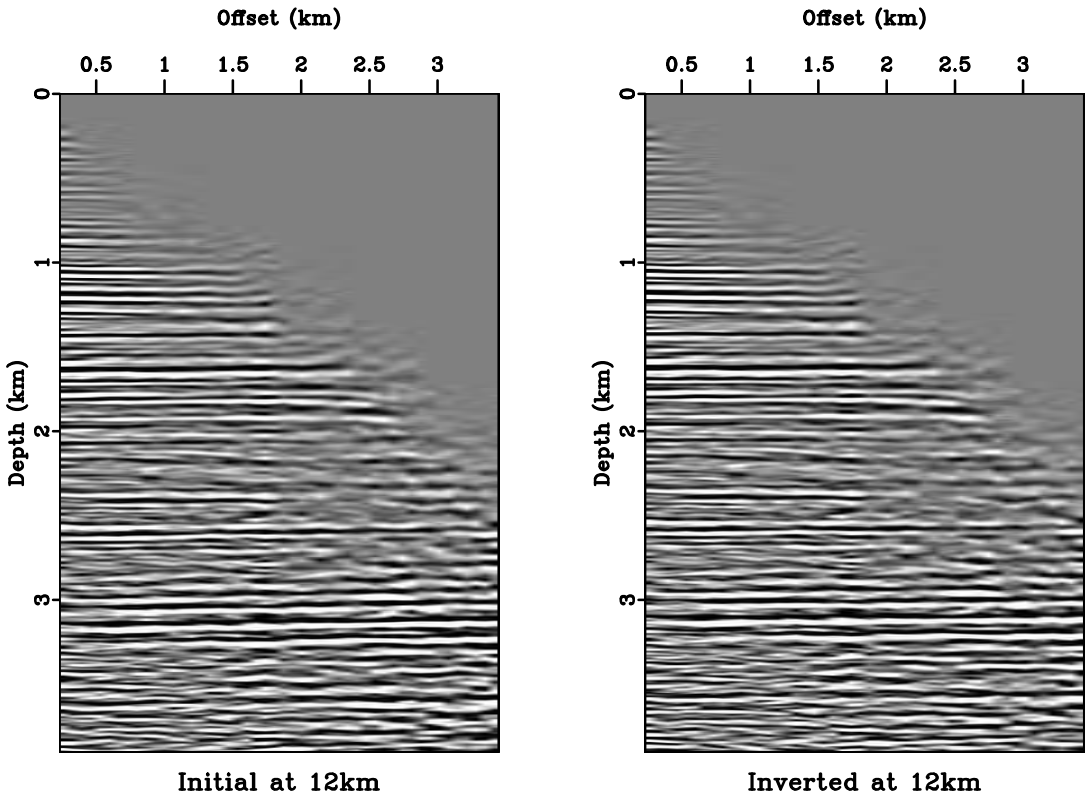


Figure 5.25: The surface offset common-image gathers at $x = 12$ km of (left) prior model and (right) inverted model. time2dep/beinew cig2a

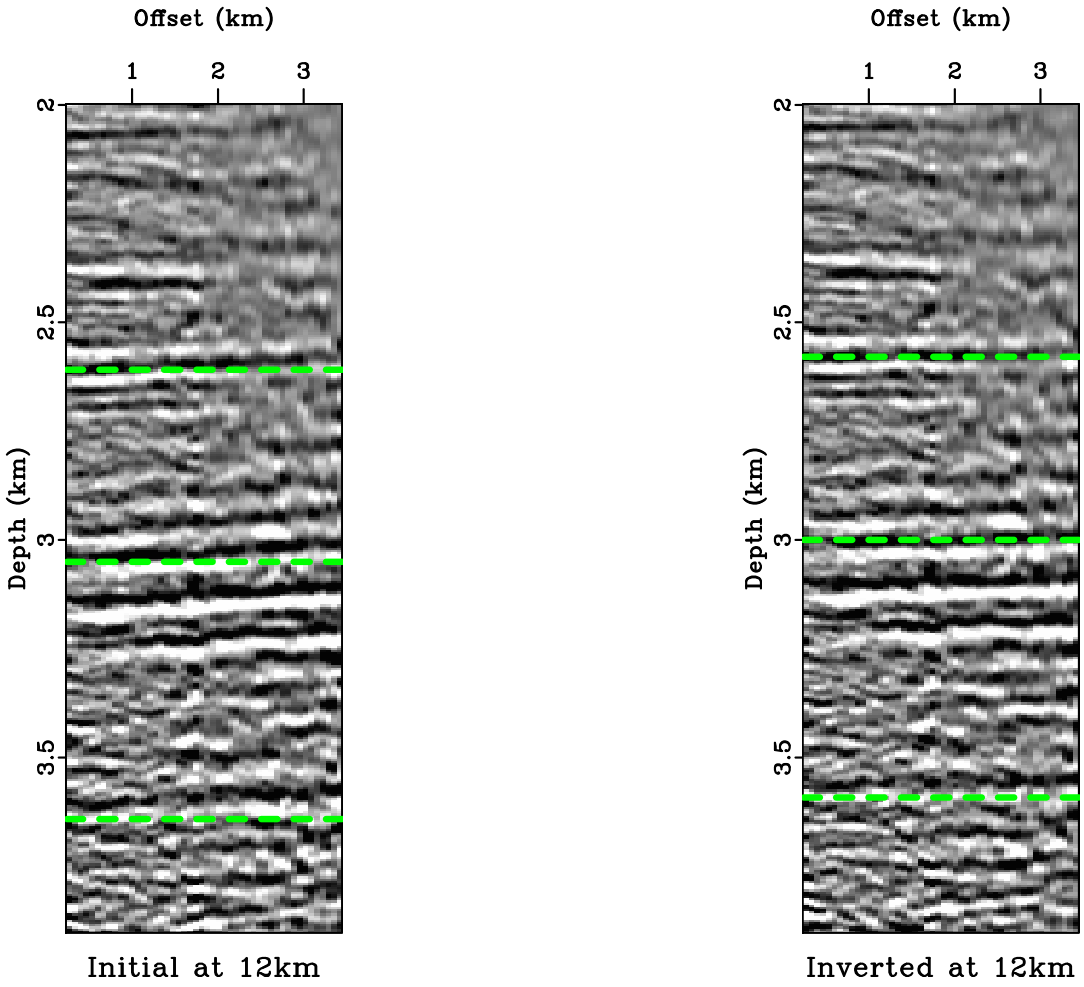


Figure 5.26: Zoom-in of deep parts of Figure 5.25. The events of prior model curve upward. time2dep/beinew cig2

to the field data example above, the time migration gives an interval velocity model through Dix inversion, as shown in Figure 5.27. After 6 iterations, the proposed time-to-depth conversion updates Dix-inverted model by decreasing the cost E to relative 1.6%. Figure 5.28 compares the inverted model and exact model, where the proposed method has greatly improved the model misfit beneath Gaussian anomaly. The time-migrated image can be mapped to depth according to both Dix-inverted model and inverted model. In Figure 5.30, although the proposed method does not result in perfectly fat reflector images as in Figure 1.10, the undesired image distortion due to Gaussian anomaly is considerably alleviated.

DISCUSSION

A 3-D extension of the proposed method is possible. Instead of a scalar x_0 we have both in-line and cross-line coordinates $\mathbf{x_0} = (x_0, y_0)$. Consequently, the geometrical spreading in 3-D becomes a matrix $\mathbf{Q}$, whose determinant relates the generalized Dix velocity and interval velocity (Cameron et al., 2007). Starting from this relationship, an iterative time-to-depth conversion can be established by following procedures similar to the ones outlined in this chapter.

The main limitation of my approach is the failure of underlying theory at caustics, which in turn limits either the depth or the extent of lateral velocity variation of the model. In practice, we could detect and mask out the caustic regions from cost function at each iteration. Another choice is to divide the original domain into several depth chunks, then apply velocity estimation and redatuming from one chunk to another recursively (Bevc, 1997). By doing so, image-ray crossing may not

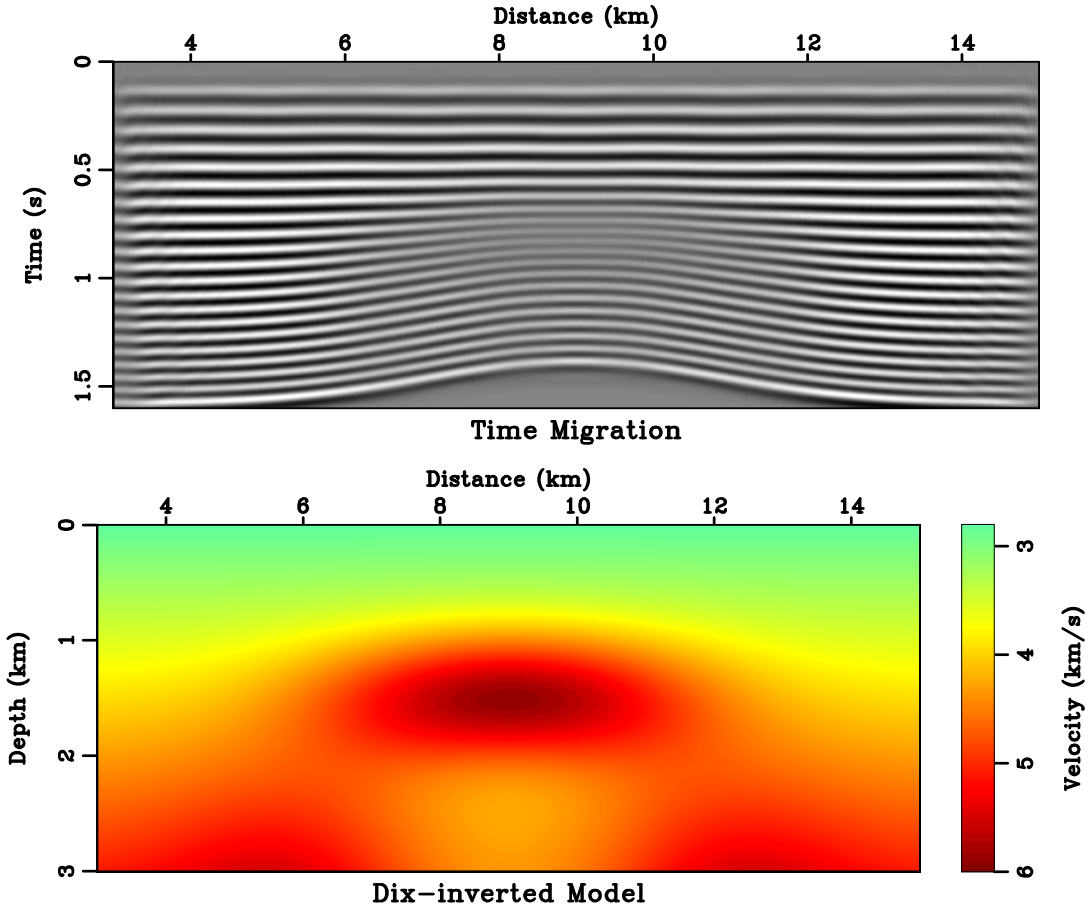


Figure 5.27: The time-migrated image (top) of redatumed data that is free of near-surface effects and the Dix-inverted prior model (bottom). time2dep/thesis tmigt

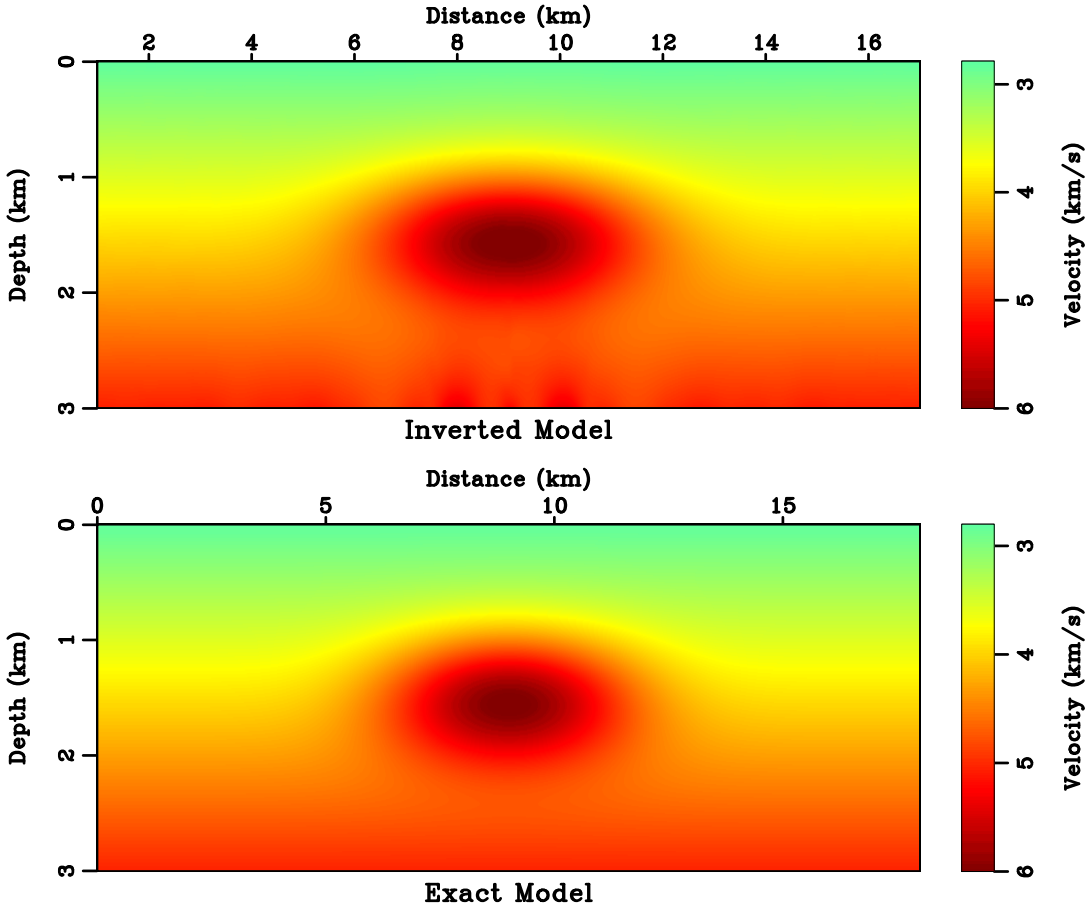


Figure 5.28: The inverted model by proposed iterative time-to-depth conversion (top) and the exact model (bottom). Compare with Figures 1.10 and 5.27. time2dep/thesis invt

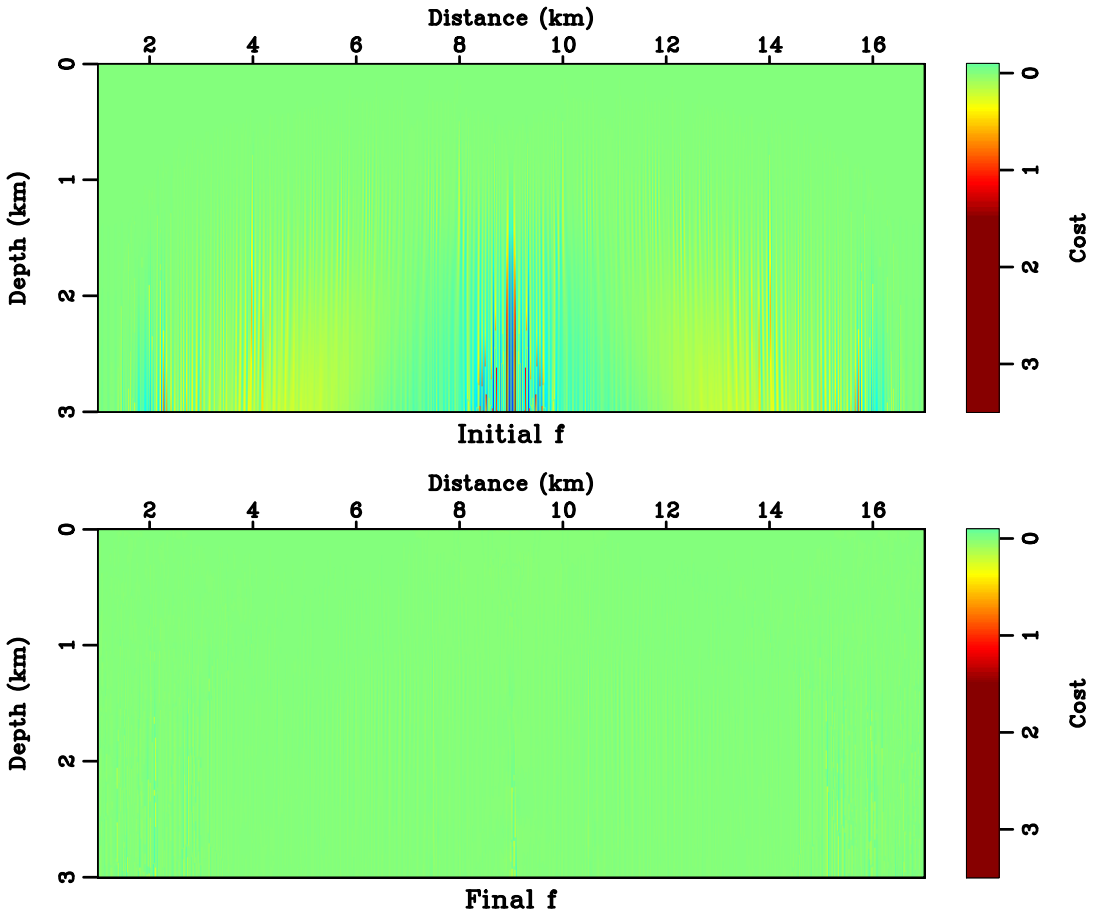


Figure 5.29: The costs of (top) prior model and (bottom) inverted model. time2dep/thesis costt

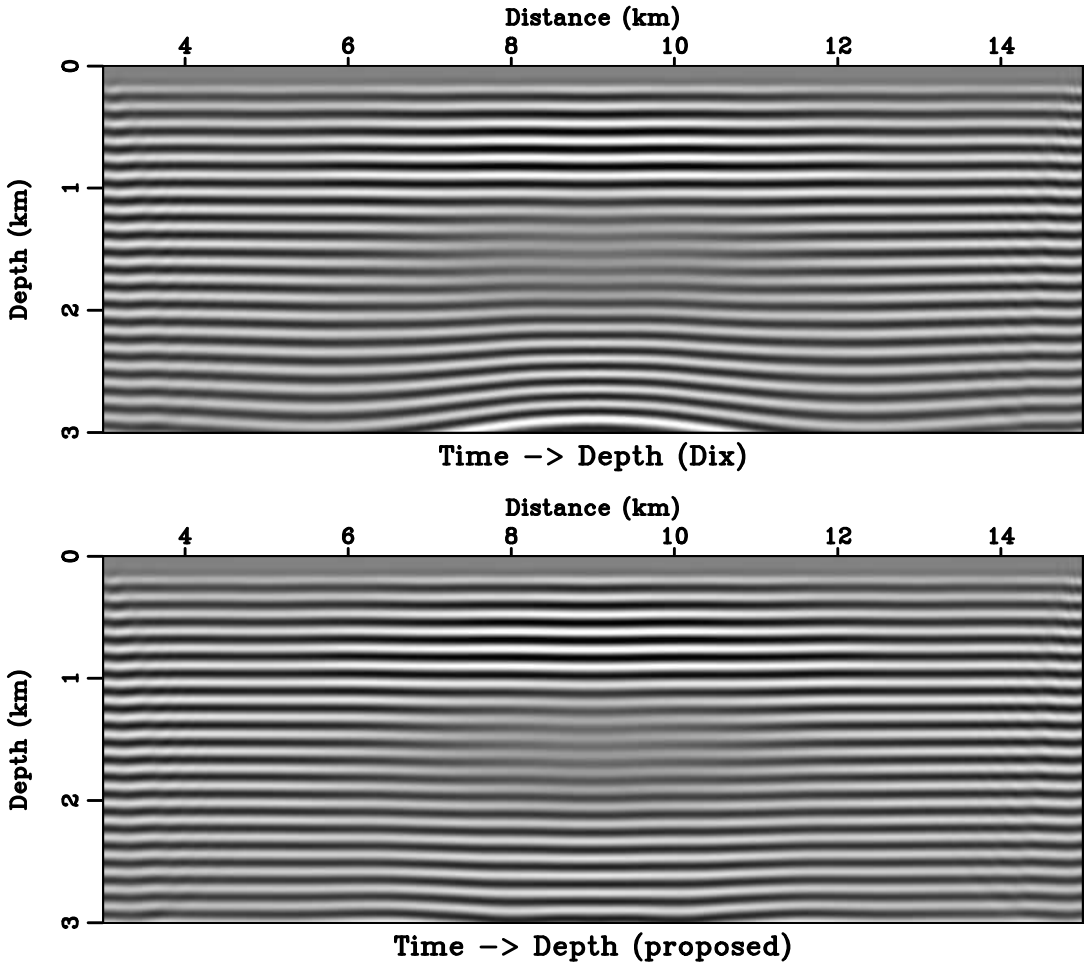


Figure 5.30: Time-migrated image in Figure 5.27 mapped to depth using (top) Dix-inverted model and (bottom) inverted model. time2dep/thesis mapdt

develop within each chunk thanks to a limited depth interval. The latter option is more attractive in both theoretical and computational aspects. Its implementation remains open for future research.

Another issue is the handling of in-flow boundaries other than the earth surface. In the constant velocity gradient and constant horizontal slowness-squared gradient examples, I avoid the problem by limiting the interval velocity model within image ray coverage. Alternatively, I could pad the input v_d laterally. The padding should simply spray the original boundaries. By doing so, image rays at the new left and right boundaries must run straightly downward as the media there are of $v(z)$ type. Although we could not expect the inversion to fix the in-flow boundary, the resulting errors might be local around that boundary. These errors can be negligible when considering migration, because the image in central region is of the most interest.

Finally, in my approach time-migration velocity not only determines the prior model but also drives the inversion. Consequently, errors in time-migration velocity have a direct influence on the accuracy of inverted model. In practice, the estimation of time-migration velocity is carried out with certain smoothing. Combined with regularization in the time-to-depth conversion, it is understandable that the resulting interval velocity model may contain limited fine-scale features and high velocity contrasts. In this regard, I suggest the method of this chapter as an efficient estimation of an initial guess for subsequent depth-imaging velocity model refinements.

CONCLUSIONS

I have introduced a novel nonlinear inversion formulation for time-to-depth velocity conversion and coordinate mapping, which reduces to solving a nonlinear system of partial differential equations iteratively. I turn one of the governing equations that relates seismic velocity to geometrical spreading of image rays into a cost function and linearize the other two equations: the eikonal equation and the orthogonality condition. Regularization provides extra constraints during inversion and fills the null space. The proposed method appears to be robust and fast converging.

Chapter 6

Conclusions

Contributions and summary

Ray theory plays an important role in exploration seismology. Although recent developments in wave-based imaging and velocity model building methods have demonstrated the potential of handling complex geologies, traveltime-based approaches are still a popular choice when it comes to efficiency, robustness and flexibility. Traveltime-based methods are often the first step in seismic processing. Wave-based methods can follow up as a second step if necessary, in order to obtain improved image and refined velocity model. Therefore, developments in traveltime-based approaches can directly decrease the overall processing cost and enhance the success rate of wave-based methods. My thesis is dedicated to improving current techniques and promoting new ideas in the field of traveltime-based methods.

The major contributions of this thesis cover three topics:

- Verification of the feasibility of first-break traveltime tomography based on double-square-root eikonal equation.

  I consider two upwind discretizations of the DSR eikonal equation and show that the explicit scheme is only conditionally convergent. On the other hand, an implicit upwind discretization is unconditionally convergent and monotonically causal. Therefore I develop a DSR eikonal solver with the implicit scheme

and a special local update strategy that handles the DSR singularity at horizontal waves. I derive theories for the adjoint-state DSR tomography and apply upwind finite-differences in its implementation. The DSR tomography is compared against traditional eikonal-based tomography and appears to be more robust and faster to converge.

- Sparse source interpolation of traveltimes through an eikonal-based traveltime source-derivative.

  I adopt a first-order partial differential equation that originates from differentiating the eikonal equation to compute the traveltime source-derivatives conveniently. Unlike methods that rely on explicit finite-difference estimations, the accuracy of the eikonal-based derivative does not depend on input source sampling. I implement a cubic Hermite traveltime interpolation with the first-order traveltime source-derivatives, which effectively improves the efficiency of Kirchhoff migrations. I also make use of the traveltime source-derivatives in Kirchhoff anti-aliasing.

- New formulation for robust time-to-depth conversion in laterally variant model. Instead of extrapolation, I formulate the problem as a nonlinear optimization directly for interval velocity. By doing so, I use inversion to gradually update the Dix-inverted model in order to account for the geometrical spreading of image rays. I further derive the linear operators required in the inversion and explain how each linearization update is obtained as a feedback of information across the whole computational domain. Finally, I employ regularization to

mitigate numerical instability of the problem.

Discussion and future work

All methods covered in this thesis belong to the category of traveltime-based seismic imaging and velocity model building. Although the image quality of traveltime-based methods such as Kirchhoff migration is not always comparable with that of wave-based approaches, a fast Kirchhoff algorithm combined with advances in computing power make interactive migration possible. This is highly desirable for practical seismic processing. In this regard, my proposed method in Chapter 4 can be one of crucial components in realizing such a possibility. My work on DSR tomography is an attempt towards prestack data usage in velocity model building. Due to first-break constraints, the proposed work is limited to near surface. However, the DSR formula is also valid for depth migration. It means a prestack reflection tomography is in theory also feasible. Following the dispersion relation prescribed by DSR eikonal equation, a kinematically correct wave-based migration as well as its adjoint can be established. The advantage of such a DSR reflection tomography might be similar to those of DSR first-break tomography. Time-to-depth conversion on the other hand completes a bridge between the time- and depth-domain seismic processings. Although the time-migration in my work is carried out by Kirchhoff algorithm, it is possible to employ a wave-type migration in the time coordinate. After time-to-depth conversion, the mapped depth image will have the same quality as one from direct depth-domain wave-based migration. For this reason, time-domain processing plus time-to-depth conversion can be a very powerful package that addresses both imaging and velocity model building.

All three methods I have developed are currently under an isotropic medium assumption. Specifically, the eikonal equation adopted in this thesis does not consider anisotropy. The traveltime source-derivative formula can be derived in anisotropic case in a similar fashion as the current work. The result might require spatial derivatives of not only velocity but also anisotropic parameters. Therefore, differentiability of these anisotropic parameters is a key assumption for successful traveltime interpolation. The DSR forward modeling must be revisited in order to show that the implicit discretization is still applicable in the presence of anisotropy. If so, an ordered upwind scheme might be feasible for solving the equation efficiently. On the tomography side, due to the under-determined nature of the problem, certain model preconditioning is recommended for constraining the extra anisotropy parameters. Similarly, the theory of time-to-depth conversion allows anisotropy conveniently, as we could replace the isotropic eikonal equation for image-rays with an anisotropic one, while the other two equations might also need modifications. However, the linearization of cost with respect to anisotropy parameters might be very different from that of velocity. The optimization must carefully address these differences such that the model update is not only numerically stable but also geologically meaningful.

Appendix

PROGRAM DOCUMENTATIONS

In the Appendix I list documentations of the main programs developed in Chapters 3-5. These programs are realized following the Implementation sections of these chapters, along with the operators described in Chapter 2.

sfdsreiko: Double-square-root eikonal solver (2D)

```
sfdsreiko < in.rsf > out.rsf flag=flag.rsf alpha=alpha.rsf velocity=y
thres=5.e-5 tol=1.e-3 nloop=10 causal=y
```

string	**alpha=**	characteristic angle
bool	**causal=y** [y/n]	if y, neglect non-causal branches
string	**flag=**	upwind stencil flag
int	**nloop=10**	number of bisection root-search
float	**thres=5.e-5**	threshold (percentage)
float	**tol=1.e-3**	tolerance for bisection root-search
bool	**velocity=y** [y/n]	if y, velocity; n, slowness-squared

sfdsrtomo: Prestack first-arrival traveltime tomography (DSR)

```
sfdsrtomo < in.rsf > out.rsf reco=reco.rsf grad=grad.rsf flag=flag.rsf
mask=mask.rsf prec=prec.rsf adj=n velocity=y velocity=y shape=n
weight=n pow=2.  verb=n niter=5 cgiter=10 thres=5.e-5 tol=1.e-3
nloop=10 eps=0.  causal=y rect#=(1,1,...)  what=
```

bool	**adj=n** [y/n]	adjoint flag (for what=l)
bool	**causal=y** [y/n]	if y, neglect non-causal branches
int	**cgiter=10**	number of conjugate-gradients
float	**eps=0.**	regularization parameter
string	**flag=**	upwind stencil flag
string	**grad=**	gradient (for what=l)
string	**mask=**	data preconditioner
int	**niter=5**	number of iterations
int	**nloop=10**	number of bisection root-search
float	**pow=2.**	power for data weighting
string	**prec=**	model preconditioner
string	**reco=**	first-breaks
int	**rect#=(1,1,...)**	smoothing radius on #-th axis
bool	**shape=n** [y/n]	if y, shaping; n, Tikhonov
float	**thres=5.e-5**	threshold (percentage)
float	**tol=1.e-3**	tolerance for bisection root-search
bool	**velocity=y** [y/n]	if y, velocity; n, slowness-squared
bool	**verb=n** [y/n]	verbosity flag
bool	**weight=n** [y/n]	data weighting
string	**what=** [tomography,linear]	what to compute

sfeikods: Fast marching with source perturbation

```
sfeikods < vel.rsf > time.rsf shotfile=shots.rsf tdl1=tdl1.rsf
tds1=tds1.rsf tdl2=tdl2.rsf tds2=tds2.rsf vel=y order=2
sweep=n br1=d1 br2=d2 br3=d3 plane1=n plane2=n plane3=n
b1=plane[2]?n1:(int)(br1/d1+0.5) b2=plane[1]?n2:(int)(br2/d2+0.5)
b3=plane[0]?n3:(int)(br3/d3+0.5) zshot=0.  yshot=o2+0.5*(n2-1)*d2
xshot=o3+0.5*(n3-1)*d3 l=1
```

int	**b1=plane[0]?n1:(int)(br1/d1+0.5)**	source box size in z (sample)
int	**b2=plane[1]?n2:(int)(br2/d2+0.5)**	source box size in y (sample)
int	**b3=plane[2]?n3:(int)(br3/d3+0.5)**	source box size in x (sample)
float	**br1=d1**	source box size in z (physical)
float	**br2=d2**	source box size in y (physical)
float	**br3=d3**	source box size in x (physical)
float	**zshot=o1+0.5*(n1-1)*d1**	source location in z
float	**yshot=o2+0.5*(n2-1)*d2**	source location in y
float	**xshot=o3+0.5*(n3-1)*d3**	source location in x
string	**shotfile=**	shot locations
int	**l=1** [1:z; 2:y; 3:x]	source perturbation direction
int	**order=2** [1,2]	accuracy order
bool	**plane1=n** [y/n]	plane-wave source in z
bool	**plane2=n** [y/n]	plane-wave source in y
bool	**plane3=n** [y/n]	plane-wave source in x
bool	**sweep=n** [y/n]	if y, fast sweeping; n, fast marching
string	**tdl1=**	1st order relative source-derivative
string	**tdl2=**	2nd order relative source-derivative
string	**tds1=**	1st order source-derivative
string	**tds2=**	2nd order source-derivative
bool	**vel=y** [y/n]	if y, velocity; n, slowness-squared

sfkirdat: 2D Prestack Kirchhoff redatuming

```
sfkirdat < in.rsf > out.rsf sgreen=sgreen.rsf rgreen=rgreen.rsf verb=n
datum= aperture=50 taper=10 length=0.025
```

int	**aperture=50**	aperture (number of traces)
float	**datum=**	datum depth
float	**length=**	filter length (second)
string	**sgreen=**	traveltime table for sources
string	**rgreen=**	traveltime table for receivers
int	**taper=10**	taper (number of traces)
bool	**verb=n** [y/n]	verbosity flag

sfkirmig: 2D Prestack Kirchhoff depth migration

```
sfkirmig < dat.rsf > mig.rsf table=tbl.rsf deriv=der.rsf adj=y cmp=y
nt= nh=1 ns=1 t0=0.  dt= h0=0.  dh= s0=0.  ds= tau=0.  aperture=90.
antialias=1.  cig=n type=
```

bool	**adj=y** [y/n]	if y, migration; n, modeling
float	**antialias=1.**	antialiasing
float	**aperture=90.**	migration aperture (degree)
bool	**cig=n** [y/n]	if y, common offset/receiver gathers
bool	**cmp=y** [y/n]	if y, CMP gather; n, shot gather
string	**deriv=**	traveltime source-derivatives
float	**h0=0.**	offset/receiver origin (for adj=n)
float	**dh=**	offset/receiver sampling (for adj=n)
int	**nh=**	number of offsets/receivers (for adj=n)
float	**t0=0.**	time origin (for adj=n)
float	**dt=**	time sampling (for adj=n)
int	**nt=**	number of time (for adj=n)
float	**s0=0.**	source origin (for adj=n)
float	**ds=**	source sampling (for adj=n)
int	**ns=**	number of sources (for adj=n)
string	**table=**	traveltime tables
float	**tau=0.**	time-shift (second)
string	**type=** [linear,partial,hermit]	type of interpolation

sfirays: Fast marching for image rays and time coordinate

```
sfirays < in.rsf > out.rsf t0=t0.rsf x0=x0.rsf f0=f0.rsf velocity=y
order=1 thres=10.
```

string	**f0=**	one-sided stencil locations
int	**order=1** [1,2]	fast-marching accuracy order
string	**t0=**	time coordinate
string	**x0=**	time coordinate
float	**thres=10.**	threshold for caustics
bool	**velocity=y** [y/n]	if y, velocity; n, slowness-squared

sftdconvert: Iterative time-to-depth velocity conversion

```
sftdconvert < in.rsf > out.rsf dix=dix.rsf t0=t0.rsf x0=x0.rsf
f0=f0.rsf grad=grad.rsf cost=cost.rsf mask=mini.rsf prec=prec.rsf
velocity=y order=1 thres=10.  niter=1 cgiter=200 shape=n eps=0.1
nline=0 verb=n tol=1.e-6 rect#=(1,1,...)
```

int	**cgiter=5**	number of conjugate-gradients
string	**cost=**	cost
string	**dix=**	Dix velocity
float	**eps=0.1**	regularization parameter
string	**f0=**	one-sided stencil locations
string	**grad=**	gradient
string	**mask=**	data preconditioner
string	**mval=**	model preconditioner
int	**niter=5**	number of iterations
int	**nline=5**	number of line-search
int	**order=1** [1,2]	fast-marching accuracy order
string	**prec=**	model preconditioner
int	**rect#=(1,1,...)**	smoothing radius on #-th axis
bool	**shape=n** [y/n]	if y, shaping; n, Tikhonov
string	**t0=**	time coordinate
string	**x0=**	time coordinate
float	**thres=10.**	threshold for caustics
float	**tol=1.e-3**	tolerance for shaping
bool	**velocity=y** [y/n]	if y, velocity; n, slowness-squared
bool	**verb=n** [y/n]	verbosity flag

Bibliography

Abma, R., J. Sun, and N. Bernitsas, 1999, Antialiasing methods in Kirchhoff migration: Geophysics, **64**, 1783–1792.

Aki, K., and P. G. Richards, 1980, Quantitative seismology: W.H. Freeman & Co.

Aldridge, D. F., 1994, Linearization of the eikonal equation (short note): Geophysics, **59**, 1631–1632.

Alkhalifah, T., 2011a, Efficient traveltime compression for 3-D prestack Kirchhoff migration: Geophysical Prospecting, **59**, 1–9.

———, 2011b, Prestack traveltime approximations: 81st Annual International Meeting, SEG, Expanded Abstracts, 3017–3021.

Alkhalifah, T., and S. Fomel, 2010, An eikonal based formulation for traveltime perturbation with respect to the source location: Geophysics, **75**, no. 6, T175–T183.

Anderson, J. E., and I. Tscankin, 1997, Dip-moveout processing by Fourier transform in anisotropic media: Geophysics, **62**, 1260–1269.

Audebert, F., 2001, 3-D prestack depth migration: Why Kirchhoff?, *in* SEP-80: Stanford Exploration Project, 1–17.

Backus, G., and F. Gilbert, 1968, The resolving power of gross earth data: Geophysical Journal of the Royal Astronomical Society, **16**, 169–205.

Baek, H., H. Calandra, and L. Demanet, 2013, Registration-based guided least-squares waveform inversion: 75th EAGE Conference & Exhibition incorporating SPE EUROPEC, Tu–07–03.

Bartel, D., M. Busby, J. Nealon, and J. Zaske, 2006, Time to depth conversion and

uncertainty assessment using average velocity modeling: 66th Annual International Meetings, SEG Expanded Abstracts, 2166–2170.

Bashkardin, V., T. J. Browaeys, S. Fomel, F. Gao, S. A. Morton, S. Terentyev, and A. Vladimirsky, 2012, Phase-space computation of multi-arrival traveltimes, Part II: Implementation and application to angle-domain imaging: 82nd Annual International Meeting, SEG, Expanded Abstracts, 1537.1.

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

Belonosova, A. V., and A. S. Alekseev, 1974, About one formulation of the inverse kinematic problem of seismics for a two-dimensional continuously heterogeneous medium: Some methods and algorithms for interpretation of geophysical data (in Russian), Nauka, 137–154.

Bergman, B., A. Tryggvason, and C. Juhlin, 2004, High-resolution seismic traveltime tomography incorporating static corrections applied to a till-covered bedrock environment: Geophysics, **69**, 1082–1090.

Berkhout, A. J., 1979, Steep dip finite-difference migration: Geophysics, **27**, 196–213.

Bertsekas, D., 1982, Enlarging the region of convergence of Newton's method for constrained optimization: Journal of Optimization Theory and Applications, **36**, 221–251.

Bevc, D., 1997, Imaging complex structures with semirecursive Kirchhoff migration: Geophysics, **62**, 577–588.

Beylkin, G., 1985, Imaging of discontinuities in the inverse scattering problem by inversion of a generalized Radon transform: Journal of Mathematical Physics, **26**, 99–108.

Biondi, B., 2001, Kirchhoff imaging beyond aliasing: Geophysics, **66**, 654–666.

Biondi, B., and W. W. Symes, 2004, Angle-domain common-image gathers for migration velocity analysis by wavefield-continuation imaging: Geophysics, **69**, 1283–1298.

Björck, A., 1996, Numerical methods for least squares problems: Society for Industrial and Applied Mathematics.

Black, J. L., and M. A. Brzostowski, 1994, Systematics of time migration errors: Geophysics, **59**, 1419–1434.

Bleistein, N., 1987, On the imaging of reflectors in the earth: Geophysics, **52**, 931–942.

Bleistein, N., J. K. Cohen, and J. W. Stockwell, 2001, Mathematics of multidimentional seismic inversion: Springer, New York.

Bortfeld, R., 1989, Geometrical ray theory: rays and traveltimes in seismic systems (second-order approximation of traveltimes): Geophysics, **54**, 342–349.

Brenders, A. J., and R. G. Pratt, 2007, Efficient waveform tomography for lithospheric imaging: implications for realistic, two-dimensional acquisition geometries and low-frequency data: Geophysical Journal International, **168**, 152–170.

Bube, K. P., and J. K. Washbourne, 2008, Wave tracing: Ray tracing for the propagation of band-limited signals: Geophysics, **73**, no. 5, VE377–VE393.

Cameron, M., S. Fomel, and J. Sethian, 2007, Seismic velocity estimation from time migration: Inverse Problems, **23**, 1329–1369.

———, 2008, Time-to-depth conversion and seismic velocity estimation using time-migration velocity: Geophysics, **73**, no. 5, VE205–VE210.

———, 2009, Analysis and algorithms for a regularized Cauchy problem arising from a nonlinear elliptic PDE for seismic velocity estimation: Journal of Computational Physics, **228**, 7388–7411.

Carcione, J. M., D. Kosloff, and R. Kosloff, 1988, Viscoacoustic wave propagation simulation in the earth: Geophysics, **53**, 769–777.

Causse, E., R. Mittet, and B. Ursin, 1999, Preconditioning of full-waveform inversion in viscoacoustic media: Geophysics, **64**, 130–145.

Chacon, A., and A. Vladimirsky, 2012a, Fast two-scale methods for Eikonal equations: SIAM Journal on Scientific Computing, **33**, no. 3, A547–A578.

———, 2012b, A parallel two-scale methods for Eikonal equations: submitted to SIAM Journal on Scientific Computing.

Chapman, C., 2002, Fundamentals of seismic wave propagation: Cambridge University Press.

Claerbout, J. F., 1985, Imaging the earth's interior: Blackwell Scientific Publications.

———, 1996, Basic earth imaging: http://sepwww.stanford.edu/sep/prof/bei10.2008.pdf.

Claerbout, J. F., and S. M. Doherty, 1970, Coarse grid calculations of waves in inhomogeneous media with application to delineation of complicated seismic structure: Geophysics, **35**, 407–418.

Cox, M., 1999, Static corrections for seismic reflection surveys: Society of Exploration Geophysics.

Dellinger, J. A., S. H. Gray, G. E. Murphy, and J. T. Etgen, 2000, Efficient 2.5-D true-amplitude migration: Geophysics, **65**, 943–950.

Dessa, J. X., S. Operto, A. Nakanishi, G. Pascal, K. Uhira, and Y. Kaneda, 2004, Deep seismic imaging of the eastern Nankai Trough, Japan, from multifold ocean bottom seismometer data by combined traveltime tomography and prestack depth migration: Journal of Geophysical Research, **109**, B02111.

Detrixhe, M., C. Min, and F. Gibou, 2013, A parallel fast sweeping method for the eikonal equation: Journal of Computational Physics, **237**, 46–55.

Dijkstra, E. W., 1959, A note on two problems in connexion with graphs: Numerische Mathematik, **1**, 269–271.

Dix, C. H., 1955, Seismic velocities from surface measurements: Geophysics, **20**, 68–86.

Duchkov, A. A., and M. V. de Hoop, 2010, Extended isochron ray in prestack depth (map) migration: Geophysics, **75**, no. 4, S139–S150.

Ehinger, A., and P. Lailly, 1995, Velocity model determination by the SMART method: 65th Annual International Meeting, SEG, Expanded Abstracts, 739–742.

Engl, H. W., M. Hanke, and A. Neubauer, 1996, Regularization of inverse problems: Kluwer Academic Publishers.

Etgen, J., 1986, Prestack reverse time migration of shot profiles, *in* SEP-50: Stanford Exploration Project, 151–170.

Etgen, J., S. H. Gray, and Y. Zhang, 2009, An overview of depth imaging in exploration geophysics: Geophysics, **74**, no. 6, WCA5–WCA17.

Farmer, P. A., I. F. Jones, H. Zhou, R. I. Bloor, and M. C. Goodwin, 2006, Application of reverse time migration to complex imaging problems: First Break, **24**, 65–73.

Fletcher, R. P., X. Du, and P. J. Fowler, 2009, Reverse time migration in tilted transversely isotropic (TTI) media: Geophysics, **74**, no. 6, WCA179–WCA187.

Fomel, S., 2002, Antialiasing of Kirchhoff operators by reciprocal parameterization: Journal of Seismic Exploration, **10**, 293–310.

———, 2003, Time-migration velocity analysis by velocity continuation: Geophysics, **68**, 1662–1672.

———, 2004, On anelliptic approximations for *qp* velocities in transversally isotropic media: Geophysical Prospecting, **52**, 247–259.

———, 2007, Shaping regularization in geophysical estimation problems: Geophysics, **72**, R29–R36.

Fomel, S., S. Luo, and H. Zhao, 2009, Fast sweeping method for the factored eikonal equation: Journal of Computational Physics, **228**, 6440–6455.

Fomel, S., and J. A. Sethian, 2002, Fast-phase space computation of multiple arrivals: Proceedings of the National Academy of Sciences of the United States of America, **99**, 7329–7334.

Fomel, S., L. Ying, and X. Song, 2013, Seismic wave extrapolation using lowrank symbol approximation: Geophysical Prospecting, **63**, no. 3, 526–536.

Fowler, P. J., X. Du, and R. P. Fletcher, 2010, Coupled equations for reverse time migration in transversely isotropic media: Geophysics, **75**, no. 1, S11–S22.

Franklin, J. B., and J. M. Harris, 2001, A high-order fast marching scheme for the linearized eikonal equation: Journal of Computational Acoustics, **9**, 1095–1109.

French, W. S., 1975, Computer migration of oblique seismic reflection profiles: Geophysics, **40**, 961–980.

Gazdag, J., 1981, Modeling of the acoustic wave equation with transform methods: Geophysics, **46**, 854–859.

Geoltrain, S., and J. Brac, 1993, Can we image complex structures with first-arrival traveltime?: Geophysics, **58**, 564–575.

Gray, S. H., 2005, Gaussian beam migration of common-shot records: Geophysics, **70**, S71–S77.

Guitton, A., A. Gboyega, and E. Diaz, 2012, Constrained full-waveform inversion by model reparameterization: Geophysics, **77**, no. 2, R117–R127.

Haddon, R. A. W., and P. W. Buchen, 1981, Use of Kirchhoff's formula for body wave

calculations in the earth: Geophysical Journal of the Royal Astronomical Society, **67**, 587–598.

Hale, D., 1984, Dipmoveout by Fourier transform: Geophysics, **49**, 741–757.

Hardy, P., 2013, Ongoing R&D in ray based tomography: well worth the effort: 83rd Annual International Meeting, SEG, Expanded Abstracts, 0878.1.

Hestenes, M. R., and E. Stiefel, 1952, Method of conjugate gradients for solving linear systems: Journal of Research of the National Bureau of Standards, **49**, 409–436.

Hill, N. R., 1990, Gaussian beam migration: Geophysics, **55**, 1416–1428.

———, 2001, Prestack Gaussian beam depth migration: Geophysics, **66**, 1240–1250.

Hubral, P., 1977, Time migration—some ray theoretical aspects: Geophysical Prospecting, **25**, 738–745.

Iversen, E., 2004, The isochron ray in seismic modeling and imaging: Geophysics, **69**, 1053–1070.

Iversen, E., and M. Tygel, 2008, Image-ray tracing for joint 3D seismic velocity estimation and time-to-depth conversion: Geophysics, **73**, no. 3, S99–S114.

Jeong, W., and R. T. Whitaker, 2008, A fast iterative method for eikonal equations: SIAM Journal on Scientific Computing, **30**, 2512–2534.

Jones, I. F., 2010, An introduction to: Velocity model building: EAGE.

Jones, I. F., R. I. Bloor, B. L. Biondi, and J. T. Etgen, 2008, Prestack depth migration and velocity model building: Society of Exploration Geophysicists.

Kim, S., 2001, An $O(N)$ level set method for eikonal equations: SIAM Journal on Scientific Computing, **22**, 2178–2193.

Knuth, D. E., 2011, The art of computer programming, volume 3: Addison-Wesley Professional.

Larner, K. L., L. Hatton, B. S. Gibson, and I. S. Hsu, 1981, Depth migration of imaged time sections: Geophysics, **46**, 734–750.

Leung, S., and J. Qian, 2006, An adjoint state method for three-dimensional transmission traveltime tomography using first arrivals: Communications in Mathematical Sciences, **4**, 249–266.

Li, S., and S. Fomel, 2013, Kirchhoff migration using eikonal-based computation of traveltime source-derivatives: Geophysics, **78**, no. 4, S211–S219.

Li, S., S. Fomel, and A. Vladimirsky, 2011, Improving wave-equation fidelity of Gaussian beams by solving the complex eikonal equation: 71st Annual International Meeting, SEG, Expanded Abstracts, 3829–3834.

Li, S., A. Vladimirsky, and S. Fomel, 2013, First-break traveltime tomography with the double-square-root eikonal equation: Geophysics, **78**, no. 6, U89–U101.

Liu, H., O. Runborg, and N. M. Tanushev, 2013, Error estimates for Gaussian beam superpositions: Mathematics of Computation, **82**, 919–952.

Liu, H. P., D. L. Anderson, and H. Kanamori, 1976, Velocity dispersion due to anelasticity: implications for seismology and mantle composition: Geophysical Journal of the Royal Astronomical Society, **47**, 41–58.

Liu, Z., and N. Bleistein, 1995, Mathematical analysis of residual moveout and velocity analysis: 65th Annual International Meeting, SEG, Expanded Abstracts, 1201–1203.

Loewenthal, D., L. Lu, R. Roberson, and J. Sherwood, 1976, The wave equation applied to migration: Geophysical Prospecting, **24**, 380–399.

Lumley, D. E., J. F. Claerbout, and D. Bevc, 1994, Anti-aliased Kirchhoff 3-D migration: 64th Annual International Meetings, SEG Expanded Abstracts, 1282–1285.

Luo, Y., and G. Schuster, 1991, Wave-equation traveltime inversion: Geophysics, **56**, 645–653.

Lynn, W. S., and J. F. Claerbout, 1982, Velocity estimation in laterally varying media: Geophysics, **47**, 884–897.

Ma, Y., D. Hale, B. Gong, and Z. Meng, 2012, Image-guided sparse-model full waveform inversion: Geophysics, **77**, no. 4, R189–R198.

Marsden, D., 1993, Static corrections—a review: The Leading Edge, **12**, 43–49.

Mendes, M., 2000, Green's function interpolation for prestack imaging: Geophysical Prospecting, **48**, 49–62.

Mulder, W. A., and A. P. ten Kroode, 2002, Automatic velocity analysis by differential semblance optimization: Geophysics, **67**, 1184–1191.

Nichols, D. E., 1994, Imaging complex structures using band-limited Green's functions: Ph.D. thesis, Stanford University.

Noble, M., P. Thierry, C. Taillandier, and H. Calandra, 2010, High-performance 3D first-arrival traveltime tomography: The Leading Edge, **29**, 86–93.

Nolet, G., 1987, Seismic tomography: With applications in global seismology and exploration geophysics: Springer, New York.

Osypov, K., 2000, Robust refraction tomography: 70th Annual International Meeting, SEG, Expanded Abstracts, 2032–2035.

Paige, C. C., and M. A. Saunders, 1982, LSQR: an algorithm for sparse linear equations and sparse least squares: ACM Transactions on Mathematical Software, **8**, 43–71.

Pei, D., 2009, Three-dimensional traveltime tomography via LSQR with regularization: 79th Annual International Meeting, SEG, Expanded Abstracts, 4004–4008.

Perrone, F., P. Sava, C. Andreoletti, and N. Bienati, 2014, Linearized wave-equation migration velocity analysis by image warping: Geophysics, **79**, no. 2, S35–S46.

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

Popov, M. M., 2002, Ray theory and Gaussian beam method for geophysicists: EDUFBA.

Popovici, A. M., 1996, Prestack migration by split-step DSR: Geophysics, **61**, 1412–1416.

Popovici, A. M., and J. Sethian, 2002, 3-D imaging using higher order fast marching traveltimes: Geophysics, **67**, 604–609.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 2007, Numerical recipes: The art of scientific computing: Cambridge University Press.

Qian, J., and W. W. Symes, 2002, An adaptive finite-difference method for traveltimes and amplitudes: Geophysics, **67**, 167–176.

Reshef, M., and D. Kosloff, 1986, Migration of common shot gathers: Geophysics, **51**, 324–331.

Rickett, J., and S. Fomel, 1999, A second-order fast marching eikonal solver: Stanford Exploration Project Report, **100**, 287–292.

Ristow, D., and T. Ruhl, 1994, Fourier finite-difference migration: Geophysics, **59**, 1882–1893.

Robein, E., 2003, Velocities, time-imaging and depth-imaging in reflection seismics: EAGE.

Sava, P., and B. Biondi, 2004, Wave-equation migration velocity analysis, I. Theory: Geophysical Prospecting, **52**, no. 5, 593–606.

Sava, P., and S. Fomel, 2003, Angle-domain common image gathers by wavefield continuation methods: Geophysics, **68**, 1065–1074.

Schleicher, J., M. Tygel, and P. Hubral, 2007, Seismic true-amplitude imaging: Society of Exploration Geophysicists.

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

Serdyukov, A. S., and A. A. Duchkov, 2013, Fast computation of traveltimes for multifold seismic data solving the double-square-root eikonal equation: 75th EAGE Conference & Exhibition incorporating SPE EUROPEC, We–02–10.

Sethian, J. A., 1996, A fast marching level set method for monotonically advancing fronts: Proceedings of the National Academy of Sciences, 1591–1595.

———, 1999, Level set methods and fast marching methods: Evolving interfaces in computational geometry, fluid mechanics, computer vision and material sciences: Cambridge University Press.

Sethian, J. A., and A. M. Popovici, 1999, 3-D traveltime computation using the fast marching method: Geophysics, **64**, 516–523.

Sethian, J. A., and A. Vladimirsky, 2001, Ordered upwind methods for static Hamilton-Jacobi equations: Proceedings of the National Academy of Sciences of the United States of America, **98**, 11069–11074.

Shen, P., and W. W. Symes, 2008, Automatic velocity analysis via shot profile migration: Geophysics, **73**, no. 5, VE49–VE59.

Sheng, J., A. Leeds, M. Buddensiek, and G. T. Schuster, 2006, Early arrival waveform tomography on near-surface refraction data: Geophysics, **71**, no. 4, U47–U57.

Sheriff, R. E., and L. P. Geldart, 1995, Exploration seismology: Cambridge University Press.

Shin, C., and D. J. Min, 2006, Waveform inversion using a logarithmic wavefield: Geophysics, **71**, no. 3, R31–R42.

Simmons, J. L., and N. Bernitsas, 1994, Nonlinear inversion of first-arrival times: 64th Annual International Meeting, SEG, Expanded Abstracts, 992–995.

Slotnick, M. M., 1959, Lessons in seismic computing: Society of Exploration Geophysics.

Stefani, J. P., 1993, Possibilities and limitations of turning ray tomography—a synthetics study: 63rd Annual International Meeting, SEG, Expanded Abstracts, 610–612.

Stoffa, P. L., J. T. Fokkema, R. M. de Luna Freire, and W. P. Kessinger, 1990, Split-step fourier migration: Geophysics, **55**, 410–421.

Stoughton, D., J. Stefani, and S. Michell, 2001, 2D elastic model for wavefield investigations of subsalt objectives, deep water Gulf of Mexico: 63rd EAGE Conference & Exhibition, A–33.

Symes, W. W., 2007, Reverse time migration with optimal checkpointing: Geophysics, **72**, no. 5, SM213–SM221.

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

Symes, W. W., and J. Qian, 2003, A slowness matching Eulerian method for multivalued solutions of eikonal equations: Journal of Scientific Computing, **19**, 501–526.

Taillandier, C., M. Noble, H. Chauris, and H. Calandra, 2009, First-arrival traveltime tomography based on the adjoint-state method: Geophysics, **74**, no. 6, WCB57–WCB66.

Tang, Y., S. Lee, A. Baumstein, and D. Hinkley, 2013, Tomographically enhanced full

waveeld inversion: 83rd Annual International Meeting, SEG, Expanded Abstracts, 1145.1.

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

Tikhonov, A. N., 1963, Solution of incorrectly formulated problems and the regularization method: Soviet Mathematics Doklady, 1035–1038.

Tromp, J., C. Tape, and Q. Liu, 2005, Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels: Geophysical Journal International, **160**, 195–216.

Tsvankin, I., 2012, Seismic signatures and analysis of reflection data in anisotropic media: Society of Exploration Geophysicists.

Ursin, B., 1982, Quadratic wavefront and traveltime approximations in inhomogeneous layered media with curved interfaces: Geophysics, **47**, 1012–1021.

Vanelle, C., and D. Gajewski, 2002, Second-order interpolation of traveltimes: Geophysical Prospecting, **50**, no. 1, 73–83.

Vanelle, C., M. Spinner, T. Hertweck, C. Jäger, and D. Gajewski, 2006, Traveltime-based true-amplitude migration: Geophysics, **71**, no. 6, S251–S259.

Červený, V., 2001, Seismic ray theory: Cambridge University Press.

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

Vidale, J., 1988, Finite-difference calculation of traveltimes: Bulletin of the Seismological Society of America, **78**, 2062–2076.

———, 1990, Finite-difference calculation of traveltimes in three dimensions: Geophysics, **55**, 521–526.

Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, no. 6, WCC1–WCC26.

Weber, O., Y. Devir, A. Bronstein, M. Bronstein, and R. Kimmel, 2008, Parallel algorithms for the approximation of distance maps on parametric surfaces: ACM Transactions on Graphics, **27**, no. 4, 104.

Weibull, W. W., and B. Arntsen, 2013, Automatic velocity analysis with reverse-time migration: Geophysics, **78**, no. 4, S179–S192.

Woodward, M., and F. Rocca, 1988, Wave-equation tomography: 58th Annual International Meeting, SEG, Expanded Abstracts, 1232–1235.

Xie, X. B., and H. Yang, 2008, The finite-frequency sensitivity kernel for migration residual moveout and its applications in migration velocity analysis: Geophysics, **73**, no. 6, S241–S249.

Xu, S., H. Chauris, G. Lambare, and M. Noble, 2001, Common-angle migration: A strategy for imaging complex media: Geophysics, **66**, 1877–1894.

Yatziv, L., A. Bartesaghi, and G. Sapiro, 2006, A fast $O(N)$ implementation of the fast marching algorithm: Journal of Computational Physics, **212**, 393–399.

Yilmaz, O., 2001, Seismic data analysis: Processing, inversion, and interpretation of seismic data: Society of Exploration Geophysicists.

Yilmaz, O., I. Tanir, and C. Gregory, 2001, A unified 3-D seismic workflow: Geophysics, **66**, 1699–1713.

Zelt, C. A., and P. J. Barton, 1998, 3D seismic refraction tomography: A comparison of two methods applied to data from the Faeroe Basin: Journal of Geophysical Research, **103**, 7187–7210.

Zhang, Y., and B. Biondi, 2013, Moveout-based wave-equation migration velocity analysis: Geophysics, **78**, no. 2, U31–U39.

Zhang, Y., and G. Zhang, 2009, One-step extrapolation method for reverse time migration: Geophysics, **74**, no. 4, A29–A33.

Zhang, Y., G. Zhang, and N. Bleistein, 2005, Theory of true-amplitude one-way wave equations and true-amplitude common-shot migration: Geophysics, **70**, no. 4, E1–E10.

Zhang, Z., L. Huang, and Y. Lin, 2012, A wave-energy-based precondition approach to full-waveform inversion in the time domain: 82nd Annual International Meeting, SEG, Expanded Abstracts, 1555.1.

Zhao, H., 2005, A fast sweeping method for eikonal equations: Mathematics of Computation, **74**, 603–627.

———, 2007, Parallel implementations of the fast sweeping method: Journal of Computational Mathematics, **25**, 421–429.

Zhou, H., L. Amundsen, and G. Zhang, 2012, Fundamental issues in full waveform inversion: 82nd Annual International Meeting, SEG, Expanded Abstracts, 4806–4810.

Zhu, T., S. Cheadle, A. Petrella, and S. Gray, 2000, First-arrival tomography: method and application: 70th Annual International Meeting, SEG, Expanded Abstracts, 2028–2031.

Zhu, X., D. P. Sixta, and B. G. Angstman, 1992, Tomostatics: turning-ray tomography + static corrections: The Leading Edge, **11**, 15–23.

Vita

Siwei Li was born in Chengdu, China in March 1987. He graduated from Shishi High School in his hometown in 2005. In 2009, he completed the degree of Bachelor of Science in Geophysics from Peking University in Beijing. Later in the same year, he came to United States to pursue the degree of Doctor of Philosophy in Geophysics at The University of Texas at Austin.

Email: llisiw@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.