Copyright
by
Vladimir Bashkardin
2014

The Dissertation Committee for Vladimir Bashkardin
certifies that this is the approved version of the following dissertation:

Phase-Space Imaging of Reflection Seismic Data

Committee:

Sergey Fomel, Supervisor

Paul Stoffa, Co-Supervisor

Scott Morton

Mrinal Sen

Karl Schultz

Phase-Space Imaging of Reflection Seismic Data

by

Vladimir Bashkardin, B.Sc.

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
August 2014

Acknowledgments

Failures are not something to be avoided.
You want to have them happen as quickly as
you can so you can make progress rapidly.

Gordon Earle Moore, 2000

The quote in the epigraph belongs to one of the founders of Intel Corporation and the author of the famous Moore's Law. He said that in an interview, explaining what comprises the very essence of engineering. I believe that failures are even more important in science. In my time at UT Austin, I certainly learned from my scientific failures as much as I did from successes. The work that I describe in this thesis took more than four years to complete. For every line of the computational code that underline the presented results here, at least two lines were discarded. For every figure displaying a successful seismic image, at least two figures had to be thrown away before the final acceptable computational imaging framework was developed. It was quite discouraging at the time, but in hindsight, I can see that it was an absolutely critical part of the research process. The presented results would not be possible without those failures. One of the main lessons that I learned as a grad student is that failures in science and computational research should be expected and welcomed. I am grateful to my teachers and advisers at UT Austin that I was put into an environment where research failures are not something to be punished for but rather a reason to learn and make progress toward better science.

Most obviously, I am thankful to my adviser, Sergey Fomel. It was quite a privilege to be selected by him as one of his first two students. I witnessed over years his extremely hard day-to-day work of building his own research group. It is an honor to be considered one of the contributors to what is now known as Texas Consortium for Computational Seismology. Not only Sergey created one of the currently most productive academic research groups in computational geophysics, but he did so firmly holding on to the principles of computational reproducibility and open access to computational experiments and data.

Paul Stoffa, Mrinal Sen, and Steve Grand contributed immensely to my education as geophysics professors. I had the most comprehensive class on introduction to seismology from Steve. Mrinal's class helped me tremendously to better understand the basics of inversion problems in geophysics. Paul kindly agreed to be a co-adviser for this thesis. I received invaluable advice from him in my early years at UT. I learned about interesting concepts of non-linear inversion in seismic problems from our work together.

My experience in the first years at UT would not be the same without Will Burnett and Jules Browayes. Will was also accepted by Sergey as one of his first PhD students. We started at the same time and shared an office for a while. Jules was a postdoc working with Sergey at the time; his office was next door to ours. The three of us discussed research problems and worked at a whiteboard many more times than I can remember. It is due to them that I recognize communication and free exchange of scientific ideas as an absolutely essential component of research. As the group grew, I had opportunities to work with other scientists and students. I am thankful to Yang Liu, Hesam Kazemeini, Yihua Cai, Karl Schleicher, and Yanadet Sripanich for our interaction and collaboration.

This research work would not be possible without the generous support of certain industry sponsors. I thank Hess and Total for providing an opportunity to work on the challenging problems described in this thesis. In my mind, this project can be viewed as a great example of how research interaction between academia and industry should be organized, since we were provided academic freedom to explore different solutions and publish results freely. On behalf of Hess, Scott Morton, Sergey Terentyev, and Thomas Cullison participated in the project. Paul Williamson and Fuchun Gao contributed from Total's side. We had numerous meetings in Houston and Austin to discuss progress and agree on solutions. I also worked as an intern in both companies. Through these internship and work on the project, I obtained some invaluable knowledge about everyday industry practices of practical seismic imaging.

I would also like to thank Keith Gray, the leader of High Performance Computing Group at BP. He provided an opportunity to work outside of School of Geosciences at UT. I spent one semester as a research assistant at Texas Advanced Computing Center working on BP's computational code optimization for accelerator hardware architectures. I was supervised by Lars Koesterke, from whom I gained better understanding of optimization intricacies. It was a crucial time for me that solidified my decision to steer my career towards HPC problems in computational seismology.

Jackson School of Geosciences and UT Austin provide unparalleled learning environment. I cherish memories of the time that I spent on the beautiful UT campus. Austin itself is a truly unique city, where I lived one of my most difficult and most interesting years. I am happy that I decided to go to UT Austin for PhD studies.

Finally, I can not thank my family enough for the support that I was given during my studies. My wife helped me stay organized and focused even in most challenging times. My son, who had been born less than a year before the thesis was written, kept me motivated to finish this work (unbeknownst to him, though). I owe the fact that this work is complete to them.

VLADIMIR BASHKARDIN

The University of Texas at Austin
March 2014

Phase-Space Imaging of Reflection Seismic Data

Publication No. ____________________

Vladimir Bashkardin, Ph.D.

The University of Texas at Austin, 2014

Supervisors: Sergey Fomel
Paul Stoffa

Modern oil and gas exploration depends on a variety of geophysical prospect tools. One of them is reflection seismology that allows to obtain interwell information of sufficient resolution economically. This exploration method collects reflection seismic data on the surface of an area of prospect interest and then uses them to build seismic images of the subsurface.

All imaging approaches can be divided into two groups: wave equation-based methods and integral schemes. Kirchhoff migration, which belongs to the second group, is an indispensable tool in seismic imaging due to its flexibility and relatively low computational cost. Unfortunately, the classic formulation of this method images only a part of the surface data, if so-called multipathing is present in it. That phenomenon occurs in complex geologic settings, such as subsalt areas, when seismic waves travel between a subsurface point and a surface location through more than one path.

The quality of imaging with Kirchhoff migration in complex geological areas can be improved if multiple paths of ray propagation are included in the integral. Multiple arrivals can be naturally incorporated into the imaging operator if it is expressed as an integral over subsurface take-off angles. In this form, the migration operator involves escape functions that connect subsurface locations with surface seismic data values through escape traveltime and escape positions. These escape quantities are functions of phase space coordinates that are simply related to the subsurface reflection system. The angle-domain integral operator produces output scattering- and dip-angle image gathers, which represent a convenient domain for subsurface analysis. Escape functions for angle-domain imaging can be simply computed with initial-value ray tracing, a Lagrangian computational technique. However, the computational cost of such a bottom-up approach can be prohibitive in practice. The goal of this work was to construct a computationally efficient phase space imaging framework. I designed several approaches to computing escape functions directly in phase space for mapping surface seismic reflection data to the subsurface angle domain.

Escape equations have been introduced previously to describe distribution of escape functions in the phase space. Initially, I employed these equations as a basis for building an Eulerian numerical scheme using finite-difference method in the 2-D case. I show its accuracy constraints and suggest a modification of the algorithm to overcome them. Next, I formulate a semi-Lagrangian approach to computing escape functions in 3-D. The second method relies on the fundamental property of continuity of these functions in the phase space. I define locally constrained escape functions and show that a global escape solution can be reconstructed from local solutions iteratively. I validate the accuracy of the proposed methods by imaging synthetic seismic data in several complex 2-D and 3-D models. I draw conclusions about efficiency by comparing the compute time of the imaging tests with the compute

time of a well-optimized conventional initial-value ray tracing.

Table of Contents

Acknowledgments **iv**

Abstract **viii**

List of Tables **xii**

List of Figures **xiii**

Chapter 1. Introduction **1**

Chapter 2. Escape equations and reduced phase space **17**

Chapter 3. Finite-difference solution of escape equations **34**

Chapter 4. Semi-Lagrangian method for computing escape functions **62**

Chapter 5. 2-D angle-domain Kirchhoff imaging with escape functions **80**

Chapter 6. 3-D angle-domain Kirchhoff imaging with escape functions **106**

Chapter 7. Conclusions **140**

Bibliography **145**

Vita **154**

List of Tables

6.1 Correspondence between parameters and figures in the imaging accuracy test for the semi-Lagrangian algorithm. 115

List of Figures

1.1 Parameterization of Kirchhoff integral imaging operator (a scheme): by surface coordinate (left), by subsurface phase angle (right). 7

1.2 Ray trajectories computed for a source point at $x = 3.0$ km and $z = 2.75$ km in the low-velocity anomaly model. 12

1.3 Exit (escape) position $\widehat{x}$ as a function of the take-off angle θ for ray trajectories computed for a source point at $x = 3.0$ km and $z = 2.75$ km in the low-velocity anomaly model. 12

1.4 Exit (escape) traveltime $\widehat{T}$ as a function of the take-off angle θ for ray trajectories computed for a source point at $x = 3.0$ km and $z = 2.75$ km in the low-velocity anomaly model. 13

2.1 Slowness vector $\mathbf{p}$ in 2-D (a scheme). 19

2.2 Slowness vector $\mathbf{p}$ in 3-D (a scheme). 20

2.3 Consolidation of characteristic lines in polar areas of the spherical coordinate system (a scheme). 23

2.4 Smoothed Marmousi model velocity profile for escape value computations. 27

2.5 Ray trajectories in Marmousi model computed for location at $x = 6.5$ km and $z = 2.0$ km. 27

2.6 Escape depth function $\widehat{z}$ for all locations at constant depth $z = 2.0$ km of Marmousi model. 28

2.7 Escape lateral position function $\widehat{x}$ for all locations at constant depth $z = 2.0$ km of Marmousi model. 29

2.8 Escape time function $\widehat{T}$ for all locations at constant depth $z = 2.0$ km of Marmousi model. 29

2.9 Escape depth function $\widehat{z}$ for all locations at constant lateral position $x = 6.5$ km of Marmousi model. 30

2.10 Escape lateral position function $\widehat{x}$ for all locations at constant lateral position $x = 6.5$ km of Marmousi model. 30

2.11 Escape time function $\widehat{T}$ for all locations at constant lateral position $x = 6.5$ km of Marmousi model. 31

2.12 Smoothed SEG/EAGE Salt model velocity volume for escape value computations. 31

2.13 Escape depth function $\widehat{z}$ at point $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. 32

2.14 Escape position function $\widehat{x}$ at point $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. 32

2.15 Escape position function $\widehat{y}$ at point $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. 33

2.16 Escape time function $\widehat{T}$ at point $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. 33

3.1 Upwind principle demonstrated on the first-order finite-difference stencil: the white point is being computed, the grey points have known values and are located in the upwind direction, the stencil is aligned with the known points in the upwind direction determined by the vector field **a** (a scheme). 38

3.2 Boundary conditions for 3-D reduced phase space (a scheme). 39

3.3 Boundary conditions for 5-D reduced phase space (a scheme). 40

3.4 Escape depth function $\widehat{z}$ computed for Marmousi model (first-order finite-difference iterative solver). 41

3.5 Escape lateral position function $\widehat{x}$ computed for Marmousi model (first-order finite-difference iterative solver). 42

3.6 Escape time function $\widehat{T}$ computed for Marmousi model (first-order finite-difference iterative solver). 42

3.7 Escape depth function $\widehat{z}$ at constant depth $z = 2.0$ km of Marmousi model (first-order finite-difference iterative solver). 43

3.8 Escape lateral position function $\widehat{x}$ at constant depth $z = 2.0$ km of Marmousi model (first-order finite-difference iterative solver). 43

3.9 Escape time function $\widehat{T}$ at constant depth $z = 2.0$ km of Marmousi model (first-order finite-difference iterative solver). 44

3.10 Escape depth function $\widehat{z}$ computed for Marmousi model (second-order finite-difference iterative solver). 44

3.11 Escape lateral position function $\widehat{x}$ computed for Marmousi model (second-order finite-difference iterative solver). 45

3.12 Escape time function $\widehat{T}$ computed for Marmousi model (second-order finite-difference iterative solver). 45

3.13 Escape depth function $\widehat{z}$ at constant depth $z = 2.0$ km of Marmousi model (second-order finite-difference iterative solver). 46

3.14 Escape lateral position function $\widehat{x}$ at constant depth $z = 2.0$ km of Marmousi model (second-order finite-difference iterative solver). . . . 46

3.15 Escape time function $\widehat{T}$ at constant depth $z = 2.0$ km of Marmousi model (second-order finite-difference iterative solver). 47

3.16 Exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (first-order finite-difference iterative solver). 49

3.17 Exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (second-order finite-difference iterative solver). 49

3.18 Surface exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (first-order finite-difference iterative solver) plotted over wavefield computed for a source at the same point. 50

3.19 Surface exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (second-order finite-difference iterative solver) plotted over wavefield computed for a source at the same point. 50

3.20 Closeup of escape depth function $\widehat{z}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). 52

3.21 Closeup of escape lateral position function $\widehat{x}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). 52

3.22 Closeup of escape time function $\widehat{T}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). 53

3.23 Closeup of escape depth function $\widehat{z}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). 53

3.24 Closeup of escape lateral position function $\widehat{x}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). 54

3.25 Closeup of escape time function $\widehat{T}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). 54

3.26 Escape depth function $\widehat{z}$ at constant depth $z = 2.0$ km of Marmousi model (hybrid iterative solver). 56

3.27 Escape depth function $\widehat{x}$ at constant depth $z = 2.0$ km of Marmousi model (hybrid iterative solver). 56

3.28 Escape depth function $\widehat{T}$ at constant depth $z = 2.0$ km of Marmousi model (hybrid iterative solver). 57

3.29 Exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (hybrid iterative solver). 57

3.30 Surface exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (hybrid iterative solver) plotted over wavefield computed for a source at the same point. 58

3.31 Surface exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (ray tracing) plotted over wavefield computed for a source at the same point. 58

3.32 Map of ray-traced points (red) of the hybrid escape solution at constant depth $z = 2.0$ km of Marmousi model. 59

4.1 Salt body in SEG/EAGE Salt model. 63

4.2 Ray trajectories computed for the location $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. 64

4.3 Local escape traveltime $\widehat{T}$ computed for $d_{max} = 1.0$ km at the location $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. . 64

4.4 Local escape traveltime $\widehat{T}$ computed for $d_{max} = 0.75$ km at the location $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km. of SEG/EAGE Salt model. 65

4.5 Local escape traveltime $\widehat{T}$ computed for $d_{max} = 0.5$ km at the location $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km. of SEG/EAGE Salt model. 65

4.6 Change of phase $\mathbf{p}$ to $\widehat{\mathbf{p}}$ defined as a rotation by angle ψ around vector $\mathbf{l}$ (a scheme). 67

4.7 Change of position $\mathbf{x}$ and phase $\mathbf{p}$ according to local values of functions $\mathbf{x}_\Delta$ and $\mathbf{q}$ (a scheme). 67

4.8 Iterative reconstruction of escape quantities (a scheme). 69

4.9 5-D interpolation of the local escape values (a scheme). 72

4.10 Distribution of the local escape functions across computer cluster nodes (a scheme). 76

4.11 Surface exit locations for all arrivals originating from point $x = 6.7$ km, $y = 7.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model (ray tracing) plotted over wavefield computed for a source at the same point and extracted at receiver line of constant $y = 7.7$ km. 79

4.12 Surface exit locations for all arrivals originating from point $x = 6.7$ km, $y = 7.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model (iterative reconstruction) plotted over wavefield computed for a source at the same point and extracted at receiver line of constant $y = 7.7$ km. . . 79

5.1 2-D subsurface reflection system, relationship between phase vectors and scattering and dip angles (a scheme). 81

5.2 2-D angle-domain migration using escape tables (a scheme). 82

5.3 Reversal of exit positions on the surface for a ray tube in 2-D due to a caustic point (a scheme). 83

5.4 Definition of a muting zone in angle-domain for a fixed depth point (a scheme). 86
5.5 Marmousi model final image. 88
5.6 Marmousi model scattering (a) and dip (b) angle gathers for lateral position 2.0 km. 89
5.7 Marmousi model scattering (a) and dip (b) angle gathers for lateral position 4.0 km. 89
5.8 Marmousi model scattering (a) and dip (b) angle gathers for lateral position 6.0 km. 90
5.9 Marmousi model scattering (a) and dip (b) angle gathers for lateral position 8.0 km. 90
5.10 Sigsbee2B model velocity. 91
5.11 Sigsbee2B model scattering (a) and dip (b) angle gathers for lateral position 7.0 km. 91
5.12 Sigsbee2B model final image. 92
5.13 Sigsbee2B model scattering (a) and dip (b) angle gathers for lateral position 11.0 km. 93
5.14 Sigsbee2B model scattering (a) and dip (b) angle gathers for lateral position 15.0 km. 93
5.15 Sigsbee2B model scattering (a) and dip (b) angle gathers for lateral position 19.0 km. 94
5.16 Sigsbee2B model scattering (a) and dip (b) angle gathers for lateral position 23.0 km. 94
5.17 Hess VTI model vertical velocity. 95
5.18 Hess VTI model Thomsen anisotropy parameter ϵ. 95
5.19 Hess VTI model Thomsen anisotropy parameter δ. 96
5.20 Hess VTI model scattering (a) and dip (b) angle gathers for lateral position 7.0 km. 96
5.21 Hess VTI model final image. 97
5.22 Hess VTI model scattering (a) and dip (b) angle gathers for lateral position 11.5 km. 98
5.23 Hess VTI model scattering (a) and dip (b) angle gathers for lateral position 17.0 km. 98
5.24 BP TTI model symmetric axis velocity (a) and axis tilt angle (b). . . 99
5.25 BP TTI model Thomsen anisotropy parameters ϵ (a) and δ (b). . . . 100

5.26 BP TTI model final image. 101

5.27 BP TTI model scattering (a) and dip (b) angle gathers for lateral position 10.0 km. 102

5.28 BP TTI model scattering (a) and dip (b) angle gathers for lateral position 20.0 km. 102

5.29 BP TTI model scattering (a) and dip (b) angle gathers for lateral position 30.0 km. 103

5.30 BP TTI model scattering (a) and dip (b) angle gathers for lateral position 40.0 km. 103

5.31 BP TTI model scattering (a) and dip (b) angle gathers for lateral position 50.0 km. 104

5.32 BP TTI model scattering (a) and dip (b) angle gathers for lateral position 60.0 km. 104

5.33 BP TTI model scattering (a) and dip (b) angle gathers for lateral position 70.0 km. 105

6.1 3-D subsurface reflection system, relationship between phase vectors and scattering, dip angles (a scheme). 107

6.2 Exit triangle defined by a set of escape values in the escape tables (a scheme). 109

6.3 3-D angle-domain migration using escape tables (a scheme). 110

6.4 Escape triangle corresponding to the travelpath though a caustic of the first type (a scheme). 111

6.5 Escape triangle corresponding to the travelpath though a caustic of the second type (a scheme). 111

6.6 Local escape function sampling in space is 50 m, angle - 5°. $d_{max} = 0.25$ km. 117

6.7 Local escape function sampling in space is 50 m, angle - 5°. $d_{max} = 0.5$ km. 118

6.8 Local escape function sampling in space is 50 m, angle - 5°. $d_{max} = 0.75$ km. 119

6.9 Local escape function sampling in space is 50 m, angle - 10°. $d_{max} =$ 0.25 km. 120

6.10 Local escape function sampling in space is 50 m, angle - 10°. $d_{max} = 0.5$ km. 121

6.11 Local escape function sampling in space is 50 m, angle - 10°. $d_{max} =$ 0.75 km. 122

6.12 Local escape function sampling in space is 75 m, angle - 5°. $d_{max} = 0.25$ km. 123

6.13 Local escape function sampling in space is 75 m, angle - 5°. $d_{max} = 0.5$ km. 124

6.14 Local escape function sampling in space is 75 m, angle - 5°. $d_{max} = 0.75$ km. 125

6.15 Local escape function sampling in space is 75 m, angle - 10°. $d_{max} = 0.25$ km. 126

6.16 Local escape function sampling in space is 75 m, angle - 10°. $d_{max} = 0.5$ km. 127

6.17 Local escape function sampling in space is 75 m, angle - 10°. $d_{max} = 0.75$ km. 128

6.18 Local escape function sampling in space is 100 m, angle - 5°. $d_{max} = 0.25$ km. 129

6.19 Local escape function sampling in space is 100 m, angle - 5°. $d_{max} = 0.5$ km. 130

6.20 Local escape function sampling in space is 100 m, angle - 5°. $d_{max} = 0.75$ km. 131

6.21 Local escape function sampling in space is 100 m, angle - 10°. $d_{max} = 0.25$ km. 132

6.22 Local escape function sampling in space is 100 m, angle - 10°. $d_{max} = 0.5$ km. 133

6.23 Local escape function sampling in space is 100 m, angle - 10°. $d_{max} = 0.75$ km. 134

6.24 Target line at x=6.7 km imaged by multi-arrival Kirchhoff angle-domain migration using escape tables built by ray tracing. 135

6.25 Target line at x=6.7 km imaged by multi-arrival Kirchhoff angle-domain migration using escape tables built by iterative reconstruction in phase space. 136

6.26 Local similarity for the two target lines at x=6.7 km shown above. . . 137

6.27 Scattering angle gather at $x = 6.7km$ and $y = 7.7km$ (horizontal axis is the scattering angle, each panel represents a scattering azimuth angle sector specified by a diagram above it). Escape tables are produced by iterative reconstruction in phase space. 138

6.28 Dip angle gather at $x = 6.7km$ and $y = 7.7km$ (horizontal axis is the dip angle, each panel represents a dip azimuth angle sector specified by a diagram above it). Escape tables are produced by iterative reconstruction in phase space. 139

Chapter 1

Introduction

Background

The modern hydrocarbon exploration relies on a variety of geological and geophysical prospect tools. For explorationists concerned with searching for oil and gas deposits, economically efficient mapping of new reservoirs is impossible without reliable interwell information. Reflection seismology is one of the prospect tools that enables geophysicists to collect such data. It is probably the most efficient instrument for hydrocarbon exploration in terms of the ratio of price to obtained resolution of interwell structural information.

Reflection seismology produces structural images of the subsurface through a series of steps (Yilmaz, 2001). First, reflection seismic data are acquired on the surface of an area of prospect interest. At this stage, a burst of energy is sent downward repeatedly from different source positions. Structural heterogeneities, such as bedding surfaces (horizons) or planar fractures (faults), scatter part of the energy back toward the surface, where it is recorded at multiple receiver positions to create seismic traces grouped into records. Raw field seismic records are impossible to utilize directly for exploration analysis (interpretation) because the acquired data exist in the surface acquisition coordinates that are unrelated to the subsurface. To make these data useful for exploration purposes, they need to be mapped (imaged) back to the depth domain. This process is known as seismic migration. At the second stage of the seismic data analysis, field reflection data are processed and prepared for imaging. And, finally, the imaging (migration) stage is commenced.

Contemporary seismic imaging is, in fact, an iterative process. Proper migration of reflection time data to subsurface depth space is only possible if an accurate depth seismic velocity is known. The latter is related to rock properties, their dispositions, and, therefore, requires some a priori knowledge of the subsurface. Since it is usually quite limited, imaging has to be done iteratively to allow refinement of the velocity model at every step.

The necessity to migrate recorded data back to depth was recognized very early in the history of seismic exploration, even before the beginning of the digital era (Bednar, 2005). In the last several decades, an strong research effort has been focused on creating better and more advanced imaging techniques (Etgen et al., 2009). Rapid progress in the practice of migration has been enabled by exponential growth of available computer resources. Modern seismic imaging is a very compute intensive process. Many of the most powerful supercomputers in the world have been built over the years solely for addressing seismic imaging tasks. Despite stunning progress both in computer hardware and in migration research, there exists no "silver bullet" imaging algorithm that would always produce optimal results most economically in all possible geological settings. Some environments, such as those related to subsalt areas, are especially troublesome for imaging (Cogan et al., 2011).

Hydrocarbon accumulations have been known to be associated with salt depositions since before the creation of reflection seismology. Interestingly, early experiments with reflected waves were aimed at finding shallow salt domes onshore along the coast of Gulf of Mexico (Karcher, 1974). Today, salt basins still remain one of the frontiers and challenges of exploration endeavors; only it is offshore, in deep water areas, covered by thick overburden (Addison et al., 2010).

The geologic term “salt” usually means a type of rock consisting primarily of the mineral halite, which is chemically known as sodium chloride. Salt has unusual mechanical properties: it is less dense than most other sedimentary rocks and it is also relatively incompressible (Hudec and Jackson, 2007). The combination of these features causes salt to flow like a fluid on the geologic time scale, if a substantial differential loading is present and there are pre-existing weak points in the overburden. Under these conditions, the salt mass will rise toward the surface. As a result of this process, significant salt volumes can be displaced and form large canopies surrounded by younger sediments. In the Gulf of Mexico, for example, Jurassic salt is often found encased by Tertiary deposits. Geologically older salt in this environment is mechanically very different from the surrounding rocks. Most notably, the velocity of seismic wave propagation is much higher in salt. Such drastic changes in media properties around salt edges create a very strong barrier for traveling seismic energy. The problem is intensified by typically complex shapes of salt bodies so that the energy can get scattered in sophisticated patterns, often hitting different parts of the salt before returning to the surface. For sediments adjacent to the salt body or situated under it (subsalt area), this complicated geologic environment gives rise to a phenomenon called multipathing. This means that the same point in the subsurface can send energy back to the surface through multiple different paths in the media. For high-quality imaging around and under salt bodies, accurate migration of all scattered energy is needed (Fehler et al., 2002). Certain imaging approaches can naturally do that but it is not yet economical to use them in the iterative fashion required in exploration practice. Other, more computationally efficient techniques need to be specifically adapted for imaging with multipathing, yet there is no trivial solution to this problem that does not carry a significant computational penalty (Bednar, 2006).

Kirchhoff imaging and traveltime computations

The assortment of imaging procedures can be generally divided into two major groups: wave equation-based methods and integral schemes. From the second group, integral-operator Kirchhoff migration remains a staple in the toolbox of imaging practitioners, even in the most complex geologic settings (Leveille et al., 2011). The ability of Kirchhoff migration to utilize the full bandwidth of the recorded signal and to produce image gathers in a target-oriented fashion enables iterative imaging and velocity model building. This type of imaging also naturally handles different types of seismic anisotropy and allows for a computationally robust implementation.

The conventional Kirchhoff imaging operator (Schneider, 1978; Schleicher et al., 2007) can be written as

$$I(\mathbf{x}) = \iint W(\mathbf{x},\mathbf{s},\mathbf{r})\,\mathbf{D}_t u\Big[\mathbf{s}_s,\mathbf{s}_r,T(\mathbf{s}_s,\mathbf{x})+T(\mathbf{s}_r,\mathbf{x})\Big]\,d\mathbf{s}_s\,d\mathbf{s}_r\ , \tag{1.1}$$

where $\mathbf{x}$ is the subsurface (image) location, u is the wavefield recorded at the surface, $\mathbf{D}_t$ is the waveform correction operator, $\mathbf{s}_s$ and $\mathbf{s}_r$ are the shot and receiver positions on the surface, T is the traveltime from the surface to $\mathbf{x}$, and W is the amplitude weight.

Fundamentally, this type of imaging rests on a high-frequency approximation of wave propagation and a necessity to compute ray-theoretical Green's functions. The classic Kirchhoff kernel uses single-valued traveltimes computed from surface locations to subsurface image points. Various efficient algorithms have been developed over the years for traveltime computations, all of which generally can be divided into two groups: Eulerian and Lagrangian methods (Engquist and Runborg, 2003; Runborg, 2007).

Finite-difference eikonal solvers (Vidale, 1990; van Trier and Symes, 1991) belong to the first group. They enable fast simultaneous computation of minimum-time arrivals from a surface point to many subsurface locations on a model depth grid. Two of the most prominent methods for solving the eikonal equation are fast sweeping (Zhao, 2005; Fomel et al., 2009) and fast marching (Sethian and Popovici, 1999; Popovici and Sethian, 2002). The latter can be computationally more efficient than the former on strongly nonuniform problems (Gremaud and Kuster, 2006). In general, eikonal solvers provide a very robust way of calculating minimum-time arrivals. However, minimum-time travelmaps are not always sufficient for imaging difficult geologic areas (Geoltrain and Brac, 1993). In many cases, if only single arrivals are considered, shortest path or maximum-energy ones are better alternatives (Audebert et al., 1997). Ray tracing techniques allow computation of many arrivals and thus enable a choice between different arrivals.

Ray tracing algorithms belong to the group of Lagrangian methods - they describe computation of traveltimes in terms of reconstruction of individual ray trajectories. There exist a plethora of such tracing techniques (Farra, 1993; Červený, 2001) which can be broken into two categories: one-point and two-point ray tracing. The latter is concerned with finding a trajectory between two fixed start and end positions. The former is also commonly known as initial-value ray tracing, since it deals with computation of a ray for a given initial position and a phase direction. One-point ray tracing is relatively straightforward to implement compared to the two-point problem, but its output is very sensitive to initial conditions in the presence of strong velocity gradients. If the background velocity model contains significant variations, rays with close initial conditions tend to diverge greatly when they pass near strong velocity nonuniformities. In such cases, so-called "shadow zones" appear when initial conditions are sampled uniformly and rays are traced independently from each other. This problem can be addressed using wavefront construction (Vinje et al., 1993) or Huygens wavefront tracing (Sava and Fomel, 2001) approaches.

Eikonal solvers are computationally more efficient than ray tracing techniques, because they define traveltime calculations in such a way that new values are obtained with the help of previously estimated points. Ray tracers, however, are more flexible in a sense that they allow for computing different types of arrivals and can limit computations to a selected few locations. Although even with this ability to choose proper imaging contributions, single-arrival Kirchhoff migration often does not provide accurate enough images (Operto et al., 2000). It has been shown that, for complex geology and in the presence of traveltime multipathing, incorporation of multiple arrivals into Kirchhoff migration is crucial for high-quality imaging (Xu et al., 2001; Brandsberg-Dahl et al., 2003).

Angle-domain Kirchhoff migration and escape functions

A different approach to Kirchhoff migration is based on the generalized Radon transform (Beylkin, 1985), which introduces accurate weights related to reflectivity for true-amplitude migration (Miller et al., 1987). However, the original formulation for integration in the surface coordinate system does not take into account the possible development of multi-valued traveltimes and requires computation of the so-called Beylkin determinant (Bleistein, 1987). Both obstacles can be removed if the integration is performed in subsurface angular coordinates over source and receiver branches. This unravels multipathing and establishes an imaging domain in which surface data are mapped to subsurface points as a function of scattering and dip angles (Xu et al., 2001; Brandsberg-Dahl et al., 2003; Bleistein et al., 2005).

The Kirchhoff integral operator for multi-arrival angle-domain imaging can be written then as

$$I(\mathbf{x}) = \iint \widehat{W}(\mathbf{x}, \mathbf{p}_s, \mathbf{p}_r)\, \mathbf{D}_t u \Big[\widehat{\mathbf{y}}(\mathbf{x}, \mathbf{p}_s), \widehat{\mathbf{y}}(\mathbf{x}, \mathbf{p}_r), \widehat{T}(\mathbf{x}, \mathbf{p}_s) + \widehat{T}(\mathbf{x}, \mathbf{p}_r) \Big]\, d\mathbf{p}_s\, d\mathbf{p}_r \ , \quad (1.2)$$

where u is the wavefield recorded at the surface, $\mathbf{p}_s$ and $\mathbf{p}_r$ are the shot and receiver phase slowness vectors respectively for a pair of rays originating from $\mathbf{x}$, $\widehat{T}$ is the ray exit (escape) traveltime at the surface exit (escape) position $\widehat{\mathbf{y}}$, $\mathbf{D}_t$ is the waveform-correction time derivative operator, and $\widehat{W}$ is the amplitude weight.

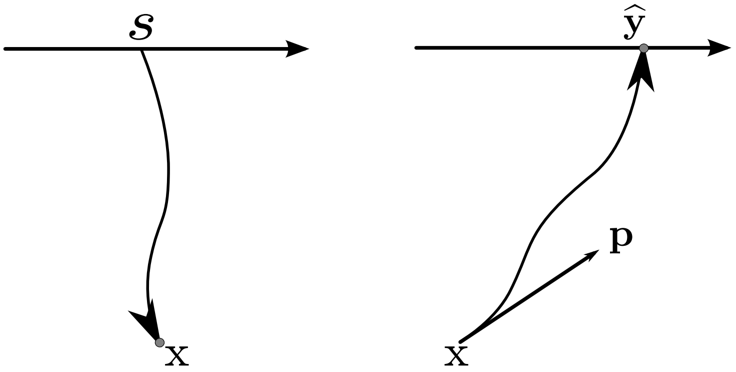


Figure 1.1: Parameterization of Kirchhoff integral imaging operator (a scheme): by surface coordinate (left), by subsurface phase angle (right). chapter-intro/. adom

The integral operator in equation (1.2) presents a bottom-up imaging approach, in which every combination of a subsurface location and a subsurface slowness vector is mapped uniquely to a pair of a surface point (exit location) and an exit ray parameter (Xu et al., 2001). Multiple arrivals are naturally unraveled when Kirchhoff integration is organized in the form of equation (1.2). If we limit contributions to $I(\mathbf{x})$ only to specific fixed directions $\mathbf{p}_s - \mathbf{p}_r$ or $\mathbf{p}_s + \mathbf{p}_r$, then we obtain scattering angle or dip angle gathers, respectively (Koren and Ravve, 2011). The first subset of gathers is an important measure of image consistency; the latter one is a convenient domain for separating specular and diffraction components of the image (Landa et al., 2008; Klokov and Fomel, 2012). Scattering angle gathers also provide a convenient image representation for AVA (amplitude versus angle) analysis.

A combination of the escape traveltime $\widehat{T}(\mathbf{x},\mathbf{p})$ and the escape position $\widehat{\mathbf{y}}(\mathbf{x},\mathbf{p})$ represents an escape function defined in the 6-D phase space $\{\mathbf{x},\mathbf{p}\}$. To implement the summation in equation (1.2), one has to compute a discrete version of the escape function, escape tables, in the reduced 5-D space with the initial directions defined by the inclination angle θ and the azimuth angle ϕ of the phase vectors $\mathbf{p}$. A conventional way of obtaining these tables would be to perform ray tracing for all spatial locations in multiple directions that span the reduced phase space. Each individual set of $\widehat{T}$ and $\widehat{\mathbf{y}}$ values of the escape tables is then produced by one traced ray. This approach, however, poses a significant practical challenge, because it adds an extra dimension to the computations compared to the classical surface-oriented formulation. This might render this approach too computationally expensive.

Escape functions in phase space

Ray trajectories represent characteristic lines in the phase space. Their evolution can be described by the following system of ray-tracing equations (Červený, 2001)

$$\dot{\mathbf{x}} = \mathbf{p} - S\left(\mathbf{x},\mathbf{p}\right)\,\nabla_{\mathbf{p}} S\ , \tag{1.3}$$

$$\dot{\mathbf{p}} = S\left(\mathbf{x},\mathbf{p}\right)\,\nabla_{\mathbf{x}} S\ , \tag{1.4}$$

$$\dot{T} = S^2\left(\mathbf{x},\mathbf{p}\right)\ , \tag{1.5}$$

where $\dot{\mathbf{f}}$ denotes $\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\sigma}$, σ is a parameter changing along the ray, and $S(\mathbf{x}, \mathbf{p})$ is the phase slowness. In the isotropic case, $S(\mathbf{x}, \mathbf{p})$ does not depend on $\mathbf{p}$.

A conventional way of solving the system (1.3-1.5) for a gridded medium is to perform numerical integration (Runborg, 2007). Background velocity is usually smoothed (Alde et al., 2002) and represented in terms of spline coefficients (Virieux and Farra, 1991). Explicit interfaces might be used along with the smoothed background, although, interfaceless fully gridded representation of the background medium is common in seismic imaging. The computational cost $\mathcal{O}_{\mathbf{Conv}}$ of obtaining the full escape table for such a model is then proportional to the following product

$$\mathcal{O}_{\mathbf{Conv}} \sim \mathcal{O}\left(N_x N_y N_z N_\theta N_\phi\right) \cdot \mathcal{O}_{\mathbf{RT}}\,, \tag{1.6}$$

where $N_x N_y N_z$ is the number of points in the spatial imaging grid, $N_\theta N_\phi$ is the number of directions needed at each point to provide sufficient angle coverage, and $\mathcal{O}_{\mathbf{RT}}$ is the computational cost of tracing one ray trajectory, which, in turn, can be expressed as

$$\mathcal{O}_{\mathbf{RT}} \sim \mathcal{O}\left(\sqrt[3]{N_x^f N_y^f N_z^f}\right) \cdot \mathcal{O}_{\mathbf{ODE}}\,, \tag{1.7}$$

where $N_x^f N_y^f N_z^f$ is the number of points on a spatial grid fine enough for accurate ray tracing, and $\mathcal{O}_{\mathbf{ODE}}$ is the computational cost of ODE integration.

Ray tracing estimates escape values at end points of ray trajectories by reconstructing them independently with small integration steps in the numerical solution of the system (1.3-1.5). This is a computationally expensive procedure for building full 5-D escape tables due to the $\mathcal{O}_{\mathbf{RT}}$ cost. A different way of describing escape functions needs to be used to negate the effect of this term in equation (1.6) and bring down the total computational cost.

It is still possible to trace rays from surface locations and then reconstruct escape functions in the subsurface coordinate system. Because individual traced rays never go exactly through nodes of the subsurface grid, interpolation between different ray branches has to be performed. In case of multipathing, a difficult problem arises - how to choose two rays which belong to the same ray branch and how to prevent incorrect interpolation between unrelated rays? Apparently, this problem does not have an elegant solution (Xu and Lambaré, 2004). Different criteria might be utilized to pair correct rays, but none of them exactly guarantee proper coupling.

Another method for obtaining multiple arrivals employs robustness and stability of eikonal solvers. In the method of slowness matching (Symes and Qian, 2003), the background model is partitioned into small sectors, in every one of which a single arrival solutions is obtained. Multiple arrivals are then reconstructed from these local solutions. At the edges of sectors, the slowness vectors are matched on both sides, so that the phase vector exactly at the boundary is continuous. However, since the solution is not known in advance, it is difficult to split the media in such a way so that important caustic points are not lost in individual local eikonal solutions.

Yet, another approach relies on properties of the high-frequency wave propagation in phase space. In the presence of velocity anomalies, wavefronts fold and develop caustic points. However, when described in the higher dimensional phase space, they represent smooth curves (Lambaré et al., 1996). Wavefront propagation can be computed with Eulerian methods in phase space directly (Benamou, 1999; Osher et al., 2002). Much like Eikonal solvers, these methods rely on using previously computed values to find new ones, thus allowing to achieve greater computational robustness compared to the Lagrangian approach of ray tracing.

Wavefront propagation and construction is just a means of achieving the ultimate goal, that is, obtaining the escape tables for the angle-domain imaging. Intermediate positions of the wavefronts are not important, only escape functions $\widehat{T}(\mathbf{x}, \mathbf{p})$ and $\widehat{\mathbf{y}}(\mathbf{x}, \mathbf{p})$ are. Therefore, a more robust approach to finding the escape tables is to compute escape functions directly in the phase space. Fomel and Sethian (2002) showed that the distribution of escape quantities in the subsurface can be described by a set of escape equations in the phase space, which spans all possible locations and directions in the subsurface.

Escape functions in the phase space are continuous, provided that the background medium is smooth. Figure 1.2 demonstrates ray trajectories computed for a source point under the low-velocity anomaly, which causes ray multipathing on the way to the surface. In this case, one surface location may have more than one ray exiting at it. If, however, the escape position $\widehat{x}$ of each ray is plotted as a function of its initial phase (or its take-off angle θ), then it becomes a continuous single-valued function (Figure 1.3). Other escape functions, such as traveltime $\widehat{T}$, have the same property (Figure 1.4). The property of smoothness and continuity of escape functions in the phase space makes it a convenient domain for computing escape tables. These functions provide a natural way of mapping all arrivals in surface reflection seismic data to the geologically meaningful subsurface scattering-dip angle system.

Problem statement and outline

Incorporation of multipathing into Kirchhoff migration has been shown to greatly improve quality of seismic images in difficult geologic areas (Koren et al., 2002; Alde et al., 2003). The classic integral in the surface coordinates does not provide a convenient way of introducing multiple arrivals into migration summation.

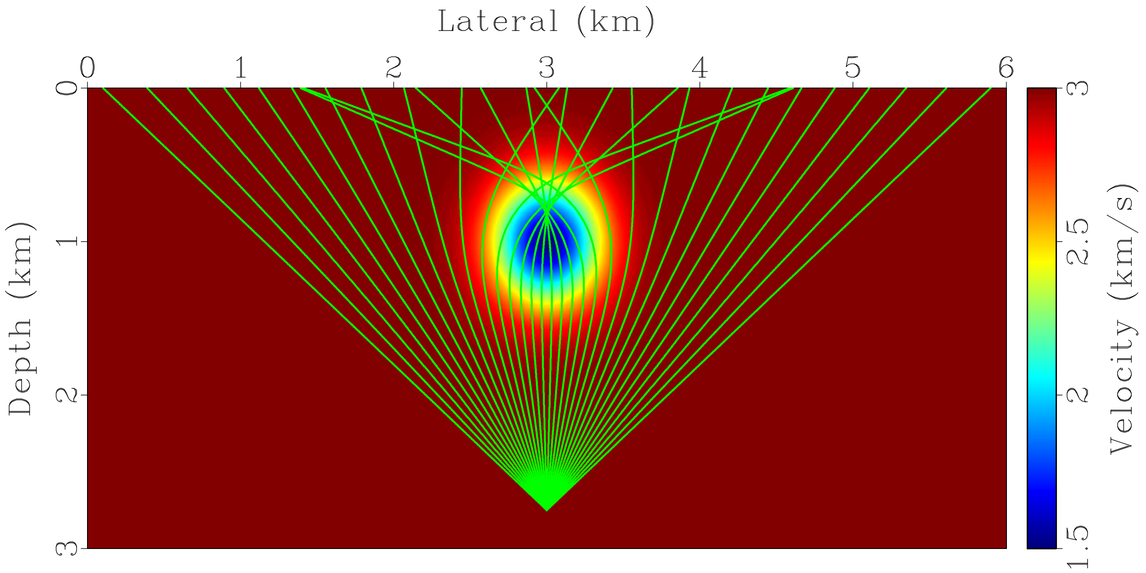


Figure 1.2: Ray trajectories computed for a source point at $x = 3.0$ km and $z = 2.75$ km in the low-velocity anomaly model. chapter-intro/cloud cloud2trajs

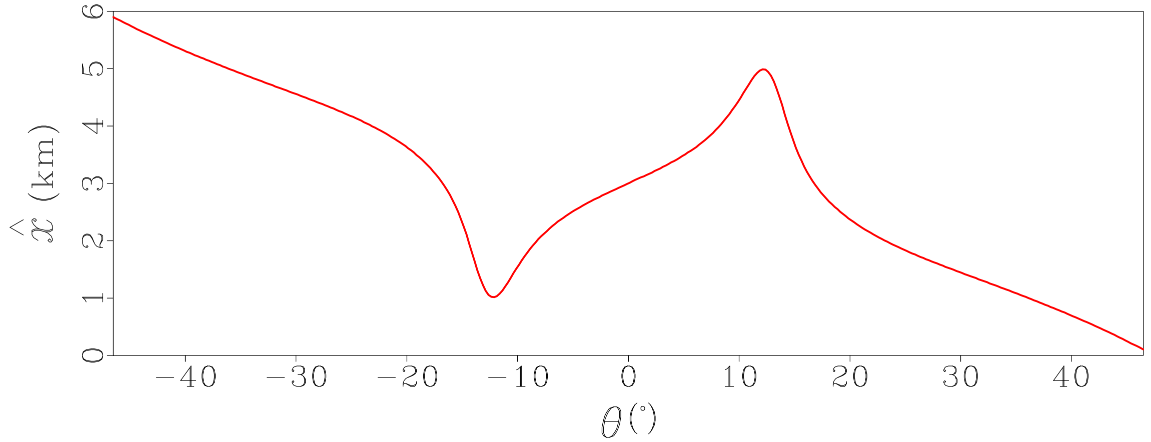


Figure 1.3: Exit (escape) position $\widehat{x}$ as a function of the take-off angle θ for ray trajectories computed for a source point at $x = 3.0$ km and $z = 2.75$ km in the low-velocity anomaly model. chapter-intro/cloud cloud2esc1

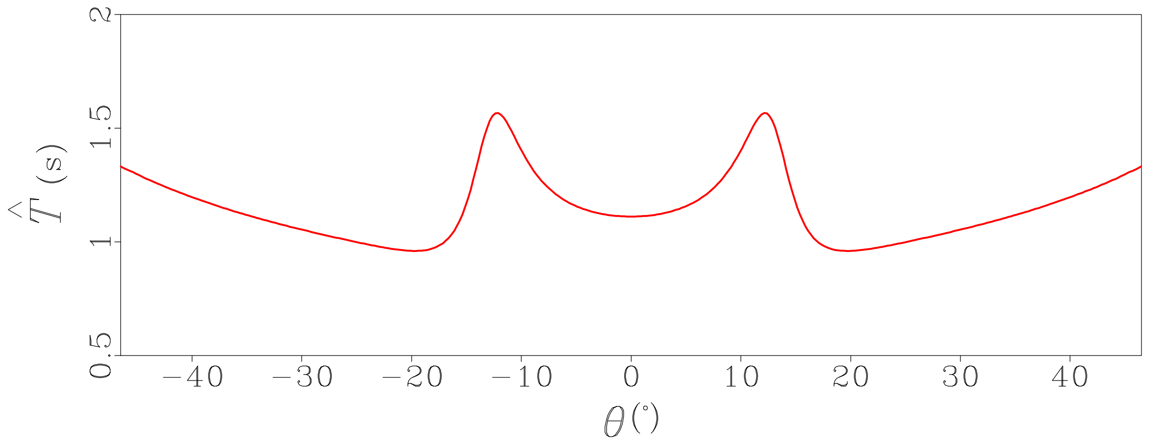


Figure 1.4: Exit (escape) traveltime $\widehat{T}$ as a function of the take-off angle θ for ray trajectories computed for a source point at $x = 3.0$ km and $z = 2.75$ km in the low-velocity anomaly model. $\boxed{\text{chapter-intro/cloud cloud2esc2}}$

Angle-domain definition of the integration naturally incorporates all arrivals, but poses a significant computational challenge due to necessity to compute these arrivals in the "bottom-up" fashion. If ray tracing is used for this purpose, then the computational overhead is such that this type of imaging is not considered as robust as the conventional single-arrival Kirchhoff migration. Due to this obstacle, multi-arrival angle-domain Kirchhoff migration has not been widely implemented and used in practice.

Angle-domain Kirchhoff integral maps surface reflection seismic data into subsurface image positions and subsurface phase directions. When combined, these two subsystems define phase space - a combination of all possible positions and directions. For a given velocity model, this mapping of reflection data to phase space is described by a set of escape functions. These functions can be computed by means of many high-frequency wave propagation methods, but escape equations provide, potentially, the most direct route to them. The equations describe how escape functions are distributed in the phase space. Thus, they allow us to construct an Eulerian-like algorithm for computing the required escape tables.

The objective of this work is to develop a computationally efficient framework for imaging surface reflection seismic data in the phase space and, consequently, in the angle domain. In Chapter 2, I provide a theoretical background for escape equations and derive their counterparts in reduced phase space, a more convenient domain for practical computation than full phase space. I provide a way of handling an important case of seismic anisotropy in escape equation computations as well. The last part of that chapter is dedicated to analysis of behavior of escape functions in reduced phase space for complex 2-D and 3-D velocity models.

In Chapter 3, I describe the process of building a finite-difference method for direct computation of 2-D escape equations. I analyze results obtained for the 2-D model used in the previous chapter. I also demonstrate accuracy problems for the pure finite-difference method and propose a modification to it, which produces a hybrid Eulerian-Lagrangian approach. I discuss computational performance constraints of the resultant algorithm and obstacles for scaling it from 2-D to 3-D problems. In Chapter 4, I design a different, semi-Lagrangian algorithm for 3-D problems. The algorithm relies on reconstruction of a global escape solution by means of iterative stepping through local escape functions and their interpolation in reduced phase space. I analyze performance properties of this approach and propose a version of it for parallel computer architectures, which enables computations for larger 3-D velocity models.

While angle-domain migration itself is not the subject of this research, it is important to investigate if escape tables produced by the algorithms from Chapters

3 and 4 can be utilized directly in imaging summation without any additional post-processing. It is also imperative to verify that migration results for these algorithms, when they are applied to complex background media, are accurate. As a frame of reference, I employ imaging results obtained through the same angle-domain migration code but using raytraced escape tables as input instead. I draw conclusions about performance improvements once optimal parameters for my phase space approach are established. The optimal set of parameters implies that imaging output for this method has comparable visual quality to that of the conventional one based on ray tracing.

In Chapter 5, I develop a method to construct 2-D angle-domain integration around escape tables. I cover implementation of such critical items for practical Kirchhoff imaging as antialiasing, phase shifts related to caustic points, noise suppression in output image, and handling of input reflection data irregularities. I demonstrate imaging results for a number of 2-D models, including two anisotropic ones. Escape tables for these tests are obtained using the algorithm designed in Chapter 3. Along with stacked images, I show scattering-angle and dip-angle gathers.

In Chapter 6, I provide some important details of the implementation of 3-D angle-domain integration. I use escape tables computed using the algorithm from Chapter 4 to obtain images and angle gathers for a challenging synthetic model based on salt tectonics. I show a number of angle gathers produced for different combinations of the algorithm parameters in order to illustrate how to choose them optimally. I conclude this chapter by comparing a target line image obtained using the new algorithm against a result based on ray tracing.

I summarize my findings in Chapter 7 and suggest possible improvements to the proposed numerical scheme for computing escape tables. I also show how this new approach is related to some other previously developed techniques.

Chapter 2

Escape equations and reduced phase space

Escape equations

In the previous chapter, I introduced escape functions $\widehat{T}(\mathbf{x},\mathbf{p})$ and $\widehat{\mathbf{y}}(\mathbf{x},\mathbf{p})$ defined in the phase space. These functions span all possible locations and directions in the subsurface and describe escape traveltime and escape positions for them. For 3-D media ($n=3$), the vector of escape position $\widehat{\mathbf{y}}$ contains three components $\{\widehat{x},\widehat{y},\widehat{z}\}$. It has only two components $\{\widehat{x},\widehat{z}\}$ for 2-D media ($n=2$).

For every point on the ray, the escape location $\widehat{\mathbf{y}}$ remains constant, therefore, the derivative of this location with respect to the evolution variable σ defined in equations (1.3)-(1.5) can be written as

$$\dot{\widehat{\mathbf{y}}} = 0\,. \tag{2.1}$$

By using the chain rule, we then can expand it into the following form

$$\begin{gathered}\dot{\widehat{\mathbf{y}}} = \nabla_{\mathbf{x}}\,\widehat{\mathbf{y}}\,\dot{\mathbf{x}} + \nabla_{\mathbf{p}}\,\widehat{\mathbf{y}}\,\dot{\mathbf{p}} = \\ \nabla_{\mathbf{x}}\,\widehat{\mathbf{y}}\;(\mathbf{p} - S\,\nabla_{\mathbf{p}}S) + \nabla_{\mathbf{p}}\,\widehat{\mathbf{y}}\;(S\,\nabla_{\mathbf{x}}\,S) = 0\;.\end{gathered} \tag{2.2}$$

Escape traveltime decreases along the ray toward the boundary, therefore, we can derive a similar expression for $\widehat{T}$:

$$\begin{gathered}\dot{\widehat{T}} = \nabla_{\mathbf{x}}\,\widehat{T}\,\cdot\,\dot{\mathbf{x}} + \nabla_{\mathbf{p}}\,\widehat{T}\,\cdot\,\dot{\mathbf{p}} = \\ \nabla_{\mathbf{x}}\,\widehat{T}\,\cdot\,(\mathbf{p} - S\,\nabla_{\mathbf{p}}S) + \nabla_{\mathbf{p}}\,\widehat{T}\,\cdot\,S\,\nabla_{\mathbf{x}}\,S = -S^2\;.\end{gathered} \tag{2.3}$$

Equations (2.2) and (2.3) are known as escape equations (Fomel and Sethian, 2002; Fomel, 2003). They have the form of a general advection system

$$\mathbf{a}\left(\mathbf{x},\mathbf{p}\right)\cdot\nabla_{\mathbf{x},\mathbf{p}}\,\widehat{x}=b\;, \tag{2.4}$$

where $\mathbf{a}\left(\mathbf{x},\theta,\phi\right)$ is the vector field defining the characteristics (rays) in phase space, $\widehat{x}$ is an escape quantity, and b is the source term, which is nonzero for quantities changing along the ray (e.g., traveltime).

Escape equations are, therefore, steady-state advection-type partial differential equations, which describe static distribution of escape time and locations for all arrivals originating from $\mathbf{x}$ with initial phase vector $\mathbf{p}$. The vector field

$$\mathbf{a}\left(\mathbf{x},\mathbf{p}\right)=\begin{pmatrix}\mathbf{p}-S\,\nabla_{\mathbf{p}}\,S\\ S\,\nabla_{\mathbf{x}}\,S\end{pmatrix} \tag{2.5}$$

has the following property

$$\nabla_{\mathbf{x},\mathbf{p}}\cdot\mathbf{a}\left(\mathbf{x},\mathbf{p}\right)=0\,. \tag{2.6}$$

which simply states the well-known fact that phase space is incompressible (Thomson and Chapman, 1985). This property means that characteristic lines do not split or merge, and, therefore, the escape quantities are continuous functions in the phase space.

For computational purposes, it is beneficial to derive similar equations in the reduced phase space, in which phase dimensions are replaced with angles associated with the phase vector direction (Osher et al., 2002). Reduced phase space has fewer dimensions, because n components of phase slowness vector $\mathbf{p}$ are related by $(n-1)$ angles.

For a 2-D medium, we define the slowness vector as

$$\mathbf{p}\left(p_x,p_z\right)=\begin{pmatrix}-S\,\sin\theta\\ -S\,\cos\theta\end{pmatrix}\;, \tag{2.7}$$

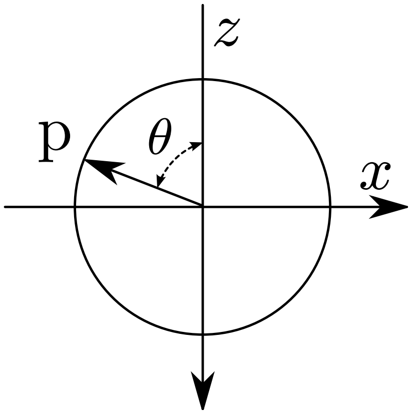


Figure 2.1: Slowness vector **p** in 2-D (a scheme). chapter-esceq/. pvec2

where θ is the angle between the phase vector and the vertical direction (Figure 2.1). Then, by using the relations

$$\begin{aligned} \frac{\partial}{\partial p_x} &= \frac{\partial \left(\tan^{-1}\frac{p_x}{p_z}\right)}{\partial p_x}\frac{\partial}{\partial \theta} = -\frac{p_z}{p_x^2+p_z^2}\frac{\partial}{\partial \theta}\,, \\ \frac{\partial}{\partial p_z} &= \frac{\partial \left(\cot^{-1}\frac{p_z}{p_x}\right)}{\partial p_z}\frac{\partial}{\partial \theta} = \frac{p_x}{p_x^2+p_z^2}\frac{\partial}{\partial \theta}\,, \end{aligned} \tag{2.8}$$

we can derive from equation (2.3) the following equation for escape traveltime in reduced phase space

$$(S\,\sin\theta - S_\theta\,\cos\theta)\;\frac{\partial \widehat{T}}{\partial x} + (S\,\cos\theta + S_\theta\,\sin\theta)\;\frac{\partial \widehat{T}}{\partial z} + (S_x\,\cos\theta - S_z\,\sin\theta)\;\frac{\partial \widehat{T}}{\partial \theta} = S^2 \tag{2.9}$$

where S_θ, S_x, and S_z are angular and spatial derivatives of the phase slowness field, respectively. This new domain is periodic in θ.

Similarly, we can obtain equations for escape location $\widehat{x}$ and escape depth $\widehat{z}$

$$\begin{aligned} (S\,\sin\theta - S_\theta\,\cos\theta)\;\frac{\partial \widehat{x}}{\partial x} + (S\,\cos\theta + S_\theta\,\sin\theta)\;\frac{\partial \widehat{x}}{\partial z} + (S_x\,\cos\theta - S_z\,\sin\theta)\;\frac{\partial \widehat{x}}{\partial \theta} = 0\,, \\ (S\,\sin\theta - S_\theta\,\cos\theta)\;\frac{\partial \widehat{z}}{\partial x} + (S\,\cos\theta + S_\theta\,\sin\theta)\;\frac{\partial \widehat{z}}{\partial z} + (S_x\,\cos\theta - S_z\,\sin\theta)\;\frac{\partial \widehat{z}}{\partial \theta} = 0\,, \end{aligned} \tag{2.10}$$

These equations are decoupled and can be computed independently from each other. Analogous equations can be formulated for other escape quantities, e.g. escape angle $\widehat{\theta}$ or escape slowness vector $\widehat{\mathbf{p}}$.

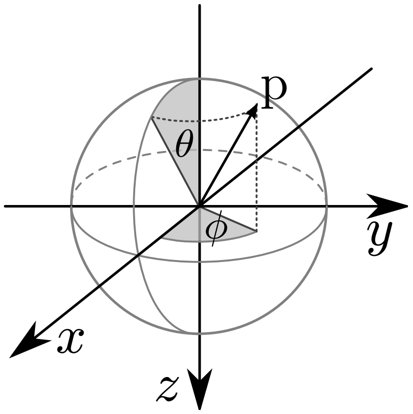


Figure 2.2: Slowness vector **p** in 3-D (a scheme). chapter-esceq/. pvec3

In a 3-D medium, the phase-vector direction is defined by two angles (Figure 2.2): θ, the angle between $\mathbf{p}$ and the z axis (inclination), and ϕ, the angle between the projection of p onto the $x-y$ plane and the x axis (azimuth). The slowness vector is then

$$\mathbf{p}\left(p_x, p_y, p_z\right) = \begin{pmatrix} S\,\sin\theta\cos\phi \\ S\,\sin\theta\sin\phi \\ -S\,\cos\theta \end{pmatrix} \;, \tag{2.11}$$

and, by changing variables from $\mathbf{p}$ to ϕ, θ and using the following relations

$$\begin{aligned} \frac{\partial}{\partial p_x} &= \frac{\partial\left(\tan^{-1}\frac{p_y}{p_x}\right)}{\partial p_x}\frac{\partial}{\partial\phi} = -\frac{p_y}{p_x^2+p_y^2}\frac{\partial}{\partial\phi} = -\frac{\cos\phi}{S\,\sin\theta}\frac{\partial}{\partial\phi}\;, \\ \frac{\partial}{\partial p_y} &= \frac{\partial\left(\tan^{-1}\frac{p_y}{p_x}\right)}{\partial p_y}\frac{\partial}{\partial\phi} = \frac{p_x}{p_x^2+p_y^2}\frac{\partial}{\partial\phi} = \frac{\sin\phi}{S\,\sin\theta}\frac{\partial}{\partial\phi}\;, \\ \frac{\partial}{\partial p_z} &= \frac{\partial\left(\tan^{-1}\left(-\frac{\sqrt{p_x^2+p_y^2}}{p_z}\right)\right)}{\partial p_z}\frac{\partial}{\partial\theta} = \frac{\sqrt{p_x^2+p_y^2}}{p_x^2+p_y^2+p_z^2}\frac{\partial}{\partial\theta} = \frac{\cos\theta}{S}\frac{\partial}{\partial\theta}\;, \end{aligned} \tag{2.12}$$

we can derive the corresponding reduced phase space equation in 3-D as

$$\begin{aligned}
&\left(-S\,\sin\theta\cos\phi+S_\theta\,\cos\theta\cos\phi-S_\phi\,\frac{\sin\phi}{\sin\theta}\right)\frac{\partial\widehat{T}}{\partial x}+\\
&\left(-S\,\sin\theta\sin\phi+S_\theta\,\cos\theta\sin\phi+S_\phi\,\frac{\cos\phi}{\sin\theta}\right)\frac{\partial\widehat{T}}{\partial y}+\\
&\left(S\,\cos\theta+S_\theta\,\sin\theta\right)\frac{\partial\widehat{T}}{\partial z}+\\
&\left(-S_x\,\cos\theta\cos\phi\;-S_y\,\cos\theta\sin\phi-S_z\,\sin\theta\right)\frac{\partial\widehat{T}}{\partial\theta}+\\
&\frac{1}{\sin\theta}\left(-S_y\,\cos\phi\;+S_x\,\sin\phi\right)\frac{\partial\widehat{T}}{\partial\phi}=S^2
\end{aligned} \tag{2.13}$$

where S_θ, S_ϕ, S_x, S_y and S_z are angular and spatial derivatives of phase slowness. In the isotropic case, S does not depend on θ or ϕ, and $S_\theta = S_\phi = 0$. Dimensions θ and ϕ are periodic. Analogously to the 2-D case, equations for the components of escape location have the same left-hand side but a different right-hand side, which is equal to zero as well.

These escape equations describe escape functions in 3-D or 5-D reduced phase space for 2-D or 3-D media respectively. They can also be expressed as a general advection system

$$\begin{aligned}
\mathbf{a}_{2-D}\left(\mathbf{x},\theta\right)\cdot\nabla_{\mathbf{x},\theta}\,\widehat{x}&=b\;,\\
\mathbf{a}_{3-D}\left(\mathbf{x},\theta,\phi\right)\cdot\nabla_{\mathbf{x},\theta,\phi}\,\widehat{x}&=b\;,
\end{aligned} \tag{2.14}$$

where the vector field of characteristic lines in the 2-D case is

$$\mathbf{a}_{2-D}\left(x,z,\theta\right)=\begin{pmatrix} S\,\sin\theta-S_\theta\,\cos\theta\\ S\,\cos\theta+S_\theta\,\sin\theta\\ S_x\,\cos\theta-S_z\,\sin\theta\end{pmatrix}\;, \tag{2.15}$$

and the vector field of characteristic lines in the 3-D case is

$$\mathbf{a}_{3-D}\left(x,y,z,\theta,\phi\right)=\begin{pmatrix} -S\,\sin\theta\cos\phi+S_\theta\,\cos\theta\cos\phi-S_\phi\,\frac{\sin\phi}{\sin\theta}\\ -S\,\sin\theta\sin\phi+S_\theta\,\cos\theta\sin\phi+S_\phi\,\frac{\cos\phi}{\sin\theta}\\ S\,\cos\theta+S_\theta\,\sin\theta\\ -S_x\,\cos\theta\cos\phi\;-S_y\,\cos\theta\sin\phi-S_z\,\sin\theta\\ \frac{1}{\sin\theta}\left(-S_y\,\cos\phi\;+S_x\,\sin\phi\right)\end{pmatrix}\;, \tag{2.16}$$

It is worth noting that $\mathbf{a}_{2-D}\left(\mathbf{x}, \theta, \right)$ has the same property as the vector field of characteristic curves in the full phase space, that is

$$\nabla_{\mathbf{x},\theta} \cdot \mathbf{a}_{2-D}\left(\mathbf{x}, \theta\right) = 0\,. \tag{2.17}$$

However, it is easy to show that, for the 3-D case,

$$\nabla_{\mathbf{x},\theta,\phi} \cdot \mathbf{a}_{3-D}\left(\mathbf{x}, \theta, \phi\right) \neq 0\,. \tag{2.18}$$

This inequality is an artifact of the chosen mathematical parameterization of characteristic curve distribution. If such a distribution is sampled evenly in the angular space $\{\theta, \phi\}$, then it looks non-uniform when viewed in the spatial domain $\{x, y, z\}$. Physically, only one line is present at the poles ($\theta = 0$ or $\theta = \pi$), while the description assumes multiple lines. These points of indeterminacy manifest themselves in the form of singular coefficients in equation (2.13). This problem is a well-known artifact of the spherical coordinate system, which often appears in other computational problems (Lapilli and Fowler, 2013). A stable numerical scheme for solving the escape equations in 5-D reduced phase space should either avoid the points of indeterminacy by shifting the angular grid or by introducing a special treatment to them (Mohseni and Colonius, 2000). Escape functions in this domain still remain continuous.

Equation coefficients for anisotropic media

Wave propagation in sedimentary rocks is frequently anisotropic. Lithification of sediments in layers and post-sedimentation processes, such as folding, often create a type of anisotropy known as Tilted Transverse Isotropy (TTI). Modern imaging practice recognizes that effects of anisotropy should be accounted for and commonly incorporates TTI model parameters into imaging workflows.

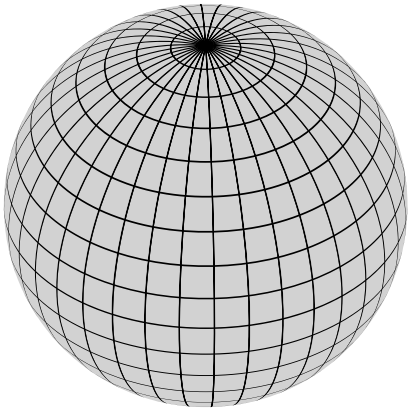


Figure 2.3: Consolidation of characteristic lines in polar areas of the spherical coordinate system (a scheme). chapter-esceq/. pred3sing

In the isotropic case, the phase slowness function has the same number of dimensions as the spatial domain. For all practical purposes, such a function can usually be stored in computer memory in its entirety. In case of anisotropic wave propagation and imaging, the dimensionality of phase slowness becomes equal to that of the corresponding reduced phase space. It is impractical to precompute a 5-D phase slowness function for a typical 3-D model. A common strategy is then to calculate phase values on the fly from the anisotropic model components as wave propagation computation proceeds.

Phase slowness function has an exact 4-parameter expression in TTI media and can be approximated with the following practical 3-parameter expression (Alkhalifah, 2000; Fomel, 2004)

$$\begin{aligned} v^2\left(\mathbf{x},\mathbf{p}\right) =& \frac{1}{2}\left(v_{\mathrm{v}}^2\,\cos^2\xi + v_{\mathrm{h}}^2\,\sin^2\xi\right) + \\ & \frac{1}{2}\sqrt{\left(v_{\mathrm{v}}^2\,\cos^2\xi + v_{\mathrm{h}}^2\,\sin^2\xi\right)^2 - \frac{8\eta}{1+2\eta}\left(v_{\mathrm{v}}\,v_{\mathrm{h}}\,\cos\xi\,\sin\xi\right)^2}\,, \end{aligned} \tag{2.19}$$

where v_{v} is the P-wave phase velocity along the symmetry axis, v_{h} is the P-wave phase velocity in the direction normal to the symmetry axis, η is the anellasticity parameter

(Alkhalifah and Tsvankin, 1995) defined through Thomsen's elastic parameters ϵ and δ (Thomsen, 1986) as

$$\eta = \frac{\epsilon - \delta}{1 + 2\delta} \,, \tag{2.20}$$

and ξ is the angle between the phase direction and the axis of symmetry, so that

$$\begin{aligned} \cos\xi &= \cos\theta \cos\theta_t + \sin\theta \cos\phi \sin\theta_t \cos\phi_t + \sin\theta \sin\phi \sin\theta_t \sin\phi_t \\ &= \cos\theta \cos\theta_t + \sin\theta \sin\theta_t \cos\left(\phi - \phi_t\right) \,, \end{aligned} \tag{2.21}$$

where θ_t and ϕ_t are the inclination and the azimuth of the symmetry axis respectively.

If equations (2.14) are used as a basis of escape tables computation, then the derivatives S_θ, S_ϕ, S_x, S_y and S_z have to be calculated to estimate the equation coefficients. These derivatives along with the phase slowness can be found from the given spatial functions $v_{\mathrm{v}}\left(\mathbf{x}\right)$, $v_{\mathrm{h}}\left(\mathbf{x}\right)$, $\eta\left(\mathbf{x}\right)$, $\theta_t\left(\mathbf{x}\right)$, and $\phi_t\left(\mathbf{x}\right)$.

Equation (2.19) can be written as

$$v^2\left(\mathbf{x}, \mathbf{p}\right) = \frac{1}{2}\left(\mathrm{A} + \mathrm{B} + \mathrm{C}\right) \,, \tag{2.22}$$

where

$$\begin{aligned} \mathrm{A} &= \mathrm{E}\,\mathrm{D} \,, \\ \mathrm{B} &= \mathrm{F}\left(1 - \mathrm{D}\right) \,, \\ \mathrm{C} &= \sqrt{\left(\mathrm{A} + \mathrm{B}\right)^2 + q\,\mathrm{D}\left(1 - \mathrm{D}\right)} \,, \\ \mathrm{D} &= \cos^2\xi \,, \\ \mathrm{E} &= v_{\mathrm{v}}^2 \,, \\ \mathrm{F} &= v_{\mathrm{h}}^2 \,, \\ q &= -\frac{8\eta}{1 + 2\eta} v_{\mathrm{v}}^2\, v_{\mathrm{h}}^2 \,. \end{aligned} \tag{2.23}$$

A derivative of v^2 with respect to some quantity u is given by the following sum

$$\frac{\partial v^2(\mathbf{x},\mathbf{p})}{\partial u} = \frac{1}{2}\left(\frac{\partial \mathrm{A}}{\partial u} + \frac{\partial \mathrm{B}}{\partial u} + \frac{\partial \mathrm{C}}{\partial u}\right) . \tag{2.24}$$

The individual components of this sum are

$$\begin{aligned}
\frac{\partial \mathrm{A}}{\partial u} &= \frac{\partial E}{\partial u}\mathrm{D} + E\frac{\partial D}{\partial u} , \\
\frac{\partial \mathrm{B}}{\partial u} &= \frac{\partial F}{\partial u}(1-\mathrm{D}) - F\frac{\partial D}{\partial u} , \\
\frac{\partial \mathrm{C}}{\partial u} &= \frac{2(\mathrm{A}+\mathrm{B})\left(\frac{\partial A}{\partial u}+\frac{\partial B}{\partial u}\right) + \frac{\partial q}{\partial u}\mathrm{D}(1-\mathrm{D}) + q\,(1-2\,\mathrm{D})\frac{\partial D}{\partial u}}{2\,\mathrm{C}} .
\end{aligned} \tag{2.25}$$

Derivatives of E, F, and q with respect to θ and ϕ are zero. Spatial derivatives of these variables are estimated numerically. The remaining component D has the following angular and spatial derivatives

$$\begin{aligned}
\frac{\partial \mathrm{D}}{\partial \theta} &= 2\,\cos\xi\Big(\cos\theta\sin\theta_t\cos(\phi-\phi_t) - \sin\theta\cos\theta_t\Big) \\
\frac{\partial \mathrm{D}}{\partial \phi} &= -2\,\cos\xi\sin\theta\sin\theta_t\sin(\phi-\phi_t) \\
\frac{\partial \mathrm{D}}{\partial x} &= 2\,\cos\xi\Big(\sin\theta\cos\theta_t\cos(\phi-\phi_t)\frac{\partial\theta_t}{\partial x} - \cos\theta\sin\theta_t\frac{\partial\theta_t}{\partial x} + \\
&\qquad \sin\theta\sin\theta_t\sin(\phi-\phi_t)\frac{\partial\phi_t}{\partial x}\Big) ,
\end{aligned} \tag{2.26}$$

and $\frac{\partial \mathrm{D}}{\partial y}$, $\frac{\partial \mathrm{D}}{\partial z}$ are completely analogous to the expression of $\frac{\partial \mathrm{D}}{\partial x}$. Spatial derivatives of θ_t and ϕ_t are estimated numerically.

Finally, a derivative of phase slowness can be found from the above expressions with this simple relation

$$\frac{\partial S}{\partial u} = -\frac{1}{2\left[v^2\right]^{\frac{3}{2}}}\frac{\partial v^2}{\partial u} . \tag{2.27}$$

Computation of the equation coefficients for 2-D anisotropic media should follow the above approach as well with the obvious simplification of the angle between the phase direction and the axis of symmetry, given as

$$\cos\xi = cos\left(\theta - \theta_t\right) \ . \tag{2.28}$$

To sum up, only five 3-D spatial functions have to be computed in advance so that the escape equation coefficients could be estimated on demand. These functions are: $v_{\mathrm{v}}^2\left(\mathbf{x}\right)$, $v_{\mathrm{h}}^2\left(\mathbf{x}\right)$, $q\left(\mathbf{x}\right)$, $\theta_t\left(\mathbf{x}\right)$, and $\phi_t\left(\mathbf{x}\right)$. For the 2-D case, there are four 2-D spatial functions. Since spatial derivatives have to be estimated as well, a practical approach would consist of storing these functions in the form of spline coefficients (de Boor, 1978).

It should be noted that the coefficient in front of $\nabla_{\mathbf{x}} \cdot \widehat{x}$ term in equation (2.14) is the group slowness. Other approximations are possible to this term other than that presented above.

Escape functions in reduced phase space

To analyze behavior of escape functions in phase space, I computed a set of constant-depth and constant-location slices of them for a popular benchmark model - Marmousi (Versteeg, 1993).

Figures 2.6, 2.7, and 2.8 show escape quantities $\widehat{z}$, $\widehat{x}$, and $\widehat{T}$ respectively computed using ray tracing for all positions and angles at the z=2 km slice in reduced phase space. Figures 2.9, 2.10 and 2.11 present same quantities for a constant lateral location.

Each location at these plots is color coded according to the exit location and time for the ray that originated from it. All escape solutions clearly exhibit areas of smoothly changing values and regions of rapidly changing values – the latter is a

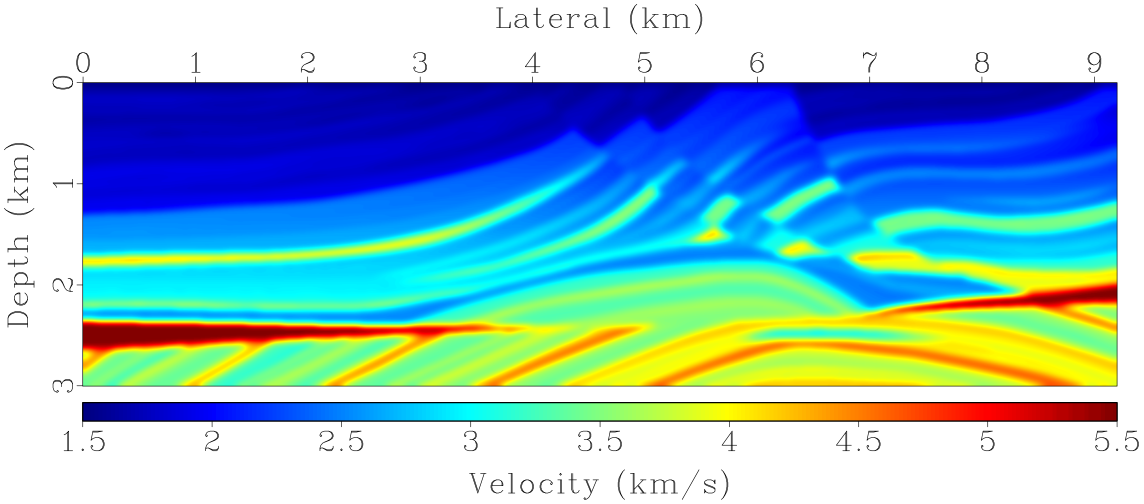


Figure 2.4: Smoothed Marmousi model velocity profile for escape value computations. chapter-esceq/marmousi marmvelescz

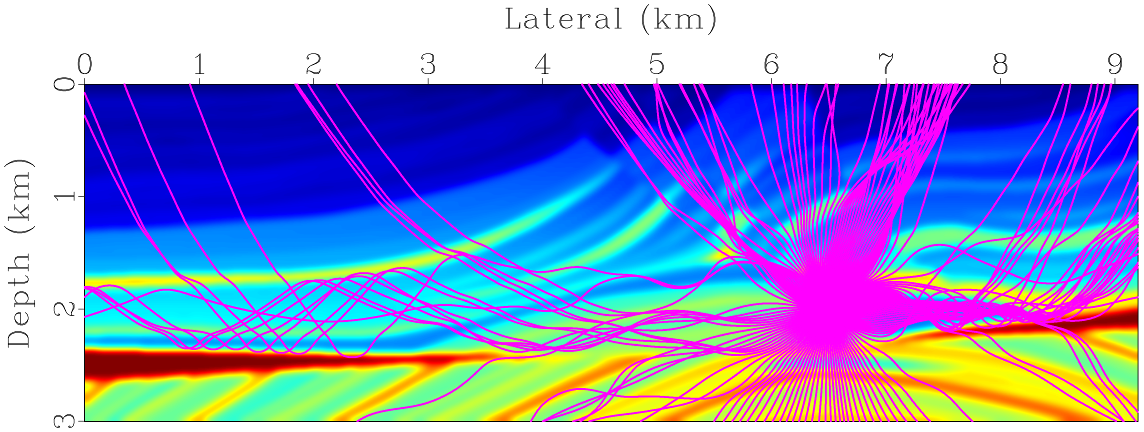


Figure 2.5: Ray trajectories in Marmousi model computed for location at $x = 6.5$ km and $z = 2.0$ km. chapter-esceq/marmousi marmtrajs

well-known effect in initial-value ray tracing, when a small change in initial conditions causes a large change in the solution (Figure 2.5).

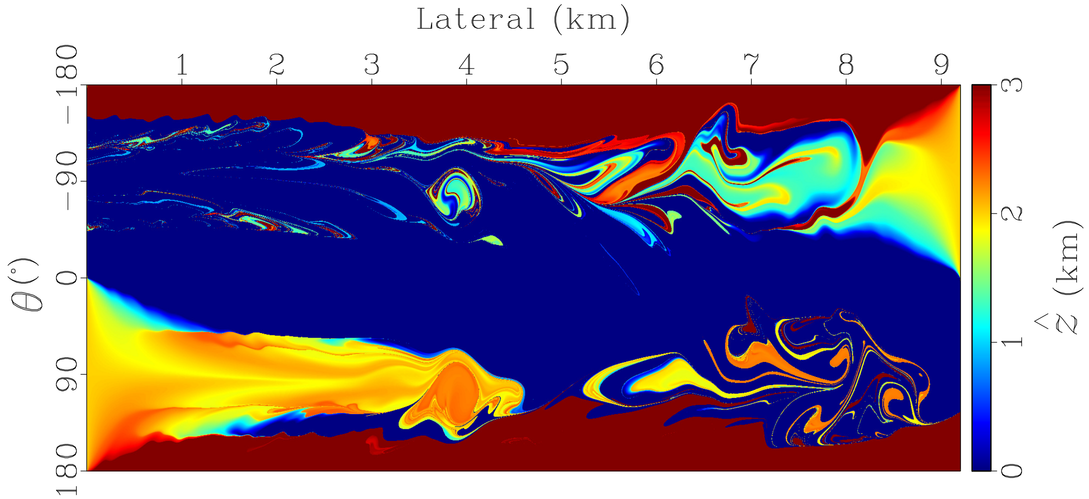


Figure 2.6: Escape depth function $\widehat{z}$ for all locations at constant depth $z = 2.0$ km of Marmousi model. chapter-esceq/marmousi marmnsz0

Escape functions behave similarly also in 5-D reduced phase space corresponding to a complex 3-D model. To test it, I computed escape quantities for a depth point inside SEG/EAGE Salt Model (Aminzadeh et al., 1997).

Figures 2.13, 2.14, 2.15, and 2.16 show escape quantities $\widehat{z}$, $\widehat{x}$, $\widehat{y}$, and $\widehat{T}$ respectively. Much like in 2-D case, the escape functions demonstrate areas of smoothly changing values and regions of rapidly changing values. The presence of a salt body and strong velocity gradients associated with it create rapid changes in escape functions.

In the next chapter, I demonstrate an Eulerian way to compute these functions by discretizing escape equations in reduced phase space.

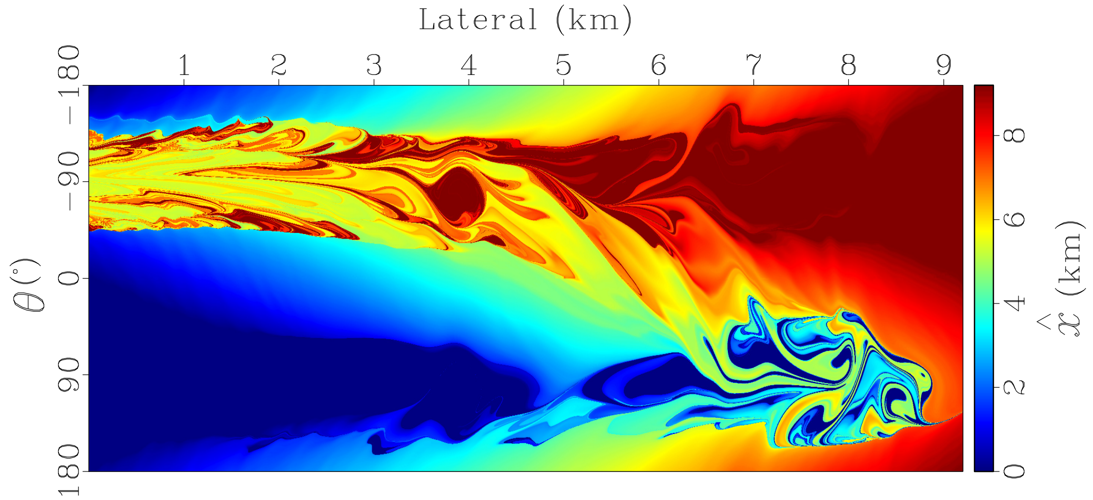


Figure 2.7: Escape lateral position function $\widehat{x}$ for all locations at constant depth $z = 2.0$ km of Marmousi model. chapter-esceq/marmousi marmnsz1

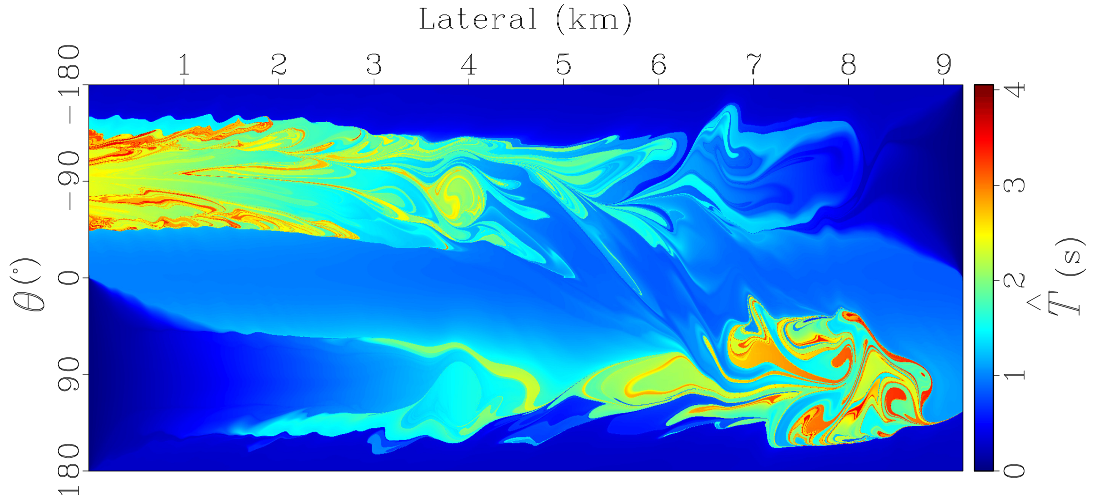


Figure 2.8: Escape time function $\widehat{T}$ for all locations at constant depth $z = 2.0$ km of Marmousi model. chapter-esceq/marmousi marmnsz2

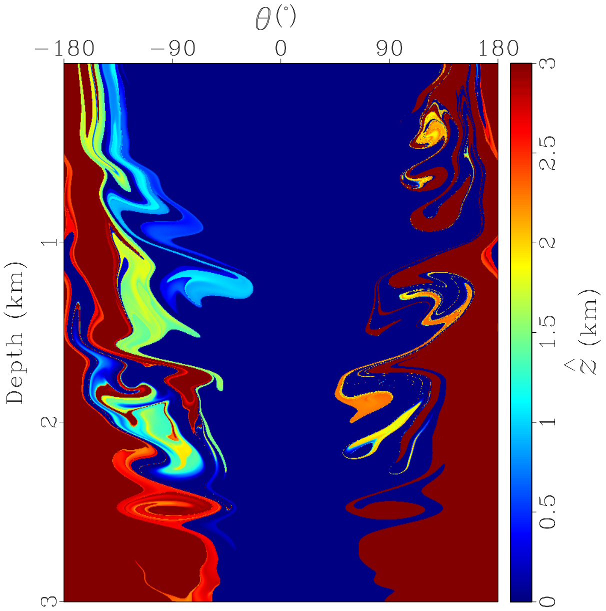


Figure 2.9: Escape depth function $\widehat{z}$ for all locations at constant lateral position $x = 6.5$ km of Marmousi model. chapter-esceq/marmousi marmnsx0

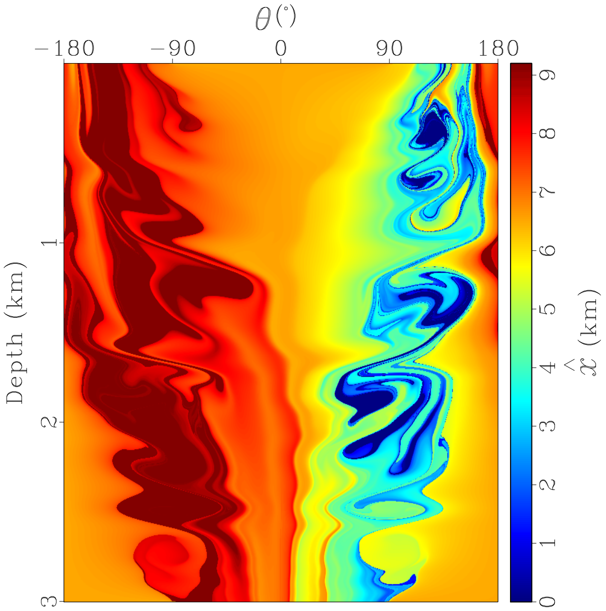


Figure 2.10: Escape lateral position function $\widehat{x}$ for all locations at constant lateral position $x = 6.5$ km of Marmousi model. chapter-esceq/marmousi marmnsx1

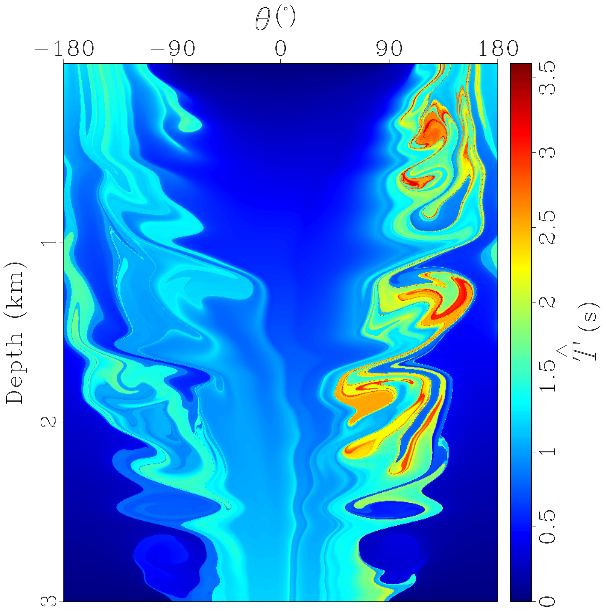


Figure 2.11: Escape time function $\widehat{T}$ for all locations at constant lateral position $x = 6.5$ km of Marmousi model. chapter-esceq/marmousi marmnsx2

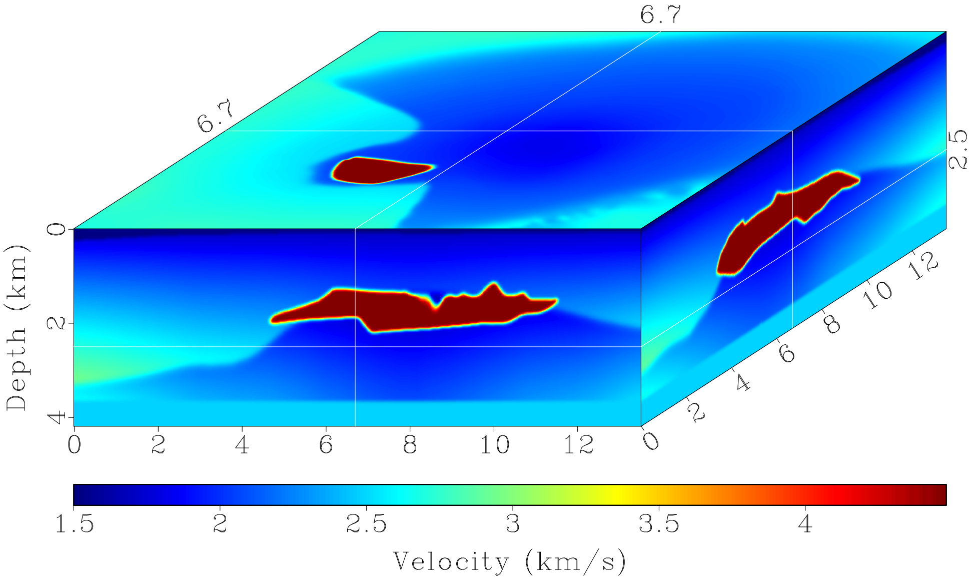


Figure 2.12: Smoothed SEG/EAGE Salt model velocity volume for escape value computations. chapter-esceq/segsalt vsaltesczxy

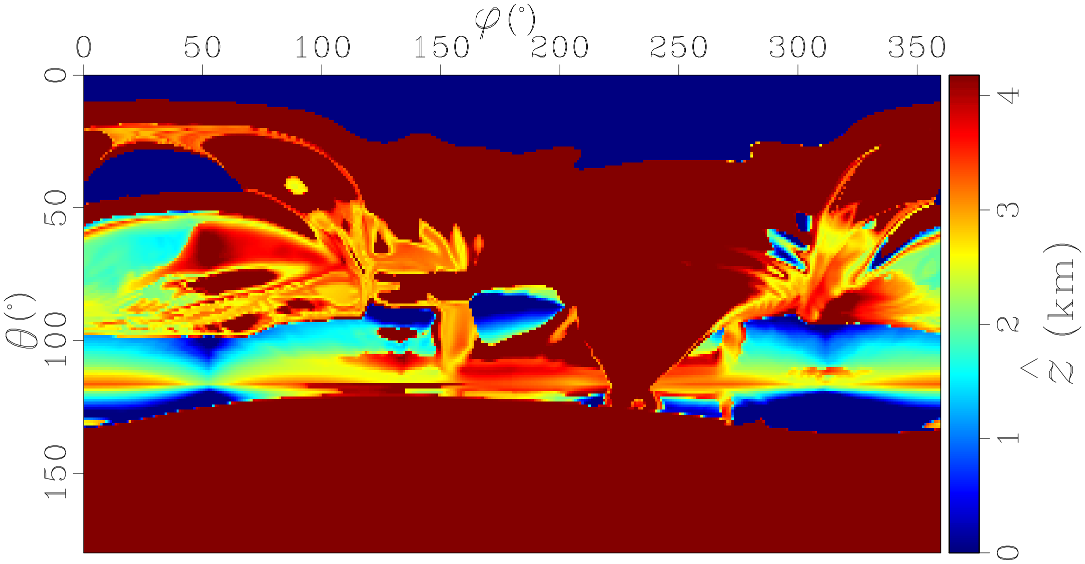


Figure 2.13: Escape depth function $\widehat{z}$ at point $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. chapter-esceq/segsalt ssaltnesczxy0

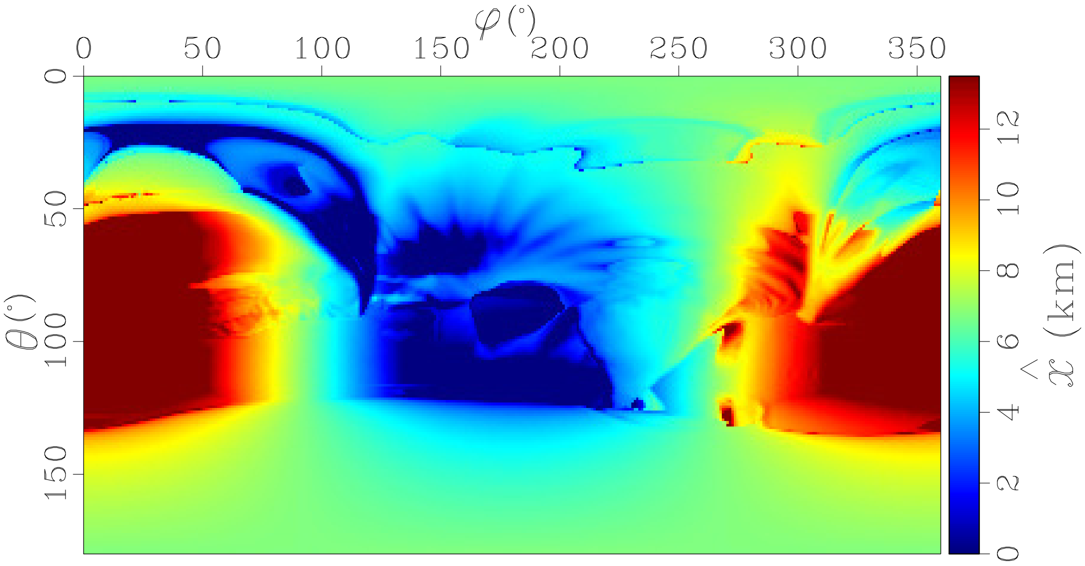


Figure 2.14: Escape position function $\widehat{x}$ at point $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. chapter-esceq/segsalt ssaltnesczxy1

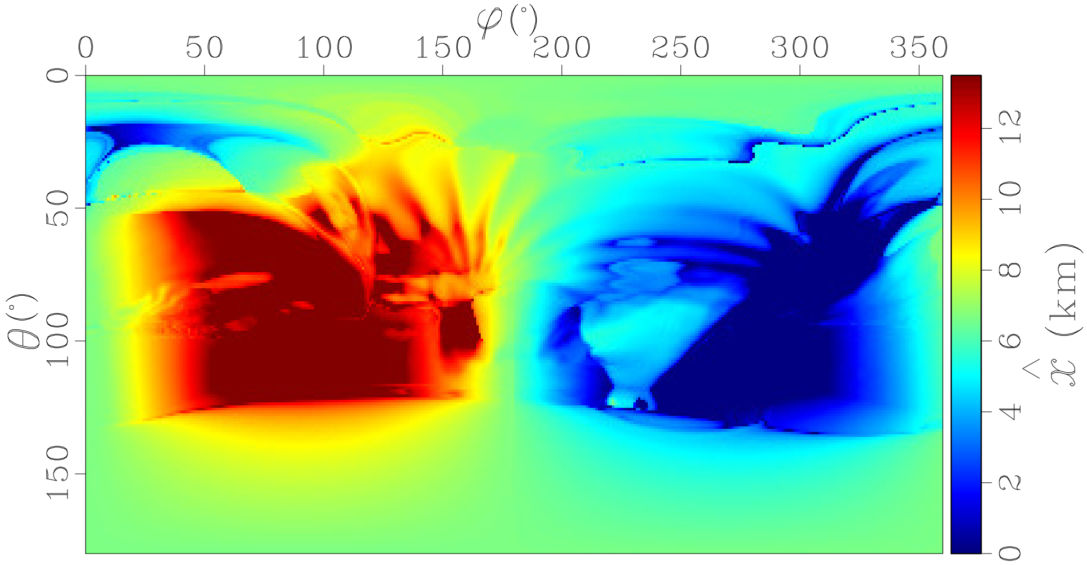


Figure 2.15: Escape position function $\widehat{y}$ at point $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. $\boxed{\text{chapter-esceq/segsalt ssaltnesczxy2}}$

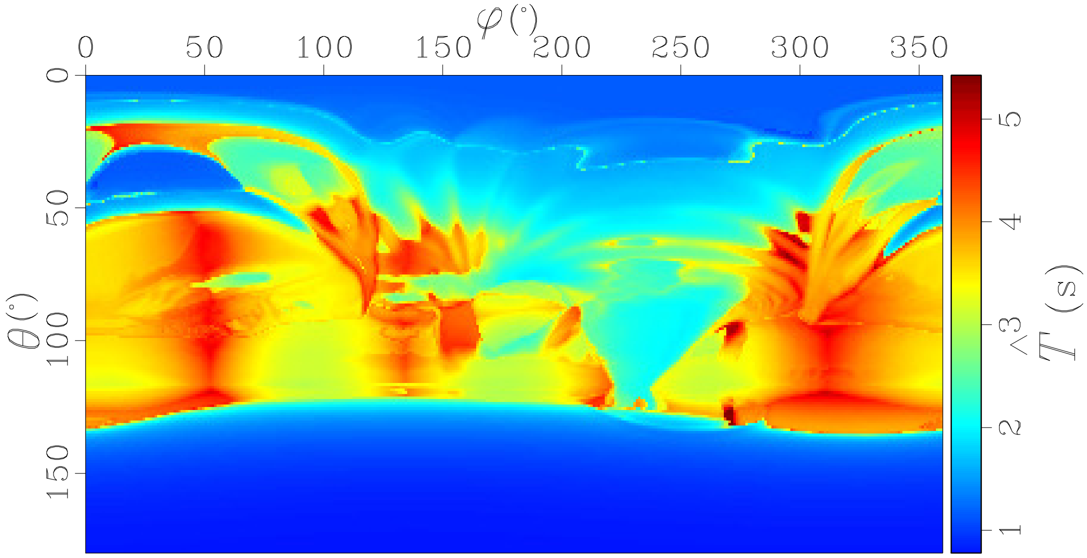


Figure 2.16: Escape time function $\widehat{T}$ at point $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. $\boxed{\text{chapter-esceq/segsalt ssaltnesczxy3}}$

Chapter 3

Finite-difference solution of escape equations

Design constraints for a numerical scheme

Many different numerical techniques have been developed for solving advection-type PDEs in the last several decades (Pletcher et al., 2012). A particular choice usually depends on the following properties of a problem in question and development constraints:

1. Geometry and dimensionality of the problem.
2. Available computational resources.
3. Development cost limits.

In the exploration industry, the last factor is especially pronounced because of the multitude of existing imaging methods. There is no single perfect imaging technique, so imaging practitioners usually have to make a choice from tens of different algorithms depending on the challenges that they face for a given exploration problem (Etgen et al., 2009). Implementation of one from the many needed software codes is a subject to a rather strict development cost upper limit – it is usually expected to be at most three person-years (Bednar, 2002). Something that requires two or three times as much effort can hardly be considered practical.

Geometry of the problem described by the reduced phase space escape equations is relatively simple – it is dictated by the type of input expected in the angle-domain migration module. Regularly sampled volumes of escape tables should be sufficient for practical imaging purposes.

Unlike the two factors described above, available computational resources are more difficult to assess. This category has been very dynamic through the history of computational sciences. For several decades, both aggregate supercomputer CPU power and memory have been growing exponentially (Danowitz et al., 2012; ITRS, 2011). There is an ongoing debate about how long this growth can continue into the future; physical limitations, such as power consumption and the smallest possible transistor size, are usually considered to be the limiting factors of the exponential growth. At the time of this writing, top supercomputers available academically and privately have aggregate computational performance in the low petaFLOPS range (Top500, 2013). Total distributed memory is in the hundreds of terabytes and is going into the petabyte range with the next generation of machines. However, these ranges are, of course, almost never readily available for use by a single computational problem. For practical purposes, it is safe to assume that a typical allocation of resources is currently in the low tens of terabytes for distributed memory and the tens to the low hundreds of teraFLOPS for aggregate CPU power.

In this chapter, I show the design of an Eulerian numerical scheme for solving the reduced phase space equations (2.14) under the constraints described above. For the sake of simplicity, I will illustrate an implementation of the designed numerical scheme for 3-D reduced phase space first. There is no principal difference between 3-D and 5-D cases. Any problems revealed in the former will be more pronounced in the latter. I will analyze accuracy and performance of the designed Eulerian scheme so as to make conclusions about the feasibility of adapting it for the realistic-scale 5-D case.

Choice of discretization technique

Fundamentally, there are two ways of discretizing equations (2.14):

- finite differences/finite volumes methods (FDM/FVM),
- finite elements methods (FEM).

FEM allows to handle complex geometries. On the other hand, FDM is generally easier to implement. The rather simple geometry of the problem in question and the strict limit of development cost make FDM a more preferable basis for a numerical scheme. In fact, the combination of these two factors are very common in seismic imaging problems, therefore, FDM is more prevalent in this computational area (Virieux et al., 2011).

A common way of finding a steady-state solution for an advection problem is to start with a non-steady formulation (Kuzmin, 2010). The non-steady formulation for 3-D reduced phase space equations can be written as

$$\frac{\partial \widehat{x}}{\partial \sigma} + \mathbf{a}_{2-D}\left(\mathbf{x}, \theta\right) \cdot \nabla_{\mathbf{x},\theta}\, \widehat{x} = b \; . \tag{3.1}$$

A steady-state solution is then obtained by stepping in σ with relatively small steps until the difference in the solution between two consecutive steps is negligible.

Reformulation to the non-steady form would allow for using one of the many efficient schemes designed in the framework of level set methods (Osher and Sethian, 1988; Shu, 1998). However, this approach has several notorious disadvantages associated with the addition of the extra dimension σ, namely

- stepping in σ is subject to CFL condition – strong variations in the vector field $\mathbf{a}_{2-D}\left(\mathbf{x},\theta\right)$ may impose a very small $\Delta\sigma$ step thus hampering computational efficiency;
- the extra dimension increases requirements to computer memory making an expansion to 5-D reduced phase space impractical.

It appears that, to preserve the computational efficiency and memory requirements, the steady-state equation has to be solved directly without evolution from the corresponding non-steady form.

Upwind finite differences

Application of finite differences method to equations (2.14) produces a system of linear equations

$$\bar{\mathbf{A}}\,\widehat{\mathbf{x}} = \mathbf{b}\ , \tag{3.2}$$

where $\bar{\mathbf{A}}$ is a sparse matrix comprising the finite-difference stencil coefficients.

While there are multiple possible finite-difference discretizations for first-order partial derivatives, the advection term in equations (2.14) should be discretized in each dimension according to the upwind principle (Courant et al., 1952; Gentry et al., 1966), i.e., the finite-difference stencil for the current point ought to be oriented toward the opposite direction of the vector field direction (Figure 3.1). The first-order upwind stencil in one dimension is given by the following simple expression (Hirsch, 2007):

$$\frac{\partial \widehat{x}}{\partial x_k} \approx \begin{cases} \dfrac{\widehat{x}_i - \widehat{x}_{i-1}}{\Delta x_k}, & a_{k,i} > 0 \\ \dfrac{\widehat{x}_{i+1} - \widehat{x}_i}{\Delta x_k}, & a_{k,i} < 0 \end{cases}\ , \tag{3.3}$$

where x_k is the k-th axis of the reduced-phase space volume, Δx_k is the grid sampling along the axis, $\widehat{x}_i$ is the escape value at the i-th node of the same dimension, and $a_{k,i}$ is the k-th component of vector $\mathbf{a}$ at the same location.

A more accurate second-order upwind stencil is

$$\frac{\partial \widehat{x}}{\partial x_k} \approx \begin{cases} \dfrac{3\,\widehat{x}_i - 4\,\widehat{x}_{i-1} + \widehat{x}_{i-2}}{2\,\Delta x_k}, & a_{k,i} > 0 \\ \dfrac{-\widehat{x}_{i+2} + 4\,\widehat{x}_{i+1} - 3\,\widehat{x}_i}{2\,\Delta x_k}, & a_{k,i} < 0 \end{cases} . \tag{3.4}$$

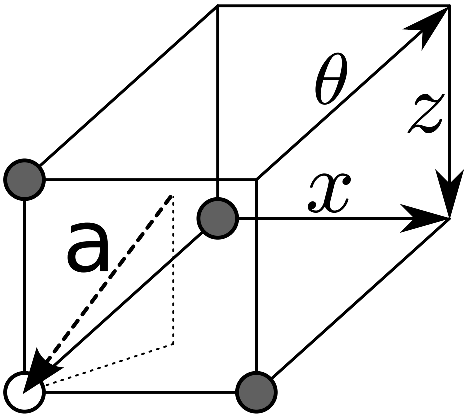


Figure 3.1: Upwind principle demonstrated on the first-order finite-difference stencil: the white point is being computed, the grey points have known values and are located in the upwind direction, the stencil is aligned with the known points in the upwind direction determined by the vector field **a** (a scheme). chapter-fdiff/. escadv2

The system (3.2) can be solved iteratively in a number of ways. The Gauss-Seidel method (Golub, 1996) is used frequently, because it allows construction of a new solution from the previous iteration "in place" without generating extra copies of the data. Note that, the matrix $\bar{\mathbf{A}}$ does not have to be computed explicitly and stored in computer memory – the matrix elements are the escape equations coefficients and can be recomputed during iterations. Therefore, only the solution vector has to be present in the computer memory.

Boundary conditions and numerical flow

While escape equations describe a steady-state solution, the iterative numerical scheme presented above realizes a numerical flow along streamlines defined by the vector field $\mathbf{a}$. This flow propagates new values into the reduced phase space volume away from the known boundary values.

If $\partial\Omega$ is the outer shell of the reduced phase space volume and $\mathbf{n}_{\partial\Omega}$ is the normal to the shell pointing in the outward direction, then the solution is determined by boundary conditions defined on the parts of the shell, where

$$\mathbf{a} \cdot \mathbf{n}_{\partial\Omega} < 0 \,.$$

Figure 3.2 shows the boundary condition areas (shaded grey areas) for 3-D reduced phase space volume. For the 5-D case, each spatial 3-D volume of constant azimuth and constant inclination has three adjacent faces dedicated the boundary conditions (Figure 3.3).

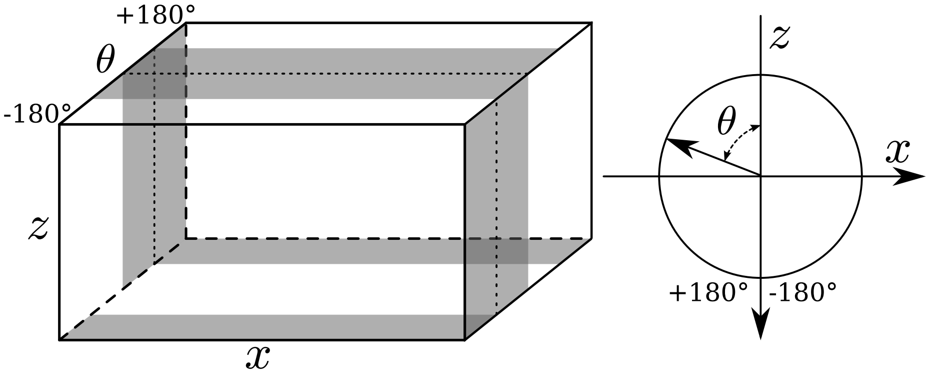


Figure 3.2: Boundary conditions for 3-D reduced phase space (a scheme). chapter-fdiff/. escbc2

Gauss-Seidel iterations consist of visiting every point of the reduced phase grid, except for the boundary-condition areas, and applying an upwind finite-difference

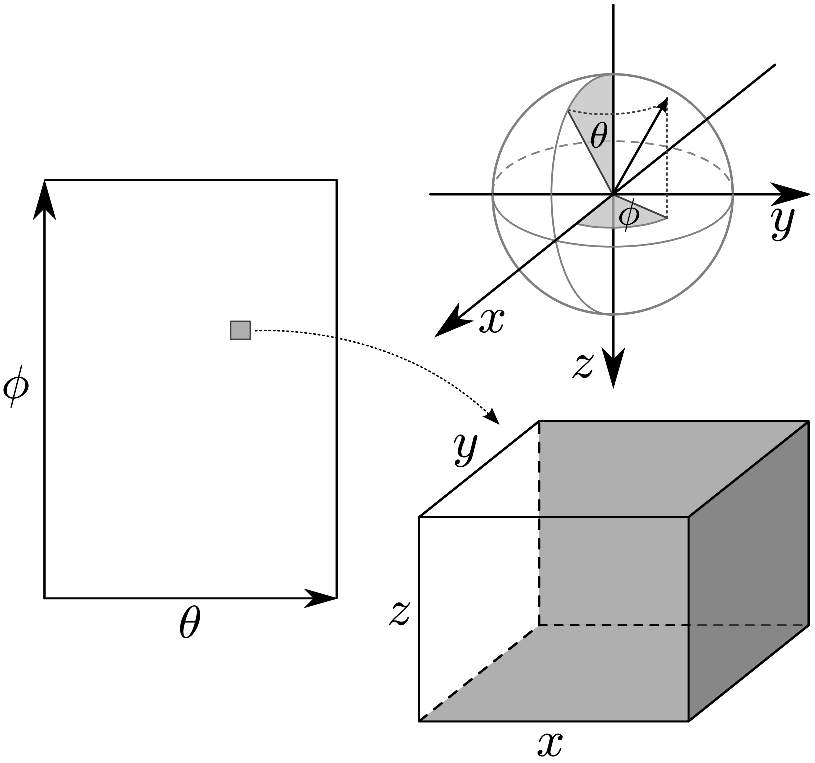


Figure 3.3: Boundary conditions for 5-D reduced phase space (a scheme). chapter-fdiff/. escbc3

stencil so as to obtain the correct escape function solution. For faster convergence, the iterative solver may incorporate alternating directions akin to those of the fast sweeping method for the eikonal equation (Zhao, 2005).

Numerical tests of the iterative finite-difference solver

Figures 3.4, 3.5, and 3.6 show escape quantities $\widehat{z}$, $\widehat{x}$, and $\widehat{T}$ respectively computed using the first-order finite-difference iterative solver. Comparison of depths slices (Figures 3.7, 3.8, and 3.9) with their ray-traced counterparts (Figures 2.6, 2.7, and 2.8) reveals that the finite-difference solution looks less detailed - it has a significant amount of numerical dissipation. The second-order finite-difference results (see Figures 3.10, 3.11, and 3.12 for full volumes; Figures 3.7, 3.8, and 3.9 for depth slices) are less dissipative but still do not expose as much detalization as ray tracing. The spatial sampling is Δx=Δz=4 m and the angular sampling is $\Delta\theta$=1°.

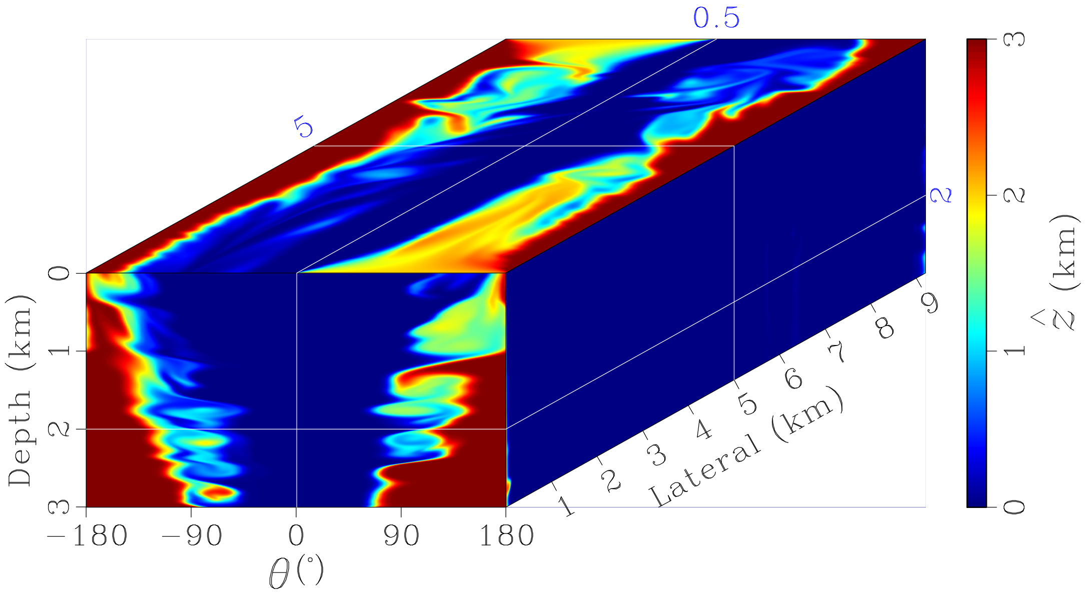


Figure 3.4: Escape depth function $\widehat{z}$ computed for Marmousi model (first-order finite-difference iterative solver). chapter-fdiff/marmousi marmfc00

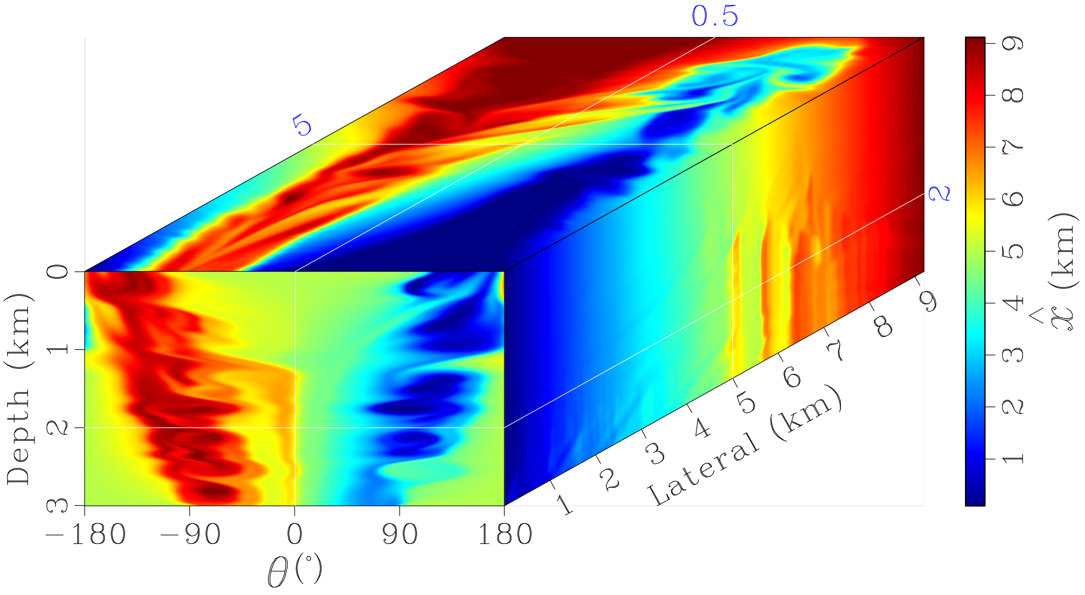


Figure 3.5: Escape lateral position function $\widehat{x}$ computed for Marmousi model (first-order finite-difference iterative solver). chapter-fdiff/marmousi marmfc01

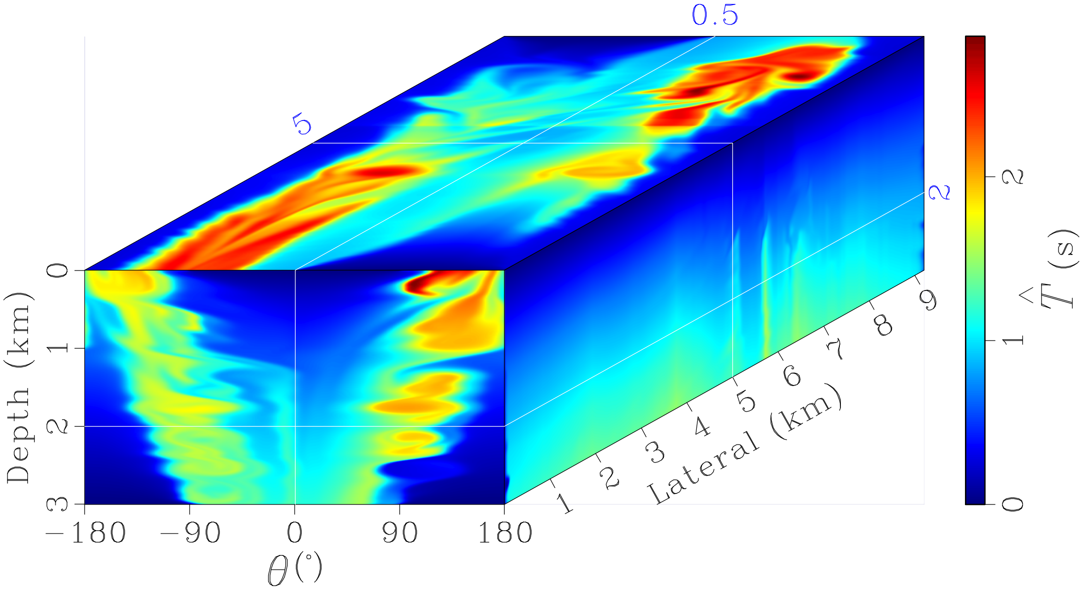


Figure 3.6: Escape time function $\widehat{T}$ computed for Marmousi model (first-order finite-difference iterative solver). chapter-fdiff/marmousi marmfc02

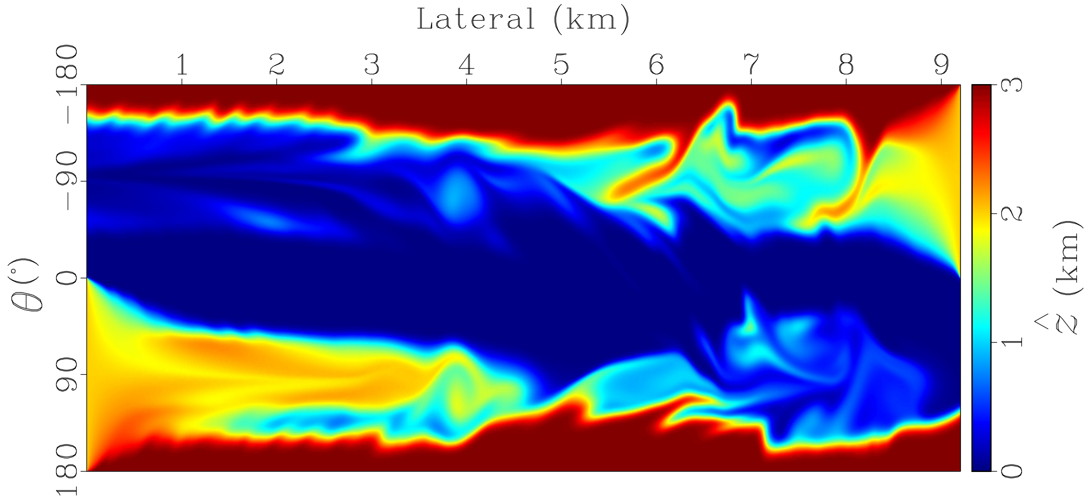


Figure 3.7: Escape depth function $\widehat{z}$ at constant depth $z = 2.0$ km of Marmousi model (first-order finite-difference iterative solver). chapter-fdiff/marmousi marmfsz00

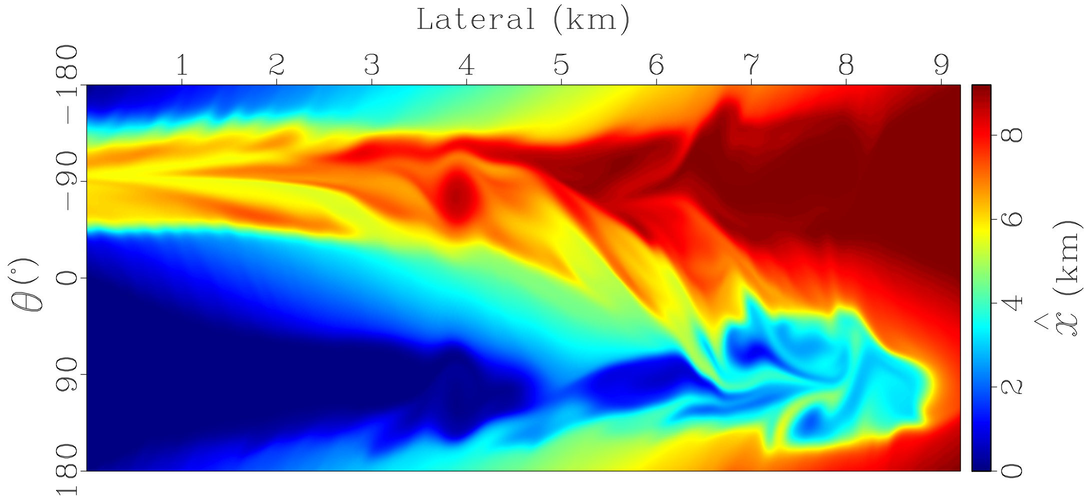


Figure 3.8: Escape lateral position function $\widehat{x}$ at constant depth $z = 2.0$ km of Marmousi model (first-order finite-difference iterative solver). chapter-fdiff/marmousi marmfsz01

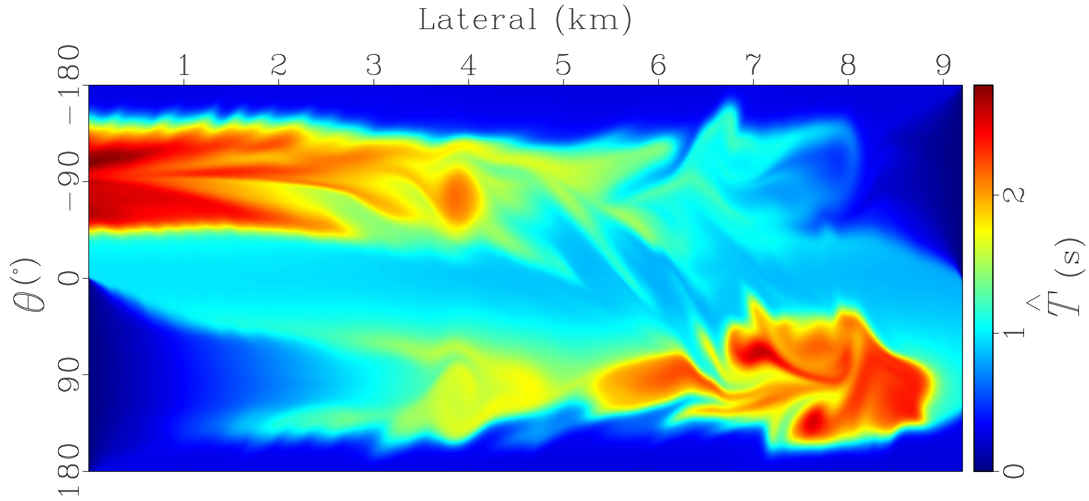


Figure 3.9: Escape time function $\widehat{T}$ at constant depth $z = 2.0$ km of Marmousi model (first-order finite-difference iterative solver). chapter-fdiff/marmousi marmfsz02

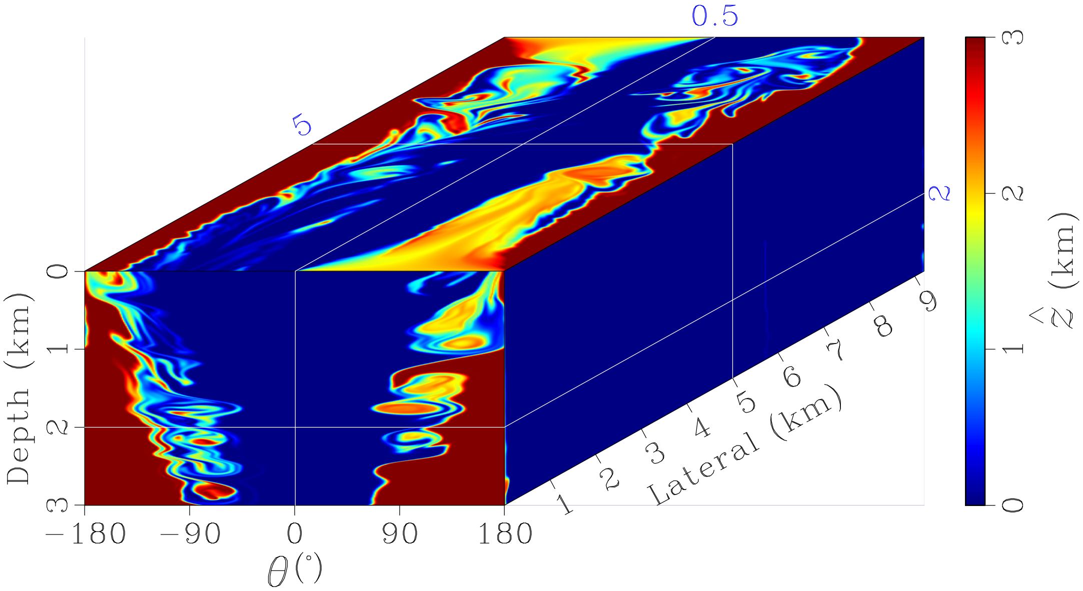


Figure 3.10: Escape depth function $\widehat{z}$ computed for Marmousi model (second-order finite-difference iterative solver). chapter-fdiff/marmousi marmfc10

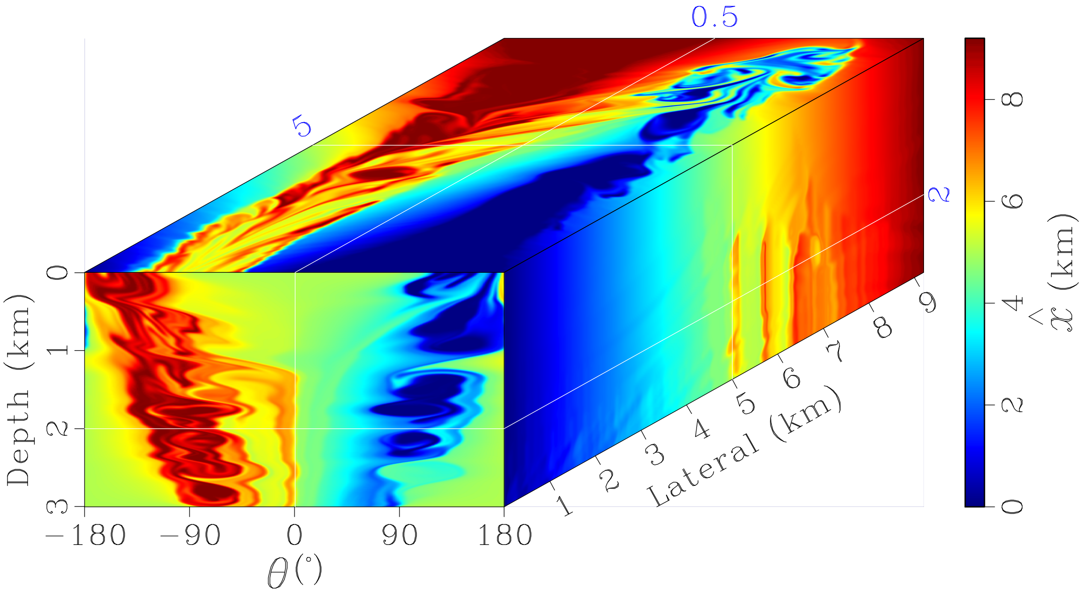


Figure 3.11: Escape lateral position function $\widehat{x}$ computed for Marmousi model (second-order finite-difference iterative solver). chapter-fdiff/marmousi marmfc11

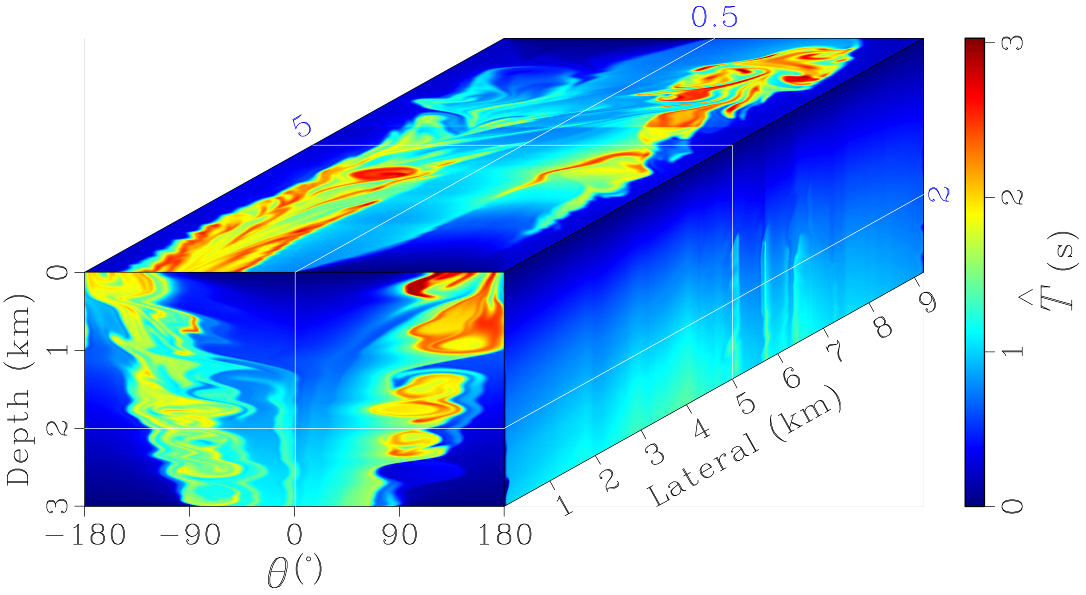


Figure 3.12: Escape time function $\widehat{T}$ computed for Marmousi model (second-order finite-difference iterative solver). chapter-fdiff/marmousi marmfc12

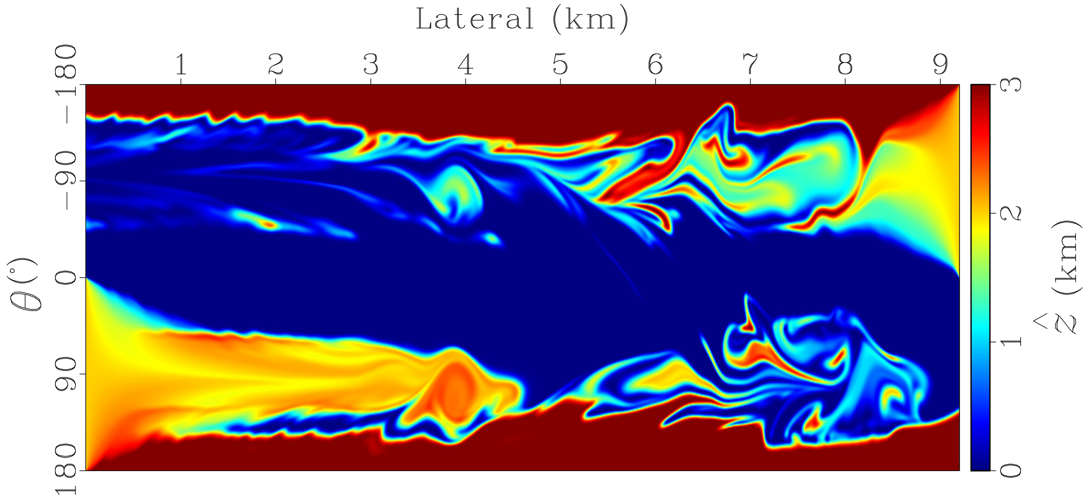


Figure 3.13: Escape depth function $\widehat{z}$ at constant depth $z = 2.0$ km of Marmousi model (second-order finite-difference iterative solver). chapter-fdiff/marmousi marmfsz10

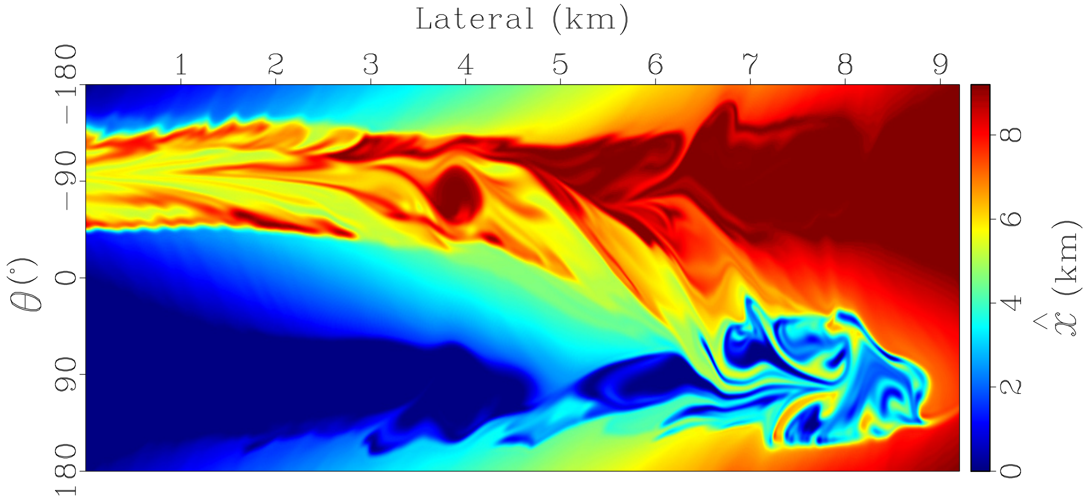


Figure 3.14: Escape lateral position function $\widehat{x}$ at constant depth $z = 2.0$ km of Marmousi model (second-order finite-difference iterative solver). chapter-fdiff/marmousi marmfsz11

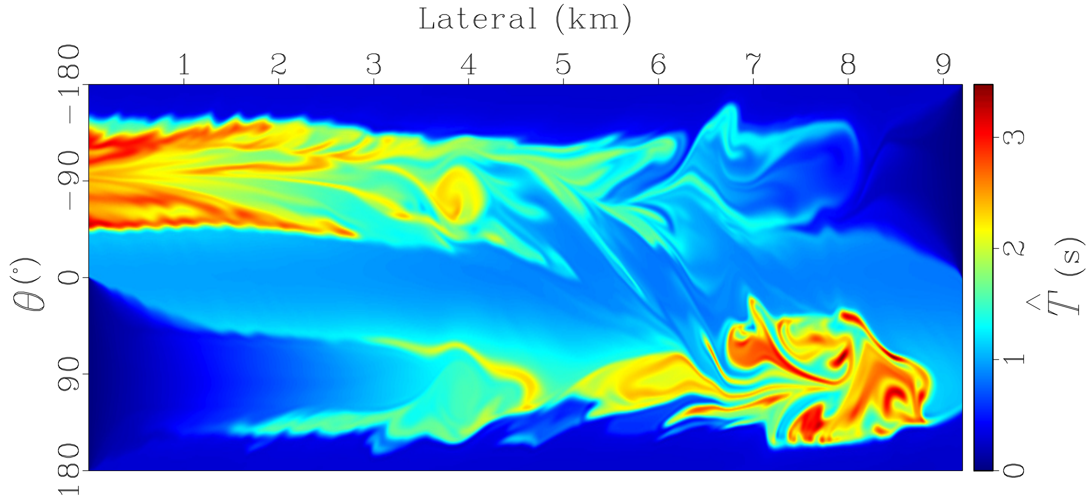


Figure 3.15: Escape time function $\widehat{T}$ at constant depth $z = 2.0$ km of Marmousi model (second-order finite-difference iterative solver). chapter-fdiff/marmousi marmfsz12

Appearance of strong dissipation in the solution results in many arrivals not appearing on the actual exit boundary of the model. Figures 3.16 and 3.17 show this effect for one depth point in the model. For these plots, I extracted escape depth values $\widehat{z}$ and escape lateral location values $\widehat{x}$ from the escape functions computed using the first-order and the second-order solvers respectively. Each pair of $\widehat{z}$ and $\widehat{x}$ comprises one exit location. Ideally, every exit location should be positioned somewhere along the boundary of the model. It is especially important to compute surface-bound arrivals with good accuracy since they are utilized in migration of surface reflection data.

Using the same depth point as a point source, I computed a wavefield with the help of an acoustic finite-difference time-domain propagator (Virieux et al., 2012). I extracted a surface-bound snapshot from the wavefield and plotted the same arrivals over it (Figures 3.18 and 3.19). It is apparent that both finite-difference solvers do not cover the arrivals on the right side of the snapshot precisely.

Modification of the iterative Eulerian solver

The presence of strong dissipation in the computational results above signals that the chosen grid resolution is not fine enough to capture the behavior of the escape functions in question. I computed a smaller subset of the functions on finer grids using ray tracing in order to investigate how escape functions behave at smaller scales. Figures 3.20, 3.21, and 3.21 show escape quantities $\widehat{z}$, $\widehat{x}$, and $\widehat{T}$ computed using Δx=Δz=1 m and the angular sampling $\Delta\theta$=0.1°. Figures 3.23, 3.24, and 3.24 demonstrate the same quantities for Δx=Δz=0.25 m and the angular sampling is $\Delta\theta$=0.02°. The second closeup reveals that the fine features in the solution can be actually resolved at very small scales - fractions of a meter and a degree. At those

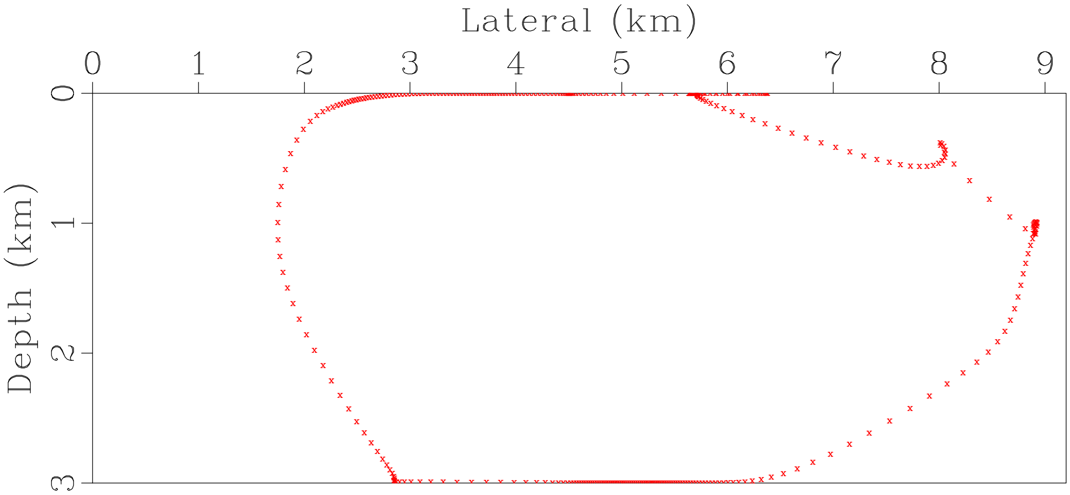


Figure 3.16: Exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (first-order finite-difference iterative solver). chapter-fdiff/marmousi marmpescs1

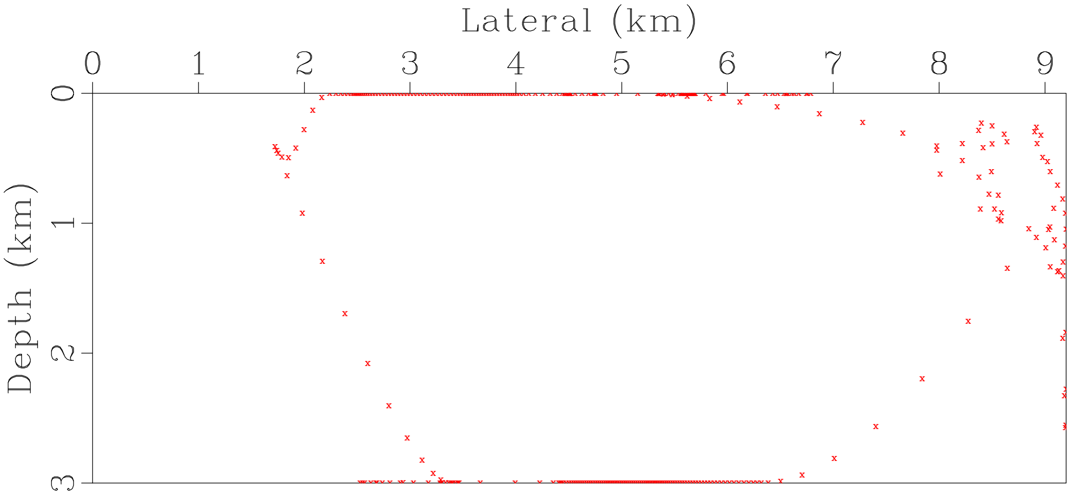


Figure 3.17: Exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (second-order finite-difference iterative solver). chapter-fdiff/marmousi marmpescs2

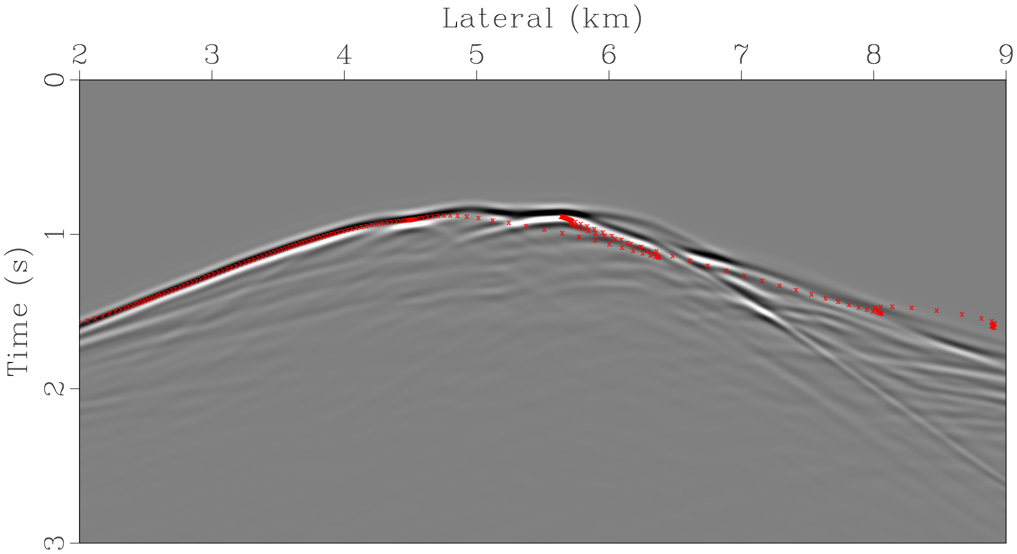


Figure 3.18: Surface exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (first-order finite-difference iterative solver) plotted over wavefield computed for a source at the same point. chapter-fdiff/marmousi marmfdescs1

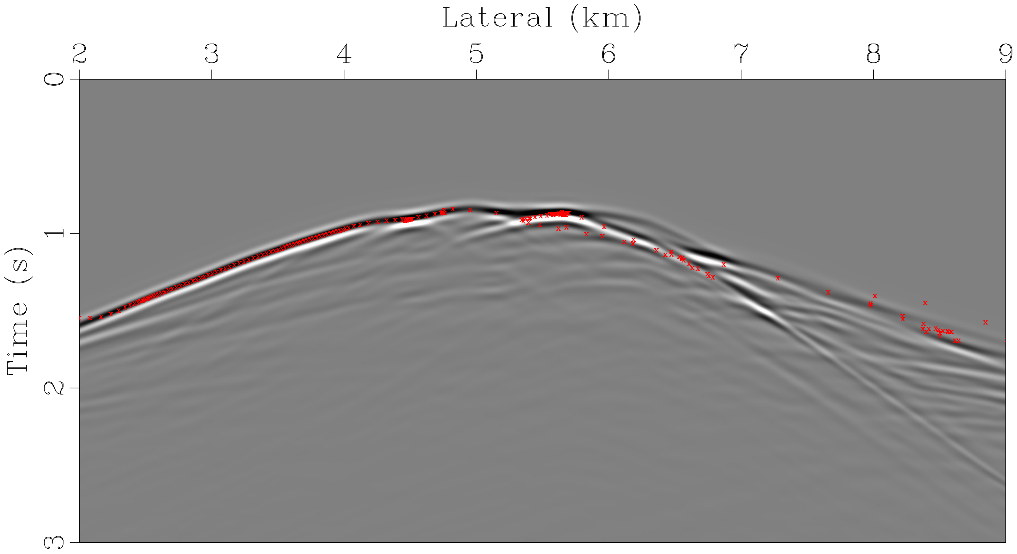


Figure 3.19: Surface exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (second-order finite-difference iterative solver) plotted over wavefield computed for a source at the same point. chapter-fdiff/marmousi marmfdescs2

scales, escape functions are continuous as predicted by the theory. At the scale of interest for imaging, however, they appear as discontinuous. Arrivals matching tests above clearly show that the accuracy of the finite-difference solver on a 4-meter,1-degree grid is insufficient. It is possible to perform same computations on a finer grid to improve the accuracy, but this would lead to the following problems:

- significant increase in computer memory required by the algorithm;
- loss of computational efficiency, since a greater number of points have to be visited by the solver;
- output sampling will be much finer than what is needed by angle-domain migration; the migration part needs escape tables sampled every 10-20 meters in space and 0.25-2° in angle in order to image a typical marine seismic data.

Therefore, it would be highly impractical to simply increase the uniform grid sampling in order to reduce the numerical dissipation.

A usual remedy for this problem in the Eulerian framework would be AMR - Adaptive Mesh Refinement (Berger and Oliger, 1984; Plewa et al., 2005) and/or application of an finite-difference stencil of a higher order (Shu, 1998). The effectiveness of higher-order, accurate discretizations is limited by the fact that boundary conditions for escape equations contain non-smooth parts due to the shape of the domain (i.e., domain corners). Refining computational grid to extremely fine scales only where necessary may improve accuracy of computations without drastic increase in computer memory requirements. However, AMR does not address the problem of spending computer time on obtaining extraneously excessive resolution. More to the point, implementation of AMR in higher dimensions can be costly (Van Straalen

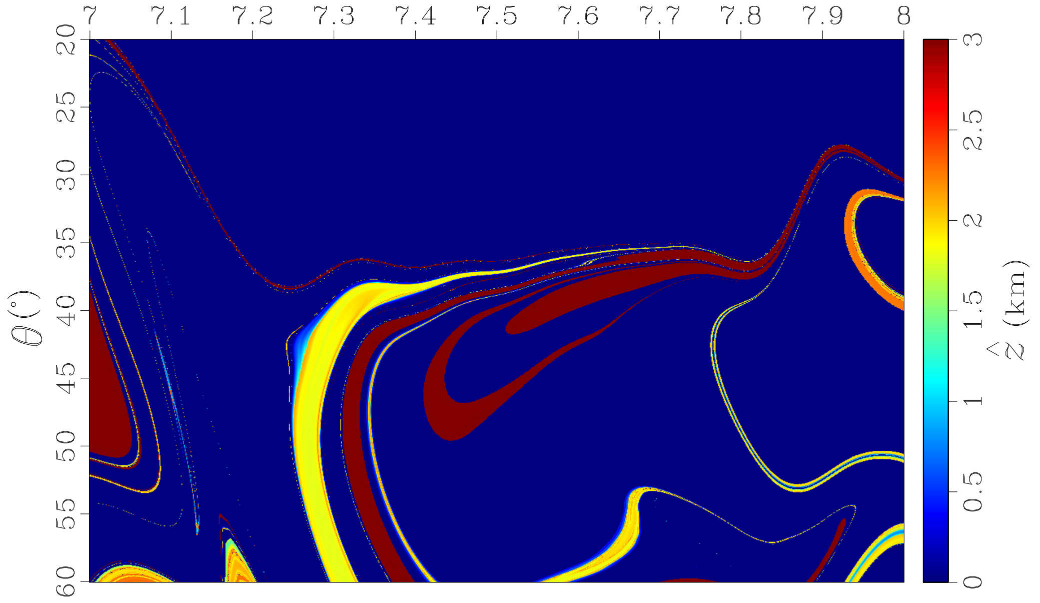


Figure 3.20: Closeup of escape depth function $\widehat{z}$ at constant depth z = 2.0 km of Marmousi model (ray tracing). chapter-fdiff/marmousi marmnz10

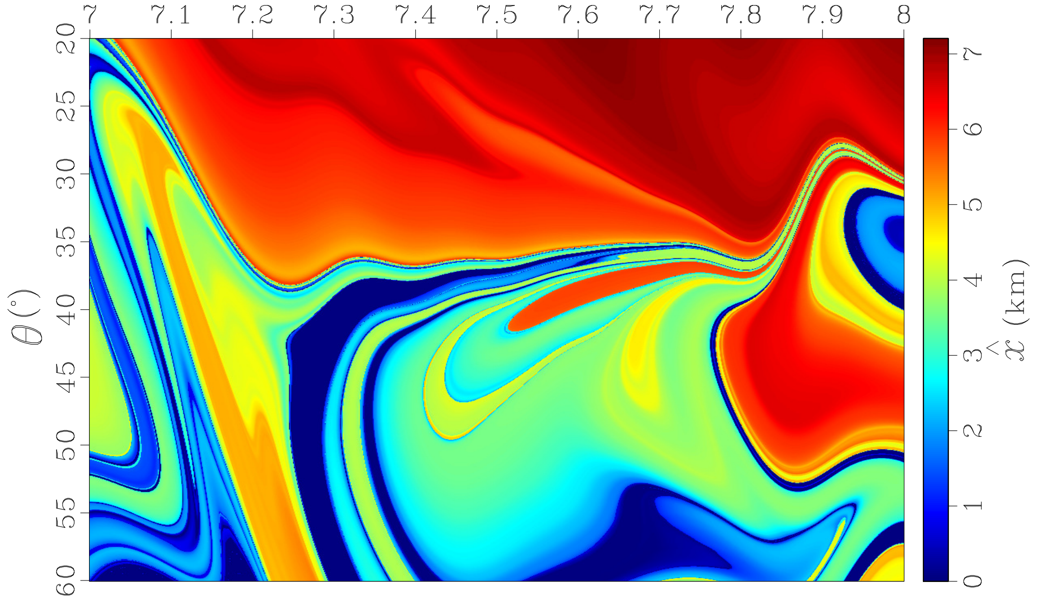


Figure 3.21: Closeup of escape lateral position function $\widehat{x}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). chapter-fdiff/marmousi marmnz11

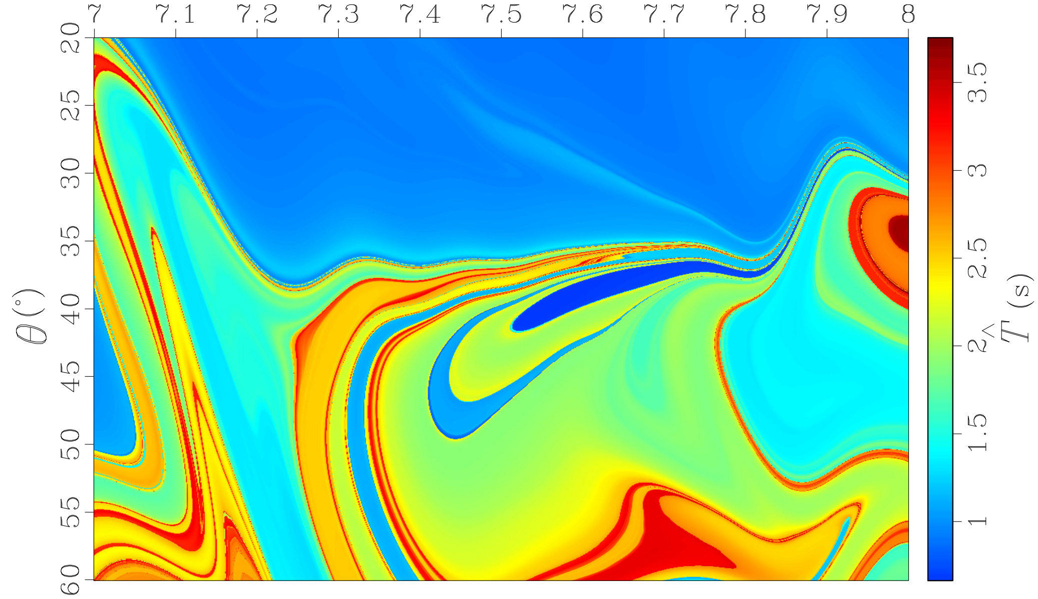


Figure 3.22: Closeup of escape time function $\widehat{T}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). chapter-fdiff/marmousi marmnz12

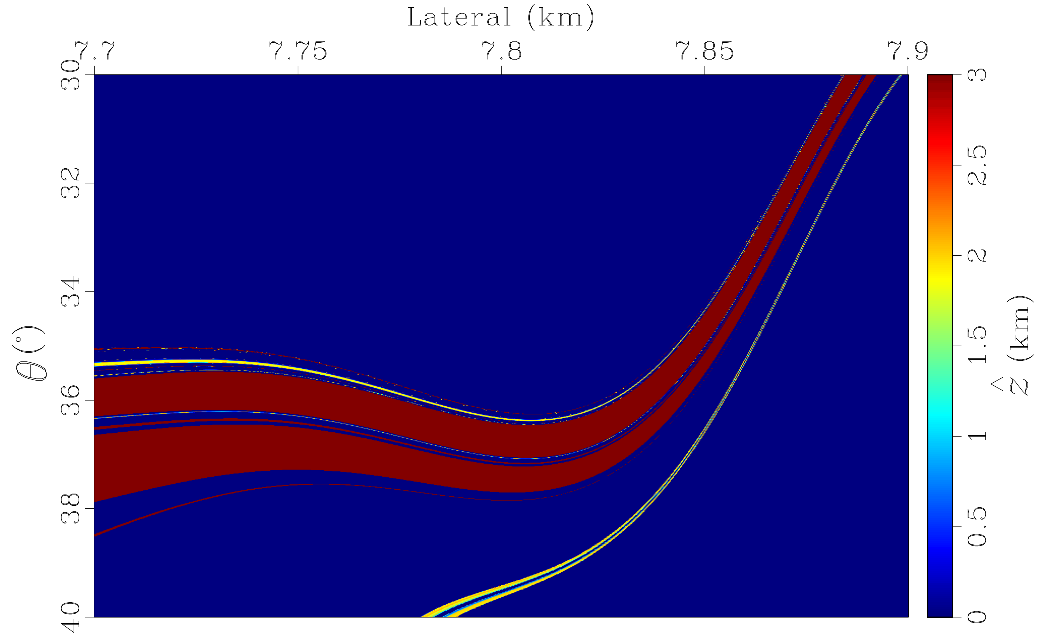


Figure 3.23: Closeup of escape depth function $\widehat{z}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). chapter-fdiff/marmousi marmnz20

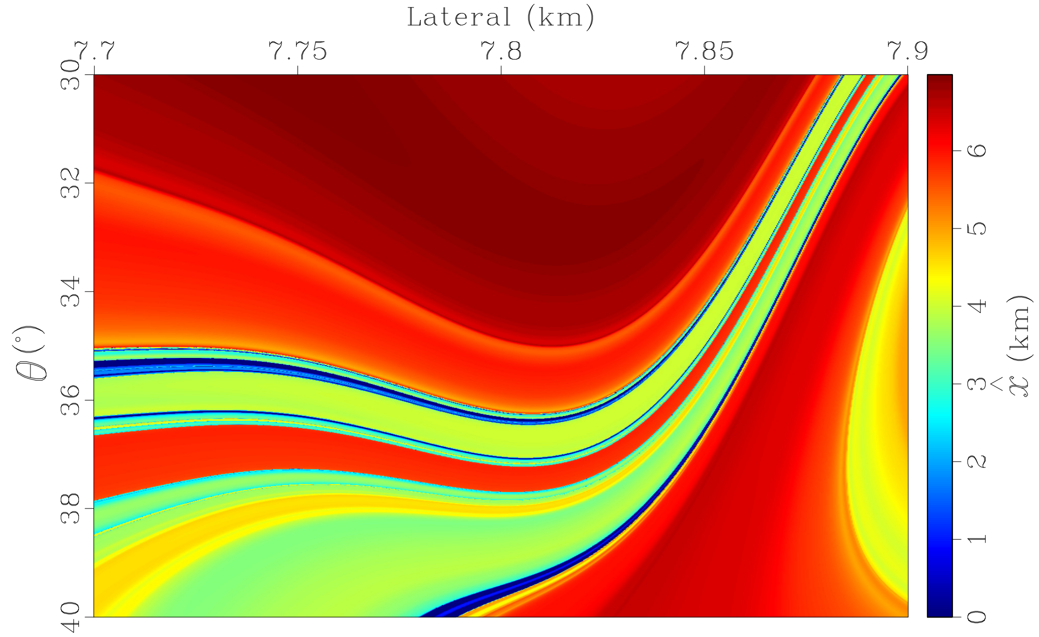


Figure 3.24: Closeup of escape lateral position function $\widehat{x}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). chapter-fdiff/marmousi marmnz21

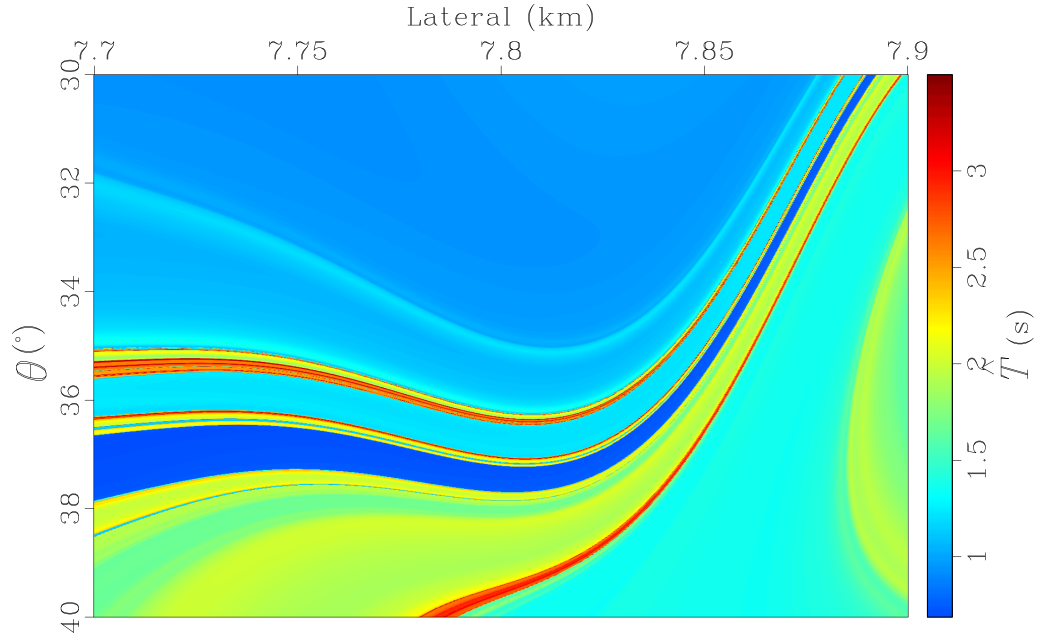


Figure 3.25: Closeup of escape time function $\widehat{T}$ at constant depth $z = 2.0$ km of Marmousi model (ray tracing). chapter-fdiff/marmousi marmnz22

et al., 2011), possibly exceeding the typical implementation cost upper limit explained in the beginning of this chapter.

A different strategy that allows the inhibition of numerical dissipation in advection problems is to use accurate Lagrangian solutions at grid points where desired accuracy cannot be acquired with low-order finite-difference stencils. Such hybrid, Eulerian-Lagrangian schemes often enable remarkable improvements in accuracy without costly grid refinement (Ferziger and Perić, 2002). Before applying the finite-difference stencil to a group of upwind points, I check if the Euclidean distances between their escape locations are smaller than some predefined threshold. If this value is exceeded, conventional ray tracing is computed for the current location in the grid. Newly obtained escape values are then locked in for future sweeping iterations.

Such a relatively simple modification allows to greatly improve the accuracy of escape computations. Figures 3.26, 3.27, and 3.28 show escape quantities $\widehat{z}$, $\widehat{x}$, and $\widehat{T}$ respectively computed using this new hybrid approach. The threshold for switching to ray tracing is set to 1.0 km in this test.

These new results demonstrate much better preservation of fine features in the escape solution. Extracted arrivals from the hybrid solution (Figure 3.30) match the ray-traced ones (Figure 3.31) well. Approximately 4% of all points in the solution need to be traced to achieve this accuracy.

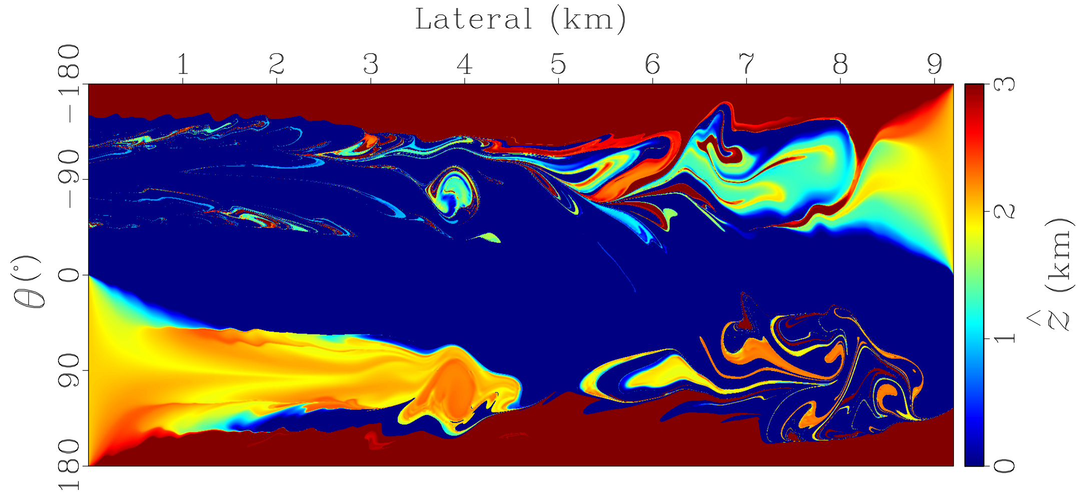


Figure 3.26: Escape depth function $\widehat{z}$ at constant depth $z = 2.0$ km of Marmousi model (hybrid iterative solver). chapter-fdiff/marmousi marmhsz0

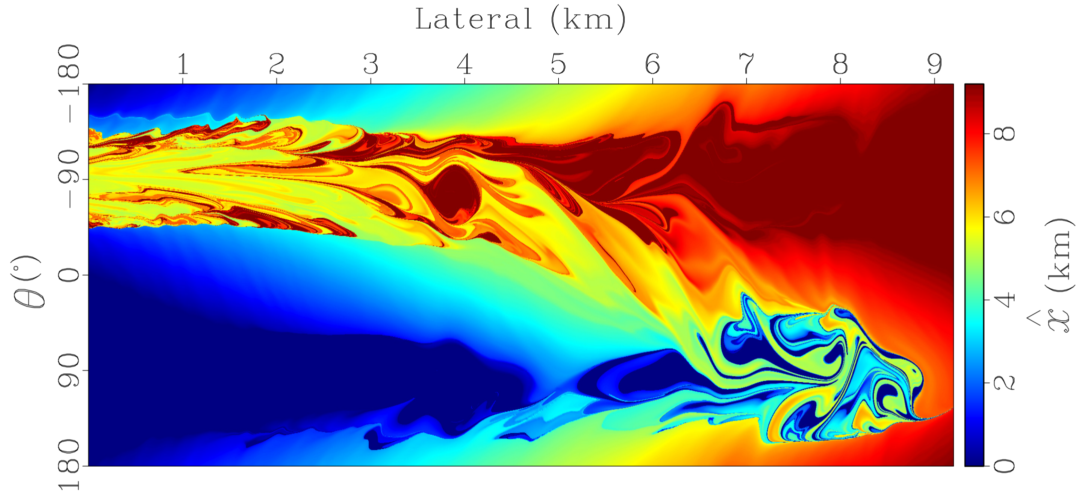


Figure 3.27: Escape depth function $\widehat{x}$ at constant depth $z = 2.0$ km of Marmousi model (hybrid iterative solver). chapter-fdiff/marmousi marmhsz1

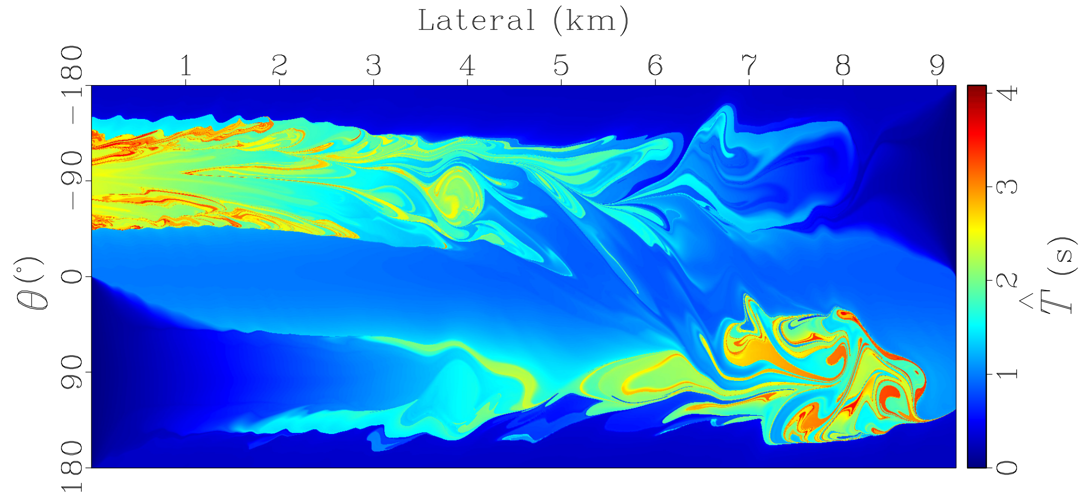


Figure 3.28: Escape depth function $\widehat{T}$ at constant depth $z = 2.0$ km of Marmousi model (hybrid iterative solver). chapter-fdiff/marmousi marmhsz2

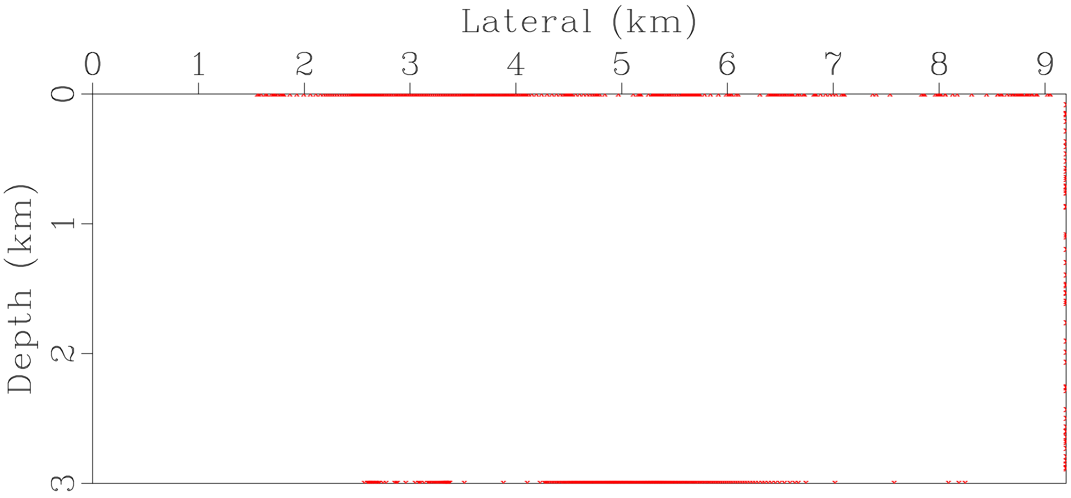


Figure 3.29: Exit locations for all arrivals originating from point x = 5.0 km and z = 2.0 km of Marmousi model (hybrid iterative solver). chapter-fdiff/marmousi marmpescs3

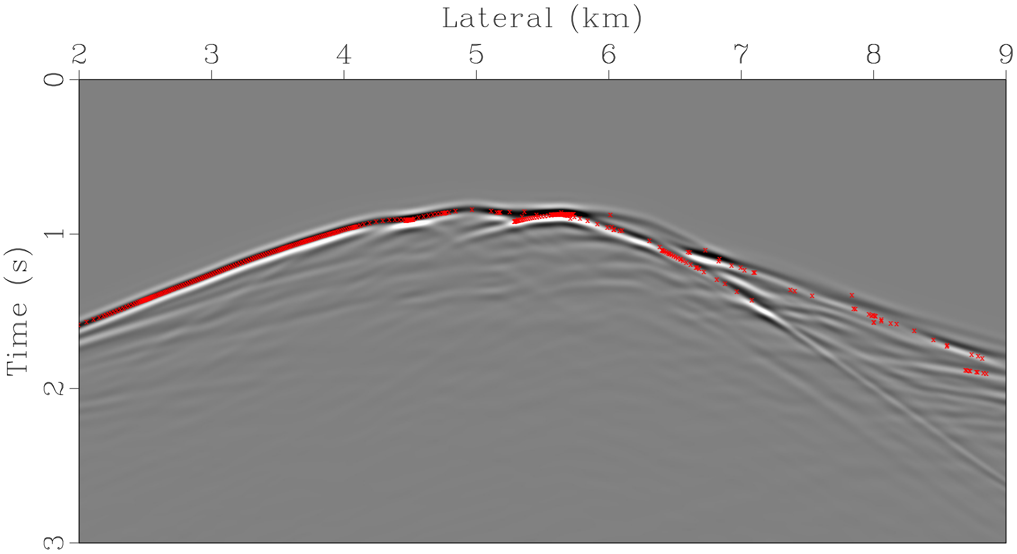


Figure 3.30: Surface exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (hybrid iterative solver) plotted over wavefield computed for a source at the same point. chapter-fdiff/marmousi marmfdescs3

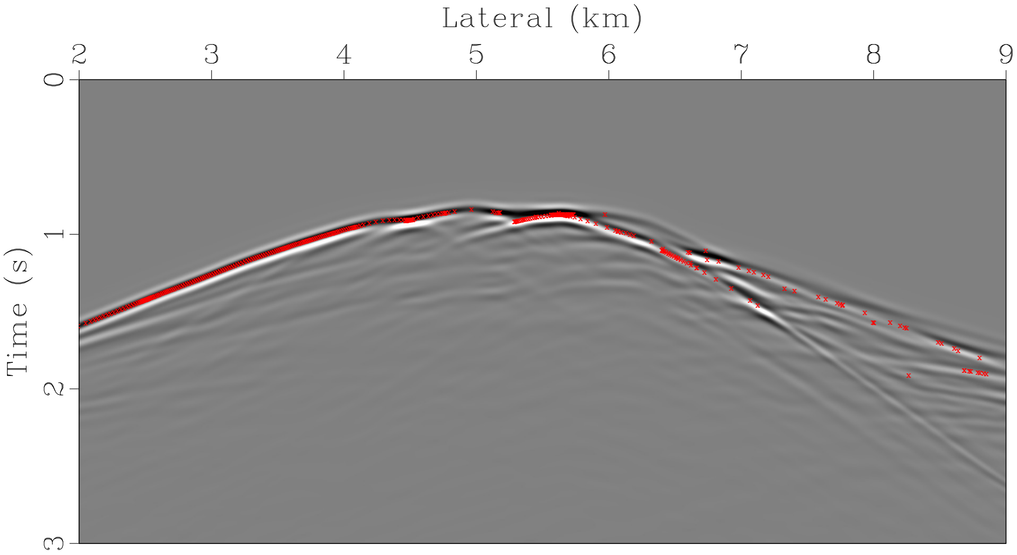


Figure 3.31: Surface exit locations for all arrivals originating from point $x = 5.0$ km and $z = 2.0$ km of Marmousi model (ray tracing) plotted over wavefield computed for a source at the same point. chapter-fdiff/marmousi marmfdescs0

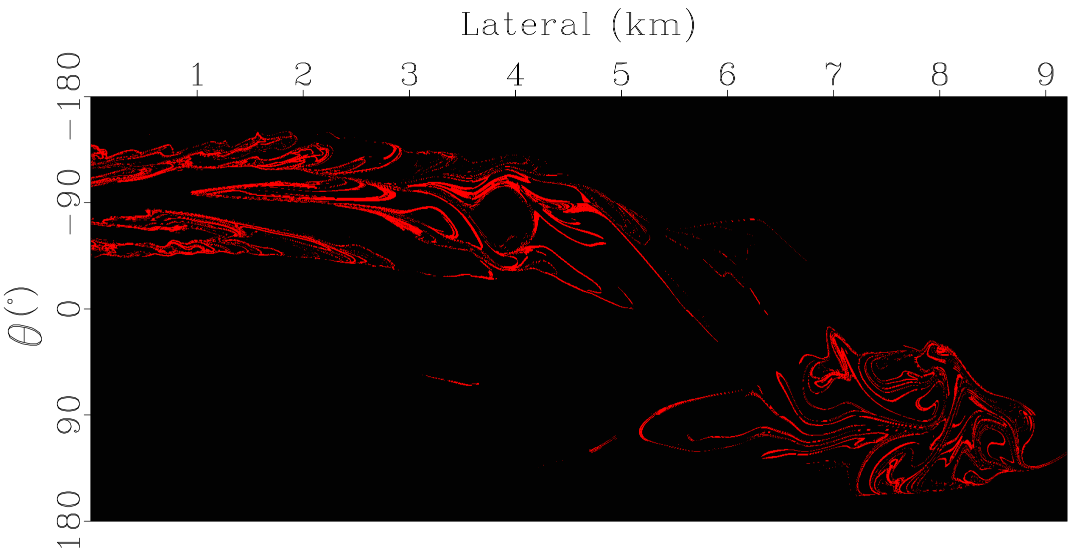


Figure 3.32: Map of ray-traced points (red) of the hybrid escape solution at constant depth $z = 2.0$ km of Marmousi model. chapter-fdiff/marmousi marmbrsz

Computational cost of the Eulerian-Lagrangian approach

The computation cost of the proposed Eulerian-Lagrangian scheme can be expressed as

$$\mathcal{O}_{\mathbf{EL}} \sim \mathcal{O}\left(N_x N_z N_\theta\right) \cdot \mathcal{O}_{\mathbf{GS}} + \mathcal{O}\left(N_r\right) \cdot \mathcal{O}_{\mathbf{RT}}\,, \tag{3.5}$$

where $N_x N_z$ is the number of points in the spatial imaging grid, N_θ is the number of phase directions, $\mathcal{O}_{\mathbf{GS}}$ is the cost of Gauss-Seidel iterations, N_r is the number of points that need to be traced, and $\mathcal{O}_{\mathbf{RT}}$ is the computational cost of tracing one ray trajectory. N_r is much smaller than the $N_x N_z N_\theta$ product, but the term $\mathcal{O}_{\mathbf{RT}}$ represents an expensive computational procedure. The test above shows that even when N_r constitutes only 4% of $N_x N_z N_\theta$ points, however, total compute time increases approximately 5 times. On top of that, it is impossible to predict which points have to be ray traced beforehand, because the structure of the escape solution is not known in advance. This makes load balancing on parallel computer architectures non-trivial. These limitations make the hybrid Eulerian-Lagrangian approach less advantageous than conventional ray tracing. It may be impractical to apply this scheme to computation of large-scale escape tables. Nevertheless, this algorithm might still be useful as a complimentary procedure to massive ray tracing. A somewhat common practice for computing "bottom-up" escape tables is to perform ray tracing on a very sparse spatial grid and then interpolate the tables to a finer grid (Ettrich et al., 2008). Instead of using a usual interpolation routine, the hybrid algorithm can be applied as a more accurate, physics-guided interpolator.

Semi-Lagrangian approach

I showed in this chapter that an Eulerian solver for escape equations can be built with the help of finite-differences. On a regularly-sampled phase space grid, it is computationally robust, but it does not appear to provide sufficient accuracy for imaging purposes. Adaptive mesh refinement can be used to preserve accuracy in the solution. However, the necessary degree of refinement might be too excessive for a typical migration task. A different way of preserving accuracy consists of providing Lagrangian treatment only to a few difficult places in the escape solution. This approach allows for a relatively simple implementation compared to AMR without having to obtain parts of the solution at very fine scales. Unfortunately, this scheme involves a significant computational penalty.

Another way of solving advection problems is known as the semi-Lagrangian method (Durran, 2010). In that framework, points in the solution are obtained with Lagrangian-type tracking of characteristic lines. But instead of performing it with small steps, large steps are allowed during computations. That is possible with the help of additional information which is either precomputed in advance or available from previously computed points. Large steps potentially enable sizable computational gains compared to the traditional Lagrangian approach.

In the next chapter, I propose a design of a semi-Lagrangian numerical scheme for computing 5-D escape tables. I show that a set of local escape functions can be defined for different subregions of a 3-D velocity model. These functions describe local displacements, rotations, and traveltime accumulations for any initial position and phase direction. The global escape functions then can be computed through iterative evaluation of the local escape functions in the reduced phase space.

Chapter 4

Semi-Lagrangian method for computing escape functions

Global and local escape functions

In the previous chapter, I showed that direct finite-difference solutions of escape equations may not provide sufficient accuracy in escape tables computations for complex velocity models. Different means of increasing the accuracy lead either to a significant loss of computational efficiency or to implementation requirements which cannot be considered practical. Nevertheless, escape equations establish a convenient framework for analyzing escape functions. These functions posses an important property - they are continuous in the phase space. In this chapter, I construct a different, semi-Lagrangian, algorithm for computing escape functions that utilizes this property at scales which are smaller than the typical velocity-model scale.

The continuity of escape functions may not be immediately apparent in complex velocity models. Figure 2.16 shows the escape traveltime $\widehat{T}$ for one depth point in SEG/EAGE model. The values have been obtained with ray tracing; the angular sampling is $\Delta\theta = \Delta\phi = 1^\circ$. Each ray was allowed to reach the boundary of the model, and its total traveltime contributed to $\widehat{T}$. We can observe rapid changes in the values of the escape traveltime, specifically in the areas corresponding to ray paths going through the salt body (Figure 4.1). High velocity contrasts associated with the edges of the salt body greatly diverge rays and cause ray paths with close initial conditions to differ from each other significantly (Figure 4.2). The escape functions at this scale look as if they are discontinuous. Their continuity becomes only apparent at much smaller scales as I demonstrated it in the previous chapter.

However, if we limit the maximum allowed travel distance for rays to some value d_{max} that is several times smaller than the average length of rays, then the corresponding localized escape traveltime starts to look much smoother (Figures 4.3,4.4,4.5). Similar changes can be demonstrated for other components of the escape function, i.e. escape location and escape phase.

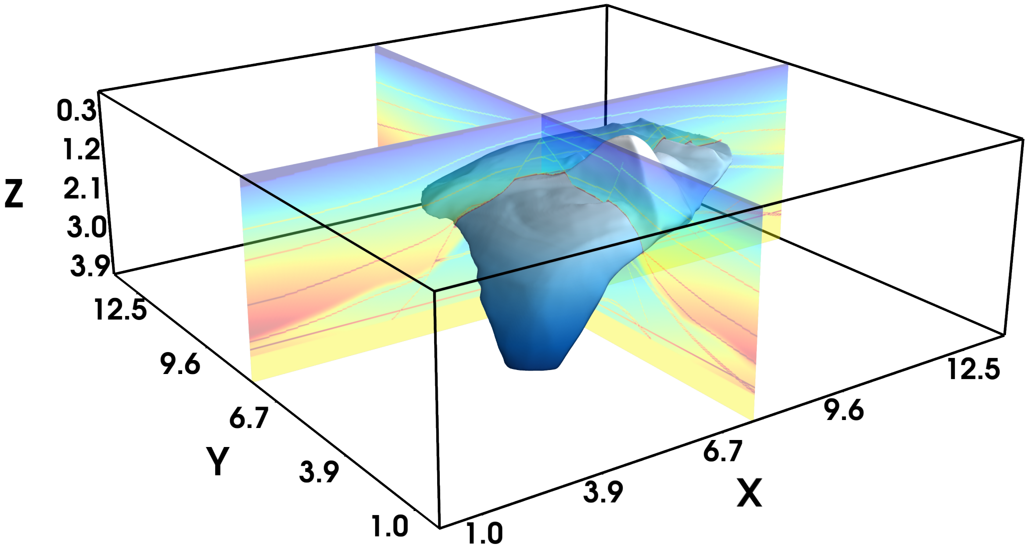


Figure 4.1: Salt body in SEG/EAGE Salt model. chapter-psint/. ssaltbod

Suppose that such local escape functions are known everywhere in the spatial area of interest. It should be possible then to construct a global escape solution for some initial conditions by traversing the phase space according to these local values. This observation leads to a numerical method described in this chapter.

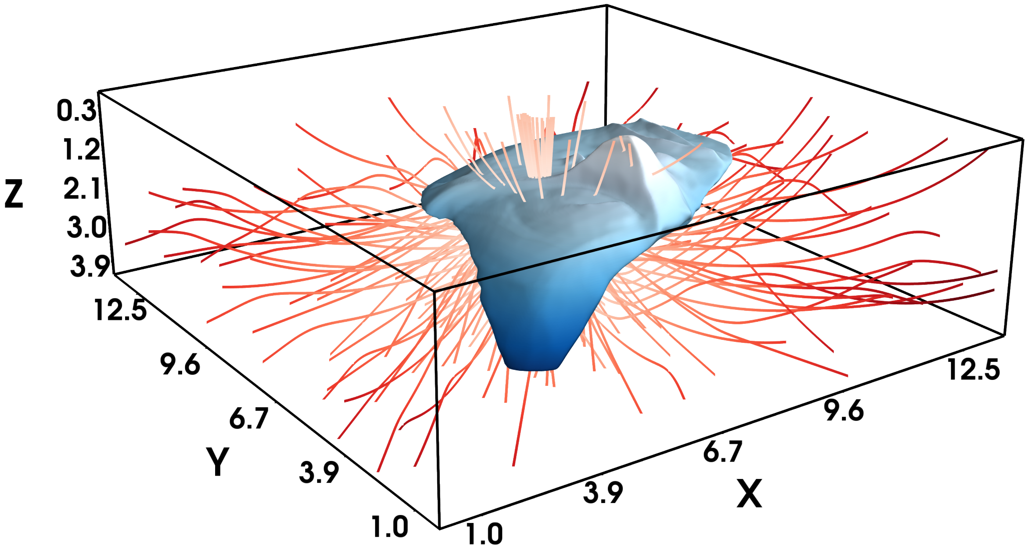


Figure 4.2: Ray trajectories computed for the location $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. chapter-psint/. ssaltbdrs

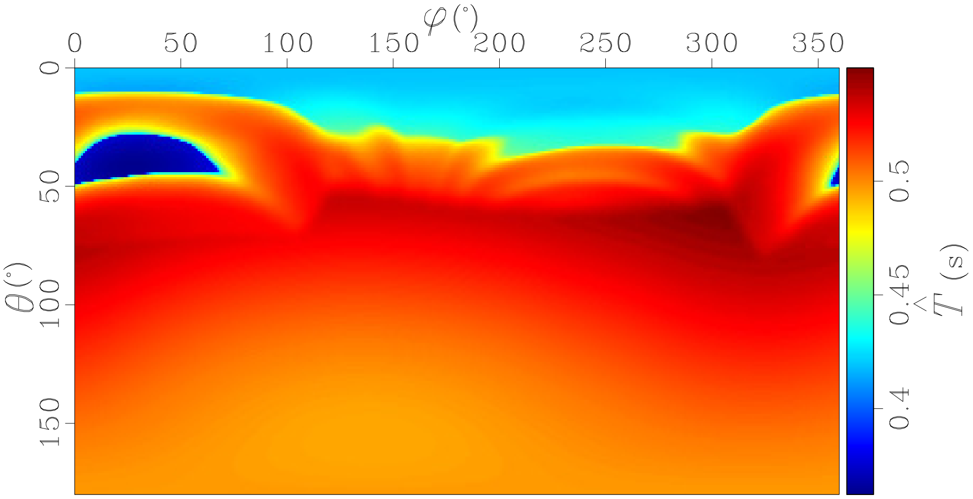


Figure 4.3: Local escape traveltime $\widehat{T}$ computed for $d_{max} = 1.0$ km at the location $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model. chapter-psint/segsalt ssaltnescmd1

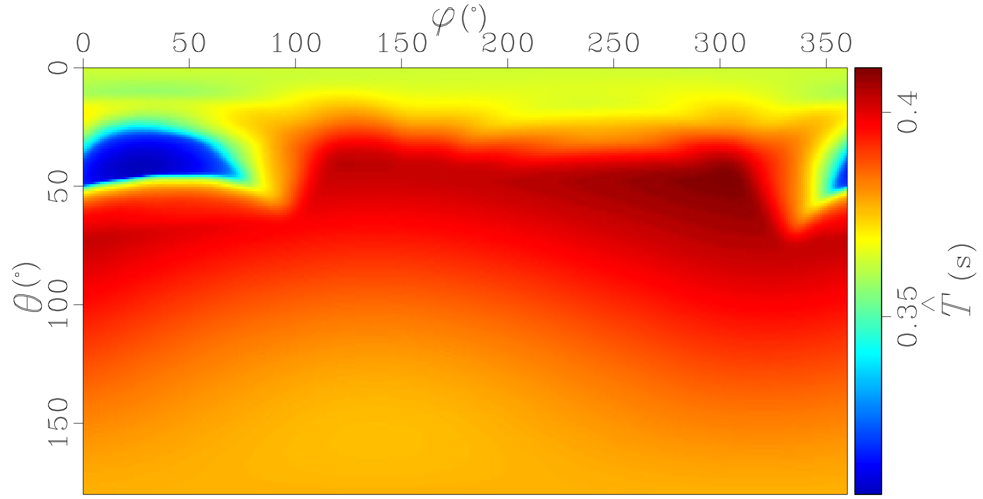


Figure 4.4: Local escape traveltime $\widehat{T}$ computed for $d_{max} = 0.75$ km at the location $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km. of SEG/EAGE Salt model. chapter-psint/segsalt ssaltnescmd2

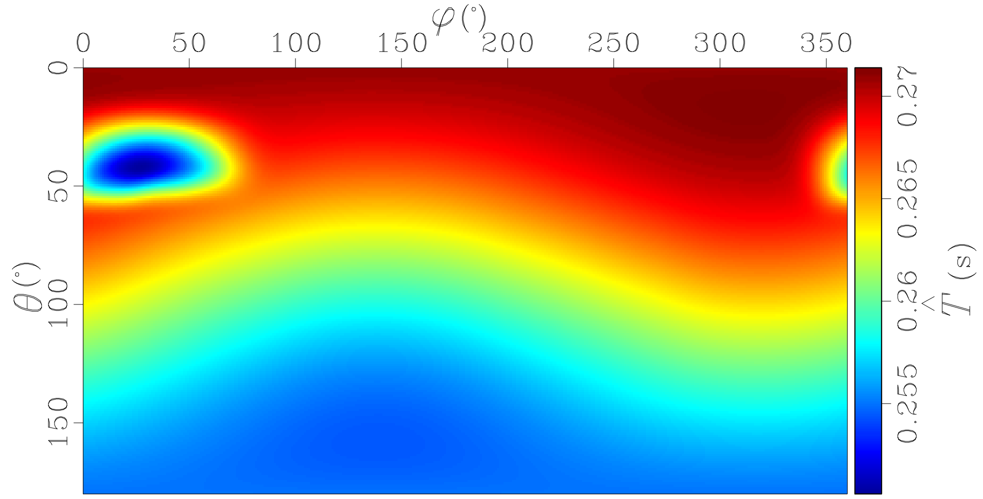


Figure 4.5: Local escape traveltime $\widehat{T}$ computed for $d_{max} = 0.5$ km at the location $x = 6.7$ km, $y = 6.7$ km, and $z = 2.5$ km. of SEG/EAGE Salt model. chapter-psint/segsalt ssaltnescmd3

Reconstruction of global escape solution from local escape functions

Let us denote the change along the characteristic line (seismic ray) in the phase space from some position $\mathbf{x}$ to $\widehat{\mathbf{x}}$ as

$$\mathbf{x}_{\Delta} = \widehat{\mathbf{x}} - \mathbf{x}\,, \;\; |\widehat{\mathbf{x}} - \mathbf{x}| = d_{max}\,. \tag{4.1}$$

The corresponding traveltime from $\mathbf{x}$ to $\widehat{\mathbf{x}}$ is $T_{\Delta} = \widehat{T} - T$, and the change of the phase from $\mathbf{p}$ to $\widehat{\mathbf{p}}$ along the same characteristic line between the same points can be written as the following product

$$\widehat{\mathbf{p}} = \mathbf{q}\,\mathbf{p}\,\mathbf{q}^{*}\,, \tag{4.2}$$

where $\mathbf{q}$ is a quaternion (Cayley, 1845; Shoemake, 1985) defined as

$$\mathbf{q} = \cos\frac{\psi}{2} + \sin\frac{\psi}{2}\left(l_x\mathbf{i} + l_y\mathbf{j} + l_z\mathbf{k}\right)\,, \tag{4.3}$$

where ψ is the angle of rotation defined as

$$\cos\psi = \frac{\mathbf{p}\cdot\widehat{\mathbf{p}}}{|\mathbf{p}|\,|\widehat{\mathbf{p}}|}\,,$$

around the axis of rotation $\mathbf{l}$ (Figure 4.6)

$$\mathbf{l} = \frac{\mathbf{p}\times\widehat{\mathbf{p}}}{|\mathbf{p}\times\widehat{\mathbf{p}}|}\,,$$

and $\mathbf{q}^{*}$ is the complex conjugate of $\mathbf{q}$.

Here, the quantities $\widehat{\mathbf{x}}$, $\widehat{\mathbf{p}}$, and $\widehat{T}$ are the local escape values for the given initial conditions $\mathbf{x}$, $\mathbf{p}$, and T. These values are determined by the local escape function, which contains information about changes in position, traveltime, and phase – $\mathbf{x}_{\Delta}$, T_{Δ} and $\mathbf{q}$ respectively.

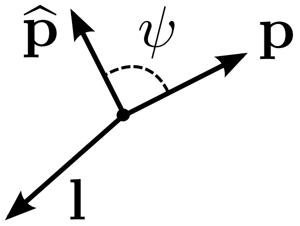


Figure 4.6: Change of phase $\mathbf{p}$ to $\widehat{\mathbf{p}}$ defined as a rotation by angle ψ around vector $\mathbf{l}$ (a scheme). chapter-psint/. quat

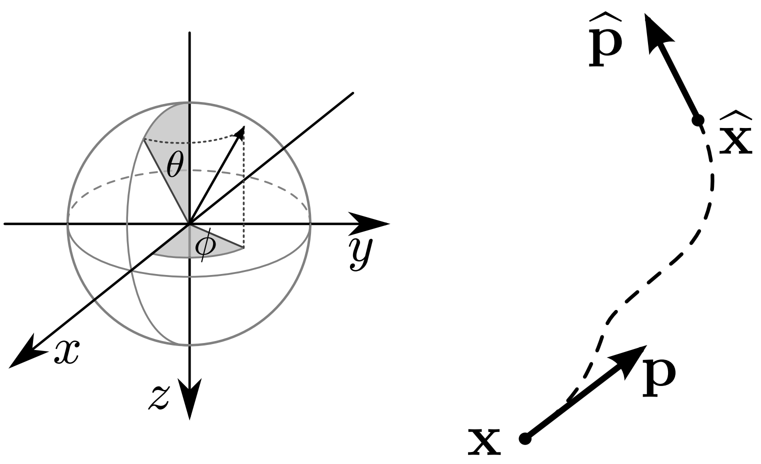


Figure 4.7: Change of position $\mathbf{x}$ and phase $\mathbf{p}$ according to local values of functions $\mathbf{x}_{\Delta}$ and $\mathbf{q}$ (a scheme). chapter-psint/. gfunc

Suppose that we know $\mathbf{x}_\Delta$, T_Δ and $\mathbf{q}$ everywhere in the phase space for all possible combinations of $\mathbf{x}$ and $\mathbf{p}$

$$\begin{aligned} \mathbf{x}_\Delta &= g_\Delta\left(\mathbf{x}, \mathbf{p}\right) \ , \\ T_\Delta &= g_T\left(\mathbf{x}, \mathbf{p}\right) \ , \\ \mathbf{q} &= g_q\left(\mathbf{x}, \mathbf{p}\right) \ . \end{aligned}$$

Then, function $g\left(\mathbf{x}, \mathbf{p}\right)$ defines local escape functions $g_\Delta\left(\mathbf{x}, \mathbf{p}\right)$, $g_T\left(\mathbf{x}, \mathbf{p}\right)$, and $g_q\left(\mathbf{x}, \mathbf{p}\right)$ everywhere in the phase space.

While the global escape functions produce exit location, time, and other quantities on the global model boundary for a combination of the initial position and the phase, the local ones generate only escape quantities constrained to some local subvolumes of the model. In order to find the global escape values for given initial conditions, the local escape function has to be evaluated consecutively. The found escape quantities from the previous estimation should be used as input for the next step. The process can be repeated until the boundary is reached.

Thus, the process of finding the global escape solution for a set of initial conditions $\widehat{\mathbf{x}}_0 = \mathbf{x}$, $\widehat{T}_0 = 0$ and $\widehat{\mathbf{p}}_0 = \widehat{\mathbf{p}}$ can be described as the following iterative procedure

$$\begin{aligned} \widehat{\mathbf{x}}_n &= \widehat{\mathbf{x}}_{n-1} + g_\Delta\left(\widehat{\mathbf{x}}_{n-1}, \widehat{\mathbf{p}}_{n-1}\right) \\ \widehat{T}_n &= \widehat{T}_{n-1} + g_T\left(\widehat{\mathbf{x}}_{n-1}, \widehat{\mathbf{p}}_{n-1}\right) \\ \mathbf{q}_n &= g_q\left(\widehat{\mathbf{x}}_{n-1}, \widehat{\mathbf{p}}_{n-1}\right) \\ \widehat{\mathbf{p}}_n &= \mathbf{q}_n \, \widehat{\mathbf{p}}_{n-1} \, \mathbf{q}_n^* \end{aligned} \tag{4.4}$$

which stops when $\widehat{\mathbf{x}}_n = \widehat{\mathbf{y}}$.

The whole computational scheme to find a global escape solution thus consists of the two major steps:

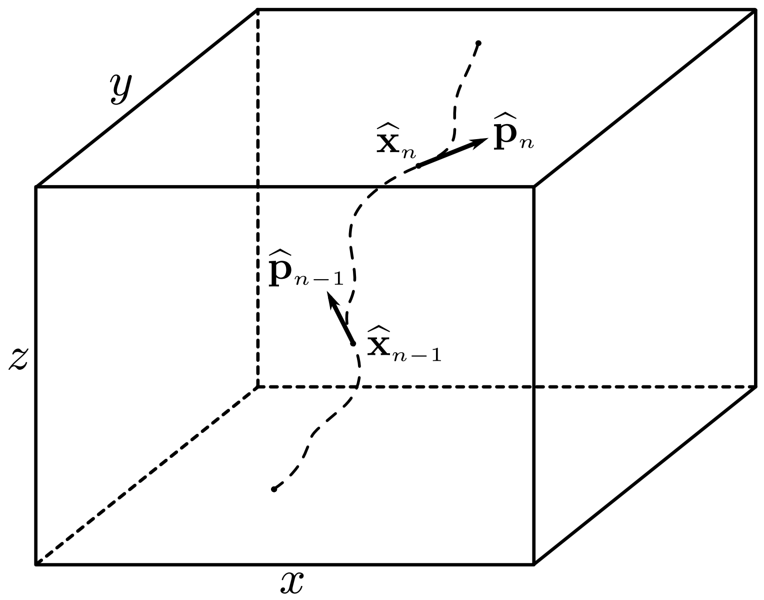


Figure 4.8: Iterative reconstruction of escape quantities (a scheme). chapter-psint/. gstepn

1. Initialization of function $g(\mathbf{x}, \mathbf{p})$.
2. Estimation of the global escape quantities by iterative reconstruction of their flow in the phase space with the help of $g(\mathbf{x}, \mathbf{p})$.

Computational cost

The computational cost of the proposed numerical method is the sum of two parts

$$\mathcal{O}_{\mathbf{Init}} + \mathcal{O}_{\mathbf{Rec}}\,, \tag{4.5}$$

where the first term is the cost of the initialization procedure and can be written as

$$\mathcal{O}_{\mathbf{Init}} \sim \mathcal{O}\left(N_x^s N_y^s N_z^s N_\theta^s N_\phi^s\right) \cdot \mathcal{O}_{\mathbf{Loc}}\,, \tag{4.6}$$

where $\mathcal{O}_{\mathbf{Loc}}$ is the cost of finding one set of values of $g(\mathbf{x}, \mathbf{p})$.

The second term in expression (4.5) is the cost of iterative reconstruction procedure in the phase space for a full escape table and can be expressed as

$$\mathcal{O}_{\mathbf{Rec}} \sim \mathcal{O}\left(N_x N_y N_z N_\theta N_\phi\right) \cdot \mathcal{O}_{\mathbf{IPS}}\,, \tag{4.7}$$

and $\mathcal{O}_{\mathbf{IPS}}$ is the cost of iterative reconstruction of one set of $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{p}}$ values. This cost can approximated as

$$\mathcal{O}_{\mathbf{IPS}} \sim \mathcal{O}\left(\sqrt[3]{N_x^s N_y^s N_z^s}\right) \cdot \mathcal{O}_{\mathbf{LPS}}\,, \tag{4.8}$$

where $\mathcal{O}_{\mathbf{LPS}}$ is the cost of finding one set of local escape values values $\widehat{\mathbf{x}}_n$ and $\widehat{\mathbf{p}}_n$ for the given pair of phase space position $\widehat{\mathbf{x}}_{n-1}$ and $\widehat{\mathbf{p}}_{n-1}$.

The total cost in equation (4.5) is primarily driven by the second term. It can be significantly smaller than the cost in equation (1.6), thus allowing for a faster implementation of Kirchhoff common-angle migration defined in equation (1.2), which is the main objective of this study.

Implementation details

In my implementation, I compute $g\left(\mathbf{x},\mathbf{p}\right)$ on a relatively sparse spatial grid $N_x^s N_y^s N_z^s$ and a sparse angular grid $N_\theta^s N_\phi^s$. The grid sampling is related to the choice of the maximum allowed local change in position d_{max} along characteristic lines. d_{max} is set to be significantly larger than the sampling of the imaging grid to allow sizable steps in the reconstruction process. It is a dominant parameter that depends on the complexity of the velocity model: the higher the gradients in the velocity are, the shorter the distance d_{max} has to be so as to avoid big interpolation errors in the reconstruction process later. This distance, in turn, controls the spacing of function $g\left(\mathbf{x},\mathbf{p}\right)$: the longer d_{max} is, the finer both spatial and angular spacings need to be.

I introduce some empirical rules for choosing the optimal set of these parameters in Chapter 6, where I compare different values for them in imaging accuracy tests.

Local escape function $g(\mathbf{x}, \mathbf{p})$ can be computed by either ray tracing or the finite-difference method described in the previous chapter. This choice is not particularly critical, because the initialization step takes relatively little compute time compared to the global escape values reconstruction stage. For practical purposes, function $g(\mathbf{x}, \mathbf{p})$ becomes a 5-D function in reduced phase space, $g_r(\mathbf{x}, \theta, \phi)$.

Function $g_r(\mathbf{x}, \theta, \phi)$ can be represented by a set of constant-inclination, constant-azimuth 3-D volumes, $g_{r,\theta,\phi}(\mathbf{x})$. The set is distributed across the 2-D inclination-azimuth $(\theta - \phi)$ angular grid. Each volume $g_{r,\theta,\phi}(\mathbf{x})$ is located at a specific node on that grid. I compute functions $g_{r,\theta,\phi}(\mathbf{x})$ independently from one another and store their values in the form of B-spline coefficients (de Boor, 1978). This particular approach will become more apparent below, where I explain the second stage of the algorithm. Note that the first stage can be trivially parallelized: each 3-D volume $g_{r,\theta,\phi}(\mathbf{x})$ can be computed by a separate node on a computer cluster.

The second stage, the iterative reconstruction, then becomes a 5-D interpolation in the reduced phase space. I perform this interpolation as a two-step process. At the second step of the process, the value of $g(\mathbf{x}, \mathbf{p})$ is computed on the angular grid θ-ϕ by means of thin-plate spline interpolation (Duchon, 1977)

$$g(\mathbf{x}, \mathbf{p}) = b_0 + b_\theta \theta + b_\phi \phi + \sum_{i=1}^{m} a_i R(\boldsymbol{\alpha}, \boldsymbol{\alpha}_i) \ , \ \boldsymbol{\alpha} = \{\theta, \phi\} \tag{4.9}$$

where b_0, b_θ, b_ϕ, and a_i are interpolation coefficients and R is the radial-basis function (RBF) (Buhmann, 2003) defined as

$$R(\boldsymbol{\alpha}, \boldsymbol{\alpha}_i) = \begin{cases} r^2 \ln r & , r \neq 0 \\ 0 & , r = 0 \end{cases} \ , \ \ r = |\boldsymbol{\alpha} - \boldsymbol{\alpha}_i| \ . \tag{4.10}$$

The interpolation coefficients are determined at the first stage from a set of points which define the RBF interpolation stencil on the angular grid. These points surround the location where the values of $g_r(\mathbf{x}, \theta, \phi)$ are to be found (Figure 4.9). The values of each of the points in the stencil are computed with 3-D cubic B-spline interpolation in constant-θ-ϕ subvolumes $g_{r,\theta,\phi}(\mathbf{x})$.

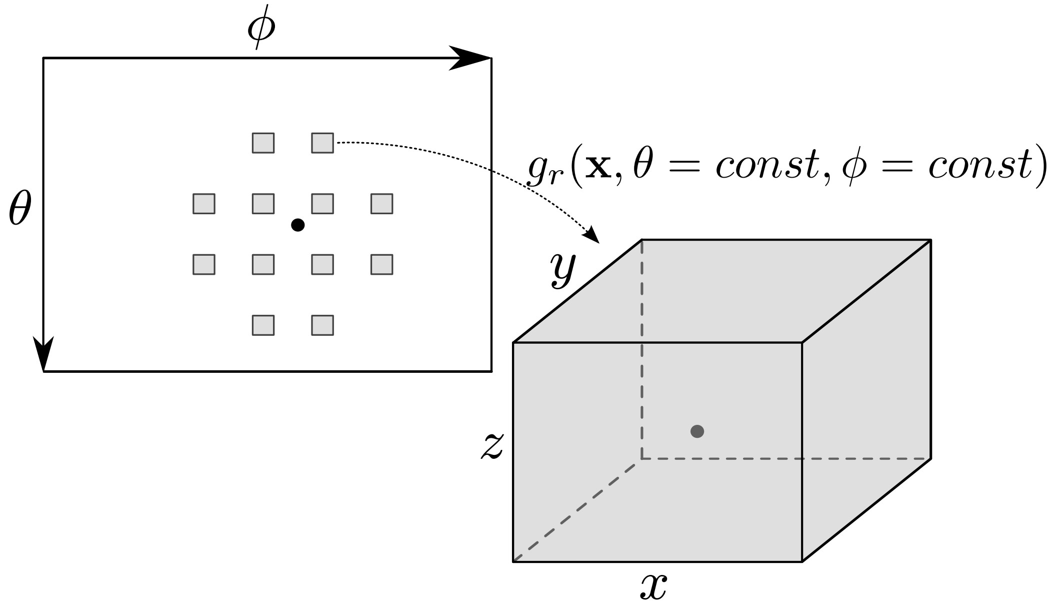


Figure 4.9: 5-D interpolation of the local escape values (a scheme). chapter-psint/. int5d

For M-point interpolation stencil, RBF coefficients for equation (4.9) are found as follows

$$\mathbf{c} = \bar{\mathbf{T}}^{-1}\mathbf{f}\,, \tag{4.11}$$

where $\mathbf{c}$ and $\mathbf{f}$ are the vector of interpolation coefficients and the vector of local escape function values respectively. They can be written as

$$\mathbf{f} = \begin{vmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ f_M \\ 0 \\ 0 \\ 0 \end{vmatrix}, \qquad \mathbf{c} = \begin{vmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_M \\ b_0 \\ b_\theta \\ b_\phi \end{vmatrix}. \tag{4.12}$$

Value f_i is one of the components of function $g_{r,\theta,\phi}(\mathbf{x})$ extracted at i-th position of the interpolation stencil.

The RBF matrix that connects function values and interpolation coefficients is

$$\bar{\mathbf{T}} = \begin{vmatrix} R_{11} & R_{12} & \cdot & \cdot & \cdot & R_{1M} & 1 & \theta_1 & \phi_1 \\ R_{21} & R_{22} & \cdot & \cdot & \cdot & R_{2M} & 1 & \theta_2 & \phi_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{M1} & R_{M2} & \cdot & \cdot & \cdot & R_{MM} & 1 & \theta_M & \phi_M \\ 1 & 1 & \cdot & \cdot & \cdot & 1 & 0 & 0 & 0 \\ \theta_1 & \theta_2 & \cdot & \cdot & \cdot & \theta_M & 0 & 0 & 0 \\ \phi_1 & \phi_2 & \cdot & \cdot & \cdot & \phi_M & 0 & 0 & 0 \end{vmatrix}, \tag{4.13}$$

where

$$R_{ij} = R\left(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j\right) .$$

The choice of RBF for 5-D interpolation enables certain flexibility in the reconstruction process: the stencil does not have to be rectangular and can potentially vary as the computation progresses. This two-stage interpolation process enables parallelization of the algorithm on computer clusters.

One attractive property of this algorithm is that all effects of seismic anisotropy are already precomputed in function $g_r(\mathbf{x}, \theta, \phi)$, which means that compute time of the interpolation stage does not depend on the type of background model. On the contrary, conventional ray tracing requires computation of the phase slowness and its derivatives in the right hand side of equations (1.3)-(1.5). For a general anisotropic model, phase slowness is a 5-D function, therefore, it is impractical to precompute it in advance. Instead, it is usually approximated during ray tracing from a number of 3-D model components, which provide sufficient description for background media. As a result, the total compute time for the same number of imaging points depends significantly on the type of the model and increases by a large factor when switching from isotropic to anisotropic ray tracing.

Implementation for parallel and distributed architectures

Much like the conventional ray tracing, the proposed approach exposes so called embarrassingly parallel logic at the most outer layers which deal with the collection of depth points that comprise the escape tables. Each point in that set can be computed independently from others according to the scheme described above.

The sequential, innermost part of the algorithm has the following steps in its core that compute one arrival for a fixed depth point and a given set initial phase angles:

1. **Form** stencil: build a list of points that belong to the RBF interpolation stencil.
2. **Extract** escape values: obtain local escape values needed by the stencil.
3. **Interpolate** escape function: form vectors $\mathbf{f}$, find RBF coefficients $\mathbf{c}$, perform interpolation and find $\mathbf{x}_\Delta$, T_Δ, and $\mathbf{q}$.
4. **Advance** escape solution: update current values of $\widehat{\mathbf{x}}_{n-1}$, $\widehat{\mathbf{p}}_{n-1}$, and $\widehat{T}_{n-1}$ to

$\widehat{\mathbf{x}}_n$, $\widehat{\mathbf{p}}_n$, and $\widehat{T}_n$ according to the scheme (4.4).

5. **Check** for completion: if $\widehat{\mathbf{x}}_n$ belongs to the model boundary, then stop; otherwise, go to step 1.

For the salt model shown above and the typical algorithm parameterization needed for high-quality imaging, the full size of function $g_r(\mathbf{x}, \theta, \phi)$ is on the order of 300 Gb. However, in a usual imaging problem, the migration aperture is spatially limited. Therefore, only a fraction of the local escape function has to be loaded into computer memory at any given time. However, if a bigger volume of the local escape function has to be used, then a distributed version of the algorithm might be needed.

In a distributed version, the local escape values are partitioned across computer cluster nodes (Figure 4.10). Every node holds a number of constant-inclination constant-azimuth 3-D parts of the local escape function. The algorithm above is then modified at step 2: the local extraction becomes a remote request-reply operation. The workload is divided between multiple server and client sides. Server processes simply wait for incoming requests to extract local values and return results to clients. Client processes traverse the phase space to obtain the final global escape solution.

Sending one request at a time is inefficient in practice, however. The request is represented by a small data structure consisting of the spatial coordinates, phase angles and a unique message identification tag. The reply is similar in length - it contains the message tag and a set of local escape values. Due to typical network latency and hardware overhead, such one-at-a-time exchange takes much more time than the direct extraction from local computer memory. To overcome this problem, I adopted the usual remedy - grouping multiple requests into one message. The client

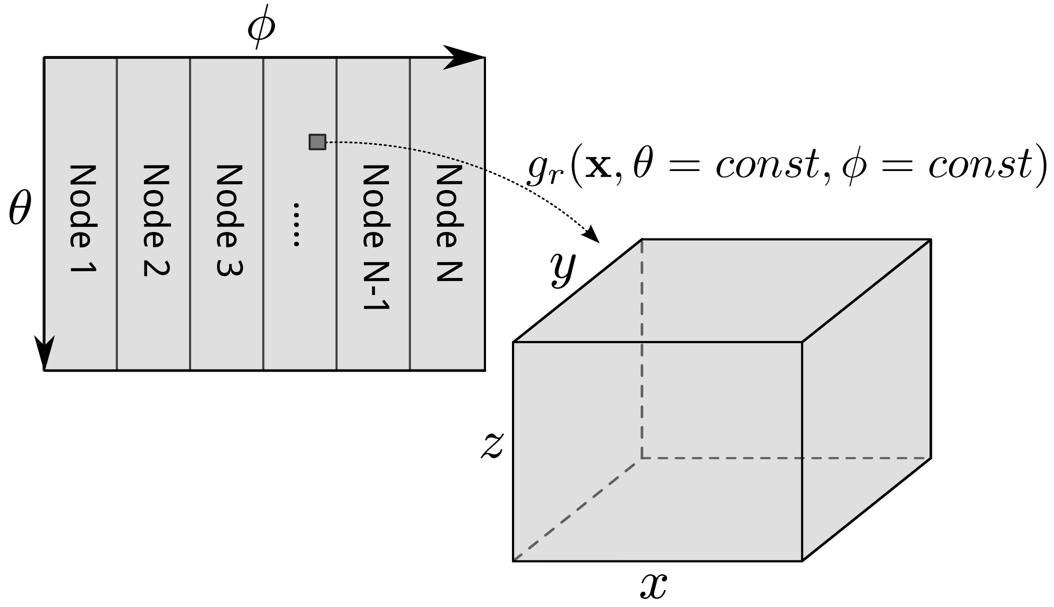


Figure 4.10: Distribution of the local escape functions across computer cluster nodes (a scheme). chapter-psint/. angrem

side then works with a queue of points instead of a single point. The modified core contains the following sequence:

1. **Add** new points: fill the queue with arrivals to be computed.
2. **Form** stencils: build a list of points that belong to all interpolation stencils in the queue, construct network request data structures for them.
3. **Sort** request: sort the list of requests by phase angles so that ones designated to the same server are continuous in memory and can be sent as one message.
4. **Send** requests: send request messages to remote server.
5. **Receive** replies: wait for the servers to return replies that contain local escape values.

6. **Reverse sort** replies: sort the list of replies so the new order matches that of the original list of stencil points.

7. **Interpolate** escape function: form vectors $\mathbf{f}$, find RBF coefficients $\mathbf{c}$, perform interpolation and find $\mathbf{x}_\Delta$, T_Δ, and $\mathbf{q}$ for all points in the queue.

8. **Advance** escape solution: update current values of $\widehat{\mathbf{x}}_{n-1}$, $\widehat{\mathbf{p}}_{n-1}$, and $\widehat{T}_{n-1}$ to $\widehat{\mathbf{x}}_n$, $\widehat{\mathbf{p}}_n$, and $\widehat{T}_n$ according to the scheme (4.4) for all points in the queue.

9. **Check** for completion: for every point in the queue, if its $\widehat{\mathbf{x}}_n$ belongs to the model boundary, then remove it from the queue; if there are uncomputed arrivals for the current depth point, go to step 1.

The above version is significantly more efficient, but it may still waste time on waiting for servers to process requests and return results. To hide this latency, I overlap computations with network exchange and construct a fully asynchronous version of the distributed algorithm. Instead of one queue, I use two queues. Using the notation from above, I build the following sequence of computational and communicative steps for the following version of the algorithm implementation:

1. Queue #1: **Add**, **Form**, **Sort**, **Send**.

2. Queue #2 (if not empty): **Receive**, **Reverse sort**, **Interpolate**, **Advance**, **Check**.

3. Queue #2: **Add**, **Form**, **Sort**, **Send**.

4. Queue #1: **Receive**, **Reverse sort**, **Interpolate**, **Advance**, **Check**.

5. Go to step 1, if there are uncomputed arrivals.

My test implementation uses Sockets Direct Protocol (SDP), which allows to utilize the advantages of Inifiniband network interconnect (Goldenberg et al., 2005) while providing natural semantics for client-server type data exchange and enabling fault tolerance. My tests show that the communication overhead in the distributed version increases the compute time by a factor of 2-2.5. Network fabric other than Infiniband may introduce a greater overhead.

Arrivals matching test

Similarly to the arrivals matching test described in the previous chapter, I performed comparisons for SEG/EAGE Salt model (Figures 4.11 and 4.12). The spatial sampling of $g_r\left(\mathbf{x},\theta,\phi\right)$ is 60 m in every of the three dimensions, the angular sampling is 5° in azimuth and inclination. d_{max} is 0.5 km. There is some discrepancy on the right side of the wavefield snapshot. Nevertheless, both tests cover all of the major arrivals to some extent. Interestingly, the iterative reconstruction produces more even coverage of the left part of the most energetic arrival. This effect can be explained by smoothing created implicitly during the iterative process. In Chapter 7, I compare both approaches by imaging a target line from the same model.

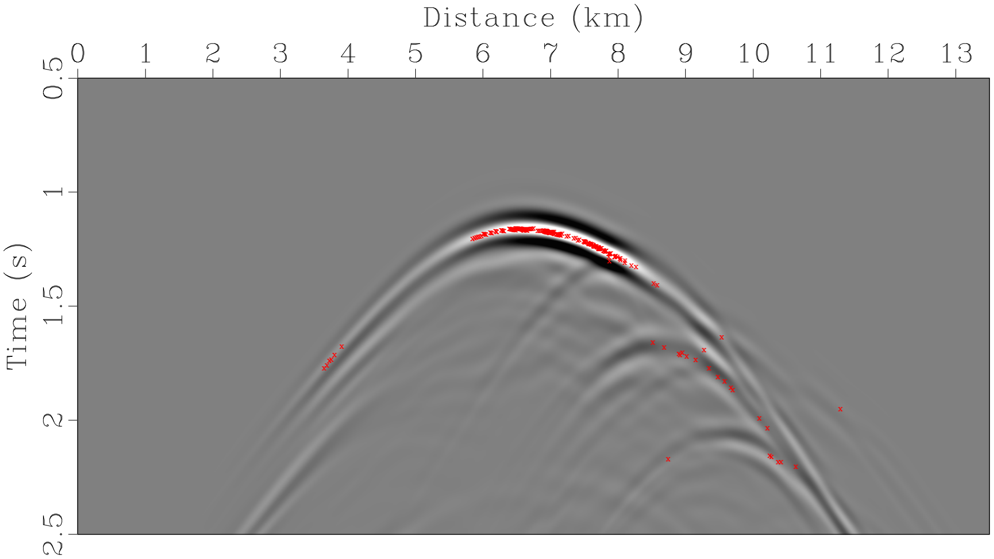


Figure 4.11: Surface exit locations for all arrivals originating from point $x = 6.7$ km, $y = 7.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model (ray tracing) plotted over wavefield computed for a source at the same point and extracted at receiver line of constant $y = 7.7$ km. chapter-psint/segsalt ssaltfdescsxl0

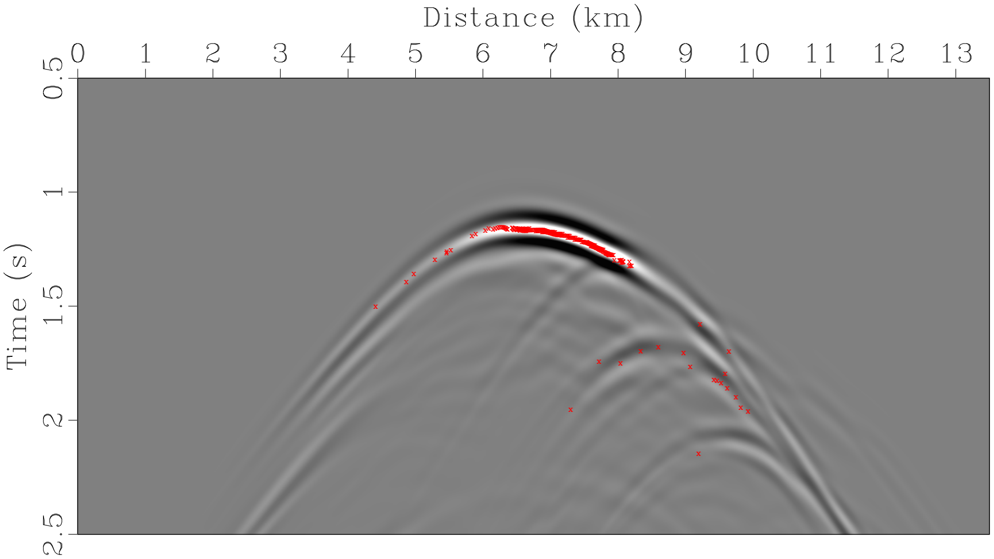


Figure 4.12: Surface exit locations for all arrivals originating from point $x = 6.7$ km, $y = 7.7$ km, and $z = 2.5$ km of SEG/EAGE Salt model (iterative reconstruction) plotted over wavefield computed for a source at the same point and extracted at receiver line of constant $y = 7.7$ km. chapter-psint/segsalt ssaltfdescsxl1

Chapter 5

2-D angle-domain Kirchhoff imaging with escape functions

Imaging using escape functions

In this chapter, I demonstrate how to implement angle-domain migration in 2-D using escape functions as input. To produce high quality image, the migration process needs to infer the following information from input escape data in order to implement the summation defined by integral (1.2):

1. Relationship of the current image point's position, subsurface scattering and dip angles to correct image contributions from the discrete surface reflection data $u\left(\mathbf{s}, \mathbf{r}, t\right)$.
2. Waveform-correction operator $\mathbf{D}_t$.
3. Amplitude weight $\widehat{W}$.
4. Antialising parameters.

I show below that all of this information can be extracted from only two functions: escape time $\widehat{T}\left(\widehat{\mathbf{x}}, \theta\right)$ and escape position $\widehat{\mathbf{y}}\left(\widehat{\mathbf{x}}, \theta\right)$. I demonstrate imaging examples for 4 different synthetic 2-D models. Each of the examples contains scattering and dip angle gathers along with stacked images. Escape tables for the examples are computed using the hybrid algorithm from Chapter 3.

Subsurface reflection system

The relationship between source and receiver phase vectors in integral (1.2) and scattering and dip angles in a 2-D media is depicted schematically in Figure 5.1.

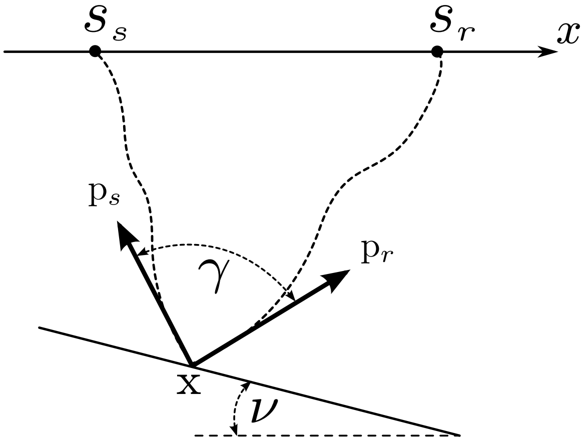


Figure 5.1: 2-D subsurface reflection system, relationship between phase vectors and scattering and dip angles (a scheme). chapter-imag2/. croper2d

The scattering angle γ can be defined as

$$\gamma = \cos^{-1}\left(\frac{\mathbf{p}_s \cdot \mathbf{p}_r}{|\mathbf{p}_s|\,|\mathbf{p}_r|}\right), \quad \gamma \in [\,0;\,\pi). \tag{5.1}$$

Correspondingly, the dip angle ν is

$$\nu = \begin{cases} -\cos^{-1}\left(-\dfrac{p^z_{s+r}}{|\mathbf{p}_{s+r}|}\right), & p^x_{s+r} > 0 \\ \cos^{-1}\left(-\dfrac{p^z_{s+r}}{|\mathbf{p}_{s+r}|}\right), & p^x_{s+r} < 0 \end{cases}, \quad \nu \in [-\pi;\,\pi) \tag{5.2}$$
$$\mathbf{p}_{s+r} = \mathbf{p}_s + \mathbf{p}_r\,.$$

When the integral (1.2) get discretized, angle-domain migration becomes a summation over a finite number of source and receiver branches defined by escape function tables. Every pair of branches gives the source location $\widehat{\mathbf{y}}(\mathbf{x},\mathbf{p}_s)$, the receiver location $\widehat{\mathbf{y}}(\mathbf{x},\mathbf{p}_s)$, and the total traveltime from the source to the receiver $\widehat{T}(\mathbf{x},\mathbf{p}_s) +$

$\widehat{T}(\mathbf{x}, \mathbf{p}_r)$. This combination allows us to find the correct contribution to the image from surface data. Because of the discrete nature of both input data and escape tables in practice, escape locations almost never exactly coincide with source and receiver positions. Instead of using individual arrivals in the summation process, I analyze couples of nearest values in the escape tables. Every such pair forms an exit segment on the surface, provided that $\widehat{z}_i = \widehat{z}_{i+1} = z_{surface}$. For every exit source segment, all source locations inside the segment are identified. Then, for a fixed source point, all of its receivers inside the exit receiver segment are found. All of the identified samples contribute to the same migration angle bin. The contribution is normalized by its hit count. Escape traveltime in each of the branches needs to be interpolated between $\widehat{T}_i$ and $\widehat{T}_{i+1}$ values.

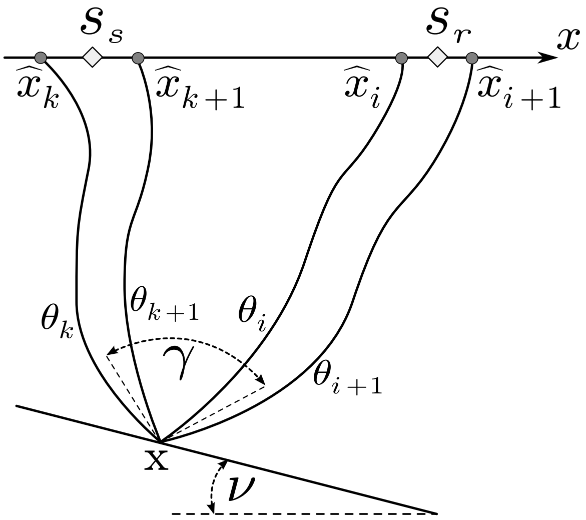


Figure 5.2: 2-D angle-domain migration using escape tables (a scheme). chapter-imag2/. cram2d

Phase shifts due to caustics

If a ray tube travels through a caustic point, then it causes seismic signal to shift its phase by $\frac{\pi}{2}$ (Chapman and Drummond, 1982). The total number of times the ray tube hit caustics is called Keller-Maslov-Arnold-Hörmander (KMAH) index. One hit corresponds to the index of 1. The aggregate phase shift of the signal is the sum of all KMAH indices in receiver and source branches times $\frac{\pi}{2}$.

I use a crude measure of the aggregate index by looking at the sign of $dx = \widehat{x}_{i+1} - \widehat{x}_i$ (Figure 5.3). The plus sign assumes KMAH to be equal to 0, the minus sign makes it 1. This technique misses some of rare cases of caustic-related phase shifts but still allows to improve the quality of imaging.

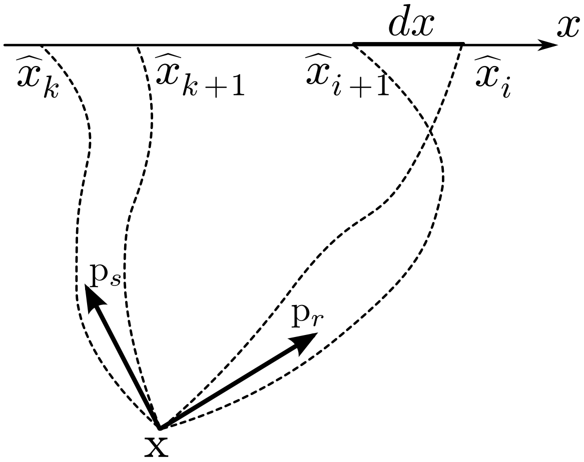


Figure 5.3: Reversal of exit positions on the surface for a ray tube in 2-D due to a caustic point (a scheme). chapter-imag2/. kmah2d

Antialiasing

The migration summation may introduce noticeable artifacts to the image, if the imaging operator accumulates data in large steps along the time axis in the steep part of the diffraction curve (Fomel, 2002). A popular approach to this problem proposed by Lumley et al. (1994) avoids aliasing by applying a low-pass triangle filter on the fly according to the slope (exit ray parameter) of the migration operator (Abma et al., 1999). The latter can be trivially found from an exit segment as

$$p_{exit} \simeq \frac{\widehat{T}_{i+1} - \widehat{T}_i}{\widehat{x}_{i+1} - \widehat{x}_i} \, . \tag{5.3}$$

Amplitude weights

For true-amplitude imaging, correct amplitude weights need to computed and applied during the summation process. Both receiver and source branch weights depend on their geometrical spreading (Koren and Ravve, 2011), which is proportional to the orthogonal cross section of a ray tube. If the exit ray parameter is also written as

$$p_{exit} = S_{exit} \sin \alpha_{exit} \, , \tag{5.4}$$

where S_{exit} is the slowness at the surface at the exit location and α_{exit} is the angle between the vertical direction and the exit ray, then, the area of the ray tube cross section can be written as

$$J_{exit} \simeq |\widehat{x}_{i+1} - \widehat{x}_i| \sqrt{1 - \left(\frac{p_{exit}}{S_{exit}}\right)^2} \, . \tag{5.5}$$

Other components of the amplitude weights, such as the obliquity factor and slowness value at the image point, do not depend on the exit quantities.

Data preparation

Before angle-domain migration can start mapping surface data to the image space, the data must be prepared accordingly. This usually includes:

1. Preprocessing data (noise attenuation, signal bandwidth enhancement, multiple removal, etc).
2. Clipping large amplitudes and removing bad data samples.
3. Differentiating data in time to compensate for the loss of bandwidth in image due to migration summation.
4. Performing causal and anticausal integration as required by the Lumley-Claerbout-Bevc antialising filter.
5. Creating a second copy of the dataset phase-shifted by $\frac{\pi}{4}$ (other phase shifts are then produced from these two copies).

Image muting and noise suppression

The migration output rarely contains any useful information at large dips and large scattering angles. To improve the image quality and computational efficiency of the migration algorithm, the output area should be constrained in the angle domain. Figure 5.4 demonstrates the construction of a mute zone and its counterpart - a pass zone, which allows migration contributions. The pass zone is a half-ellipse with the two axes defined by the maximum scattering angle γ_{max} and the maximum dip angle $\pm\nu_{max}$. For even better results, γ_{max} and ν_{max} should gradually decrease from the top to the bottom of the image domain.

Marmousi model

I implemented an imaging algorithm based on the choices described above. This migration program takes regularly sampled escape tables as input and generates

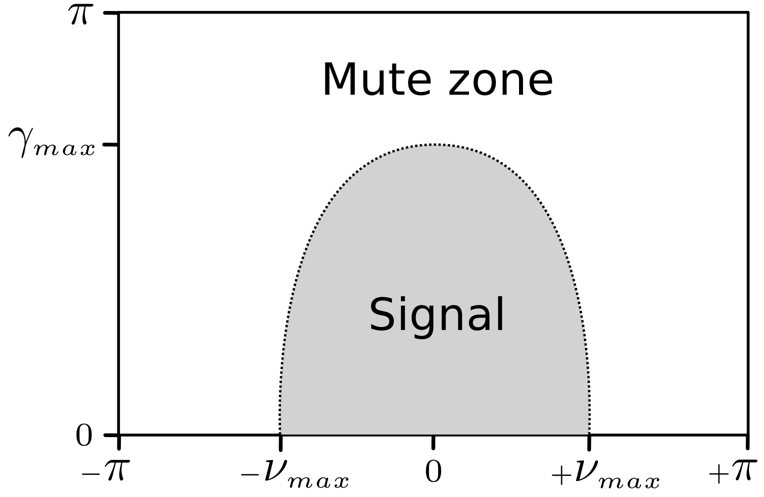


Figure 5.4: Definition of a muting zone in angle-domain for a fixed depth point (a scheme). chapter-imag2/. amute

scattering and dip angle gathers for a specified image area.

I computed escape tables for Marmousi benchmark model (Versteeg, 1993) using the hybrid Eulerian-Lagrangian solver. The spatial sampling of the escape tables is Δx=Δz=4 m and the angular sampling is $\Delta\theta$=1°. The threshold for switching to ray tracing is set to 1.0 km.

The final image is shown on Figure 5.5. A series of scattering angle gathers is presented on Figures 5.6(a)-5.9(a) and dip angle gathers are on Figures 5.6(b)- 5.9(b).

Sigsbee2B model

Sigsbee2B (Figure 5.10) is another popular 2-D imaging isotropic benchmark (Paffenholz et al., 2002). It contains a salt body and creates a challenge of imaging target subsalt features. The escape tables sampling for this model is Δz = 7.62 m, Δx=11.43 m, and $\Delta\theta$=0.5°. The final image is presented on Figure 5.12. A series of scattering angle gathers is presented on Figures 5.11(a), 5.13(a)-5.16(a) and dip angle gathers are on Figures 5.11(b), 5.13(b)-5.16(b).

Hess VTI model

Hess VTI is an anisotropic imaging benchmark created by Hess Corporation (Figures 5.17, 5.18, 5.19). It contains a salt body and a low-amplitude target reservoir adjacent to the salt. The spatial sampling of the escape tables is Δx=Δz=6.096 m and the angular sampling is $\Delta\theta$=1°. The final image is presented on Figure 5.21. A series of scattering angle gathers is presented on Figures 5.20(a), 5.22(a), 5.23(a) and dip angle gathers are on Figures 5.20(b), 5.22(b), 5.23(b).

BP TTI model

BP TTI (Shah, 2008) is another anisotropic imaging benchmark (Figures 5.24(a), 5.24(b), 5.25(a), 5.25(b)). It contains several salt bodies and areas of strong TTI anisotropy. The spatial sampling of the escape tables is Δx=Δz=12.5 m and the angular sampling is $\Delta\theta$=1°. The final image is presented on Figure 5.26. A series of scattering angle gathers is presented on Figures 5.27(a)-5.33(a) and dip angle gathers are on Figures 5.27(b)- 5.33(b).

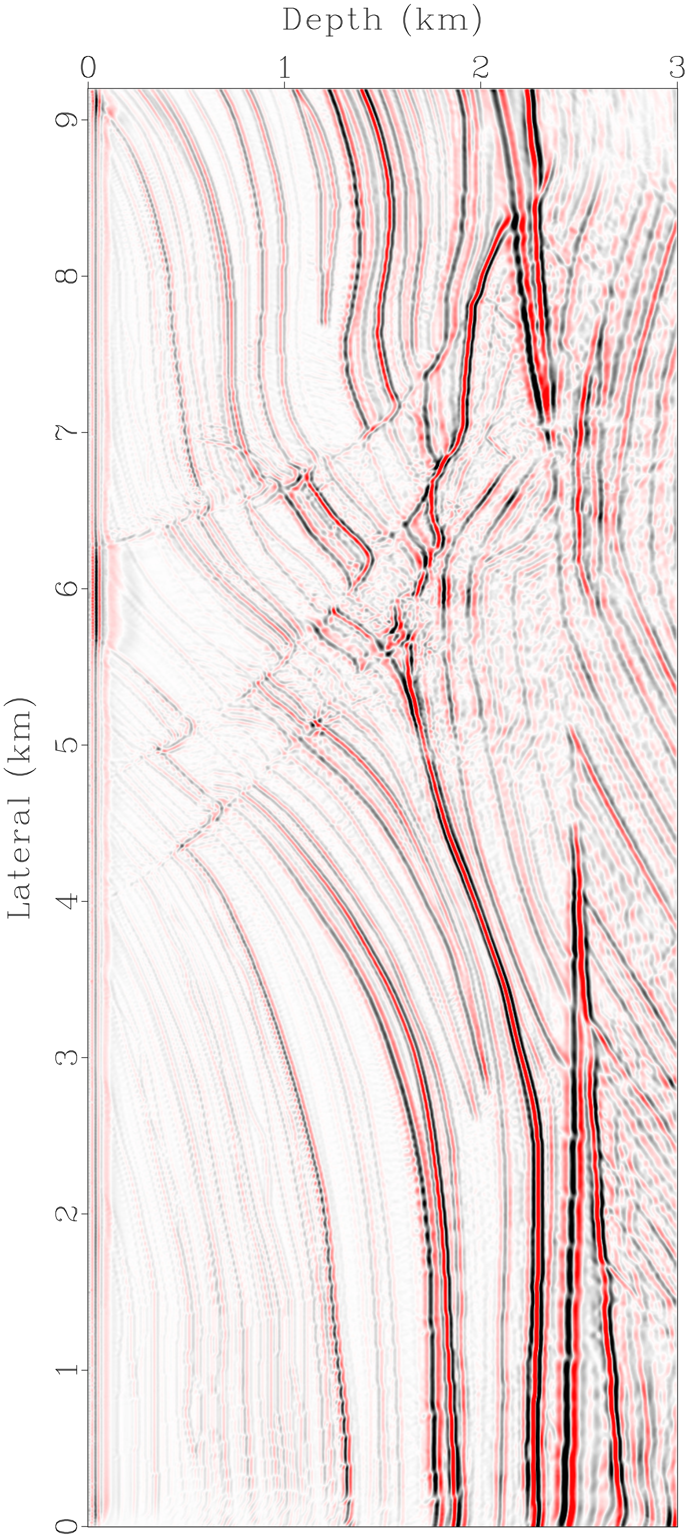
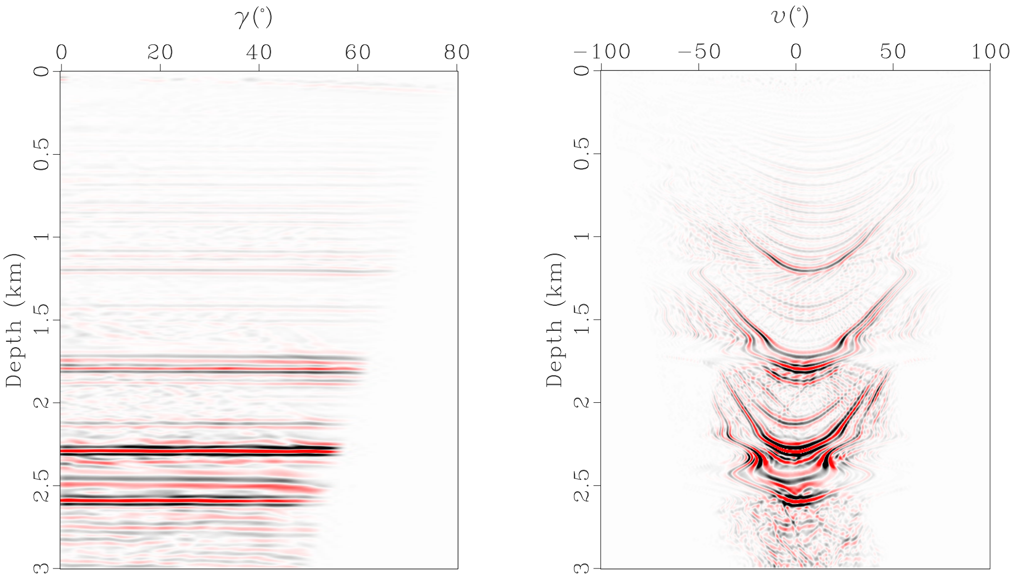


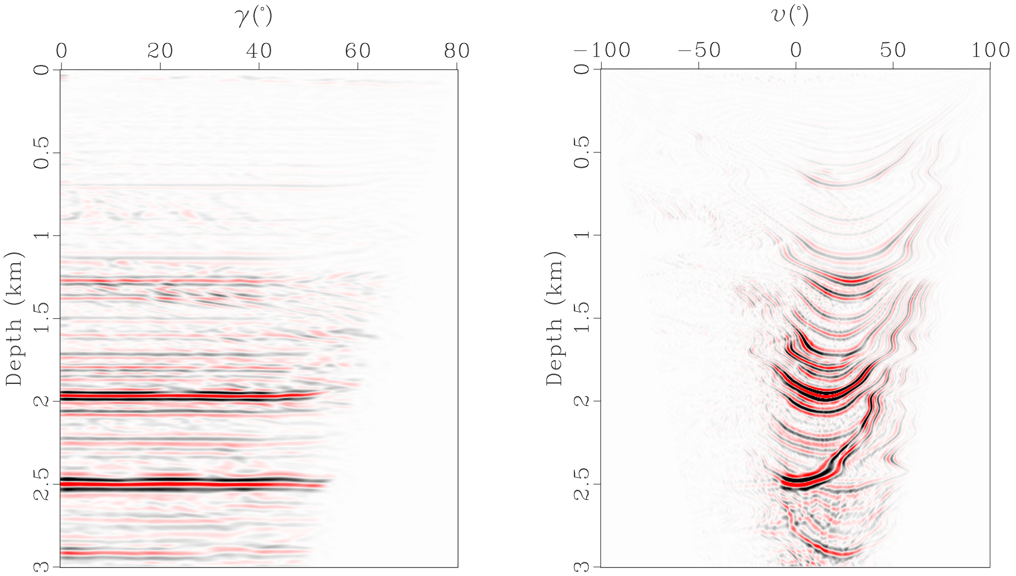
Figure 5.5: Marmousi model final image. chapter-imag2/marmousi marmdcrstk



(a)

(b)

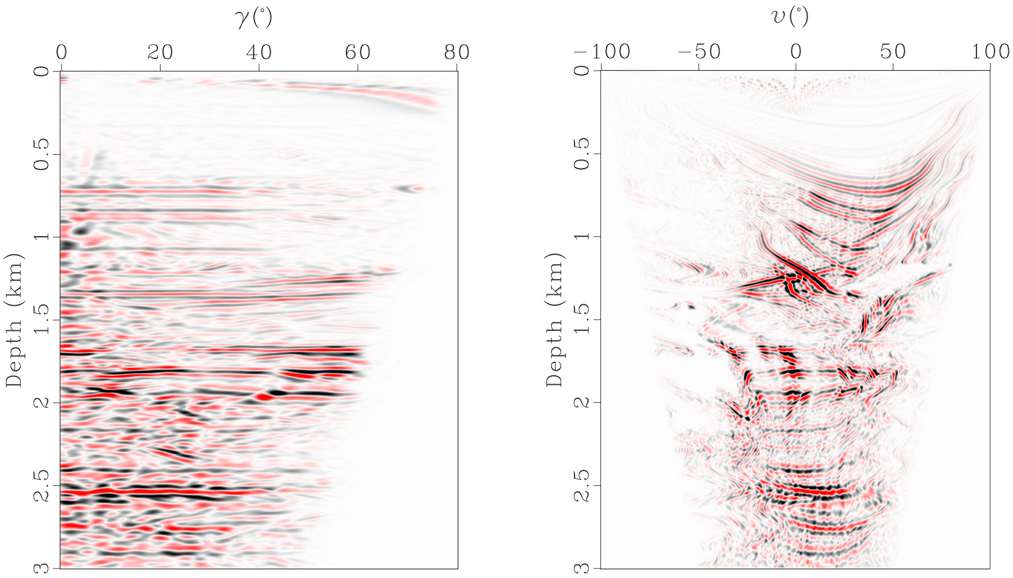
Figure 5.6: Marmousi model scattering (a) and dip (b) angle gathers for lateral position 2.0 km. chapter-imag2/marmousi marmocig0,marmdcig0



(a)

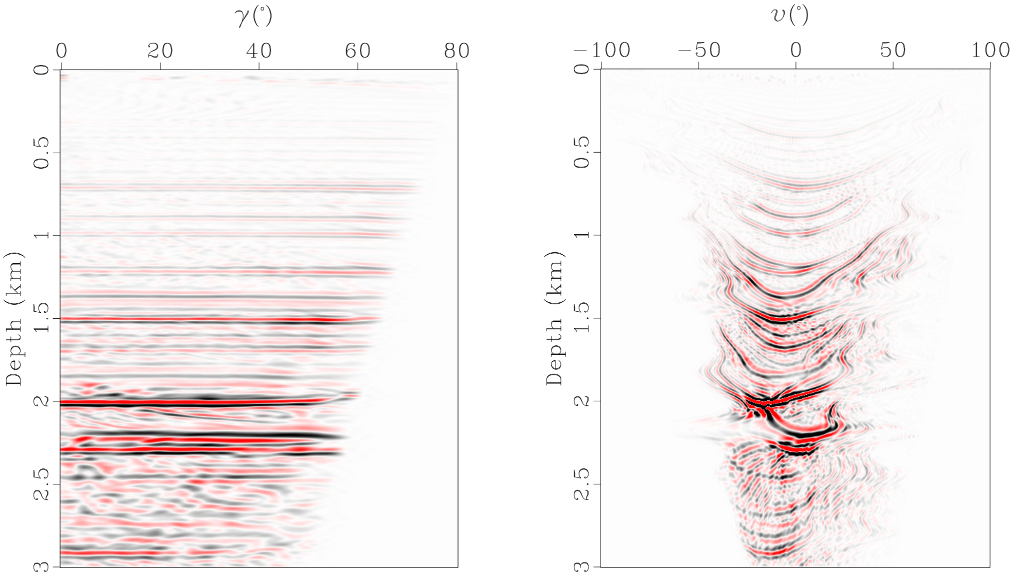
(b)

Figure 5.7: Marmousi model scattering (a) and dip (b) angle gathers for lateral position 4.0 km. chapter-imag2/marmousi marmocig1,marmdcig1



(a) (b)

Figure 5.8: Marmousi model scattering (a) and dip (b) angle gathers for lateral position 6.0 km. chapter-imag2/marmousi marmocig2,marmdcig2



(a) (b)

Figure 5.9: Marmousi model scattering (a) and dip (b) angle gathers for lateral position 8.0 km. chapter-imag2/marmousi marmocig3,marmdcig3

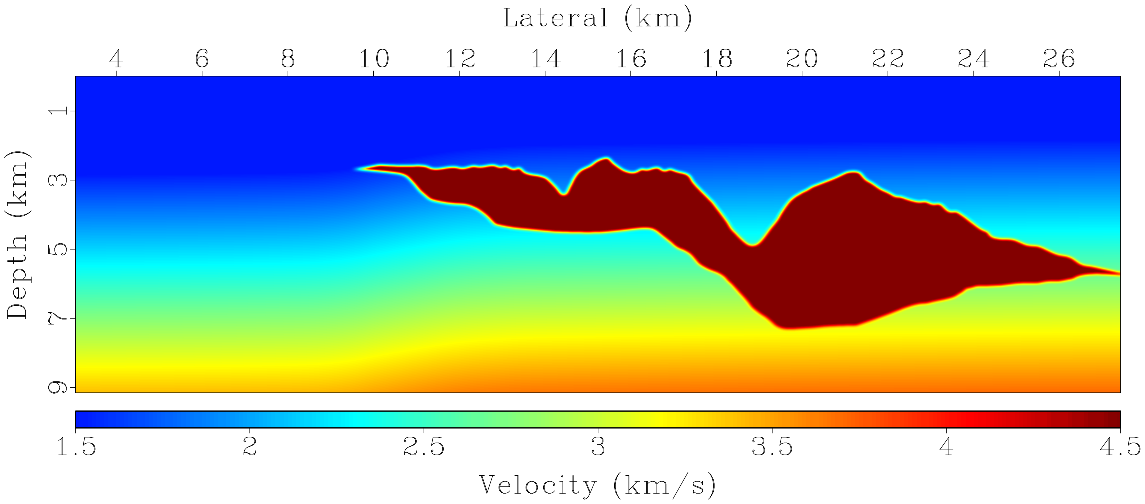
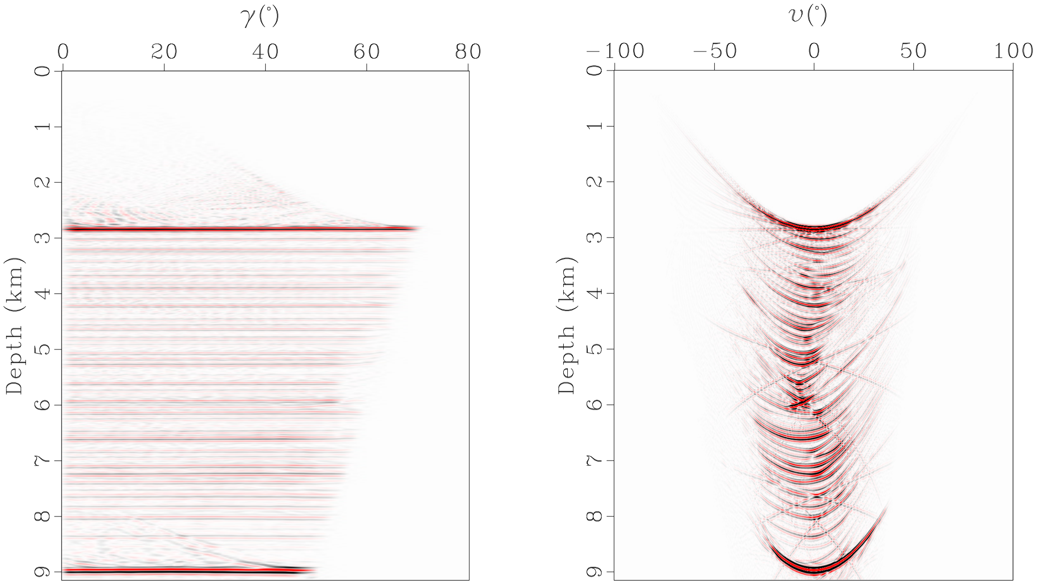


Figure 5.10: Sigsbee2B model velocity. chapter-imag2/sigsbee sigsbvel



(a) (b)

Figure 5.11: Sigsbee2B model scattering (a) and dip (b) angle gathers for lateral position 7.0 km. chapter-imag2/sigsbee sigsbocig0,sigsbdcig0

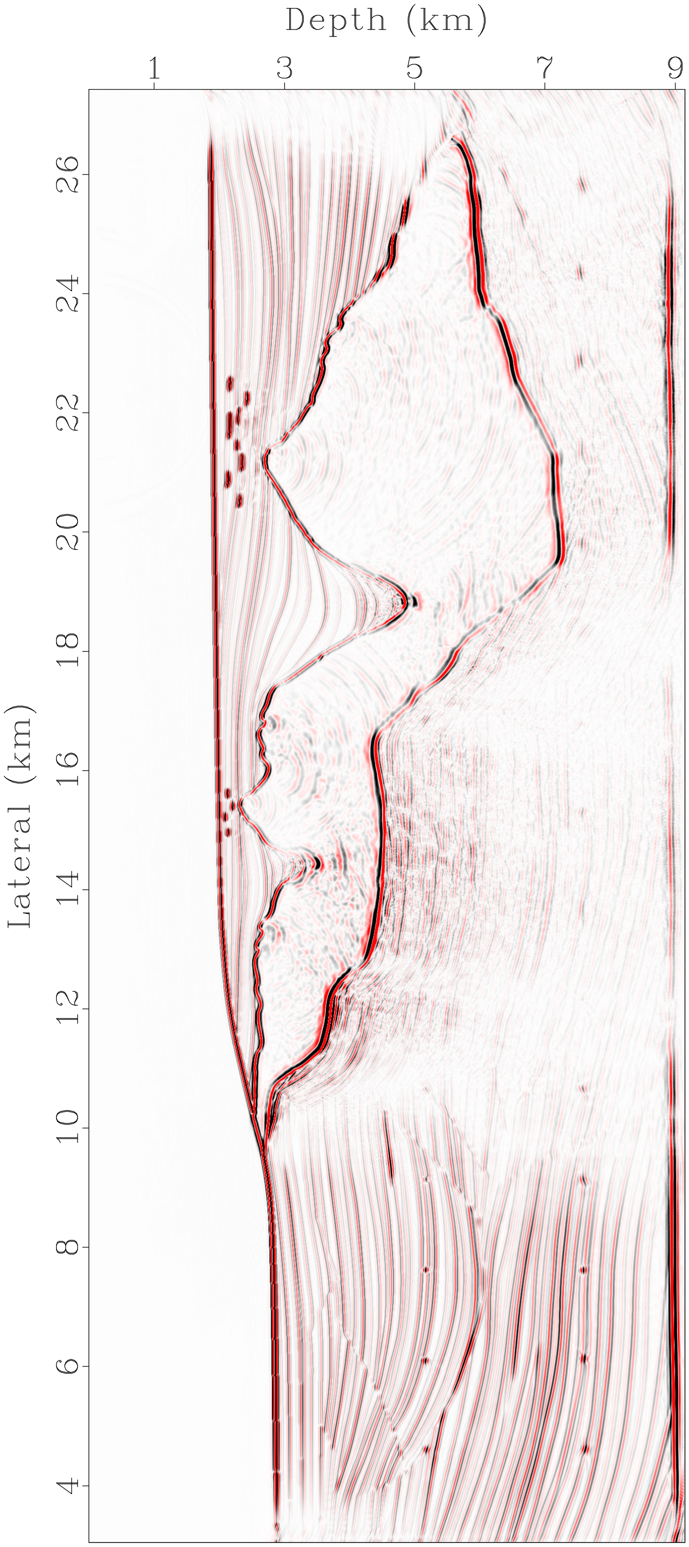


Figure 5.12: Sigsbee2B model final image. chapter-imag2/sigsbee sigsbdcrstk

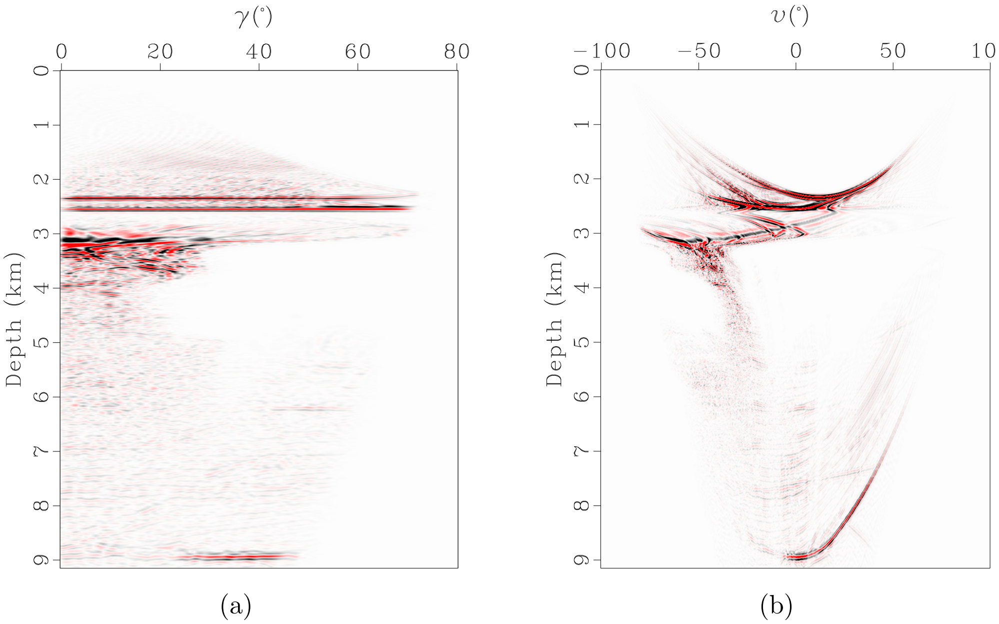


Figure 5.13: Sigsbee2B model scattering (a) and dip (b) angle gathers for lateral position 11.0 km. chapter-imag2/sigsbee sigsbocig1,sigsbdcig1

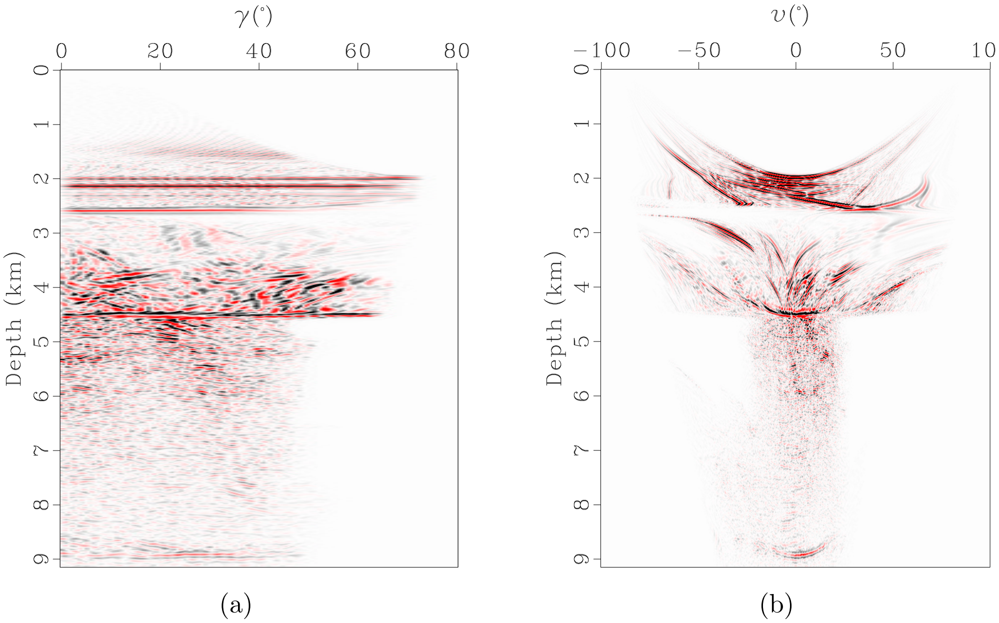
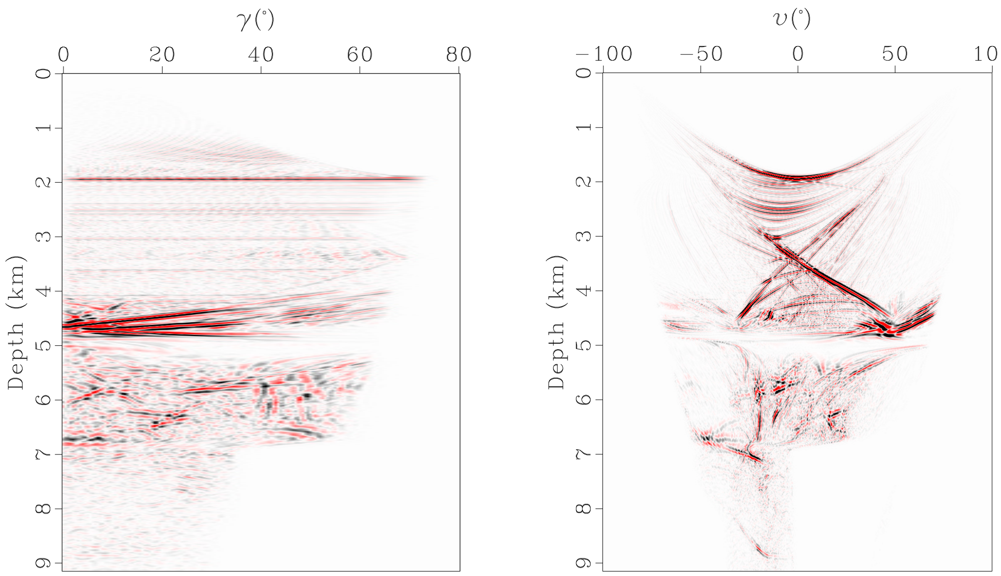
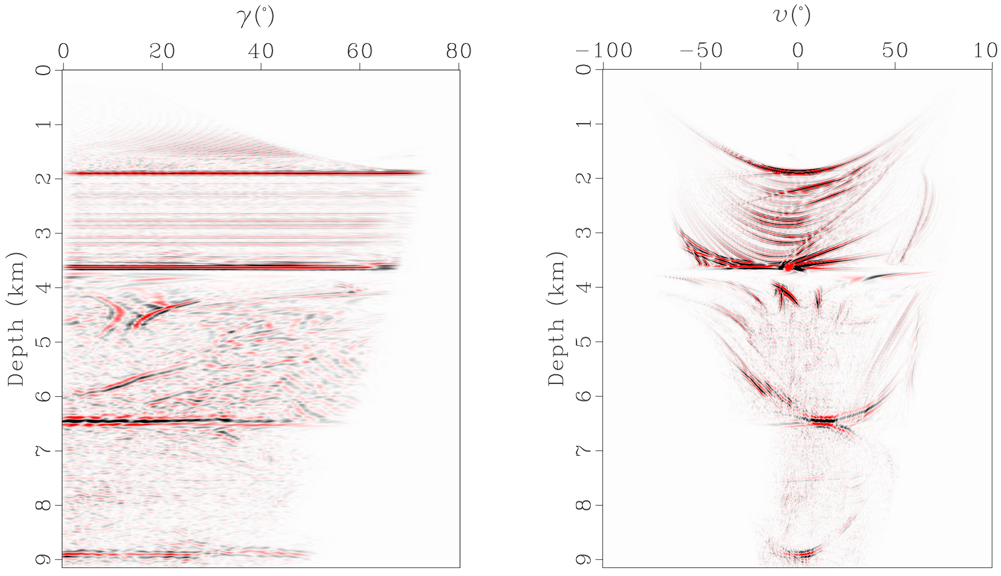


Figure 5.14: Sigsbee2B model scattering (a) and dip (b) angle gathers for lateral position 15.0 km. chapter-imag2/sigsbee sigsbocig2,sigsbdcig2



(a) (b)

Figure 5.15: Sigsbee2B model scattering (a) and dip (b) angle gathers for lateral position 19.0 km. chapter-imag2/sigsbee sigsbocig3,sigsbdcig3



(a) (b)

Figure 5.16: Sigsbee2B model scattering (a) and dip (b) angle gathers for lateral position 23.0 km. chapter-imag2/sigsbee sigsbocig4,sigsbdcig4

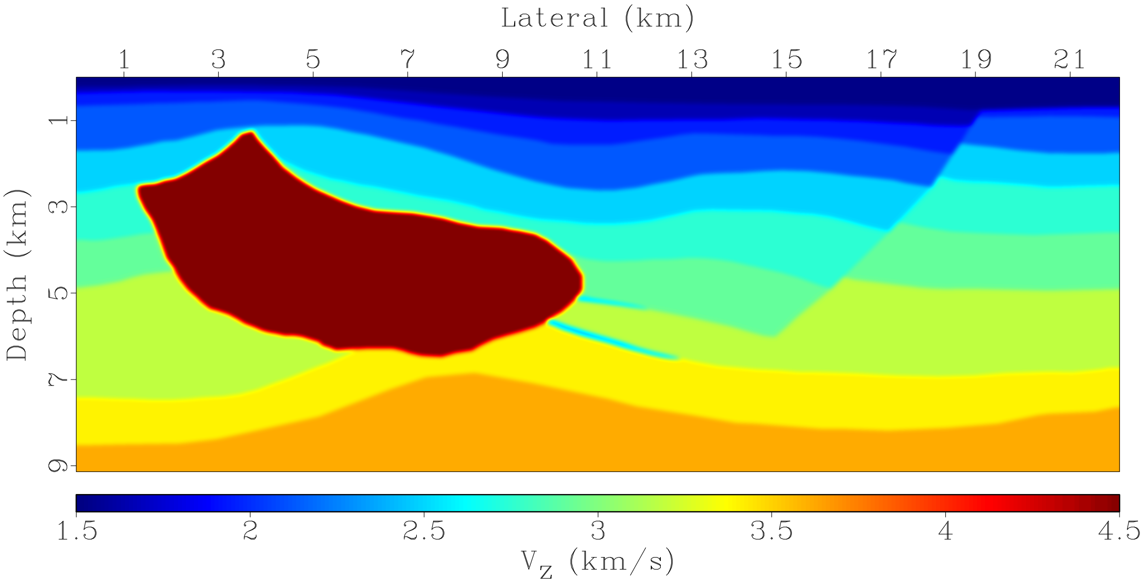


Figure 5.17: Hess VTI model vertical velocity. chapter-imag2/hessvti hessvtivz

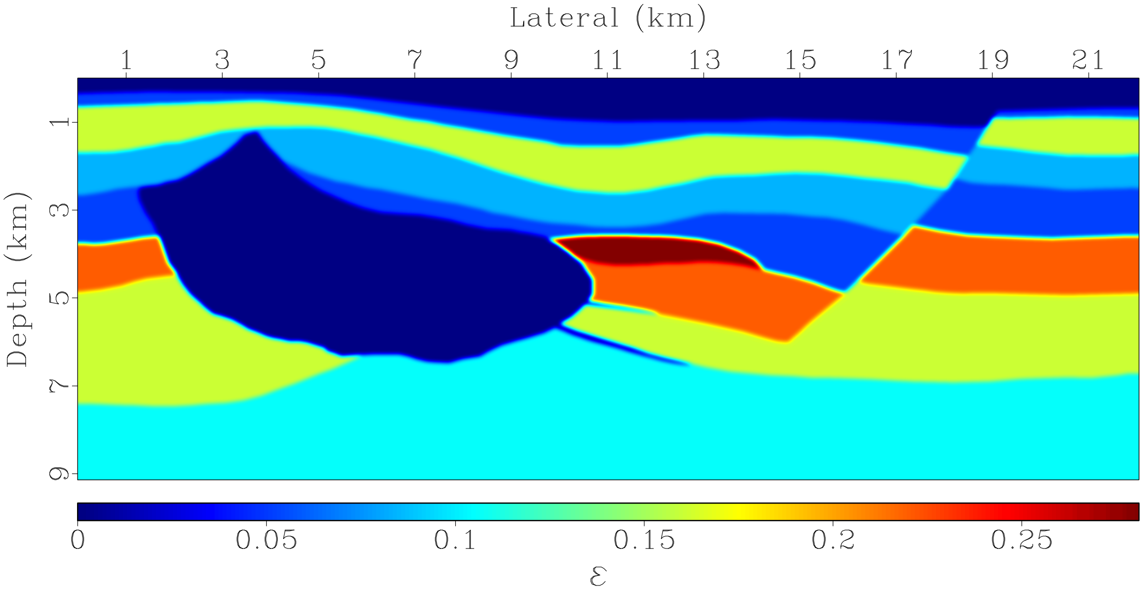


Figure 5.18: Hess VTI model Thomsen anisotropy parameter ϵ. chapter-imag2/hessvti hessvtieps

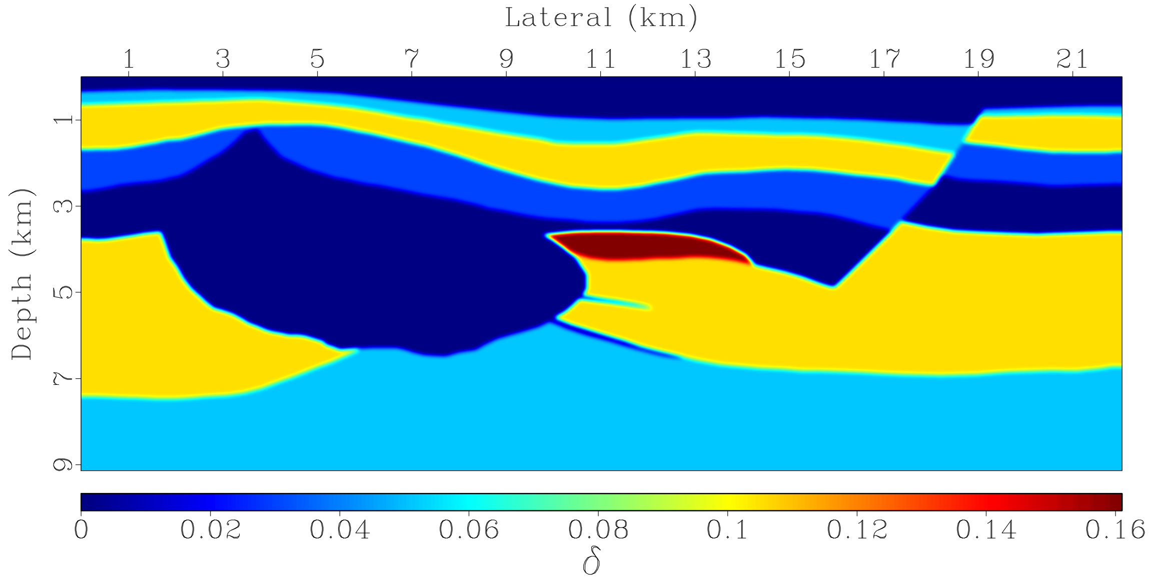
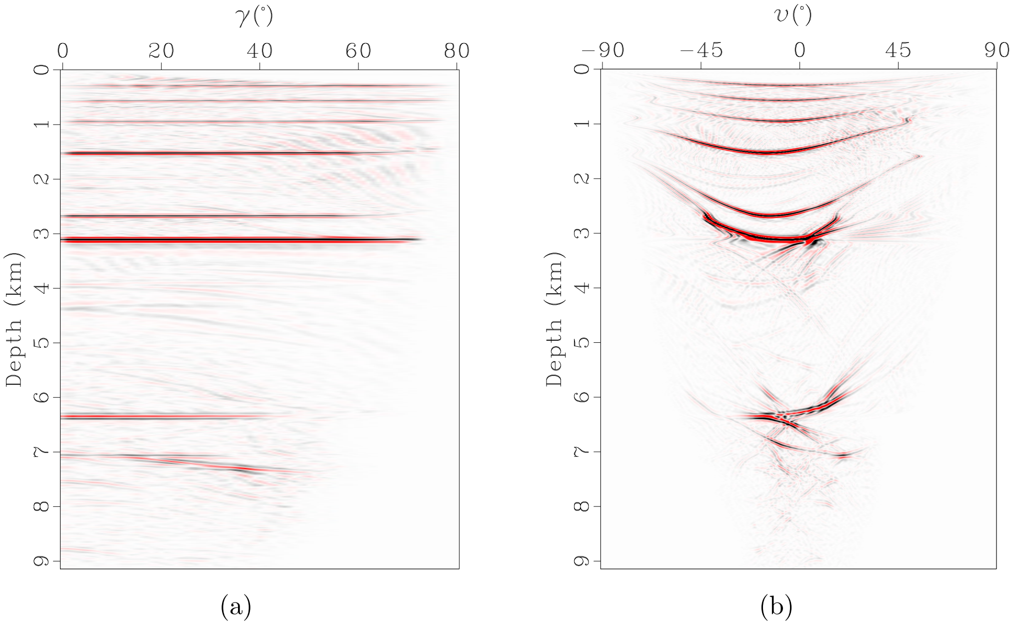


Figure 5.19: Hess VTI model Thomsen anisotropy parameter δ. chapter-imag2/hessvti hessvtidel



(a) (b)

Figure 5.20: Hess VTI model scattering (a) and dip (b) angle gathers for lateral position 7.0 km. chapter-imag2/hessvti hessvtiocig0,hessvtidcig0

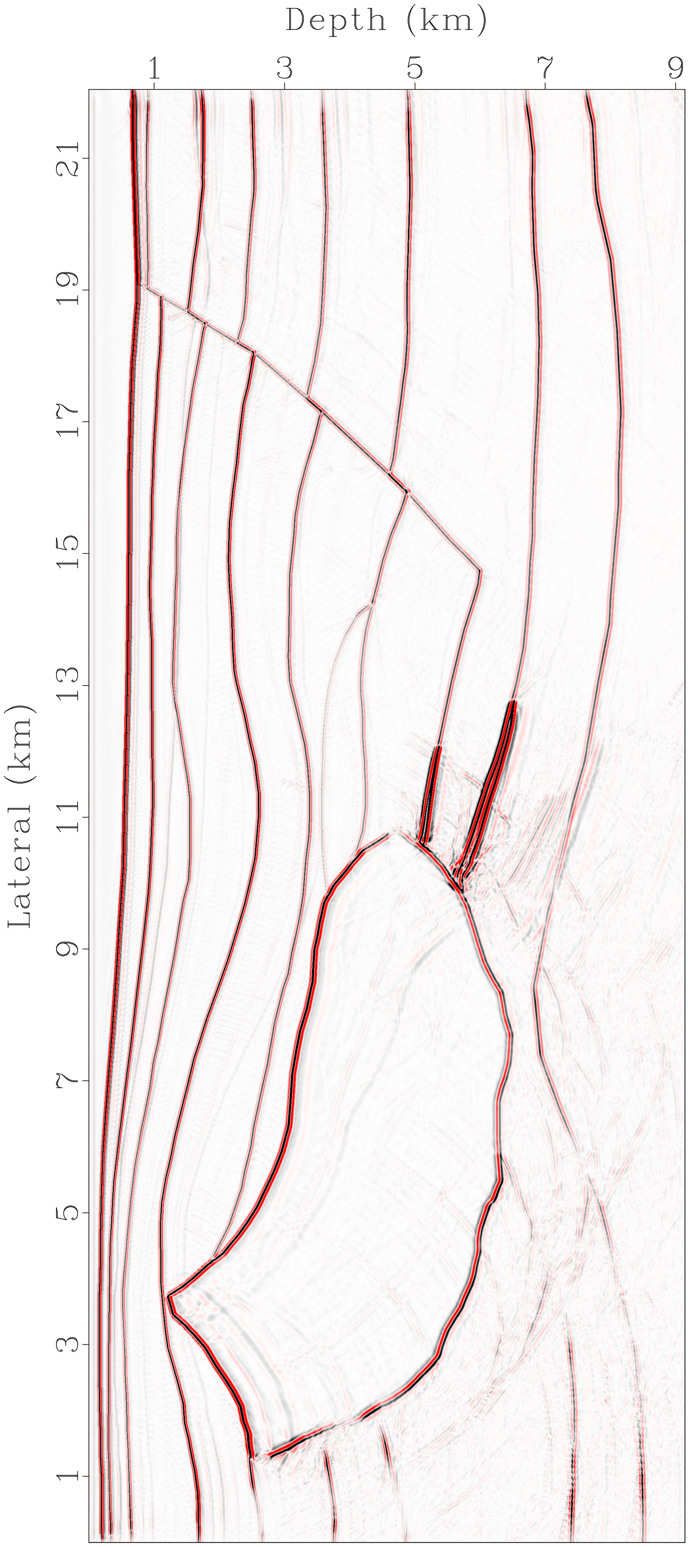
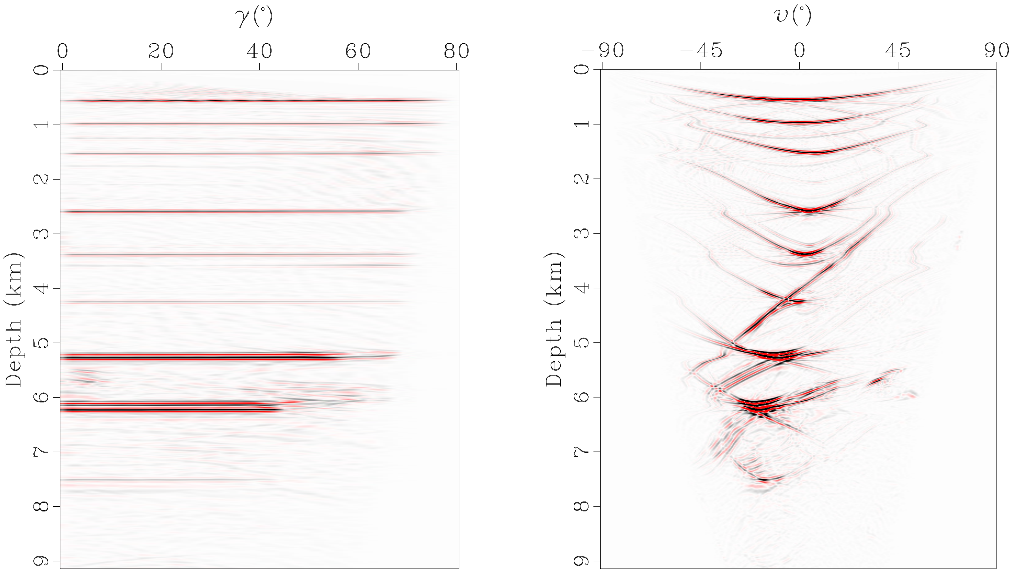
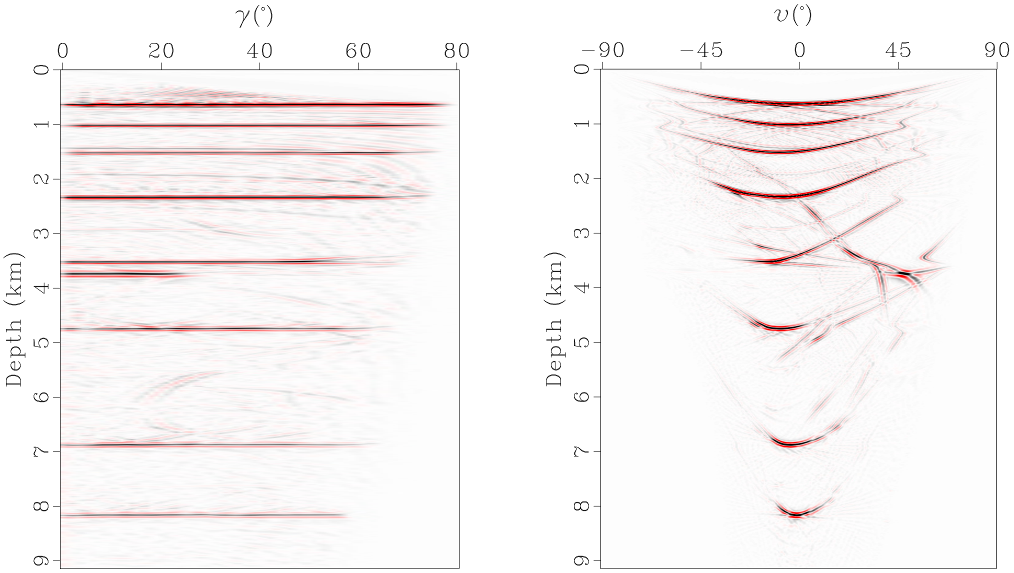


Figure 5.21: Hess VTI model final image. chapter-imag2/hessvti hessvtidcrstk



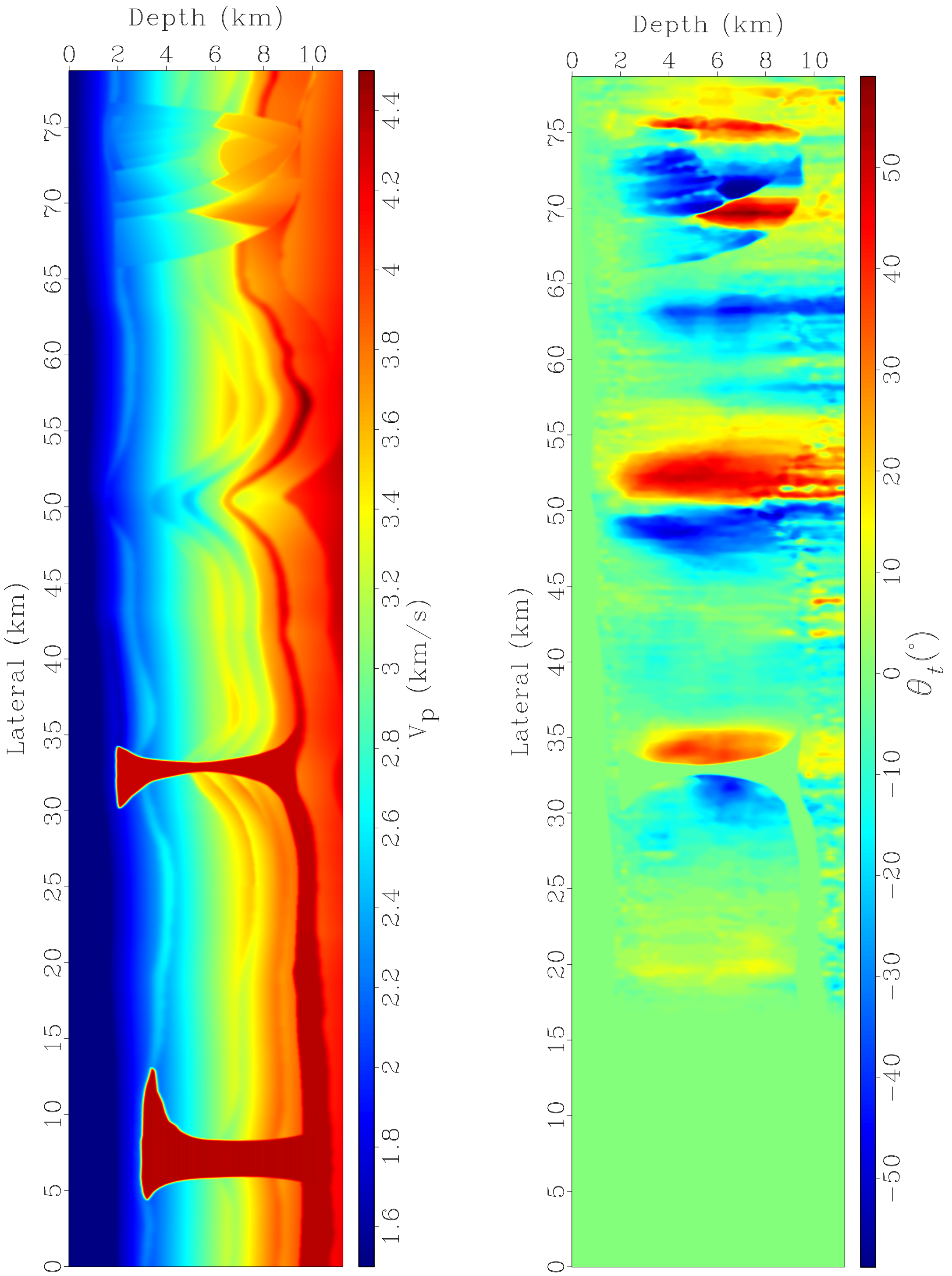
(a) (b)

Figure 5.22: Hess VTI model scattering (a) and dip (b) angle gathers for lateral position 11.5 km. chapter-imag2/hessvti hessvtiocig1,hessvtidcig1



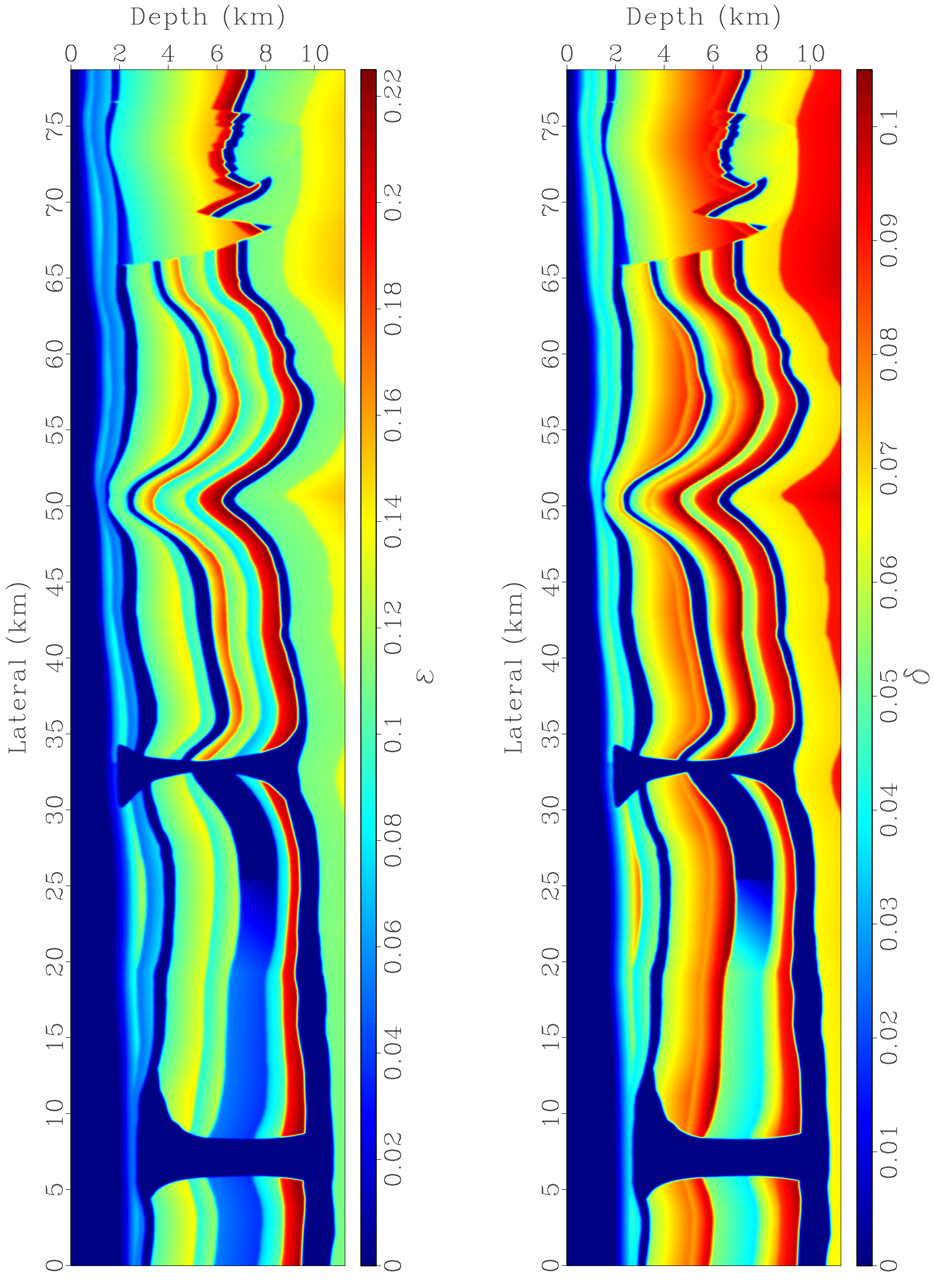
(a) (b)

Figure 5.23: Hess VTI model scattering (a) and dip (b) angle gathers for lateral position 17.0 km. chapter-imag2/hessvti hessvtiocig2,hessvtidcig2



(a) (b)

Figure 5.24: BP TTI model symmetric axis velocity (a) and axis tilt angle (b). chapter-imag2/bptti bpttivp,bpttitheta



(a) (b)

Figure 5.25: BP TTI model Thomsen anisotropy parameters ϵ (a) and δ (b). chapter-imag2/bptti bpttieps,bpttidel

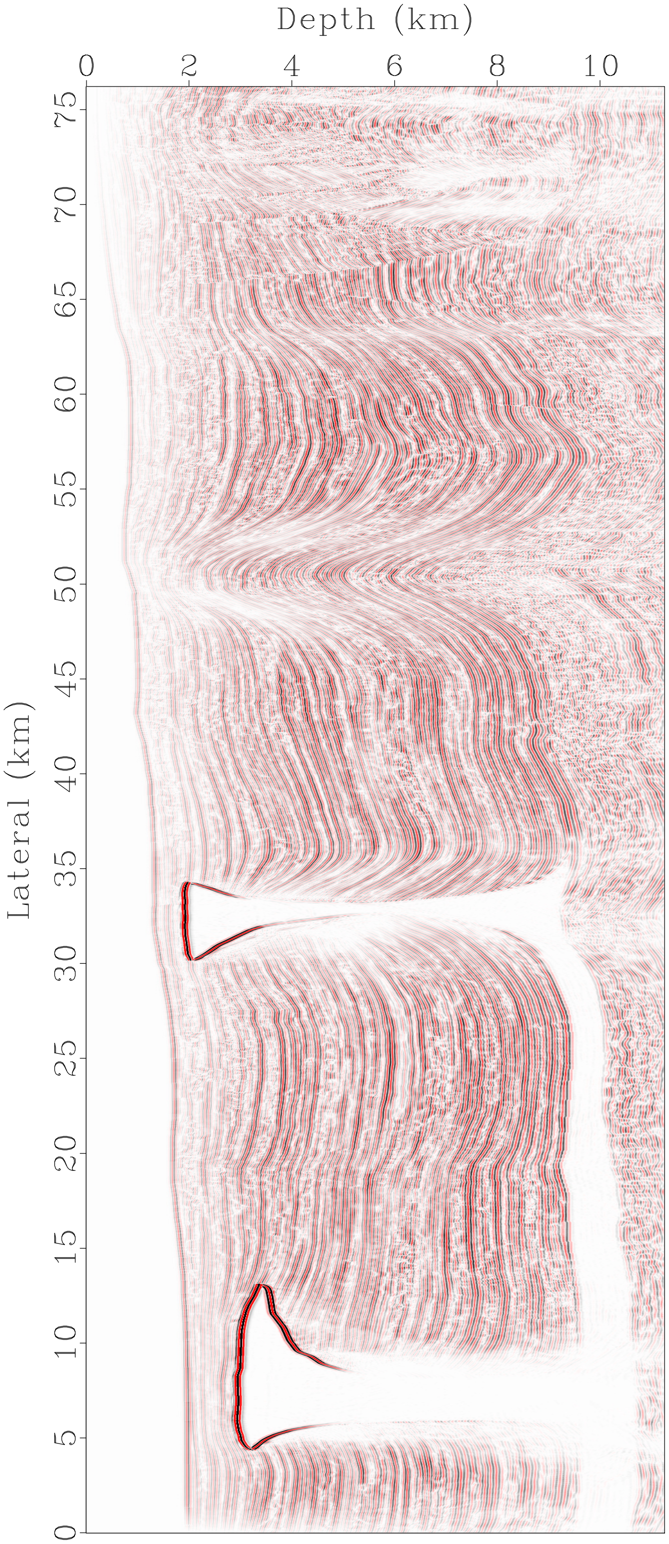
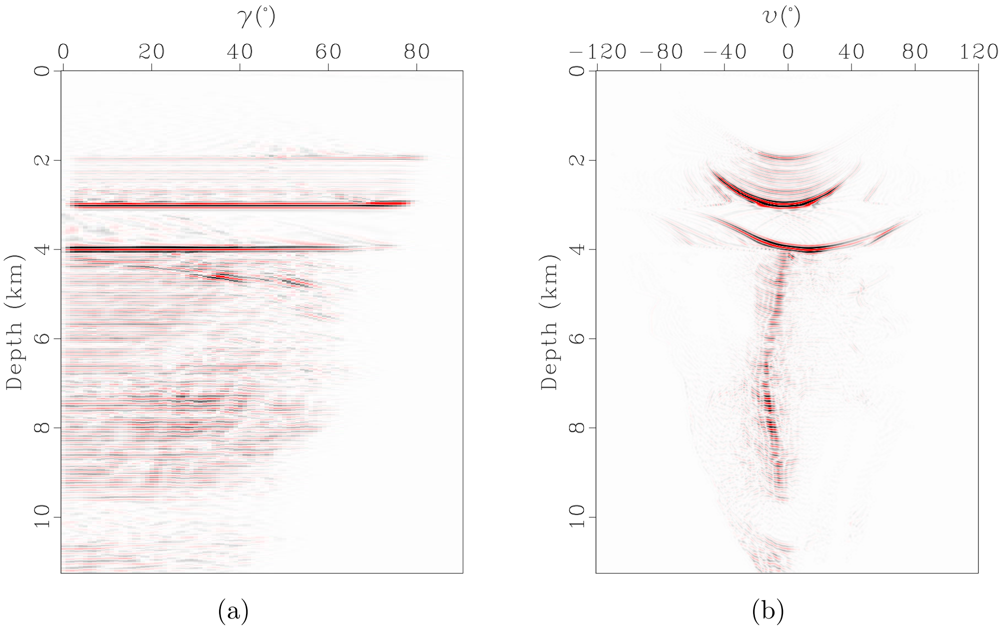
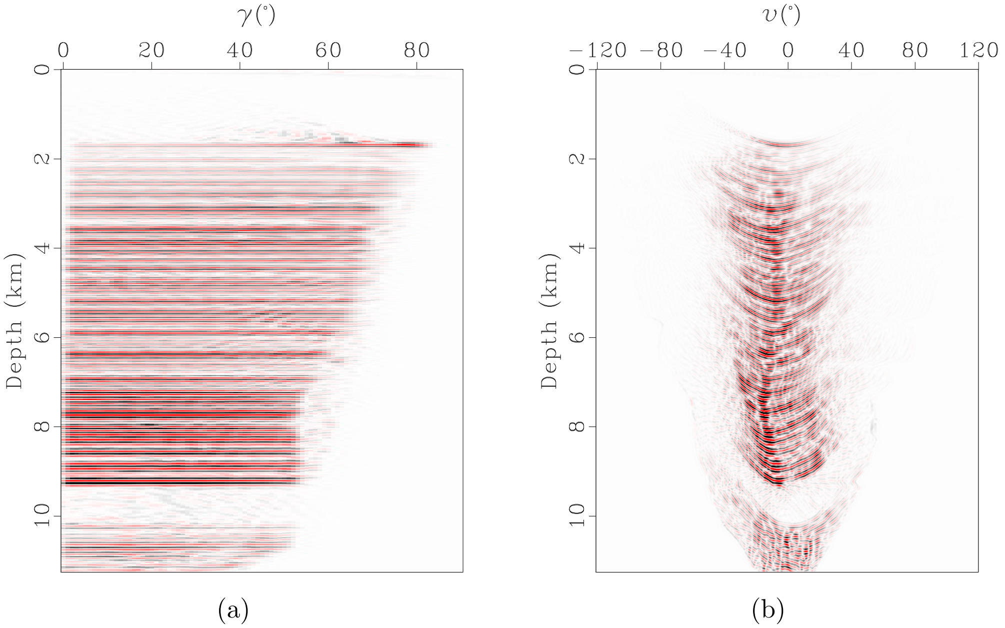


Figure 5.26: BP TTI model final image. chapter-imag2/bptti bpttidcrstk



(a) (b)

Figure 5.27: BP TTI model scattering (a) and dip (b) angle gathers for lateral position 10.0 km. chapter-imag2/bptti bpttiocig0,bpttidcig0



(a) (b)

Figure 5.28: BP TTI model scattering (a) and dip (b) angle gathers for lateral position 20.0 km. chapter-imag2/bptti bpttiocig1,bpttidcig1

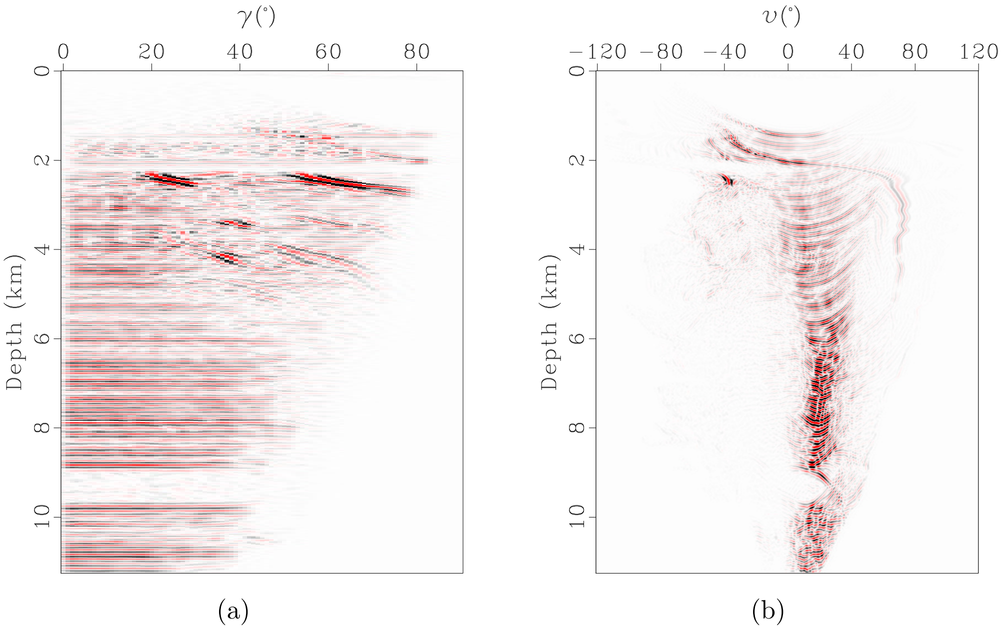


Figure 5.29: BP TTI model scattering (a) and dip (b) angle gathers for lateral position 30.0 km. chapter-imag2/bptti bpttiocig2,bpttidcig2

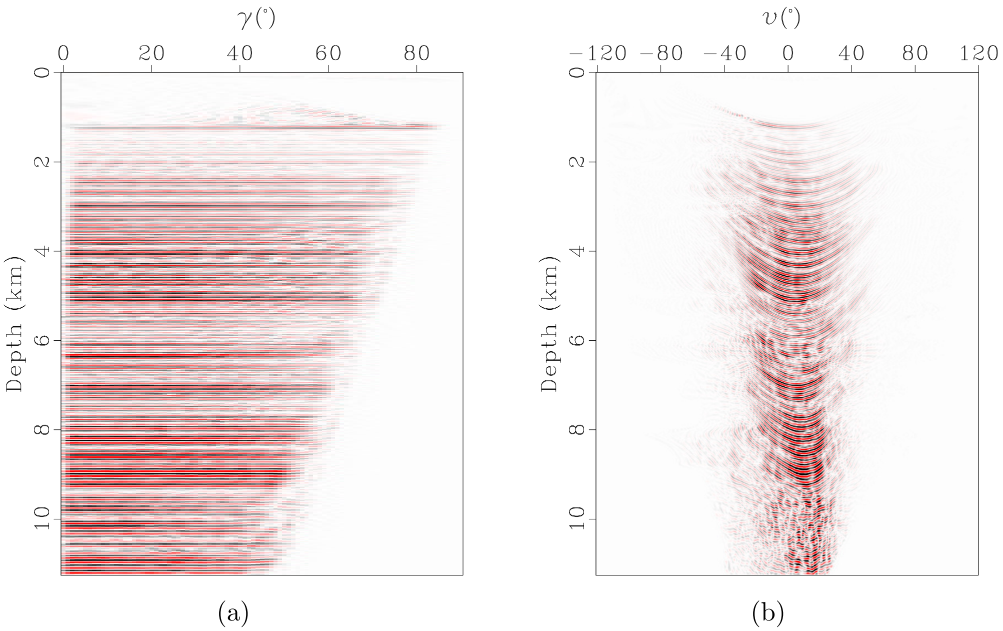
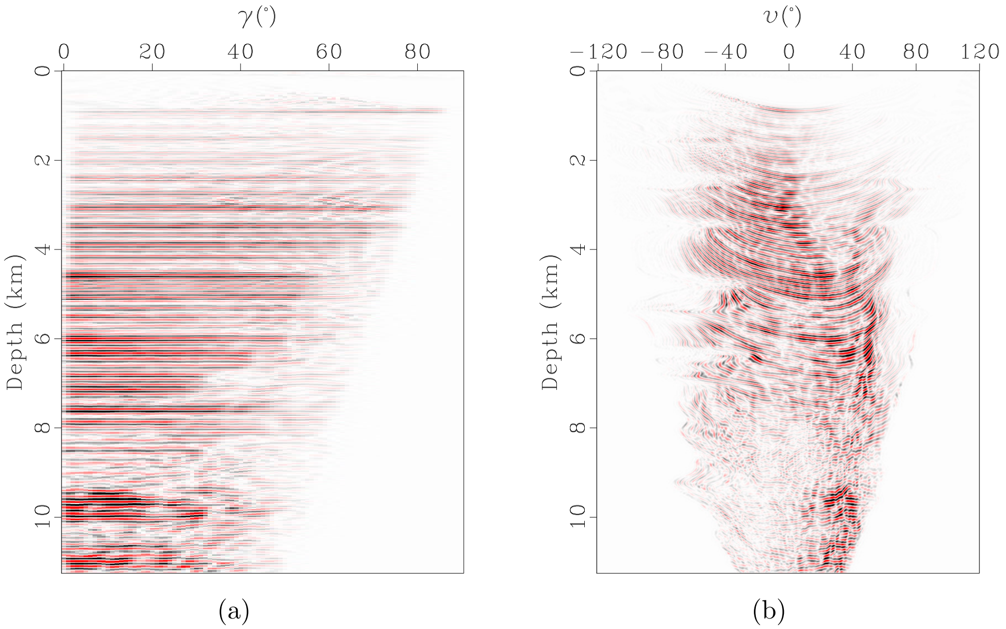
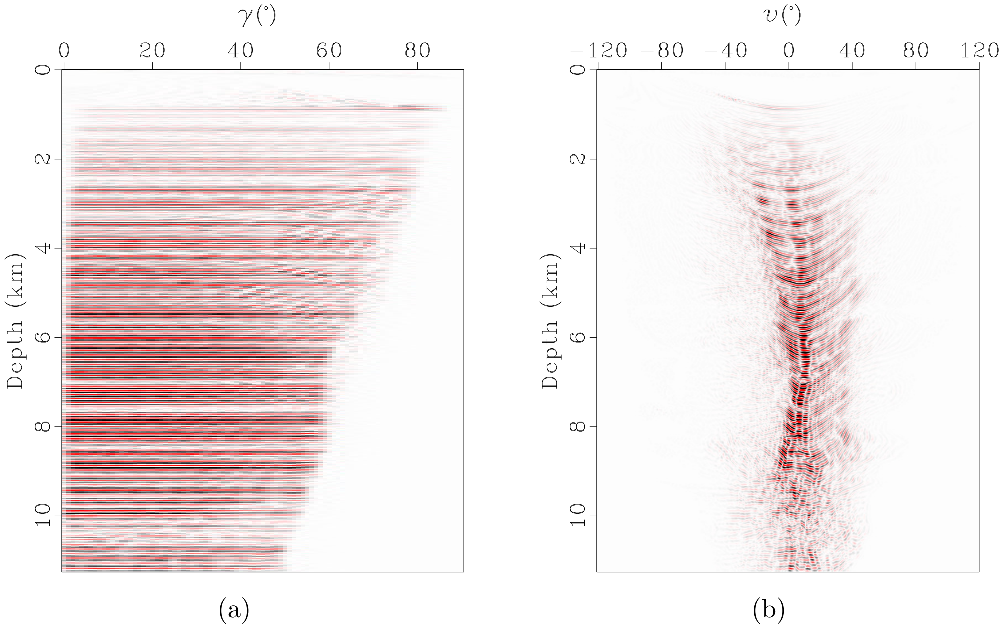


Figure 5.30: BP TTI model scattering (a) and dip (b) angle gathers for lateral position 40.0 km. chapter-imag2/bptti bpttiocig3,bpttidcig3



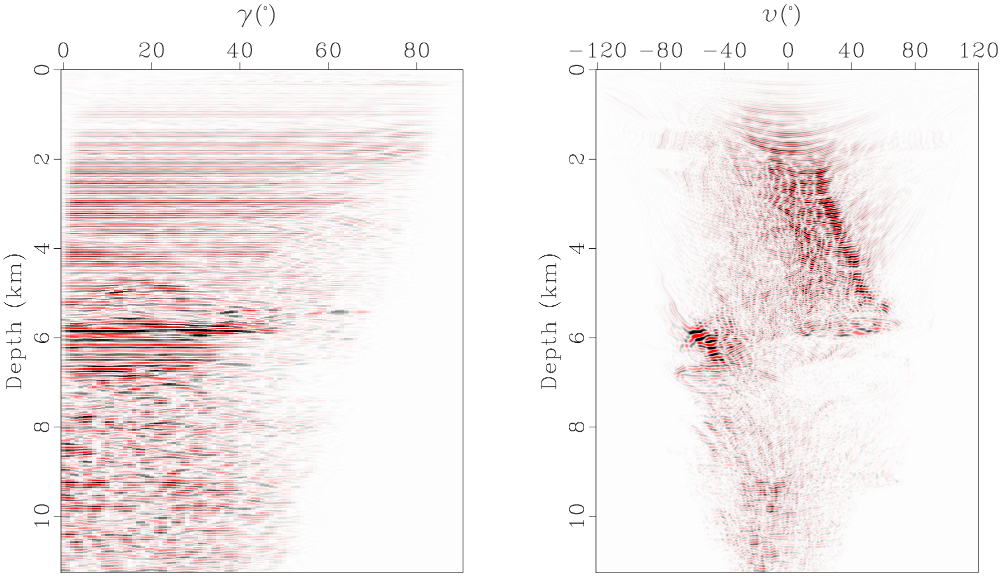
(a) (b)

Figure 5.31: BP TTI model scattering (a) and dip (b) angle gathers for lateral position 50.0 km. chapter-imag2/bptti bpttiocig4,bpttidcig4



(a) (b)

Figure 5.32: BP TTI model scattering (a) and dip (b) angle gathers for lateral position 60.0 km. chapter-imag2/bptti bpttiocig5,bpttidcig5



(a) (b)

Figure 5.33: BP TTI model scattering (a) and dip (b) angle gathers for lateral position 70.0 km. chapter-imag2/bptti bpttiocig6,bpttidcig6

Chapter 6

3-D angle-domain Kirchhoff imaging with escape functions

Imaging using escape functions

In this chapter, I demonstrate how to implement angle-domain migration in 3-D using escape functions as input. Similarly to the 2-D case, the auxiliary information for migration is found out from the input escape time function $\widehat{T}\left(\widehat{\mathbf{x}}, \theta, \phi\right)$ and the input escape position function $\widehat{\mathbf{y}}\left(\widehat{\mathbf{x}}, \theta, \phi\right)$ only.

I demonstrate imaging examples for the same synthetic 3-D salt model used in the previous chapters. Escape tables for the examples are computed using the numerical scheme described in Chapter 4. This new algorithm has several important parameters which control computational time and imaging quality. In fact, there is a tradeoff between them, so it is important to establish an optimal range of their values, which would allow good computational performance without sacrificing imaging accuracy.

It should be noted that the angle-domain migration with "bottom-up" traveltimes has an irregular access pattern to input seismic data. It is image oriented as opposed to the classic data-oriented migration defined as a summation over acquisition coordinates. The irregular data access becomes an engineering challenge in the 3-D case. However, it has been shown that it could be efficiently implemented for large datasets (Ettrich et al., 2008; Koren et al., 2008). In the imaging tests below, the size of the input data is relatively small, so my implementation simply stores a local copy of it at every compute node involved with migration.

Subsurface reflection system

The relationship between source and receiver phase vectors in integral (1.2) and scattering and dip angles for a 3-D media is depicted in Figure 6.1.

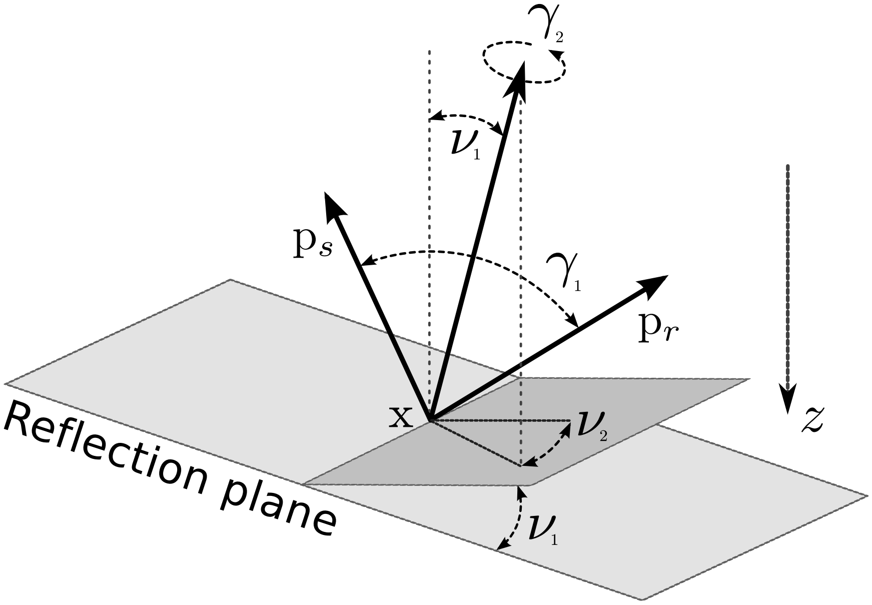


Figure 6.1: 3-D subsurface reflection system, relationship between phase vectors and scattering, dip angles (a scheme). chapter-imag3/. refl3d

The vectors associated with the dip and scattering directions are given respectively as

$$\begin{aligned} \mathbf{p}_{s+r} &= \mathbf{p}_s + \mathbf{p}_r \,, \\ \mathbf{p}_{r-s} &= \mathbf{p}_r - \mathbf{p}_s \,. \end{aligned} \tag{6.1}$$

The dip angle can be defined as

$$\nu_1 = \begin{cases} -\cos^{-1}\left(-\dfrac{\mathbf{p}_{s+r}\cdot\mathbf{n}_z}{|\mathbf{p}_{s+r}|}\right), \; p^y_{s+r} \geq 0 \\ \cos^{-1}\left(-\dfrac{\mathbf{p}_{s+r}\cdot\mathbf{n}_z}{|\mathbf{p}_{s+r}|}\right), \; p^y_{s+r} < 0 \end{cases} , \quad \nu_1 \in [-\pi;\, \pi)\,, \tag{6.2}$$

where $\mathbf{n}_z$ is the unit vector in the direction of axis z. The dip azimuth angle is

$$\nu_2 = \begin{cases} \cos^{-1}\left(\dfrac{\mathbf{p}_{s+r}^{xy} \cdot \mathbf{n}_x}{|\mathbf{p}_{s+r}^{xy}|}\right), \; p_{s+r}^{y} \geq 0 \\ \pi - \cos^{-1}\left(\dfrac{\mathbf{p}_{s+r}^{xy} \cdot \mathbf{n}_x}{|\mathbf{p}_{s+r}^{xy}|}\right), \; p_{s+r}^{y} < 0 \end{cases} , \quad \nu_2 \in [0;\, \pi)\,, \tag{6.3}$$

where $\mathbf{p}_{s+r}^{xy}$ is the projection of vector $\mathbf{p}_{s+r}$ onto $x-y$ plane and $\mathbf{n}_x$ is the unit vector in the direction of axis x.

The scattering angle can be defined as

$$\gamma_1 = \cos^{-1}\left(\frac{\mathbf{p}_s \cdot \mathbf{p}_r}{|\mathbf{p}_s|\,|\mathbf{p}_r|}\right), \quad \gamma_1 \in [\,0;\, \pi). \tag{6.4}$$

Following Sava and Fomel (2005), I define the following projections onto the reflection plane

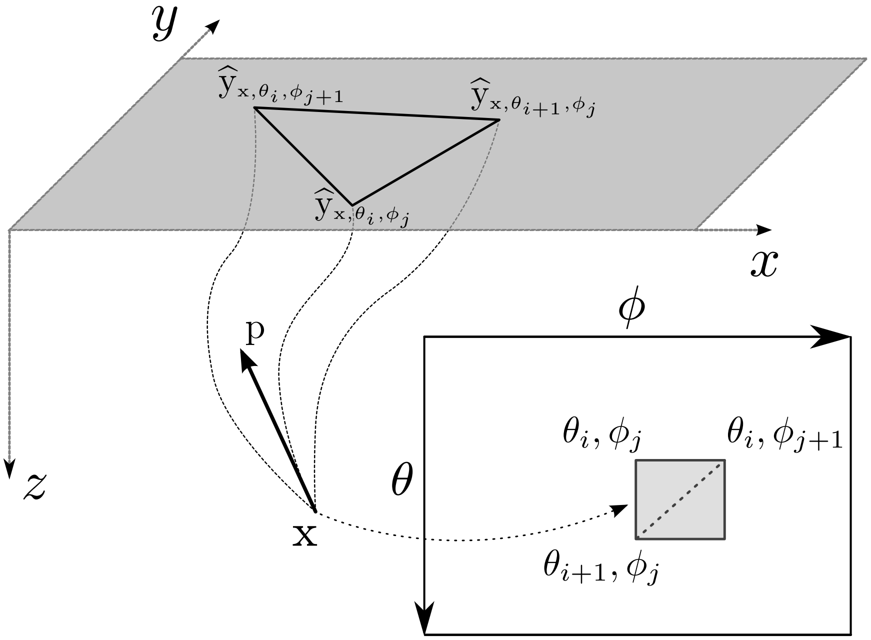
$$\begin{aligned} \mathbf{v} &= \mathbf{p}_{s+r} \times \mathbf{n}_x\,, \\ \mathbf{u} &= \mathbf{p}_{s+r} \times \mathbf{p}_{r-s}\,. \end{aligned} \tag{6.5}$$

The scattering azimuth angle is then defined as

$$\gamma_2 = \begin{cases} \cos^{-1}\left(\dfrac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}|\,|\mathbf{v}|}\right), \; (\mathbf{u} \times \mathbf{v}) \cdot \mathbf{p}_{s+r} \geq 0 \\ 2\pi - \cos^{-1}\left(\dfrac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}|\,|\mathbf{v}|}\right), \; (\mathbf{u} \times \mathbf{v}) \cdot \mathbf{p}_{s+r} < 0 \end{cases} , \quad \gamma_2 \in [0;\, 2\pi)\,. \tag{6.6}$$

For a fixed image point in 3-D case, the corresponding angular part of the escape tables defines a grid of exit location and exit times. One cell in the grid contains four sets of escape variables. Any three of them form an exit triangle on the surface (Figure 6.2), which I choose as a basis element for finding imaging contributions from surface seismic data.

The summation process (Figure 6.3) is similar to that of the 2-D case, but instead of exit segments, exit triangles are analyzed. For a given source or receiver



y
$\widehat{y}_{x,\theta_i,\phi_{j+1}}$
$\widehat{y}_{x,\theta_{i+1},\phi_j}$
$\widehat{y}_{x,\theta_i,\phi_j}$
x
ϕ
p
θ_i, ϕ_j
θ_i, ϕ_{j+1}
z
θ
x
θ_{i+1}, ϕ_j

Figure 6.2: Exit triangle defined by a set of escape values in the escape tables (a scheme). $\boxed{\text{chapter-imag3/. pang3d}}$

position on the surface, corresponding exit triangles are found that contain that point. To avoid overly excessive search process, all the exit triangles are sorted into search bins in advance. Only a fraction of all the precomputed triangles has to be checked against containing the point. For every found triangle, I use linear interpolation in barycentric coordinates to estimate the escape quantity at the point.

Phase shifts due to caustics

In 3-D, a ray tube can go through two types of caustics. Effects of both types can be recognized through a simple analysis of the escape triangle. I use the following definitions for the sides of the triangle:

$$\begin{aligned} \mathbf{a} &= |\widehat{\mathbf{y}}_{i+1,j} - \widehat{\mathbf{y}}_{i,j}| \ , \\ \mathbf{b} &= |\widehat{\mathbf{y}}_{i,j+1} - \widehat{\mathbf{y}}_{i,j}| \ , \end{aligned} \tag{6.7}$$

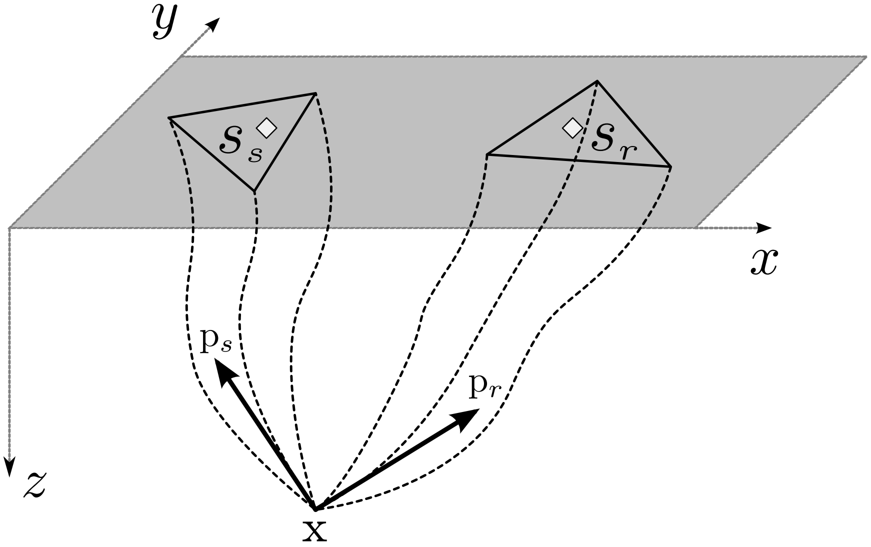


Figure 6.3: 3-D angle-domain migration using escape tables (a scheme). chapter-imag3/. cram3d

A caustic point of the first type, which corresponds to the increase of KMAH index by one, makes the product $\mathbf{a} \cdot \mathbf{b}$ negative (Figure 6.4). The second type of caustic (KMAH index of two) causes both vectors to flip, so a different dot product has to be looked at - the one between the current azimuth vector and whichever vector of the two corresponds to the constant-azimuth direction in the angular escape grid (Figure 6.4).

Antialiasing and amplitude weights

Exit triangles approximate locally the diffraction time-space surface along which the summation is performed. For the triangle filter in antialiasing procedure, a slope of the surface is required, which can be computed from two directional traveltime derivatives along the sides of an escape triangle. Using normalized versions of

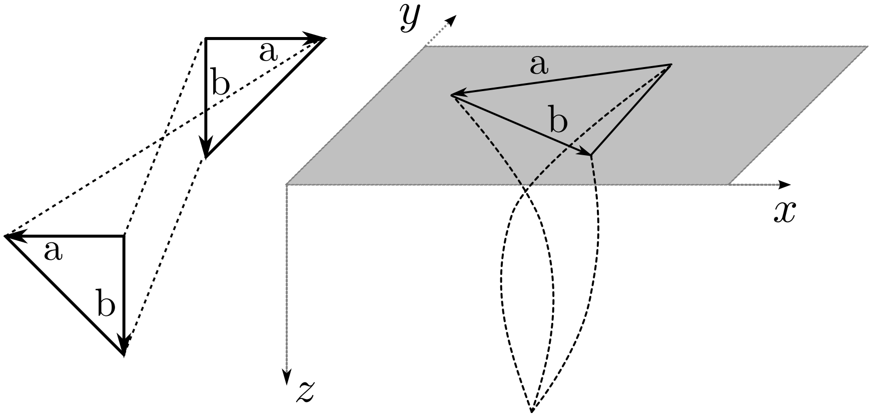


Figure 6.4: Escape triangle corresponding to the travelpath though a caustic of the first type (a scheme). chapter-imag3/. kmah3d1

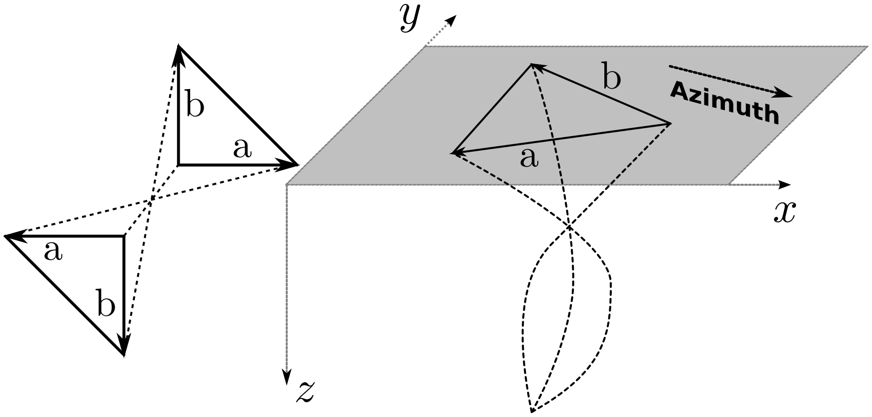


Figure 6.5: Escape triangle corresponding to the travelpath though a caustic of the second type (a scheme). chapter-imag3/. kmah3d2

vectors **a** and **b**, components of the slope vector can be found as

$$\left| \begin{array}{c} p^{x}_{exit} \\ p^{y}_{exit} \end{array} \right| = \left| \begin{array}{cc} \dfrac{a_x}{|\mathbf{a}|} & \dfrac{a_y}{|\mathbf{a}|} \\ \dfrac{b_x}{|\mathbf{b}|} & \dfrac{b_y}{|\mathbf{b}|} \end{array} \right|^{-1} \left| \begin{array}{c} \dfrac{\widehat{T}_{i+1,j} - \widehat{T}_{i,j}}{|\mathbf{a}|} \\ \dfrac{\widehat{T}_{i,j+1} - \widehat{T}_{i,j}}{|\mathbf{b}|} \end{array} \right| . \tag{6.8}$$

The geometrical spreading required for amplitude weights can be approximated as

$$J_{exit} \simeq s_{\Delta} \sqrt{1 - \left(\frac{|p_{exit}|}{S_{exit}} \right)^2} , \tag{6.9}$$

where S_{exit} is the slowness at the surface at the exit location and s_{Δ} is the area of the exit triangle.

Other considerations for 3-D imaging

If exit triangle sizes become small on the surface compared to the sampling of the acquisition system, then some angle bins might be left empty in the migration process. This effect is especially pronounced for sparse shot sampling and image locations that are close to the surface. This situation is not something unique to angle-domain Kirchhoff migration, but is also encountered in other imaging methods (Tang et al., 2011). To avoid loss of image quality, angle bin spacing needs to be adjusted with depth and the image has to be interpolated later onto the desired angle grid.

Imaging approximations based on high-frequency asymptotics are known to break down in very complex models, i.e. when velocity variations demonstrate high gradients. In such cases, some contributions to the image may be erroneous to the extent that they dominate the constructive part of the sum. This usually happens only in a few places and can be seen as noise on gathers after imaging. Noise attenuation measures, such as bandpass filtering, should be applied to gathers afterwards. Muting in the angle plane, akin to that shown in the previous chapter, may also be necessary. Large exit triangles should be discarded as they can not approximate the diffraction summation surface correctly.

SEG/EAGE Salt model imaging tests

I conducted a series of imaging experiments using an implementation of 3-D angle-domain migration based upon the principles that I described above. The goal of these experiments was to find an optimal set of parameters for the semi-Lagrangian algorithm for computing escape tables. I use imaging results based on ray tracing as a reference. The best set of parameters for the semi-Lagrangian counterpart should yield migration results of comparable quality while achieving a gain in computational efficiency. I use the same salt model as in Chapter 4. Prestack data is a full-azimuth 45-shot dataset with sparse shot sampling (0.96 km in both directions).

The semi-Lagrangian algorithm from Chapter 4 has several parameters which affect accuracy of imaging and/or computational efficiency of escape tables computation:

- Maximum displacement distance in local escape functions d_{max}.
- Spatial sampling of local escape functions $\Delta_{x,y,z}$ (for simplicity, I use same sampling for all of the three axes).
- Angular sampling of local escape functions $\Delta_{\theta,\phi}$ (again, for simplicity purposes, I use same sampling for azimuth and inclination).

To find an optimal combination of these parameters, I performed a series of imaging tests for the same salt model. I use one location in the model that has good illumination in the subsalt area for the modeled acquisition system. Using this same location, I generate scattering angle gathers for different combinations of d_{max}, $\Delta_{x,y,z}$, and $\Delta_{\theta,\phi}$. I then compare these results against a ray tracing based one. All of the results are presented below in 18 figures. I use 6 for establishing references between values of the three parameters and each individual figure. Each figure contains four panels as follows (from left to right):

1. A scattering angle gather computed for a given set of parameters.
2. Local similarity (Fomel, 2007) computed against the ray tracing result.
3. Comparison of stack traces: solid line - ray tracing, dashed line - iterative reconstruction.
4. Local similarity between the two traces.

From these experiments, I find that image quality in the shallow part of the gather allows greater values for $\Delta_{x,y,z}$ and $\Delta_{\theta,\phi}$ than the upper limit of 100 m and 10°. It can possibly accept bigger d_{max} step as well. The bottom of the salt body, however, starts to fall apart at these limits. The subsalt is then the most sensitive part of the image to the parameterization. It appears that to capture phases of the subsalt reflectors correctly, $\Delta_{x,y,z}$ must be less than 75 m, $\Delta_{\theta,\phi}$ must be less than 10°, and the value of d_max must be in the vicinity of 0.5 km.

Figures 6.24 and 6.25 show stacked images of a target line obtained from migration with escape tables built using ray tracing and iterative reconstruction in

	$d_{max} = 0.25$ km	$d_{max} = 0.5$ km	$d_{max} = 0.75$ km
$\Delta_{x,y,z} = 50$ m, $\Delta_{\theta,\phi} = 5°$	Figure 6.6	Figure 6.7	Figure 6.8
$\Delta_{x,y,z} = 50$ m, $\Delta_{\theta,\phi} = 10°$	Figure 6.9	Figure 6.10	Figure 6.11
$\Delta_{x,y,z} = 75$ m, $\Delta_{\theta,\phi} = 5°$	Figure 6.12	Figure 6.13	Figure 6.14
$\Delta_{x,y,z} = 75$ m, $\Delta_{\theta,\phi} = 10°$	Figure 6.15	Figure 6.16	Figure 6.17
$\Delta_{x,y,z} = 100$ m, $\Delta_{\theta,\phi} = 5°$	Figure 6.18	Figure 6.19	Figure 6.20
$\Delta_{x,y,z} = 100$ m, $\Delta_{\theta,\phi} = 10°$	Figure 6.21	Figure 6.22	Figure 6.23

Table 6.1: Correspondence between parameters and figures in the imaging accuracy test for the semi-Lagrangian algorithm.

phase space respectively. Local escape functions for the second image are sampled with $\Delta_{x,y,z} = 60$ m and $\Delta_{\theta,\phi} = 5°$; d_{max} is 0.5 km. Comparison of the two images (Figure 6.26) shows a very good agreement in the upper part and around the salt body. Both imaging methods capture subsalt reflectors (where illumination allows it) with only slight disagreement in the amplitudes. If escape functions are accessed locally during computations, such parameterization allows leads to an order of magnitude of improvement over ray tracing in compute time for the iterative reconstruction approach.

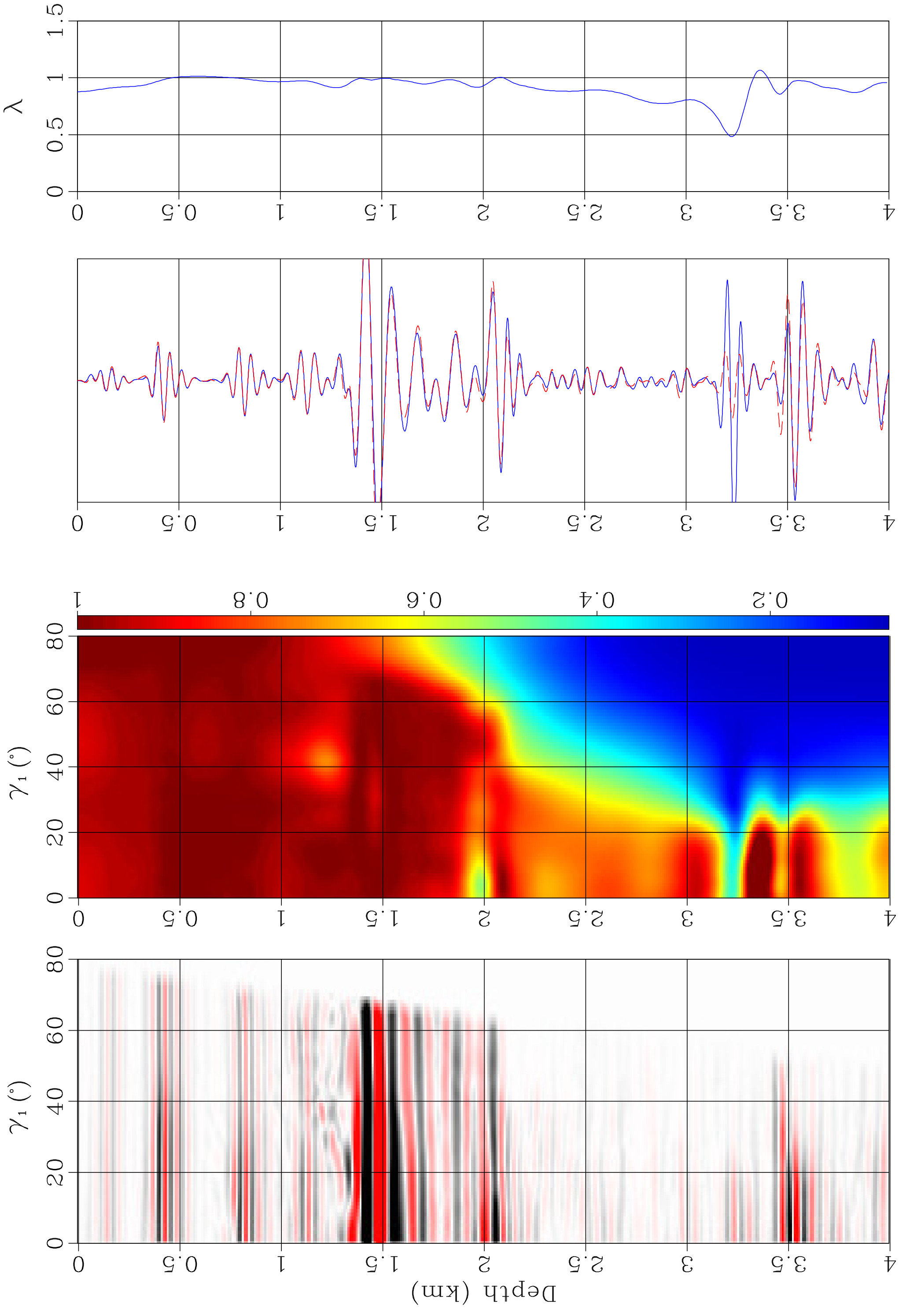


Figure 6.6: Local escape function sampling in space is 50 m, angle - 5°. $d_{max} = 0.25$ km. chapter-imag3/ssaltg ssaltsoamig000

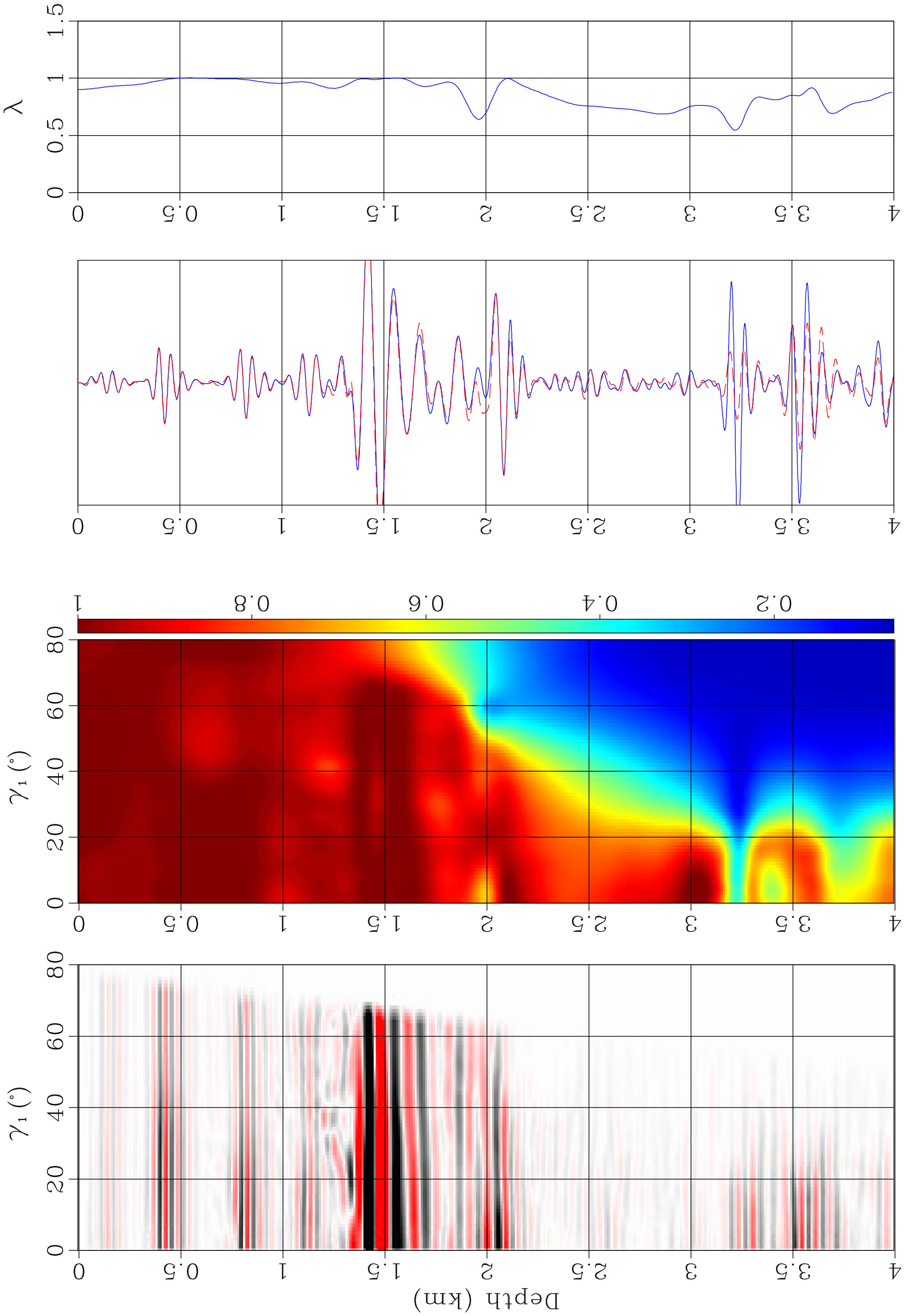


Figure 6.7: Local escape function sampling in space is 50 m, angle - 5°. $d_{max} = 0.5$ km. $\boxed{\text{chapter-imag3/ssaltg ssaltsoamig001}}$

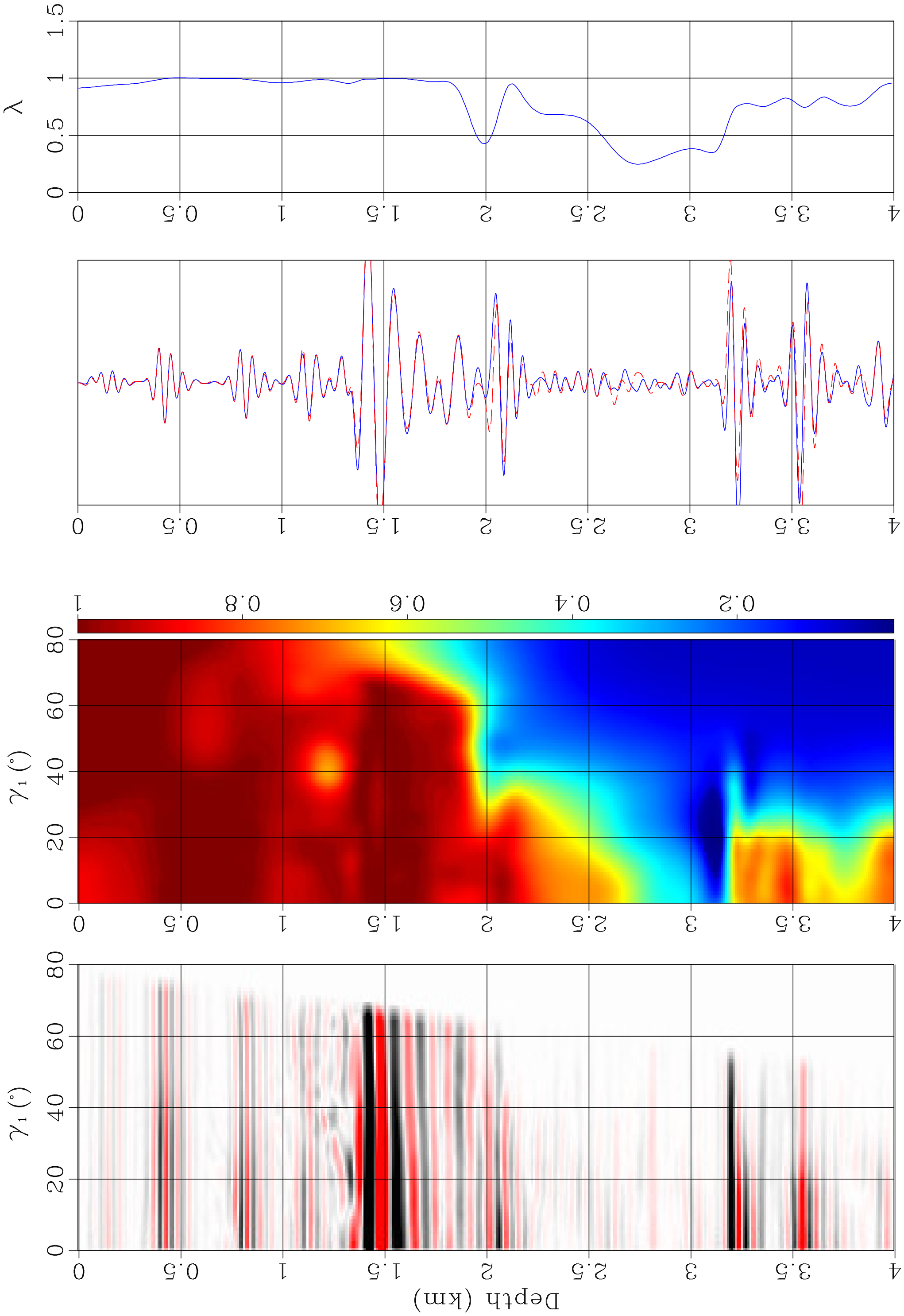


Figure 6.8: Local escape function sampling in space is 50 m, angle - 5°. $d_{max} = 0.75$ km. chapter-imag3/ssaltg ssaltsoamig002

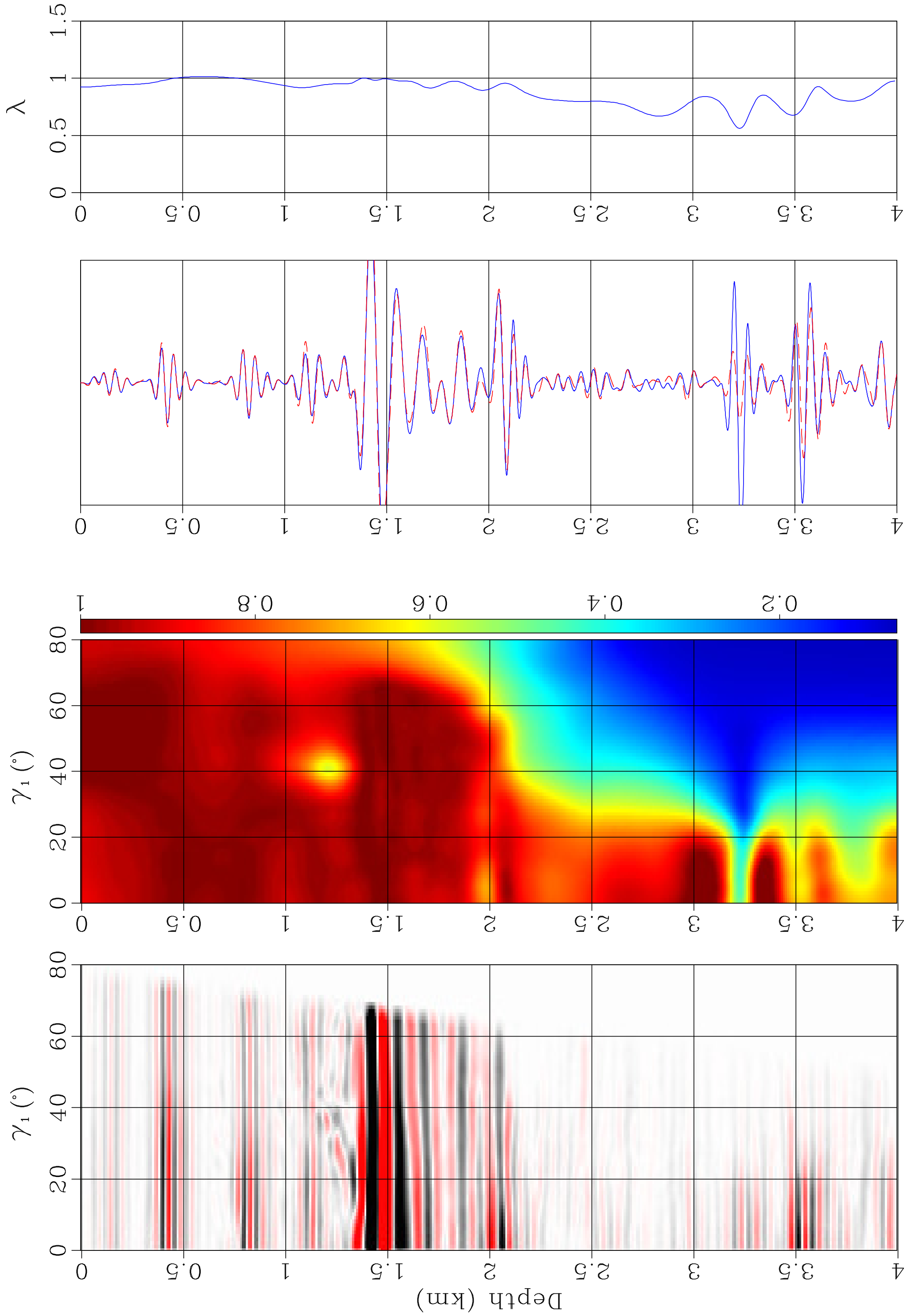


Figure 6.9: Local escape function sampling in space is 50 m, angle - 10°. $d_{max} = 0.25$ km. chapter-imag3/ssaltg ssaltsoamig010

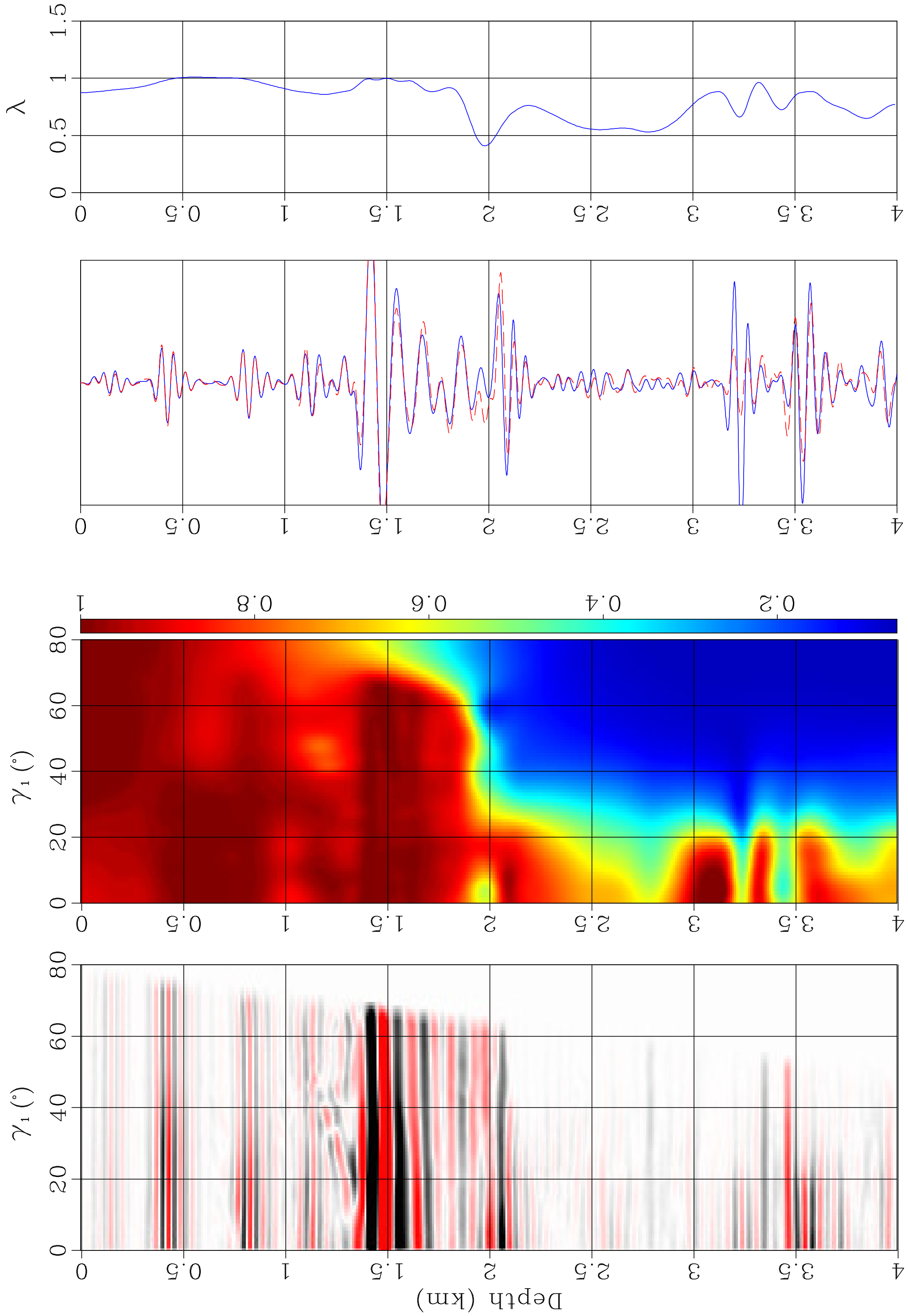


Figure 6.10: Local escape function sampling in space is 50 m, angle - 10°. $d_{max} = 0.5$ km. chapter-imag3/ssaltg ssaltsoamig011

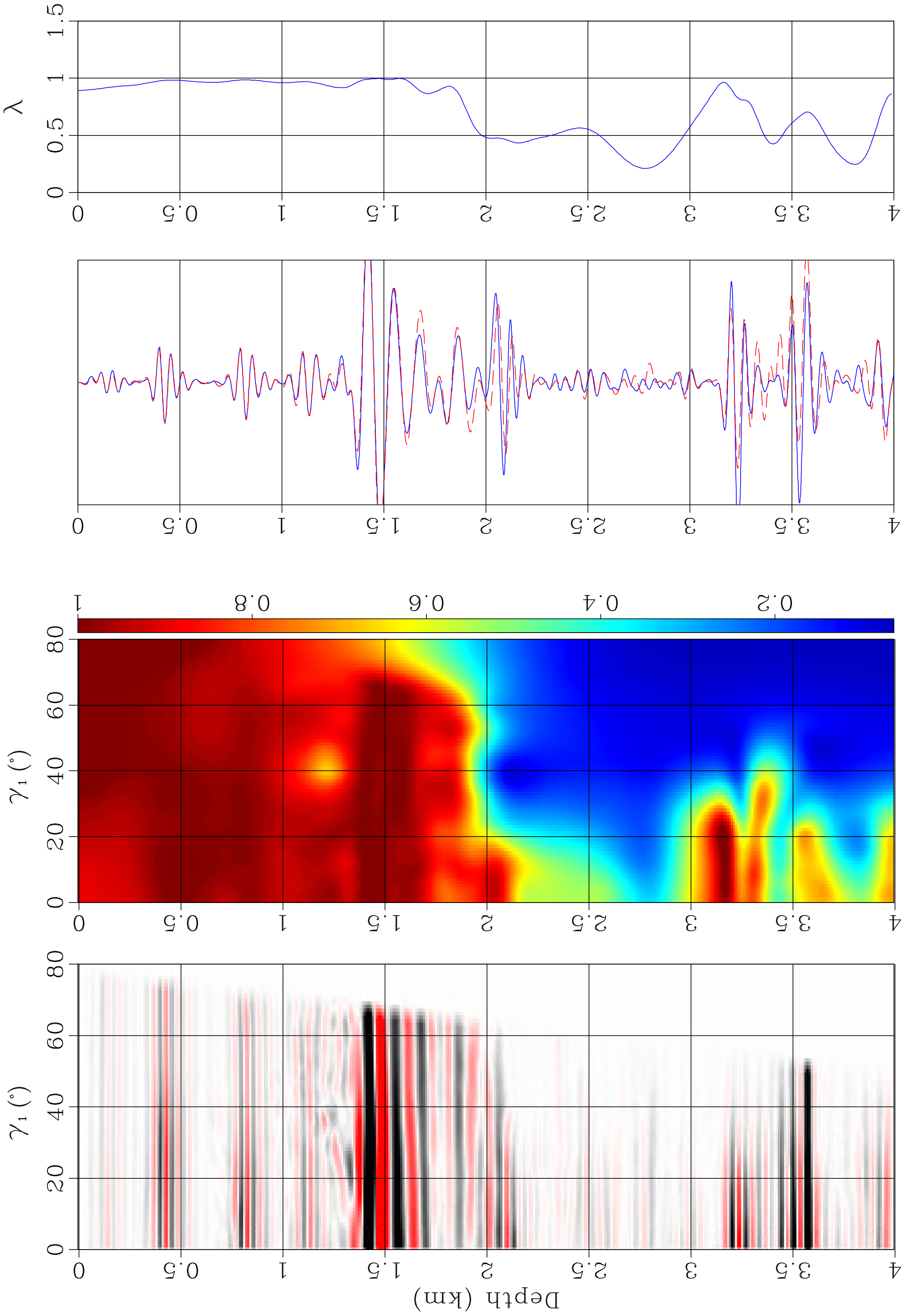


Figure 6.11: Local escape function sampling in space is 50 m, angle - 10°. $d_{max} = 0.75$ km. chapter-imag3/ssaltg ssaltsoamig012

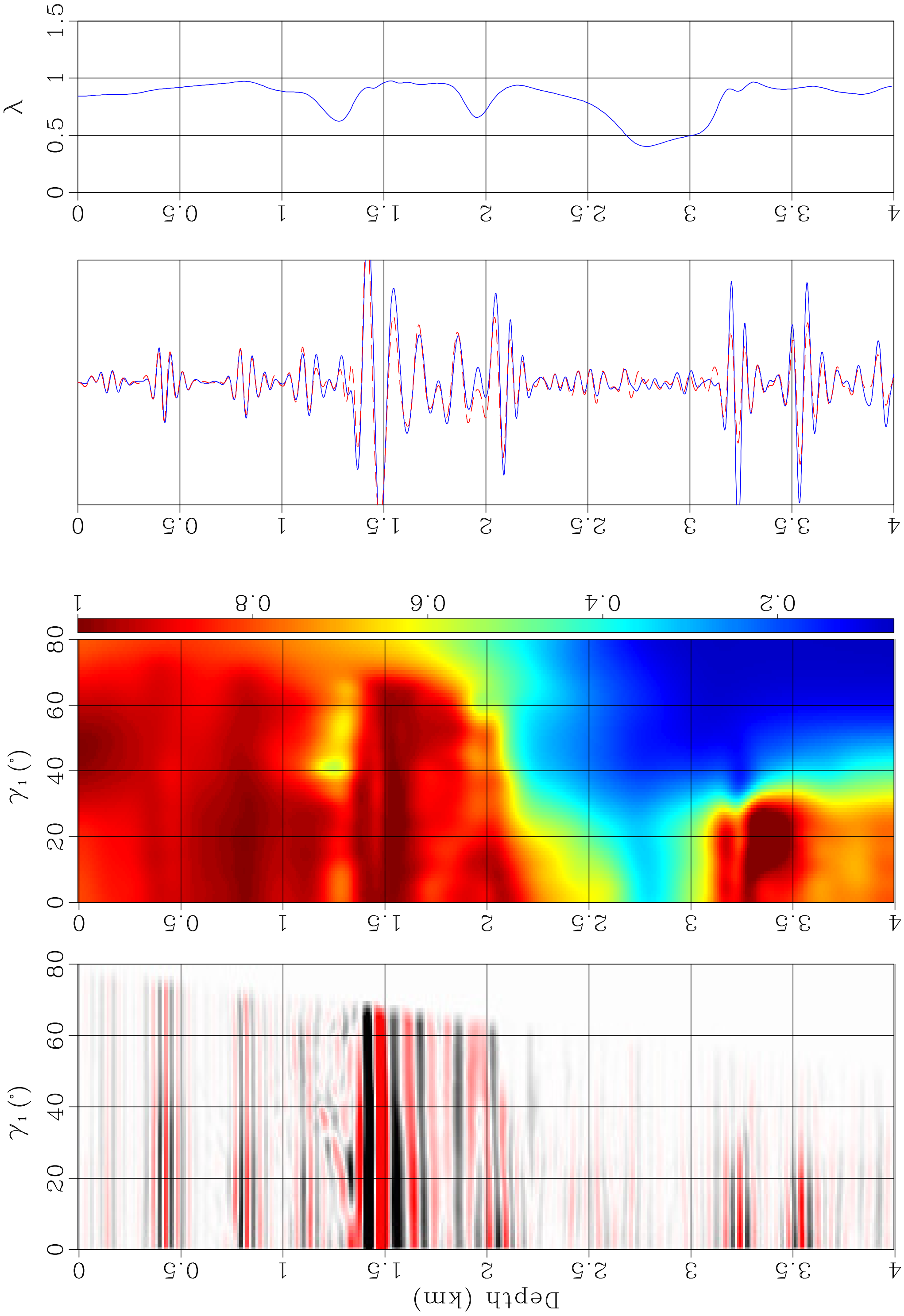


Figure 6.12: Local escape function sampling in space is 75 m, angle - 5°. $d_{max} = 0.25$ km. chapter-imag3/ssaltg ssaltsoamig100

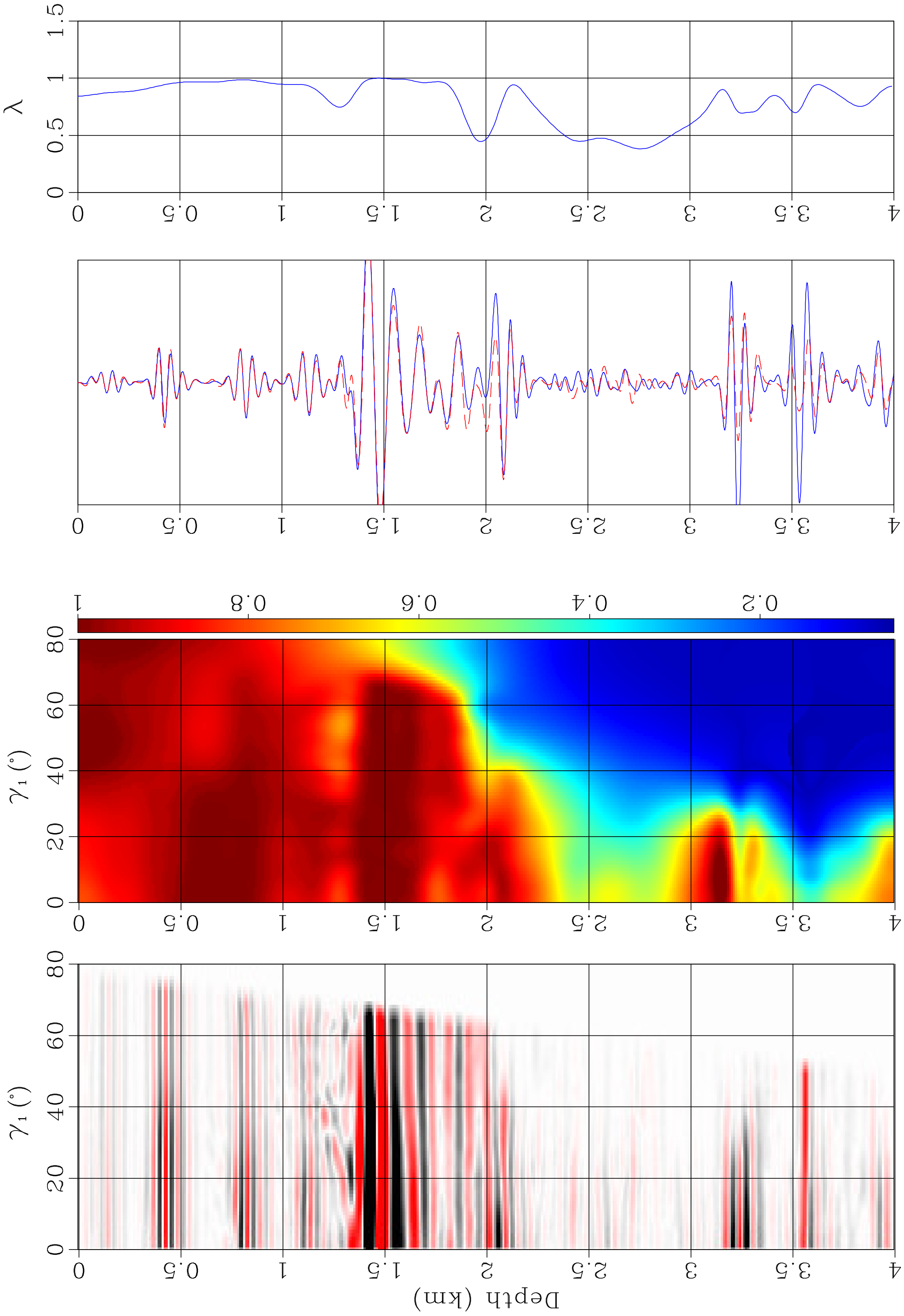


Figure 6.13: Local escape function sampling in space is 75 m, angle - 5°. $d_{max} = 0.5$ km. chapter-imag3/ssaltg ssaltsoamig101

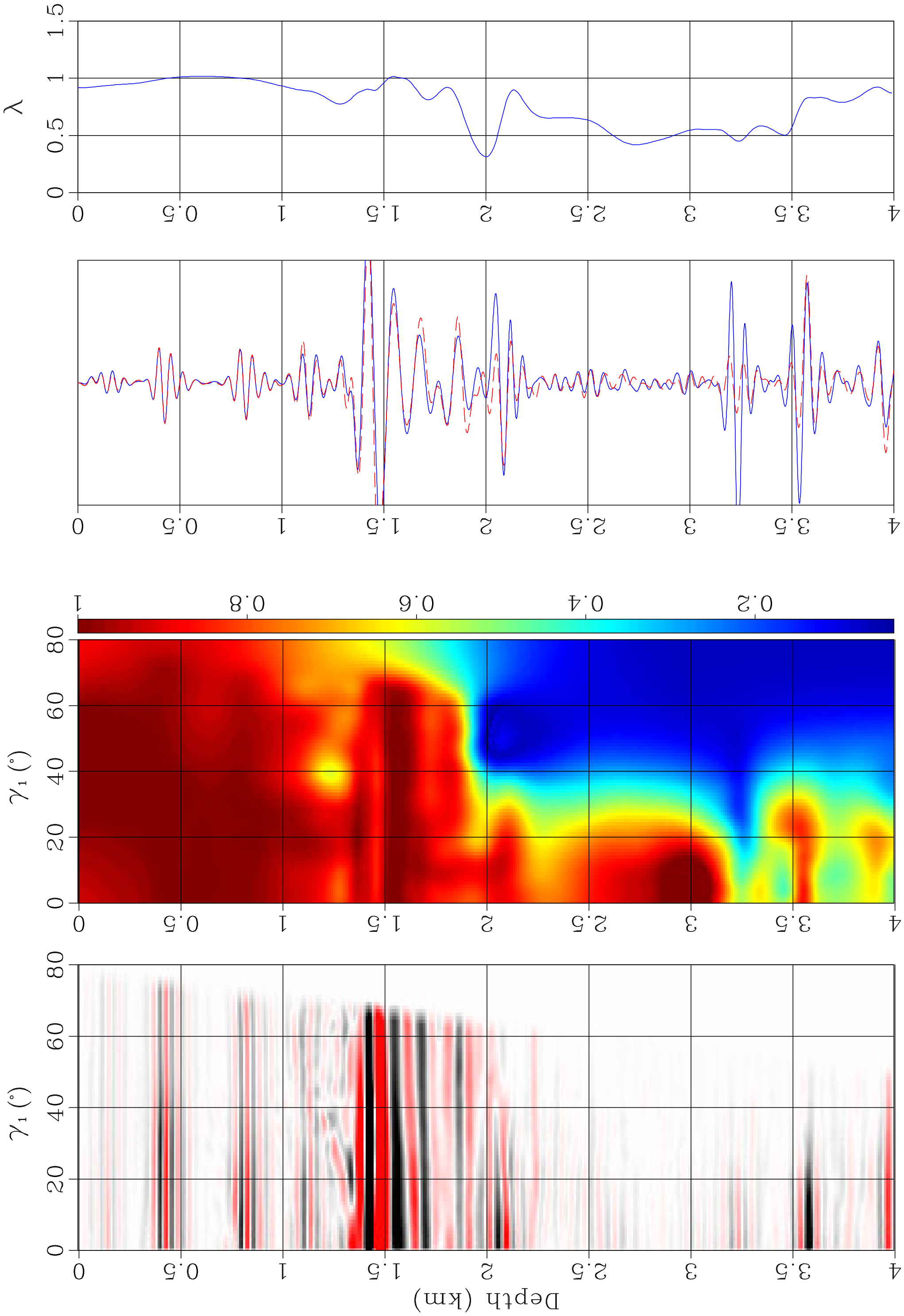


Figure 6.14: Local escape function sampling in space is 75 m, angle - 5°. $d_{max} = 0.75$ km. chapter-imag3/ssaltg ssaltsoamig102

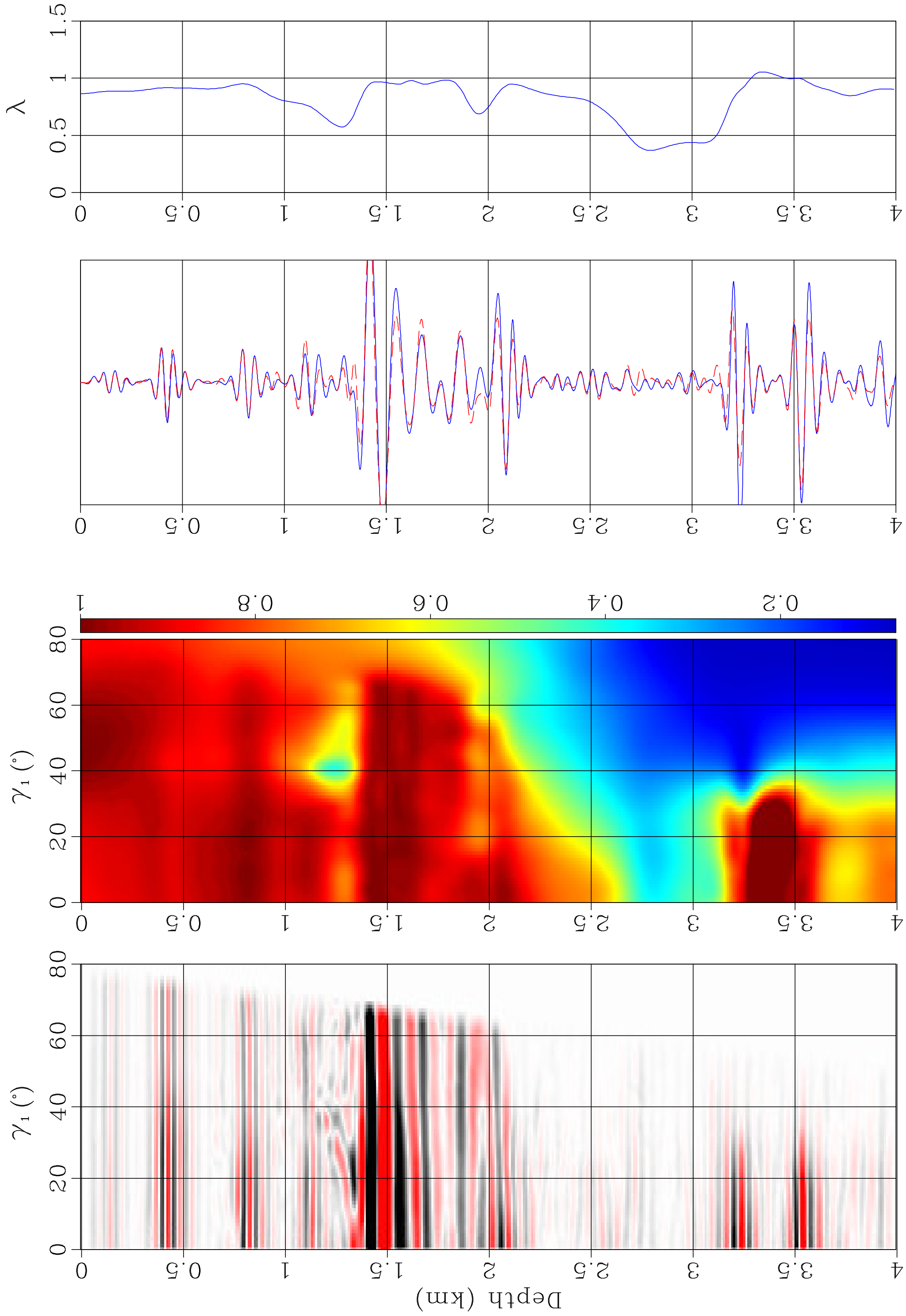


Figure 6.15: Local escape function sampling in space is 75 m, angle - 10°. $d_{max} = 0.25$ km. chapter-imag3/ssaltg ssaltsoamig110

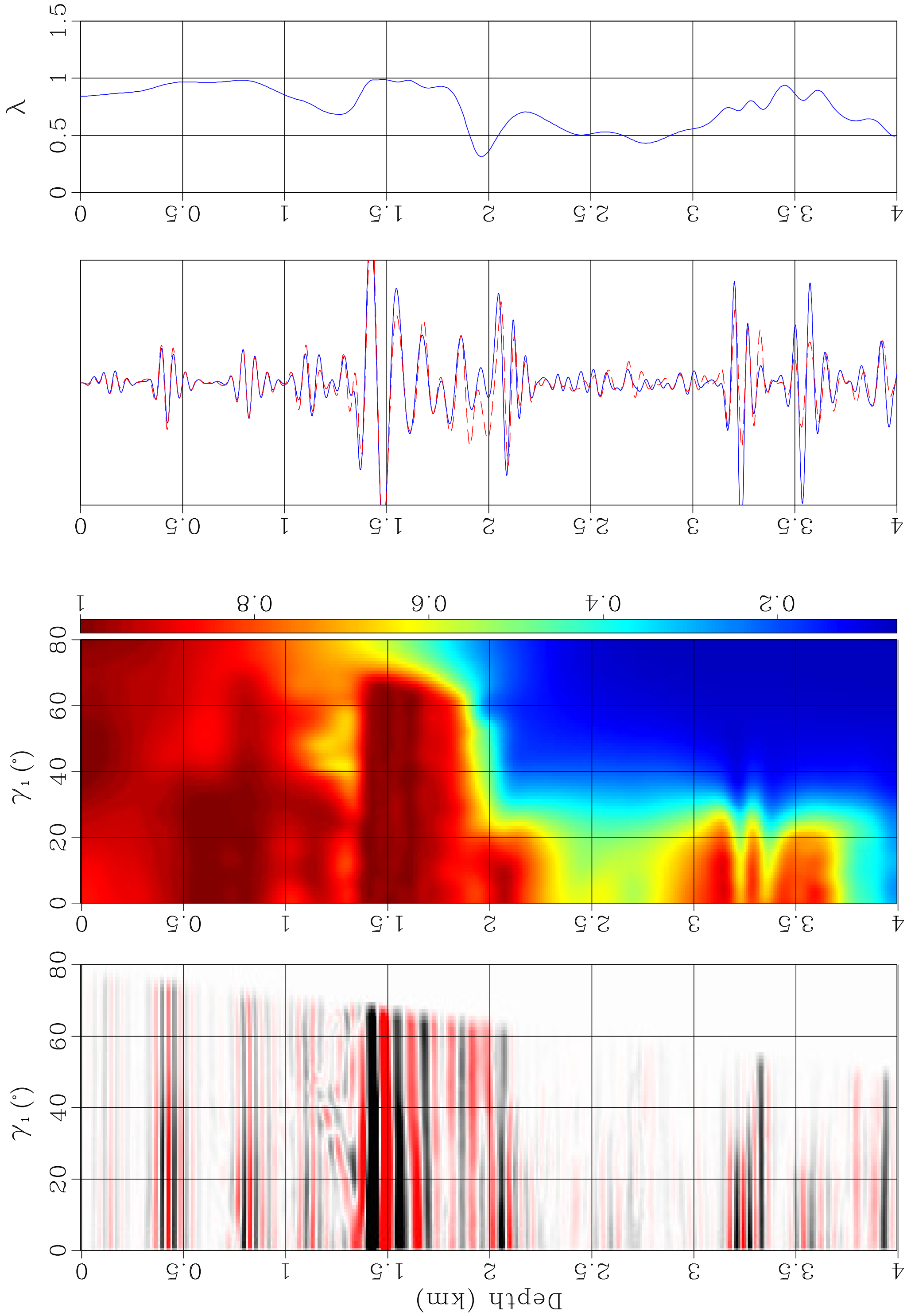


Figure 6.16: Local escape function sampling in space is 75 m, angle - 10°. $d_{max} = 0.5$ km. chapter-imag3/ssaltg ssaltsoamig111

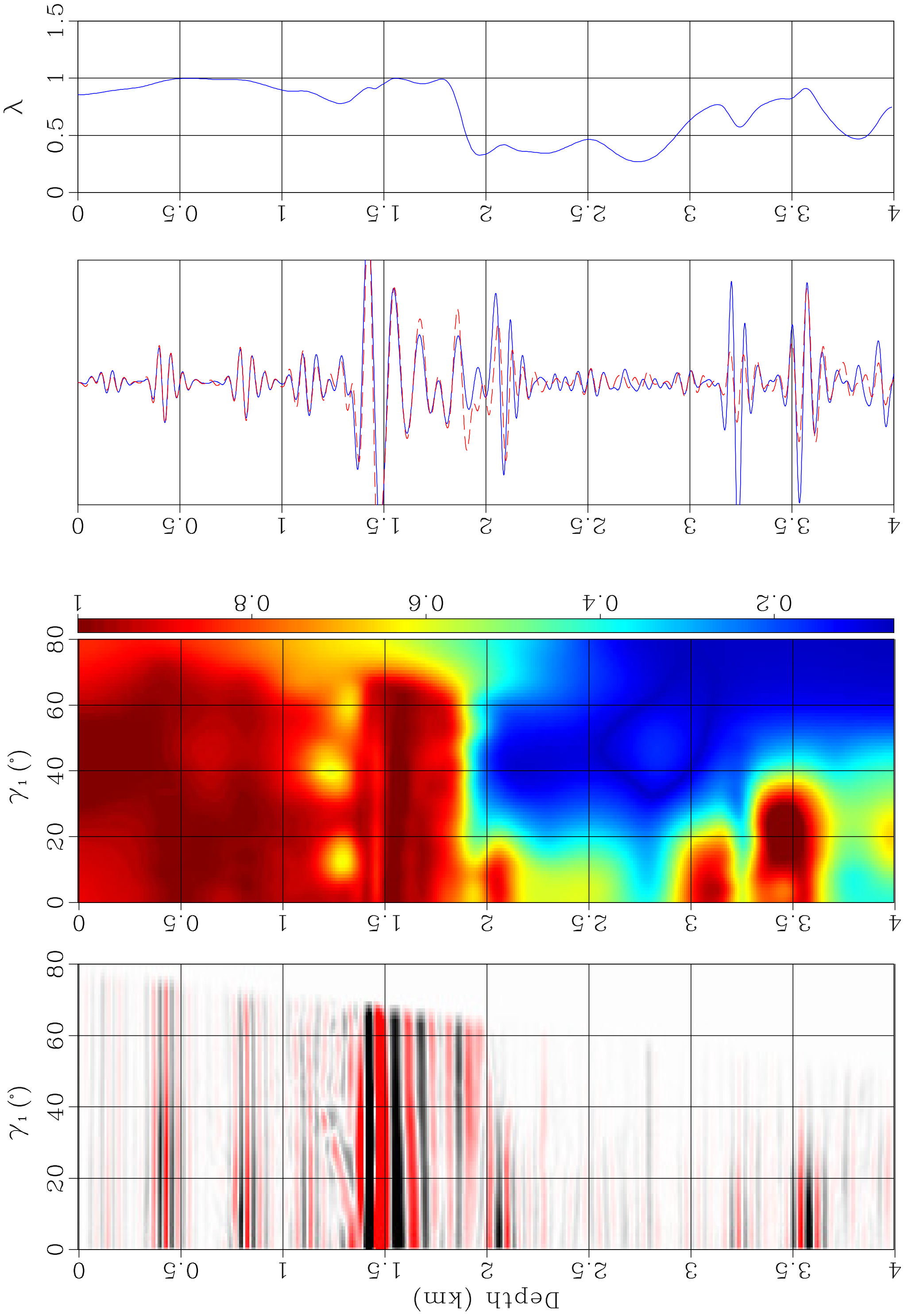


Figure 6.17: Local escape function sampling in space is 75 m, angle - 10°. $d_{max} = 0.75$ km. chapter-imag3/ssaltg ssaltsoamig112

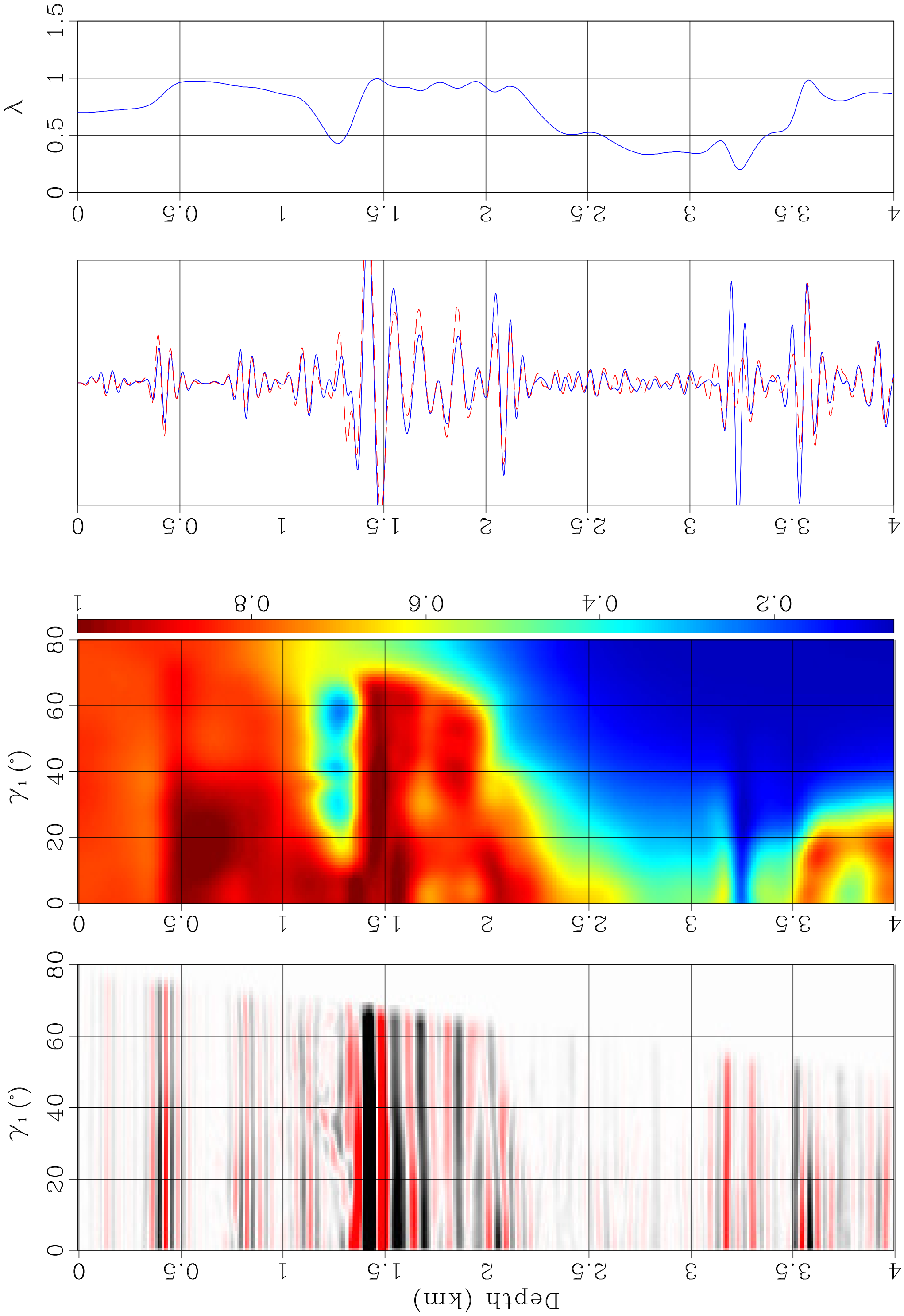


Figure 6.18: Local escape function sampling in space is 100 m, angle - 5°. $d_{max} = 0.25$ km. chapter-imag3/ssaltg ssaltsoamig200

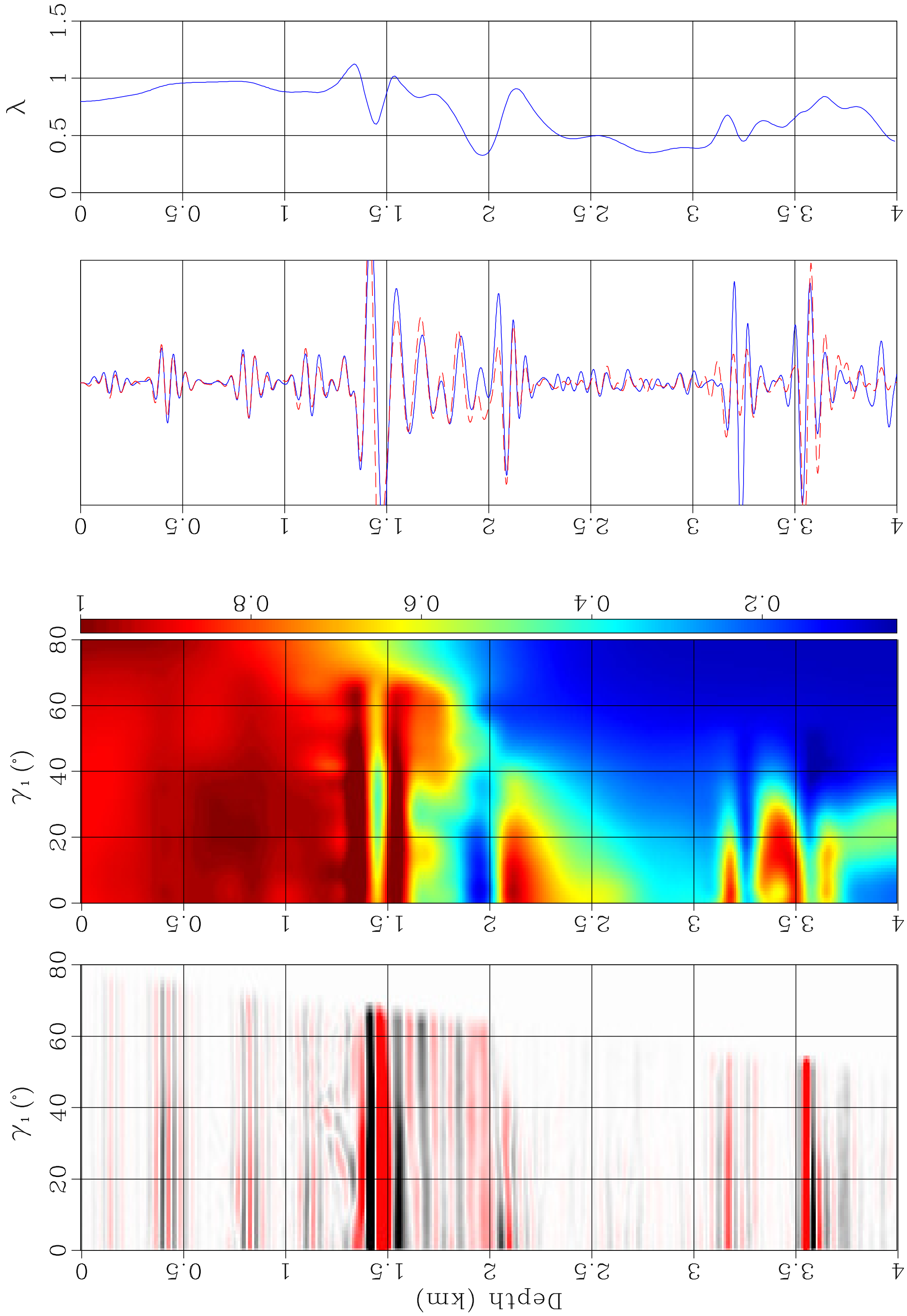


Figure 6.19: Local escape function sampling in space is 100 m, angle - 5°. $d_{max} = 0.5$ km. chapter-imag3/ssaltg ssaltsoamig201

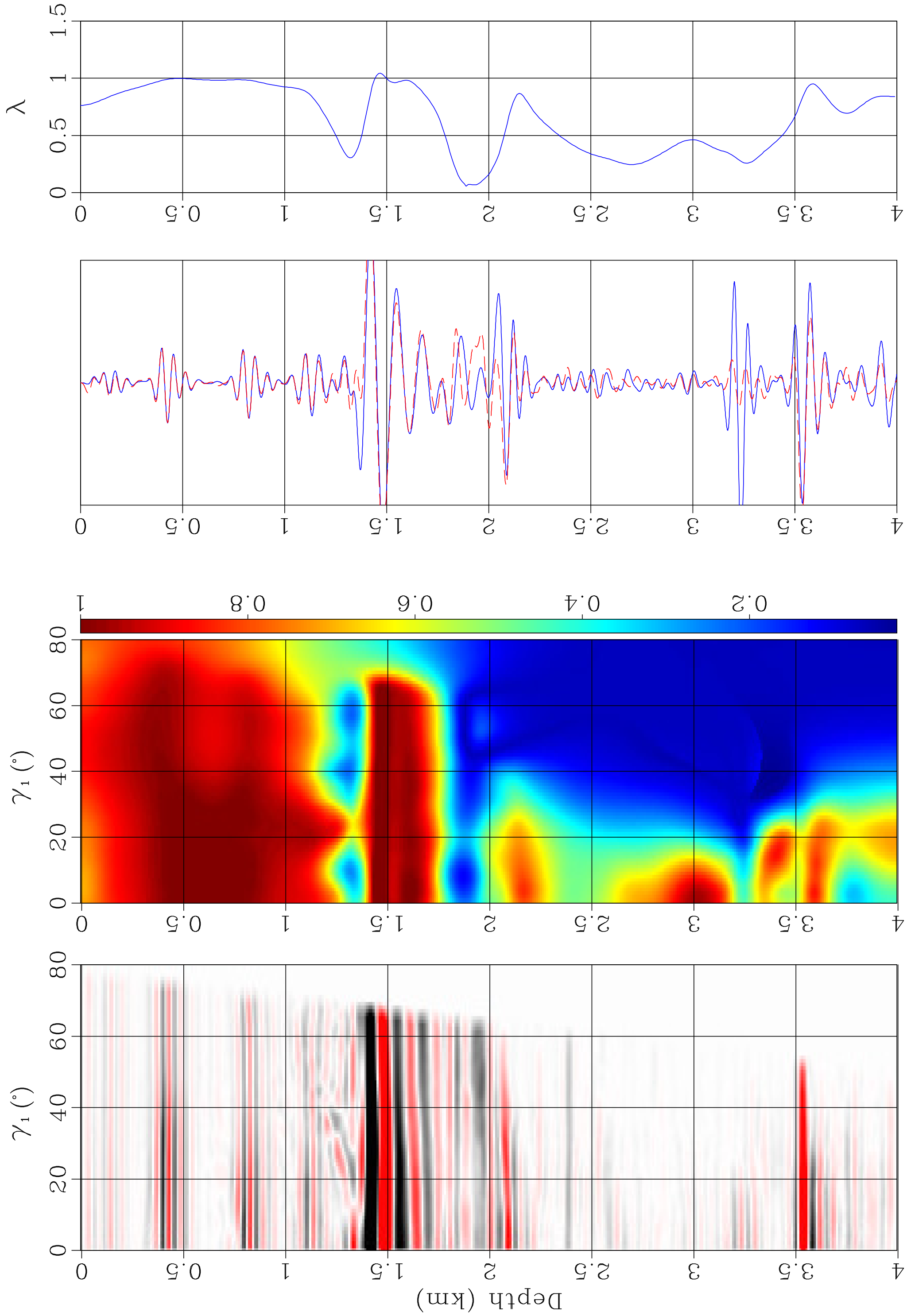


Figure 6.20: Local escape function sampling in space is 100 m, angle - 5°. $d_{max} = 0.75$ km. chapter-imag3/ssaltg ssaltsoamig202

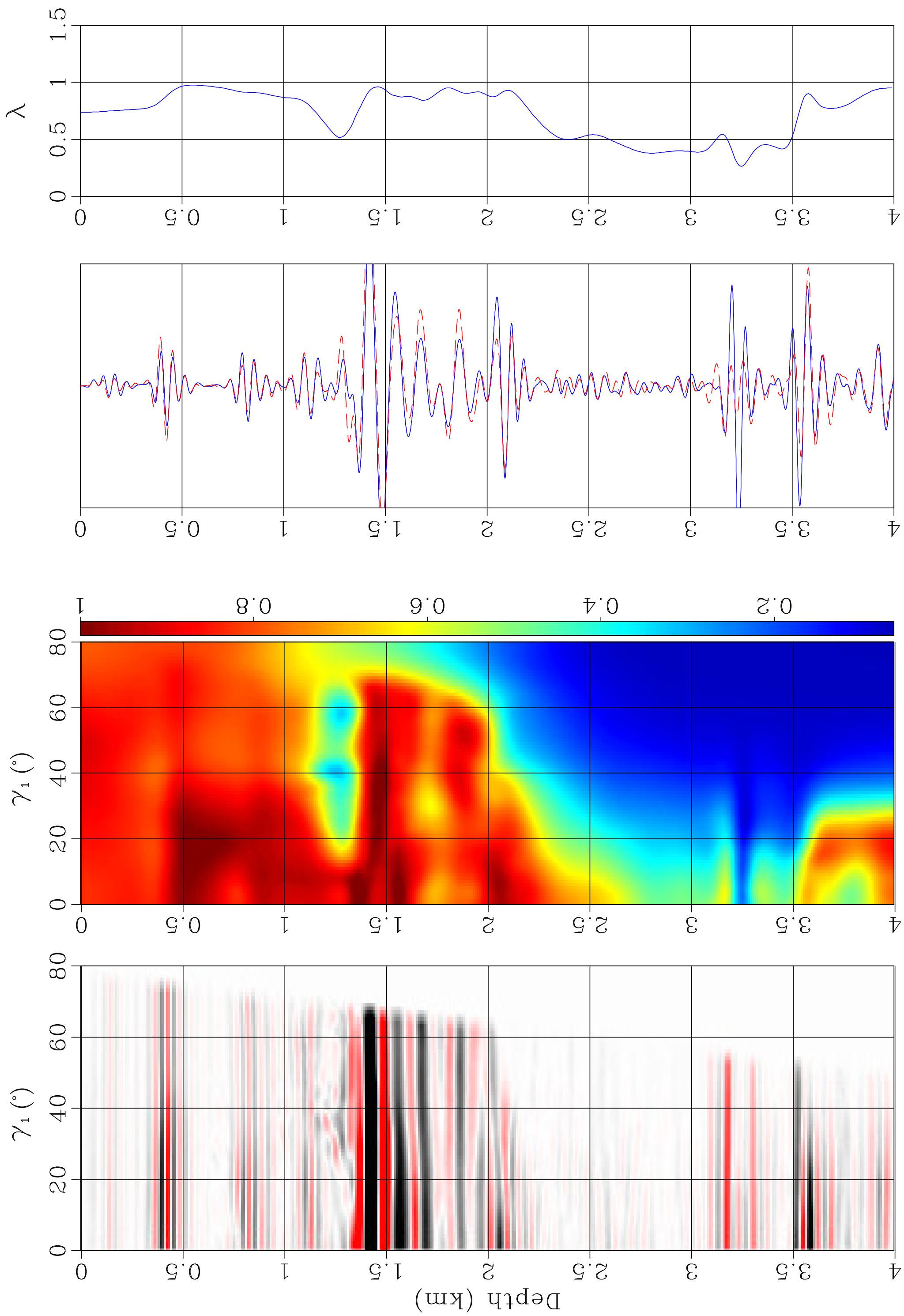


Figure 6.21: Local escape function sampling in space is 100 m, angle - 10°. $d_{max} = 0.25$ km. chapter-imag3/ssaltg ssaltsoamig210

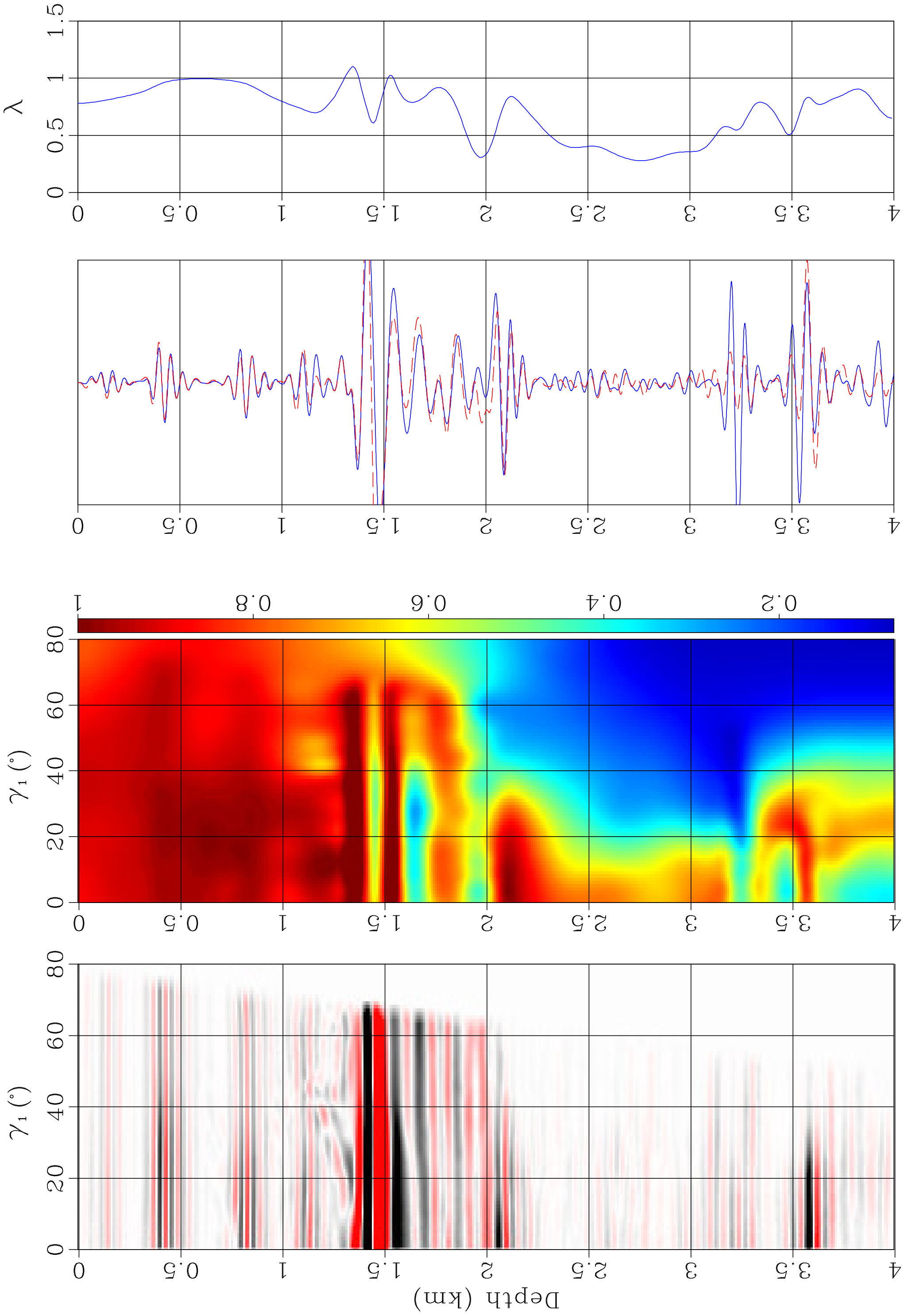


Figure 6.22: Local escape function sampling in space is 100 m, angle - 10°. $d_{max} = 0.5$ km. chapter-imag3/ssaltg ssaltsoamig211

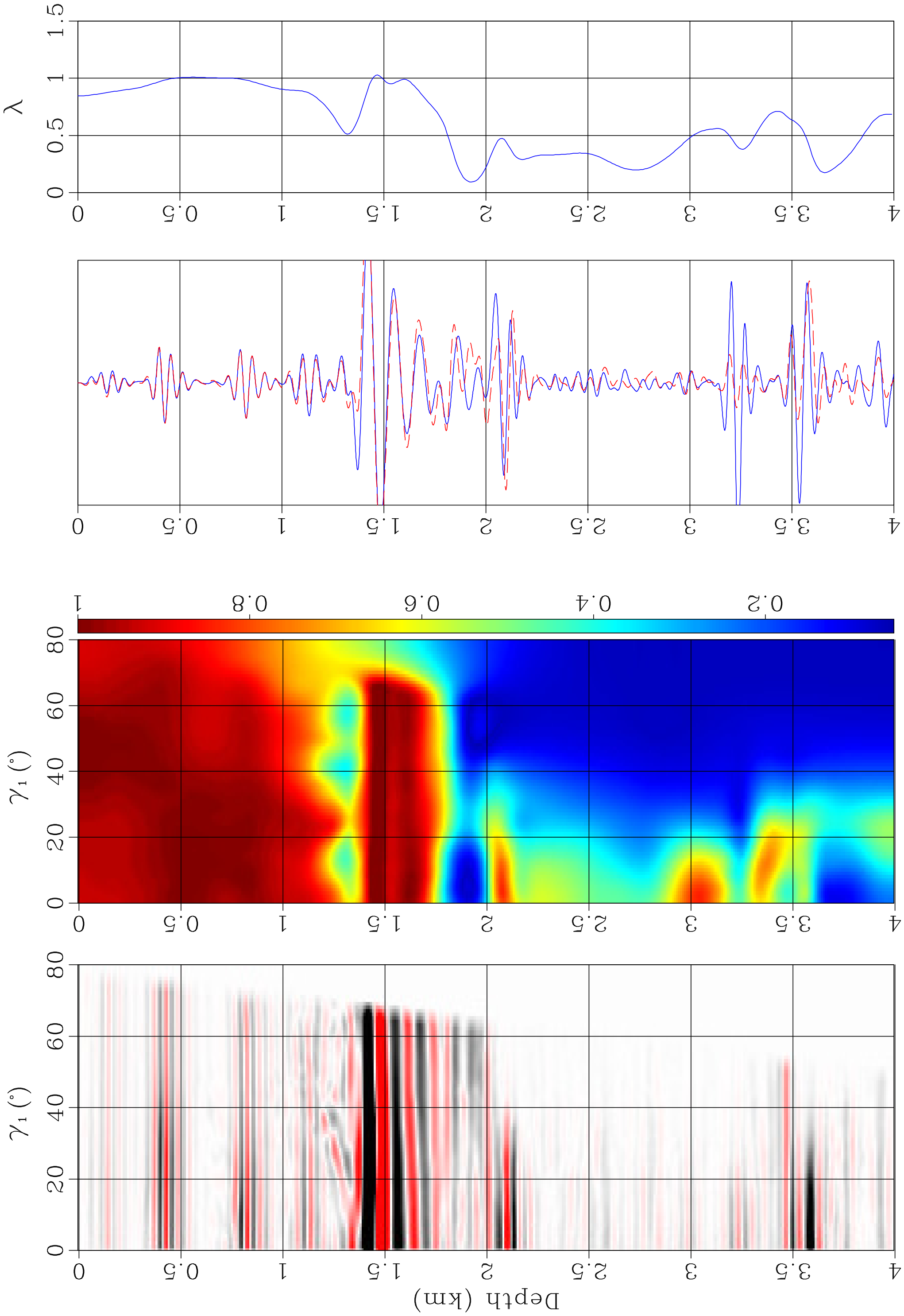


Figure 6.23: Local escape function sampling in space is 100 m, angle - 10°. $d_{max} = 0.75$ km. chapter-imag3/ssaltg ssaltsoamig212

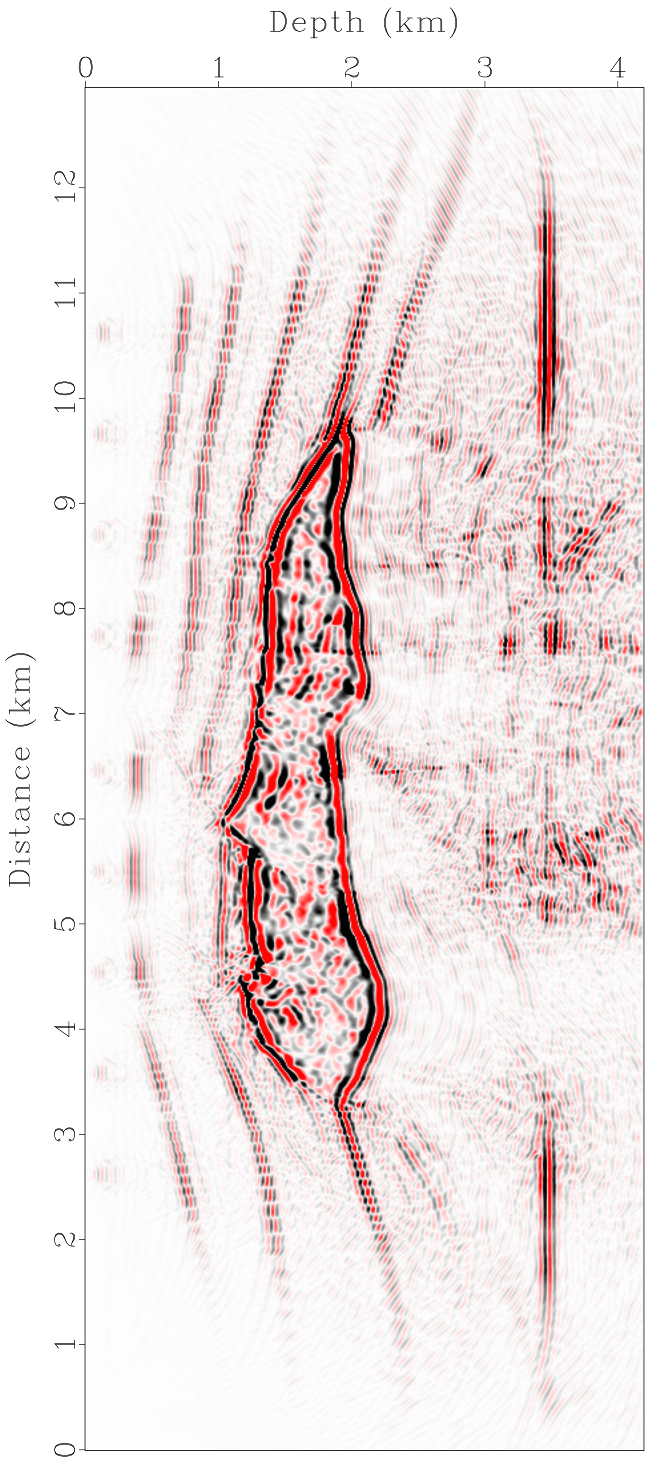


Figure 6.24: Target line at x=6.7 km imaged by multi-arrival Kirchhoff angle-domain migration using escape tables built by ray tracing. chapter-imag3/ssaltx ssaltnmig

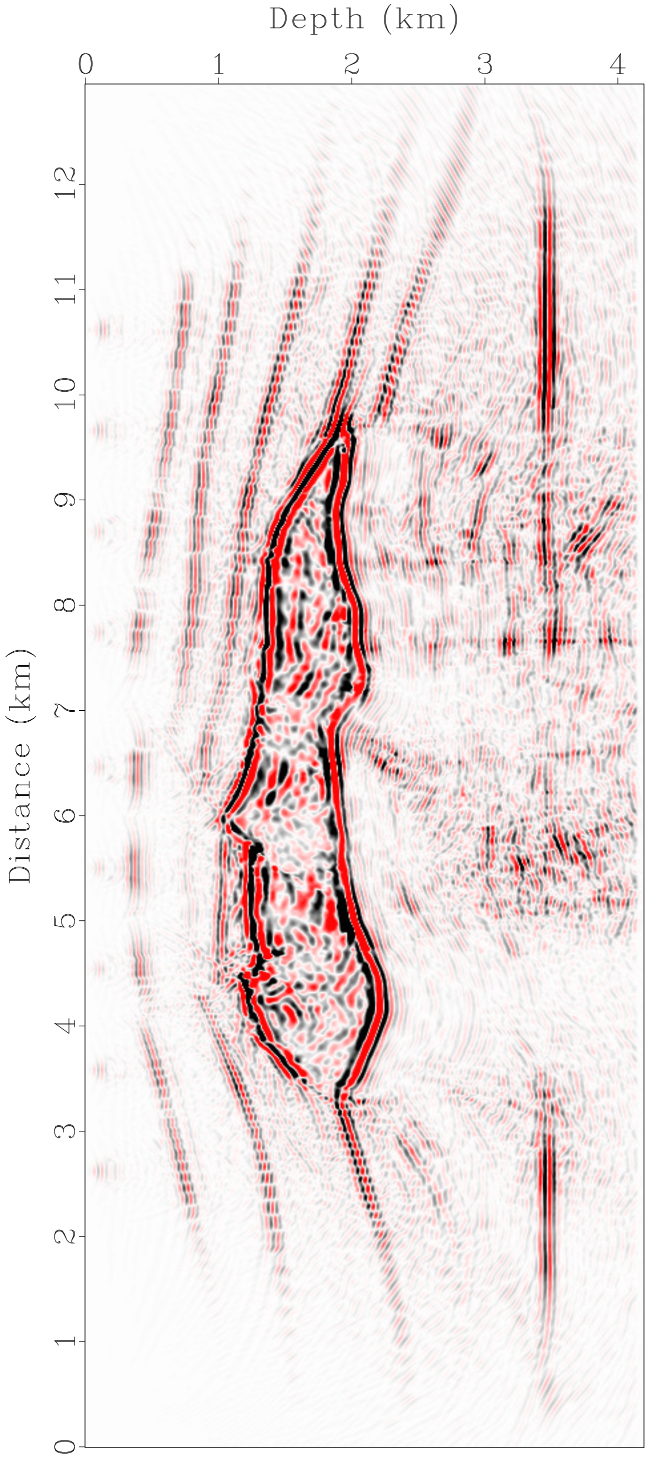


Figure 6.25: Target line at x=6.7 km imaged by multi-arrival Kirchhoff angle-domain migration using escape tables built by iterative reconstruction in phase space. chapter-imag3/ssaltx ssaltsmig

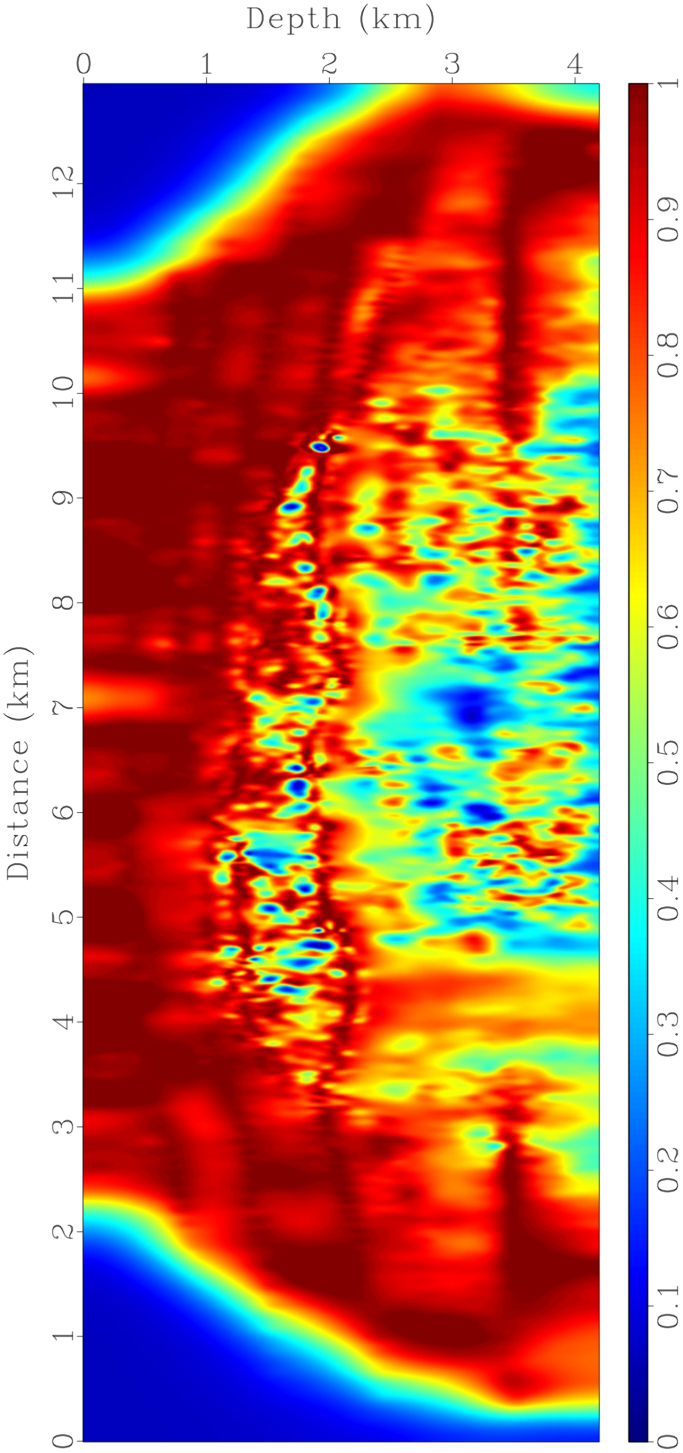


Figure 6.26: Local similarity for the two target lines at x=6.7 km shown above. chapter-imag3/ssaltx ssaltmigsim

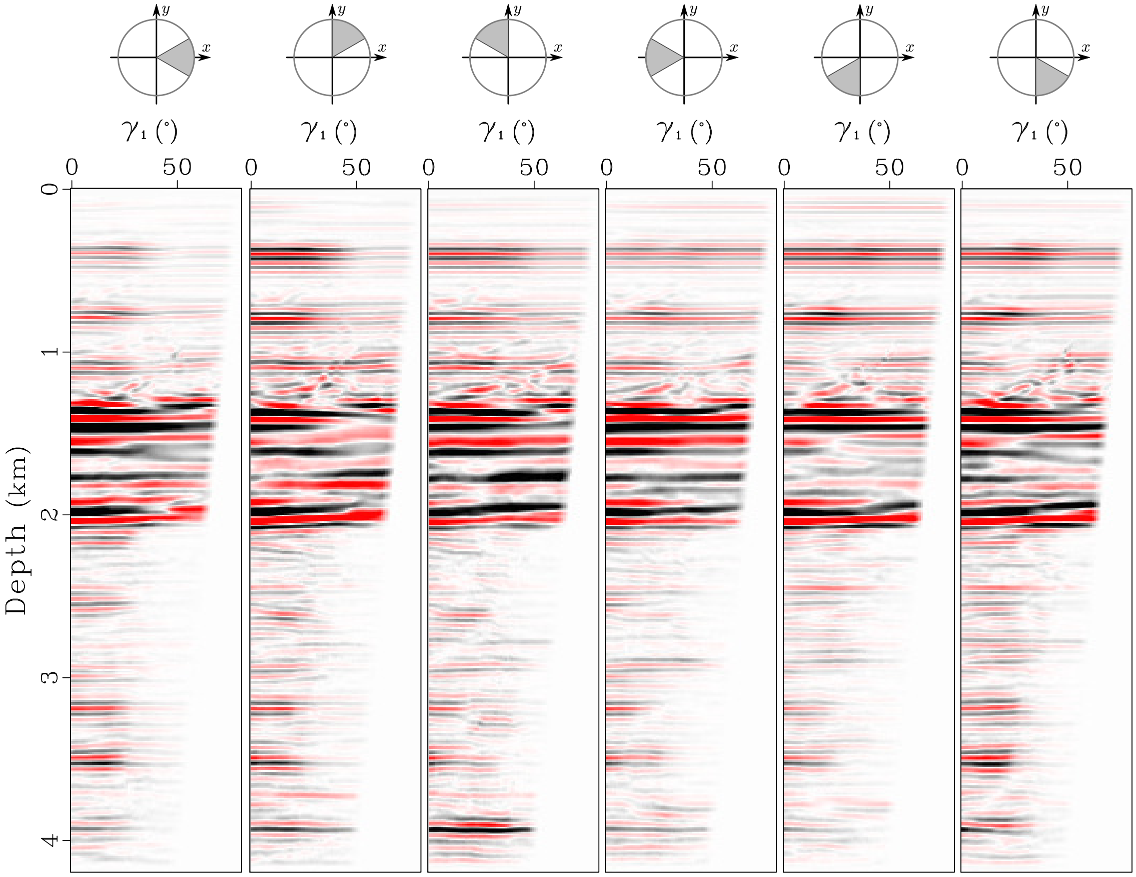


Figure 6.27: Scattering angle gather at $x = 6.7km$ and $y = 7.7km$ (horizontal axis is the scattering angle, each panel represents a scattering azimuth angle sector specified by a diagram above it). Escape tables are produced by iterative reconstruction in phase space. chapter-imag3/ssaltx ssaltsoagath

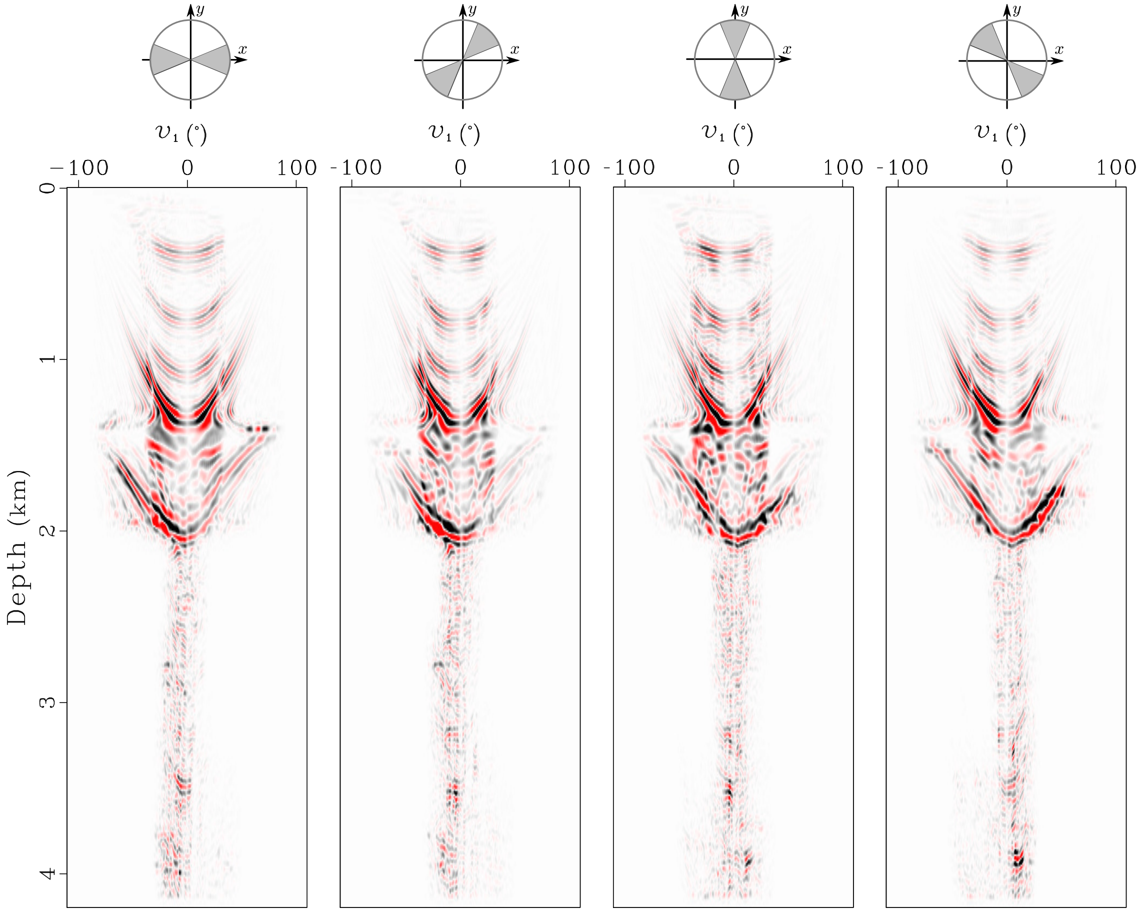


Figure 6.28: Dip angle gather at $x = 6.7km$ and $y = 7.7km$ (horizontal axis is the dip angle, each panel represents a dip azimuth angle sector specified by a diagram above it). Escape tables are produced by iterative reconstruction in phase space. chapter-imag3/ssaltx ssaltsdagath

Chapter 7

Conclusions

Summary of results

I have developed in this dissertation two numerical methods for computing escape function tables in the phase space. The hybrid Eulerian-Lagrangian approach uses finite-differences on a reduced phase space grid. Areas of high-gradient values in escape functions are treated separately with Lagrangian ray tracing to ensure accuracy. While this method achieves high accuracy in escape solutions, its computational efficiency is limited. Utilization of other accuracy preserving techniques, such as AMR, in this framework does not appear feasible in 3-D due to complexity and high cost of implementation. However, the proposed hybrid scheme can still be used as an accurate, physics-guided interpolator for building fine-sampled escape function tables from results obtained on a coarse grid by other methods.

I have also introduced an alternative, semi-Lagrangian method for obtaining escape tables. This approach is based on the fundamental property of continuity of escape functions in the phase space. I showed that, if escape functions are constrained in space, then the global solution can be found by reconstructing their values through iterative parsing of the localized solutions. In practice, this translates into a 5-D interpolation problem in reduced phase space. The extent of these local functions should be such that they expose sufficient smoothness for accurate interpolation. This approach is relatively simple to implement in 3-D. It also scales to large 3-D problems by means of distributed parallel computations. All effects of anisotropy are readily included in local escape functions, which means that compute time at the interpolation stage does not depend on the type of the background velocity model. Only initialization of the local functions has this dependence, but for large image targets, its compute time is much smaller than that of the iterative reconstruction. Even with the most conservative parameterization required for imaging complex models, the semi-Lagrangian approach demonstrated about an order of magnitude improvement in computational efficiency compared to conventional ray tracing.

Both techniques operate directly in the phase space and produce escape tables defined in terms of phase space coordinates: positions in space and directional angles. These angles are directly related to the geologically meaningful subsurface reflection system and describe a unique one-to-one mapping of surface seismic reflection data to subsurface angle domain. I showed in this work how to utilize escape tables in angle-domain migration in practice. Although the imaging process was not the main objective of this research, it was important to show that the minimal set of computed escape quantities can be used for quality imaging. I demonstrated how exit slope, geometrical spreading, and approximate KMAH index can be estimated from escape positions and escape traveltime. These parameters are needed in the migration process to estimate correct amplitude weights, phase shifts, and antialiasing filters. Ultimately, only imaging quality should be used as a measure of quality of traveltime computations. I demonstrated that the two proposed methods allow to obtain accurate imaging outputs for several 2-D and 3-D complex benchmark models.

Possible extensions and future work

While my current implementation displays principle features of the iterative reconstruction algorithm with good computational efficiency, there are several possible improvements which may make the computational cost even lower. I can suggest the following extensions:

1. The d_{max} parameter should be variable depending on the local complexity of the velocity model. That way, large steps through the phase space can be performed in relatively simple parts of the model, and small steps can be done in other parts in order to preserve accuracy of computations.
2. Similarly, spatial sampling $\Delta_{x,y,z}$ of local escape functions can be made variable depending on the local model features. This should be easier to implement than AMR, because the structure of the grid depends on the model and is known in advance.
3. A different interpolation scheme can be implemented that operates in the angular plane $\{\theta, \phi\}$. The plane is a projection of the surface of the sphere of phase directions. Approximation methods specifically designed for interpolation on the sphere might produce more accurate results in the iterative reconstruction (Wahba, 1981).

On the migration side, the following extensions might improve imaging quality

1. Wavefront construction (WFC), a standard approach to ray tracing type computations (Vinje et al., 1993), can help in avoiding oversampling or undersampling of escape tables. Strictly speaking, exact positions of wavefronts are not known

in the iterative reconstruction process. Every arrival instead is traced independently with the biggest allowed steps. Nevertheless, WFC-like results can still be achieved, if, for every three neighboring arrivals, their corresponding escape triangle on the surface is analyzed. If the area of the triangle is too large, then a new arrival should be inserted. The new arrival must originate in-between the three original ones. Now, the four arrivals form two escape triangles on the surface, so the analysis procedure should be repeated for each of them independently. The process can proceed recursively until the surface wavefront is sampled adequately.

2. Filtering data by escape slope (ray parameter) can improve the quality of output image gathers. Xu et al. (2001) originally suggested that angle-domain imaging should produce artifact-free gathers. This has been a point of debate in the research community since then. Stolk and Symes (2004) showed that angle-domain migration may still produce artifacts in a complex model setting. This happens due to the fact that surface seismic data in such environments contain conflicting dips. Phase-space escape functions at the core of the angle-domain Kirchhoff integral define one-to-one mapping from image location/phase angle to surface location/time/ray parameter. We usually ignore the ray parameter for practical reasons. It might be too expensive to decompose 5-D surface data into two slope components and try to manipulate 7-D datasets during the imaging process. However, if a fast and efficient compression method is developed for storing such datasets, then angle-domain migration can benefit significantly from using true unique one-to-one mapping from surface data to the subsurface image space (Koren and Ravve, 2011).

Connections to other methods

My iterative reconstruction of global escape values in the phase space is related to the phase flow method of Ying and Candès (2006). It employs similar principles in order to construct wavefront positions in the phase space. In my approach, I do not aim at tracking wavefronts with small time steps. Instead, I reconstruct individual characteristic lines with the biggest possible step that still preserves sufficient accuracy for imaging problems. The algorithm also bears some resemblance to the slowness matching method of Symes and Qian (2003). The slowness matching approach utilizes the property of continuity of the phase vector as well. However, the background solution is defined in terms of localized minimum-time solutions in space only, as opposed to local escape functions in phase space.

Bibliography

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

Addison, F., K. Kennelley, and F. Botros, 2010, SS: Thunder Horse and Atlantis deepwater frontier developments in the Gulf of Mexico: Future challenges for deepwater developments: Offshore Technology Conference.

Alde, D., M. Fehler, S. Hildebrand, L. Huang, and H. Sun, 2002, Determining the optimally smoothed slowness model for ray-tracing based migration using multiple-valued traveltime tables: SEG Technical Program Expanded Abstracts, 1168–1171.

———, 2003, Comparison of images of the Marmousi model for ray-tracing based Kirchhoff migration using single- and multiple-valued traveltimes, amplitudes, and KMAH phases: SEG Technical Program Expanded Abstracts, 1777–1780.

Alkhalifah, T., 2000, An acoustic wave equation for anisotropic media: Geophysics, **65**, 1239–1250.

Alkhalifah, T., and I. Tsvankin, 1995, Velocity analysis for transversely isotropic media: Geophysics, **60**, 1550–1566.

Aminzadeh, F., B. Jean, and T. Kunz, 1997, 3-D Salt and Overthrust models: Society of Exploration Geophysicists.

Audebert, F., D. Nichols, T. Rekdal, B. Biondi, D. Lumley, and H. Urdaneta, 1997, Imaging complex geologic structure with single-arrival Kirchhoff prestack depth migration: Geophysics, **62**, 1533–1543.

Bednar, J. B., 2002, ”You need to be well rounded in the fundamentals” - An interview with Bee Bednar: CSEG Recorder, **27**, 16–30.

———, 2005, A brief history of seismic migration: Geophysics.

———, 2006, Modeling, migration and velocity analysis in simple and complex structure: Panorama Technologies.

Benamou, J.-D., 1999, Direct computation of multivalued phase space solutions for Hamilton-Jacobi equations: Communications on Pure and Applied Mathematics, **52**, 1443–1475.

Berger, M. J., and J. Oliger, 1984, Adaptive Mesh Refinement for hyperbolic partial differential equations: Journal of Computational Physics, **53**, 484–512.

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

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

Bleistein, N., Y. Zhang, S. Xu, G. Zhang, and S. H. Gray, 2005, Migration/inversion: think image point coordinates, process in acquisition surface coordinates: Inverse Problems, **21**, 1715–1744.

Brandsberg-Dahl, S., M. V. de Hoop, and B. Ursin, 2003, Focusing in dip and AVA compensation on scattering-angle/azimuth common image gathers: Geophysics, **68**, 232–254.

Buhmann, M. D., 2003, Radial Basis Functions: Theory and Implementations: Cambridge University Press.

Cayley, A., 1845, On certain results relating to Quaternions: Philosophical Magazine, **26**, 141–145.

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

Chapman, C. H., and R. Drummond, 1982, Body-wave seismograms in inhomogeneous media using Maslov asymptotic theory: Bulletin of the Seismological Society of America, **72**, S277–S317.

Cogan, M., J. Gardner, and N. Moldoveanu, 2011, Subsalt imaging challenges - A deepwater imaging analysis: 73rd EAGE Conference & Exhibition, EAGE, Expanded Abstract.

Courant, R., E. Isaacson, and M. Rees, 1952, On the solution of nonlinear hyperbolic differential equations by finite differences: Communications on Pure and Applied Mathematics, **5**, 243–255.

Danowitz, A., K. Kelley, J. Mao, J. P. Stevenson, and M. Horowitz, 2012, CPU DB: Recording microprocessor history: Queue, **10**, 10:10–10:27.

de Boor, C., 1978, A practical guide to splines: Springer.

Duchon, J., 1977, Splines minimizing rotation-invariant semi-norms in Sobolev spaces, *in* Constructive Theory of Functions of Several Variables: Springer, volume **571** *of* Lecture Notes in Mathematics, 85–100.

Durran, D. R., 2010, Numerical methods for fluid dynamics: With applications to geophysics: Springer. Texts in Applied Mathematics.

Engquist, B., and O. Runborg, 2003, Computational high frequency wave propagation: Acta Numerica, **12**, 181–266.

Etgen, J., S. H. Gray, and Y. Zhang, 2009, An overview of depth imaging in exploration geophysics: Geophysics.

Ettrich, N., D. Merten, and S. K. Foss, 2008, True-amplitude angle migration in complex media: 70th EAGE Conference & Exhibition, EAGE, Expanded Abstract.

Farra, V., 1993, Ray tracing in complex media: Journal of Applied Geophysics, **30**, 55–73.

Fehler, M. C., S. T. Hildebrand, L. J. Huang, and D. X. Alde, 2002, Wavefront construction Kirchhoff migration with ray-amplitude corrections: 64th EAGE Conference & Exhibition, EAGE, Expanded Abstract.

Ferziger, J. H., and M. Perić, 2002, Computational methods for fluid dynamics: Springer.

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

———, 2003, Angle-domain seismic imaging and the oriented wave equation: SEG Technical Program Expanded Abstracts, 893–896.

———, 2004, On anelliptic approximations for qP velocities in VTI media: Geophysical Prospecting, **52**, 247–259.

———, 2007, Local seismic attributes: Geophysics, **72**, A29–A33.

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.

Gentry, R. A., R. E. Martin, and B. J. Daly, 1966, An Eulerian differencing method for unsteady compressible flow problems: Journal of Computational Physics, **1**, 87–118.

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

Goldenberg, D., M. Kagan, R. Ravid, and M. S. Tsirkin, 2005, Zero copy Sockets Direct Protocol over Infiniband – preliminary implementation and performance analysis: 13th Symposium on High Performance Interconnects, 128–137.

Golub, G., 1996, Matrix computations: Johns Hopkins University Press.

Gremaud, P. A., and C. M. Kuster, 2006, Computational study of fast methods for the Eikonal equation: SIAM Journal on Scientific Computing, **27**, 1803–1816.

Hirsch, C., 2007, Numerical computation of internal and external flows, Volume 1: Fundamentals of computational fluid dynamics: Elsevier/Butterworth-Heinemann.

Hudec, M. R., and M. P. A. Jackson, 2007, Terra infirma: Understanding salt tectonics: Earth-Science Reviews, **82**, 1–28.

ITRS, 2011, 2011 International Technology Roadmap for Semiconductors: Executive Summary: International Semiconductor Roadmap Committee.

Karcher, J. C., 1974, The reflection seismograph: its invention and use in the discovery of oil and gas fields.

Klokov, A., and S. Fomel, 2012, Separation and imaging of seismic diffractions using migrated dip-angle gathers: Geophysics, **77**, S131–S143.

Koren, Z., and I. Ravve, 2011, Full-azimuth subsurface angle domain wavefield decomposition and imaging Part I: Directional and reflection image gathers: Geophysics, **76**, S1–S13.

Koren, Z., I. Ravve, E. Ragoza, A. Bartana, and D. Kosloff, 2008, Full-azimuth angle domain imaging: SEG Technical Program Expanded Abstracts, **27**, 2221–2225.

Koren, Z., S. Xu, and D. Kosloff, 2002, Target-oriented common reflection angle migration: SEG Technical Program Expanded Abstracts, **21**, 1196–1199.

Kuzmin, D., 2010, A guide to numerical methods for transport equations: University Erlangen-Nuremberg.

Lambaré, G., P. S. Lucio, and A. Hanyga, 1996, Two-dimensional multivalued traveltime and amplitude maps by uniform sampling of a ray field: Geophysical Journal International, **125**, 584–598.

Landa, E., S. Fomel, and M. Reshef, 2008, Separation, imaging, and velocity analysis of seismic diffractions using migrated dip-angle gathers: SEG Technical Program Expanded Abstracts, 2176–2180.

Lapilli, C., and P. Fowler, 2013, Rotation parameters for model building and stable parameter inversion in orthorhombic media: SEG Technical Program Expanded Abstracts, 4656–4660.

Leveille, J. P., I. F. Jones, Z.-Z. Zhou, B. Wang, and F. Liu, 2011, Subsalt imaging for exploration, production, and development: A review: Geophysics, **76**, WB3–WB20.

Lumley, D. E., J. F. Claerbout, and D. Bevc, 1994, Anti-aliased Kirchhoff 3-D migration: SEG Technical Program Expanded Abstracts, **13**, 1282–1285.

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

Mohseni, K., and T. Colonius, 2000, Numerical treatment of polar coordinate singularities: Journal of Computational Physiscs, **157**, 787–795.

Operto, M. S., S. Xu, and G. Lambaré, 2000, Can we quantitatively image complex structures with rays?: Geophysics, **65**, 1223–1238.

Osher, S., L.-T. Cheng, M. Kang, H. Shim, and Y.-H. Tsai, 2002, Geometric optics in a phase-space-based level set and Eulerian framework: Journal of Computational Physics, **179**, 622–648.

Osher, S., and J. A. Sethian, 1988, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations: Journal of Computational Physics, **79**, 12–49.

Paffenholz, J., B. McLain, J. Zaske, and P. Keliher, 2002, Subsalt multiple attenuation and imaging: Observations from the Sigsbee2B synthetic dataset: SEG Technical Program Expanded Abstracts 2002, 2122–2125.

Pletcher, R. H., J. C. Tannehill, and D. Anderson, 2012, Computational fluid mechanics and heat transfer: Taylor & Francis.

Plewa, T., T. J. Linde, and V. G. Weirs, 2005, Adaptive mesh refinement – theory and applications: Springer.

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

Runborg, O., 2007, Computational high frequency wave propagation: Communications in Computational Physics, **2**, 827–880.

Sava, P., and S. Fomel, 2001, 3-D traveltime computation using Huygens wavefront tracing: Geophysics, **66**, 883–889.

———, 2005, Coordinate-independent angle-gathers for wave equation migration: SEG Technical Program Expanded Abstracts, 2052–2055.

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

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

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

Shah, H., 2008, The 2007 BP anisotropic velocity-analysis benchmark: 70th EAGE Conference & Exhibition, EAGE, Workshop, `http://software.seg.org/`.

Shoemake, K., 1985, Animating rotation with quaternion curves: SIGGRAPH '85 Proceedings of the 12th annual conference on Computer graphics and interactive techniques, **19**.

Shu, C.-W., 1998, Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory schemes for hyperbolic conservation laws, *in* Advanced Numerical Approximation of Nonlinear Hyperbolic Equations: Springer Berlin Heidelberg, volume **1697** *of* Lecture Notes in Mathematics, 325–432.

Stolk, C. C., and W. W. Symes, 2004, Kinematic artifacts in prestack depth migration: Geophysics, **69**, 562–575.

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.

Tang, B., S. Xu, and Y. Zhang, 2011, Aliasing in RTM 3D angle gathers: SEG Technical Program Expanded Abstracts, 3310–3314.

Thomsen, L., 1986, Weak elastic anisotropy: Geophysics, **51**, 1954–1966.

Thomson, C. J., and C. H. Chapman, 1985, An introduction to Maslov's asymptotic method: Geophysical Journal International.

Top500, 2013, Top 500 supercomputer sites, November 2013: `http://top500.org/lists/2013/11/`.

Van Straalen, B., P. Colella, D. T. Graves, and N. Keen, 2011, Petascale block-structured AMR applications without distributed meta-data: Proceedings of the 17th international conference on Parallel processing - Volume Part II, 377–386.

van Trier, J., and W. W. Symes, 1991, Upwind finite-difference calculation of traveltimes: Geophysics, **56**, 812–821.

Versteeg, R. J., 1993, Sensitivity of prestack depth migration to the velocity model: Geophysics, **58**, 873–882.

Vidale, J. E., 1990, Finite-difference calculation of traveltimes in three dimensions: Geophysics, **55**, 521–526.

Vinje, V., E. Iversen, and H. Gjøystdal, 1993, Traveltime and amplitude estimation using wavefront construction: Geophysics, **58**, 1157–1166.

Virieux, J., H. Calandra, and R.-E. Plessix, 2011, A review of the spectral, pseudospectral, finite-difference and finite-element modelling techniques for geophysical imaging: Geophysical Prospecting, **59**, 794–813.

Virieux, J., V. E. V. Cruz-Atienza, R. Brossier, E. Chaljub, O. Coutant, S. Garambois, D. Mercerat, V. Prieux, S. Operto, A. Ribodetti, and J. Tago, 2012, Modelling seismic wave propagation for geophysical imaging, *in* Seismic Waves - Research and Analysis: InTech.

Virieux, J., and V. Farra, 1991, Ray tracing in 3-D complex isotropic media: An analysis of the problem: Geophysics, **56**, 2057–2069.

Wahba, G., 1981, Spline interpolation and smoothing on the sphere: SIAM Journal on Scientific and Statistical Computing, **2**, 5–16.

Xu, S., H. Chauris, G. Lambaré, and M. Noble, 2001, Common-angle migration: A strategy for imaging complex media: Geophysics, **66**, 1877–1894.

Xu, S., and G. Lambaré, 2004, Fast migration/inversion with multivalued ray fields: Part 1 – Method, validation test, and application in 2D to Marmousi: Geophysics, **69**, 1311–1319.

Yilmaz, Ö., 2001, Seismic data analysis: Society of Exploration Geophysicists.

Ying, L., and E. J. Candès, 2006, The phase flow method: Journal of Computational Physics, **220**, 184–215.

Zhao, H.-K., 2005, A fast sweeping method for Eikonal equations: Mathematics of Computation, **74**, 603–627.

Vita

Vladimir Bashkardin was born in Yoshkar-Ola, Soviet Union in 1980. He graduated from Public School No.9 in Yoshkar-Ola, Russia in 1997, and went on to complete the degree of Mining Engineer with specialization in Geophysical Exploration from Gubkin Russian State University of Oil and Gas in 2002. After a six-year career as a software engineer in the exploration industry, he came to The University of Texas at Austin in 2007 to pursue the degree of Doctor of Philosophy in Geophysics.

Permanent address: 225 Westlake Park Blvd, Houston, TX 77079

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.