

SPECTRAL FACTORIZATION OF WAVEFIELDS AND WAVE OPERATORS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF GEOPHYSICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

James Edward Rickett
June 2001

© Copyright 2001 by James Edward Rickett

All Rights Reserved

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jon F. Claerbout
(Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Biondo Biondi

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jerry M. Harris

Approved for the University Committee on Graduate Studies:

Abstract

Spectral factorization is the problem of finding a minimum-phase function with a given power spectrum. Minimum-phase functions have the property that they are causal with a causal (stable) inverse. In this thesis, I factor multidimensional systems into their minimum-phase components. Helical boundary conditions resolve any ambiguities over causality, allowing me to factor multi-dimensional systems with conventional one-dimensional spectral factorization algorithms.

In the first part, I factor passive seismic wavefields recorded in two-dimensional spatial arrays. The result provides an estimate of the acoustic impulse response of the medium that has higher bandwidth than autocorrelation-derived estimates. Also, the function's minimum-phase nature mimics the physics of the system better than the zero-phase autocorrelation model. I demonstrate this on helioseismic data recorded by the satellite-based Michelson Doppler Imager (MDI) instrument, and shallow seismic data recorded at Long Beach, California.

In the second part of this thesis, I take advantage of the stable-inverse property of minimum-phase functions to solve wave-equation partial differential equations. By factoring multi-dimensional finite-difference stencils into minimum-phase components, I can invert them efficiently, facilitating rapid implicit extrapolation without the azimuthal anisotropy that is observed with splitting approximations.

The final part of this thesis describes how to calculate diagonal weighting functions that approximate the combined operation of seismic modeling and migration. These weighting functions capture the effects of irregular subsurface illumination, which can be the result of either the surface-recording geometry, or focusing and defocusing of the seismic wavefield as

it propagates through the earth. Since they are diagonal, they can be easily both factored and inverted to compensate for uneven subsurface illumination in migrated images. Experimental results show that applying these weighting functions after migration leads to significantly improved estimates of seismic reflectivity.

Preface

All of the figures in this thesis are marked with one of the three labels: [ER], [CR], or [NR], with a possible additional label, [M]. The first three labels define to what degree the figure is reproducible from the source code, and parameter files provided with the web version of this thesis¹.

ER denotes *Easily Reproducible*. You can reproduce such a figure from the source code, and parameter files provided with the web version of this thesis. I assume you have a UNIX workstation with a Fortran90 compiler and the current SEP environment (SEPlib 5.1) installed. As well as the tar-ball of the chapter directory itself, you may also need to download the `src`, `data`, and `Adm` directories. Rebuilding ER figures may also require non-proprietary data from the SEP data library. Before this document was placed on the web, someone other than myself destroyed and rebuilt all the ER figures in this thesis.

CR denotes *Conditionally Reproducible*. The commands are in place to reproduce these figures. However, destroying and rebuilding them may take a long period of time (greater than half an hour) and/or a large amount of computer memory (greater than 256Mb). Although noone has attempted to verify this certification completely, before this document was placed on the web a script was run to check that the make rules are in place to rebuild all the CR figures.

NR denotes *Non-Reproducible*. Figures in this class include schematic diagrams, and figures borrowed from previous authors. No rules exist to rebuild this kind of figure.

¹<http://sepwww.stanford.edu/public/docs/sep109>

Figures labeled with an [M] may be viewed as a “movie” in the web version of this thesis. The detailed comparison between different panels of the same figure is often easier if they can be viewed in this way: readers can then flip back-and-forth between panels without having to readjust their point of focus. Since these movies are designed to be viewed on a computer monitor rather than the printed page, they may have a different color scheme than the original figure. The presence of the movie file does not affect how I classify a figure’s reproducibility.

Acknowledgements

This thesis would not have been possible without the help of many people.

Firstly, I would like to thank my advisors Jon Claerbout and Biondo Biondi for their generosity with ideas, and for providing a research environment in the Stanford Exploration Project that is both challenging and exceptionally rewarding.

Many thanks are also due to all the fellow students who have shared time with me at Stanford. A few deserve special mention, however. Bob Clapp has continually helped me in many things geophysical, and all things computer-related. Sergey Fomel provided insight and advice that I am really beginning to appreciate since he has left. Antoine Guittot has put up with me as an office mate for the last three years, and never even gloated (too much) about the success of Les Bleus. Paul Sava was also a great office mate, and has continued to be someone I harass daily with questions about angle gathers and make rules.

Numerous results in this thesis would not have been possible without datasets donated by others. I am grateful to the SOI team for providing the MDI dataset in Chapter 2, Bryan Kerr for generously allowing me to include results from his Long Beach experiment in Chapter 2, and BP Amoco for releasing to Stanford the dataset in Chapter 7.

Finally, I would like to thank my family: my parents and sisters for their continued love and support, even though I have moved to the other side of the world; and Elisabeth her patience, love, and generosity in sharing her life with me.

Contents

Abstract	iv
Preface	vi
Acknowledgements	viii
1 Introduction	1
1.1 Spectral factorization	2
1.1.1 Multi-dimensional spectral factorization on a helix	2
1.2 Applications of multi-dimensional spectral factorization	3
1.2.1 Spectral factorization of seismic wavefields	4
1.2.2 Spectral factorization of partial differential equations	4
1.2.3 Spectral factorization of linear modeling operators	4
2 Spectral factorization of seismic oscillations	6
2.1 Introduction to helioseismology	7
2.1.1 Time-distance helioseismology	8
2.2 Model of stochastic oscillations	9

2.2.1	Kolmogorov review	10
2.2.2	Multi-dimensional factorization	11
2.2.3	Theoretical comparison between time-distance functions	12
2.2.4	On the assumption of translational invariance	13
2.3	Application to SOHO/MDI dataset	13
2.3.1	Time-distance functions compared	14
2.4	Application to a terrestrial passive seismic dataset	18
2.4.1	Long Beach experimental setup and preprocessing	19
2.4.2	Time-distance functions compared	19
2.5	Conclusions	21
3	Helical factorization of the Helmholtz equation	23
3.1	Introduction	23
3.2	Direct solution of Poisson's equation	24
3.3	The Helmholtz equation	26
3.3.1	Wave extrapolation	29
3.3.2	Propagating waves with the <code>Wavemovie</code> program	31
3.3.3	Reducing the filter length	33
3.4	Conclusions	34
4	Helical factorization of paraxial extrapolators	35
4.1	Introduction	35
4.2	Implicit extrapolation theory	37

4.2.1	Matrix representation of implicit operators	38
4.2.2	Helical boundary conditions	39
4.2.3	Cross-spectral factorization	40
4.2.4	Polynomial division	41
4.3	Synthetic examples	42
4.4	Conclusions	44
5	$V(x, y, z)$ and non-stationary inverse convolution	45
5.1	Introduction	45
5.2	Smooth lateral variations in velocity	46
5.2.1	Non-stationary recursive filtering	46
5.2.2	Non-stationary factorization	47
5.2.3	Numerical example	48
5.3	Strong lateral variations in velocity	51
5.3.1	Stability analysis	51
5.3.2	Numerical examples	53
5.4	Conclusions	53
6	Migration and shot illumination	56
6.1	Introduction to shot-profile migration	56
6.2	Shot-profile migration and modeling	58
6.2.1	Exploding-reflector migration by downward continuation	58
6.2.2	Exploding-reflector modeling by upward continuation	60

6.2.3	Shot-profile migration and modeling	61
6.2.4	True-amplitude shot-modeling	63
6.3	Towards true-amplitude migration	64
6.3.1	Exploding-reflector modeling as a pseudo-unitary operator	65
6.4	Compensating for irregular shot illumination	67
6.4.1	Shot illumination examples	72
6.4.2	Application to true-amplitude migration	74
6.5	Conclusions	76
7	Model versus data normalization	77
7.1	Introduction	77
7.1.1	Preconditioning and spectral factorization	78
7.1.2	Diagonal weighting functions for recursive prestack depth migration .	79
7.2	Model-space weighting functions	79
7.2.1	Three choices of reference image	80
7.2.2	Stabilizing the denominator	81
7.2.3	Numerical comparison	81
7.2.4	Computational cost	87
7.3	Data-space weighting functions	88
7.3.1	Combining model-space and data-space weighting functions	89
7.3.2	Numerical comparisons	90
7.4	Discussion	92
7.4.1	Limitations	93

7.5	Conclusion	93
A	Non-stationary inverse convolution	94
A.1	Introduction	94
A.2	Theory	95
A.2.1	Stationary convolution and inverse convolution	95
A.2.2	Non-stationary convolution and combination	97
A.2.3	Adjoint non-stationary convolution and combination	98
A.2.4	Inverse non-stationary convolution and combination	99
A.2.5	The stability of non-stationary inverse filtering	100
A.3	Conclusions	102
B	Common-image gathers for shot-profile migration	103
B.1	Introduction	103
B.2	Multi-offset imaging for shot-profile migration	106
B.2.1	Shot to midpoint transformations	106
B.2.2	Computational considerations	108
B.3	Interpretation of common-image gathers	108
B.3.1	Effect of velocity on common-image gathers	109
B.3.2	Effect of shot sampling on common-image gathers	109
B.4	Conclusions	110
C	Normalization by operator fold	113
C.1	Introduction	113

C.2	Normalization of interpolation operators	113
C.3	Normalization of Kirchhoff operators	116
C.4	Operator fold for finite-frequency migration operators	117
C.5	Conclusion	118
	Bibliography	119

List of Tables

3.1	Comparison of the mathematical form of various wavefield extrapolators.	29
7.1	Comparison of the reflector strength for different choices of illumination-based weighting function.	86
A.1	Recursive formulae for non-stationary (NS) inverse operators.	99

List of Figures

1.1	Illustration of helical boundary conditions mapping a two-dimensional function (a) onto a helix (b), and then unwrapping the helix (c) into an equivalent one-dimensional function (d). Figure by Sergey Fomel.	3
2.1	Raw full-disc solar dopplergram from 9 July 1996, 09.00. Colormap shows line-of-site velocity with approximately 0.02 km/s uncertainty. Figure from the Solar Oscillations Investigation (SOI) website, http://soi.stanford.edu .	7
2.2	Cube of MDI data.	15
2.3	Three-dimensional minimum phase impulse response.	16
2.4	Schematic showing some of the raypaths visible in helioseismic time-distance plots.	16
2.5	Time-distance impulse responses computed by (a) three-dimensional autocorrelation, and (b) three-dimensional Kolmogorov spectral factorization. Traces have been binned as a function of radius from the origin, averaging the impulse response over azimuth.	17
2.6	Two-dimensional amplitude spectra of impulse responses shown in Figure 2.5 above. The impulse responses were computed by (a) three-dimensional auto-correlation, and (b) three-dimensional Kolmogorov spectral factorization.	18
2.7	Five seconds of passive seismic data from the Long Beach experiment after notch-filtering.	20

2.8	Impulse response calculated by (a) 3-D autocorrelation, and (b) spectral factorization. Compare with Figure 2.5.	21
3.1	Deconvolution by a filter whose autocorrelation is the two-dimensional Laplacian operator. This amounts to solving the Poisson equation. After (Claerbout, 1997).	26
3.2	Amplitude of impulse response of polynomial division with minimum-phase factorization of the Helmholtz equation. The top panel shows the location of the impulse. The bottom panel shows the impulse response. Helical boundary conditions mean the second bump in the impulse response corresponds to energy propagating to the next depth step.	31
3.3	Comparison of the 45° wave equation (left) with the helical factorization of the Helmholtz equation (right).	32
3.4	Helmholtz equation factorization with different values for the ‘one-sixth’ parameter, β . From left, $\beta = 0, 1/12, 1/8$ and $1/6$	32
3.5	Helmholtz equation factorization with different 3×3 finite-difference representations of the Laplacian. From left, $\gamma = 0, 1/2, 2/3$ and 1	33
4.1	Vertical slice through broad-band impulse response of 45° wave equation, showing the distinctive cardioid.	43
4.2	Depth-slice of centered impulse response corresponding to a dip of 50° . Panel (a) shows the result of employing an x - y splitting approximation, and panel (b) shows the result of the helical factorization. Note the azimuthally isotropic nature of panel (b). Evanescent energy has been removed by dip-filtering prior to migration.	43
4.3	Depth-slice of offset impulse response corresponding to a dip of 45° . Note the helical boundary conditions on the fast spatial axis.	44

5.1	Schematic illustrating non-stationary factorization process. Step 1: Build the extrapolation matrix. Step 2: Extract the filter corresponding to the current spatial location. Step 3: Factor this filter. Step 4: Insert factors into lower and upper triangular matrices. Step 5: Move to next spatial location and repeat from step 2.	48
5.2	Velocity model with strong lateral velocity variations. Velocity varies between 1.5 km/s in the near surface sediments to 4.5 km/s in the salt.	49
5.3	Result of migrating a subvolume from the zero-offset 3-D SEG/EAGE salt model with the non-stationary helical factorization.	50
5.4	Instability in 2-D helical factorization of implicit 2-D system.	54
6.1	A single migrated shot record (a) over the Marmousi (Bourgeois et al., 1991) velocity model (b).	61
6.2	Relationship between migration, modeling, their pseudo-inverses, and their (pseudo-unitary) industrial-strength counterparts.	65
6.3	Frequency ($f - k_x$) response of cascaded modeling and migration in a constant velocity medium. Panel (a) shows amplitude spectrum of dip-limited impulsive input. Panel (b) shows spectrum after exploding-reflector modeling and migration.	66
6.4	Synthetic Marmousi datasets generated with one-way wave modeling: panel (a) shows the exploding-reflector dataset, and panel (b) shows the zero-offset dataset. The ellipses highlight the increased vertical streaking in the true zero-offset dataset.	68
6.5	Stacked section from the Mobil AVO dataset (Clapp, 1999). Vertical streaking is visible throughout the section.	69

6.6 Marmousi reflectivity: panel (a) shows the initial model, panel (b) shows the reflectivity estimate by migrating the exploding-reflector data shown in Figure 6.4 (a), and panel (c) shows the reflectivity estimate obtained by migrating the zero-offset data from Figure 6.4 (b).	70
6.7 Normalization by shot illumination: the panel (a) shows shot profile migration with the correct velocity model of 20 split-spread shots over the Marmousi model, panel (b) shows shot illumination, and panel (c) shows the normalized image.	73
6.8 Normalization by shot illumination: the panel (a) shows shot profile migration with the <i>incorrect</i> velocity model of 20 split-spread shots over the Marmousi model, panel (b) shows shot illumination, and panel (c) shows the normalized image.	75
7.1 Velocity (in km/s) model used to generate the synthetic Amoco 2.5-D dataset.	82
7.2 Synthetic shot-gathers from the Amoco 2.5-D dataset ($s_x = 18.2$ km). Panel (a) shows the gather generated by full two-way finite-difference modeling. Panel (b) shows the gather generated by linear one-way modeling. Panel (c) shows the same two-way equation gather as panel (a) but with a top-mute to enable easier comparison with panel (b).	83
7.3 Comparison of calibration images: (a) original migration, (b) original migration after modeling and migration, (c) random image after modeling and migration, and (d) flat event image after modeling and migration.	84
7.4 Comparison of weighting functions: (a) shot illumination as described in Chapter 6, (b) reference model was migrated image, (c) reference model was random image, and (d) reference model consisted of purely flat events.	85
7.5 Normalized peak amplitude of 3.9 km reflector after migration (solid line), and then normalization by flat-event illumination (dashed-line). The ideal result would be a constant amplitude of 1.	86

7.6	Results of full L_2 inversion of the Amoco 2.5-D dataset with FFD modeling/migration. Panel (a) shows results after four iterations, and panel (b) shows results after ten iterations.	87
7.7	Normalized standard deviation of flat reflector versus iteration number. After four iterations the noise-level causes degradation of amplitude reliability. . . .	88
7.8	Data-space weighting function for the gather shown in Figure 7.2 (a). Darker shades of grey correspond to higher amplitude.	91
7.9	Common offset-section corresponding to $h = 500$ m (a), and its data-space weighting function (b). Darker shades of grey correspond to higher amplitude.	91
7.10	Normalized peak amplitude of 3.9 km reflector after migration of the raw data (solid line), and after migration of a dataset that has been normalized by an appropriate data-space weighting function (dashed-line). Compare with Figure 7.5.	92
A.1	Impulsive input (a) and response (b) to non-stationary filtering with filter-bank defined in equation (A.16).	101
B.1	Illustration of offset before (a), during (b), and after (c,d) migration.	105
B.2	Imaging offsets with $\Delta h = \Delta x$ based on equation (B.1) alone.	107
B.3	Imaging offsets with $\Delta h = \Delta x/2$. The midpoint lies between propagation grid nodes.	107
B.4	Migrated images produced with data having a fully-sampled shot axis (a), and a very sparsely-sampled shot axis (b).	110
B.5	Offset-domain common-image gathers corresponding to $x = 4000$ m. Panels (a), (b), and (c) were migrated with velocity models that were correct, 6% too low, and 6% too high, respectively.	111

B.6 Angle-domain common-image gathers corresponding to $x = 4000$ m. Panels (a), (b) and (c) were migrated with velocity models that were correct, too low, and too high, respectively.	111
B.7 Poorly illuminated offset-domain common-image gathers corresponding to $x = 4000$ m. Panels (a), (b), and (c) were migrated with velocity models that were correct, 6% too low, and 6% too high, respectively. Shot spacing was 500 m, instead of 25 m for Figure B.5.	112
B.8 Poorly illuminated angle-domain common-image gathers corresponding to $x = 4000$ m. Panels (a), (b) and (c) were migrated with velocity models that were correct, 6% too low, and 6% too high, respectively.	112
C.1 Comparison of normalizing adjoint linear interpolation with the row-sum of $\mathbf{A}'\mathbf{A}$ as opposed to its diagonal. Top panel shows the model. Center panel shows 100 interpolated data points. Lower panel shows reestimated data: dashed-line after diagonal normalization, dotted-line after column-based normalization. The solid-line shows the true model.	115
C.2 Same as Figure C.1, but with only 30 data points. The diagonal-based approach (dashed-line) is less robust in areas of poor coverage than the row-sum approach (dotted-line).	116

Chapter 1

Introduction

Many functions in mathematics are reversible: for example, if you consider the function "add seven to a number", you can reverse its effect by subtracting seven to recover the original number. Other functions are not reversible: for example, if you apply the function "multiply a number by zero", there is nothing you can do to recover the original value. In general, functions are only reversible if there is a one-to-one mapping between their inputs and outputs.

If I take a number and square it, I can then try and reverse the process by taking a square root. Unfortunately, however, an ambiguity exists over the original sign of the number. Since both positive and negative numbers have the same square, the squaring process is not reversible.

In this thesis, I am interested in taking square roots not of single numbers, but of multi-dimensional wavefields and operators. The wavefields are very large, containing tens of millions of pixels, and the operators represent wave-propagation through very complicated structures. However, a fundamental problem remains the same: when taking the square root of nine, do we choose three, or minus three?

1.1 Spectral factorization

Calculating the autocorrelation of a function is similar to calculating the square of a number. However, rather than just losing information about the sign, we lose information about the function's phase. Rather than there being two numbers that both have the same square, there is an infinite family of functions that share the same autocorrelation. This leaves us with a much larger ambiguity.

The simplest way to resolve the ambiguity is to set the phase to zero. Unfortunately, this solution results in *acausal* functions — energy starts to arrive before time zero. This is somewhat non-intuitive, and inappropriate for most physical systems, as it implies a ball moves before it's kicked.

A second alternative to resolve the ambiguity is to insist on a causal function, but one whose energy is packed as close to time zero as possible. This is known as the *minimum-phase* solution. As well as being causal and relatively compact in time, minimum-phase functions have another interesting property: the inverse of a minimum-phase function is also minimum phase, and hence causal. Because of these properties, it turns out that many physical systems fit the minimum phase model.

There is only one possible minimum-phase function with a given autocorrelation, and spectral factorization is the problem of determining that unique function.

One-dimensional solutions to the spectral factorization problem are well known: for example, Claerbout (1992) describes several approaches, including the Kolmogorov algorithm (Kolmogorov, 1939) which I briefly review in Chapter 2. For multi-dimensional signals, however, the problem itself is less clear: what exactly constitutes a causal function in multi-dimensional spaces?

1.1.1 Multi-dimensional spectral factorization on a helix

Claerbout (1998b) describes the isomorphic process by which multi-dimensional functions can be mapped into equivalent one-dimensional functions. The process depends on the concept

of the helical boundary conditions, and is best illustrated by Figure 1.1, which shows a small five-point filter on a two-dimensional space, being mapped into an equivalent one-dimensional filter.

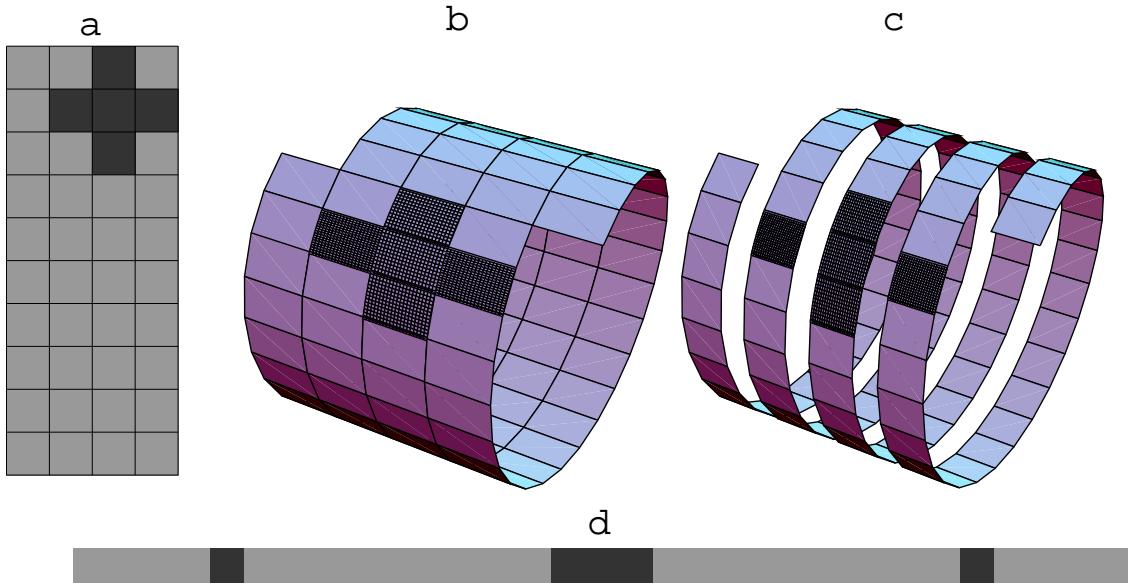


Figure 1.1: Illustration of helical boundary conditions mapping a two-dimensional function (a) onto a helix (b), and then unwrapping the helix (c) into an equivalent one-dimensional function (d). Figure by Sergey Fomel. [\[int-helix\]](#) [NR]

Under such a transformation, the concepts of causality and minimum-phase become clear. One-dimensional spectral factorization algorithms can be directly applied to the multi-dimensional helical functions.

1.2 Applications of multi-dimensional spectral factorization

This thesis describes three applications of multi-dimensional spectral factorization that are relevant to exploration geophysics.

1.2.1 Spectral factorization of seismic wavefields

The first part (Chapter 2) covers the multidimensional spectral factorization of passive seismic wavefields. I show that calculating the 3-D spectral factorization of a passive seismic wavefield amounts to estimating the medium's 3-D impulse response. Helioseismic dopplergrams provide a high-quality passive seismic dataset that has uniquely dense spatial coverage. This enables me to compare the solar impulse response from derived by spectral factorization with that derived by cross-correlation. The spectral factorization results contain significantly higher bandwidth, both temporally and spatially. I then show it is possible to derive active-source seismograms from terrestrial passive seismic data by crosscorrelating and factorizing data collected during a small passive seismic experiment conducted in Long Beach, CA.

1.2.2 Spectral factorization of partial differential equations

In the second part, I use multidimensional spectral factorization to facilitate solving multidimensional partial differential equations (PDEs) with application to exploration seismology. As an illustration of the methodology, in Chapter 3, I show how the Helmholtz operator can be factored into two filters, which then can be used to propagate waves. With more practical interests in mind, in Chapter 4, I apply the methodology to the problem of implicit finite-difference modeling and migration by wavefield extrapolation. Chapter 5 describes how the method can be applied in areas where the velocity varies laterally.

1.2.3 Spectral factorization of linear modeling operators

Most industrial-strength geophysics involves filtering the recorded data with the adjoint of the physical process that created it. If the true earth model is \mathbf{m} and \mathbf{A} is the physical forward modeling operator, then the image we compute is $\mathbf{A}'\mathbf{A}\mathbf{m}$. In the final part of this thesis, I form approximations to the $\mathbf{A}'\mathbf{A}$ operators associated with prestack depth migration that are diagonal in physical space. Since the approximations are diagonal, I can easily compute two factors simply by taking the square root of their diagonal elements. These factors are also

diagonal, and hence easily invertible. These diagonal factors can then be applied directly to the migrated image to produce an image whose amplitude more closely resembles those of the true earth model. Alternatively, the factors can be applied in concert with the original operator, to produce a new dimensionless composite operator, which is more easily invertible with iterative linear solvers.

In Chapter 6, I discuss how to calculate the shot-illumination cheaply during shot-profile migration. I then show that for sparse-shot geometries with dense receiver coverage this weighting function can completely compensate for illumination problems on flat events.

Lastly, in Chapter 7, I compare alternative methods of computing appropriate diagonal model-space and data-space weighting functions appropriate for generic linear operators, and discuss how model-space and data-space weights can be calculated and applied simultaneously.

Chapter 2

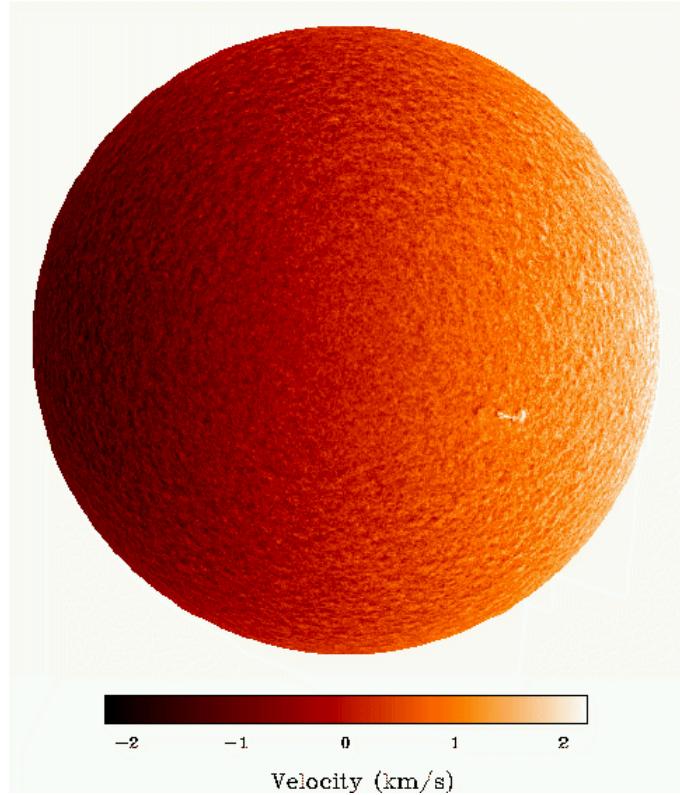
Spectral factorization of seismic oscillations

Helioseismologists study the acoustic oscillations visible on the surface of the sun to determine the sun's internal structure. The source function for the oscillations is stochastic, generated by turbulence in the convective region of the solar interior. To remove the effects of the stochastic source function, helioseismologists calculate *time-distance* curves by crosscorrelating passively observed seismograms.

In this chapter, I show how multi-dimensional spectral factorization can improve the spatial and temporal resolution of time-distance curves significantly. Time-distance curves calculated in this way have the same spatial and temporal bandwidth as the original data, rather than the decreased bandwidth obtained by crosscorrelating traces. Additionally, the spectral factorization impulse response is minimum phase, as opposed to the zero phase time-distance curves produced by crosscorrelation.

In the final section, I show crosscorrelation and spectral factorization results from a small terrestrial passive 3-D survey, recorded in Long Beach, CA. In total, about four hours of 60-channel passive data were recorded and processed. Although no clear reflection events are visible, a coherent dispersive ground-roll event is visible, and the correlograms do resemble active source seismograms.

Figure 2.1: Raw full-disc solar dopplergram from 9 July 1996, 09.00. Colormap shows line-of-site velocity with approximately 0.02 km/s uncertainty. Figure from the Solar Oscillations Investigation (SOI) website, <http://soi.stanford.edu>.
 solar-dopplergram [NR]



2.1 Introduction to helioseismology

Helioseismologists study the internal structure of the sun by observing acoustic oscillations on its surface. One of their instruments, the Michelson Doppler Imager (MDI), aboard the SOHO satellite, measures the Doppler shift of solar absorption lines formed in the lower part of the solar atmosphere (Scherrer et al., 1995). This provides line-of-sight velocity measurements for points on the sun's surface that can be used to study solar oscillations. The MDI instrument records a single dopplergram every minute; every dopplergram contains 1024×1024 pixels, and the field of view can be adjusted to either include the full solar disc, or zoomed to image a smaller region at higher resolution. Figure 2.1 shows a raw single full-disc dopplergram. The colormap indicates line-of-site velocity. Both the overall left-to-right trend (due to the sun's rotation), and the radial pattern (an artifact of the line-of-sight measurement), are removed or compensated for before the data is analyzed.

Although it is difficult to see on a single image, the stochastic nature of the solar oscillations is also apparent in Figure 2.1. Oscillations are continually excited by turbulent convective cells within the solar interior. Unlike most terrestrial seismologists who work with sources that can be localized in time and space, helioseismologists have to work with a stochastic source function, and oscillations that are continually being excited.

Helioseismologists often decompose the stochastic wavefields into spherical harmonics (Gough and Toomre, 1991; Harvey, 1995; Kosovichev, 1999). In the spherical harmonic domain, the chaotic nature of the source manifests itself in a random phase spectrum, whereas the deterministic solar structure manifests itself in the amplitude spectrum. The spherical harmonic amplitude spectrum, therefore, provides an excellent tool for studying the whole sun at one time. However, small-scale events are only described by harmonic modes of very high-order. Spherical harmonic functions are therefore inefficient for studying small, localized area's of the sun's surface.

2.1.1 Time-distance helioseismology

Time-distance helioseismology is based upon crosscorrelating oscillatory dopplergram traces from different locations on the surface of the sun (Duvall et al., 1993). The crosscorrelation between two such traces provides information about the ray-paths that propagate energy between the two locations. This allows helioseismologists to study the kinematics of acoustic waves traveling between the two trace locations, facilitating a family of techniques that are proving very successful for studying a range of solar phenomena at a large range of scales. For example, time-distance measurements can be used to estimate both near surface flow velocities associated with super-granulation (Kosovichev and Duvall, 1997), which are very difficult to resolve with spherical harmonic analysis, and meridional circulation deep within the convective zone (Giles et al., 1997).

The process of picking traveltimes from time-distance curves is a critical element of these studies. Both signal-to-noise levels and signal bandwidth can limit the resolution of traveltime picks. Signal-to-noise can be increased by stacking individual crosscorrelograms with similar offsets. This amounts to taking the multi-dimensional autocorrelation of the original data,

and unfortunately, has the side-effect that it reduces the spatial and temporal bandwidth of the derived impulse response, by essentially squaring the (ω, k_x, k_y) amplitude spectrum.

I show that this problem can be avoided by looking at the multi-dimensional minimum-phase factor of the autocorrelogram rather than the autocorrelogram itself. The minimum phase time-distance impulse response has the same spectra as the original data, as opposed to its square.

2.2 Model of stochastic oscillations

Under an assumption of translational invariance, we can model the acoustic oscillation recorded in solar dopplergrams as a stochastic source function convolved with a Green's function impulse response. After a three-dimensional Fourier transform, the convolution becomes a simple multiplication:

$$D(k_x, k_y, \omega) = S(k_x, k_y, \omega)G(k_x, k_y, \omega), \quad (2.1)$$

where D is the observed data, S is the source function, and G is the impulse response. We are interested in the three-dimensional time-space acoustic impulse response, $g(x, y, t)$. As it stands, however, equation (2.1) has many more unknowns than knowns, so additional assumptions are required before we can estimate G .

Secondly, we assume $s(x, y, t)$ is white in space and time, or equivalently, $S\bar{S} = 1$, where the \bar{S} denotes the complex conjugate of S . If this is not true in practice, spectral color from the source function will leak into the derived impulse response.

Under this assumption, equation (2.1) reduces to the statement that the power spectrum of the impulse response equals the power spectrum of the data,

$$|D|^2 = |G|^2. \quad (2.2)$$

While defining the amplitude spectrum of G , this equation places no constraints on its phase, and so we need an additional assumption to ensure a unique solution.

We will assume that G is a minimum-phase function, where a minimum-phase function

is defined as a causal function with a causal convolutional inverse. It turns out that many physical systems (from electronics to acoustics) have this property. For example, consider the function that maps stress to strain and vice versa: if you specify the stress in a solid, the strain will react accordingly, leading to a causal function relating stress to strain. However, you could also specify the strain in the solid, and then the observed-stress would be a causal function of the strain. Since these two functions are inverse processes, they clearly satisfy the minimum-phase definition.

If this model holds true, then estimating the impulse response reduces to estimating a minimum-phase function with the same (ω, k_x, k_y) spectrum as the original data: or equivalently, multi-dimensional spectral factorization.

2.2.1 Kolmogorov review

Kolmogorov (1939) spectral factorization provides a highly efficient Fourier method for calculating a minimum phase time domain function with a given power spectrum. The Kolmogorov spectral factorization method is commonly referred to as the Hilbert transform method within the signal processing community [e.g. Robinson and Treitel (1980)], although they are equivalent.

Following Claerbout (1992), we will describe the method briefly with Z transform notation. In this notation, $Z = e^{i\omega\Delta t}$ is the unit delay operator, and functions can be evaluated either in the frequency domain as functions of ω , or in the time domain as the coefficients of the polynomial in Z . Causal functions can, therefore, be written as polynomials with non-negative powers of Z , whereas anti-causal functions contain non-positive powers of Z .

The spectral factorization problem can be summarized as given a power spectrum, $S(Z)$, we must find a minimum phase function such that

$$\bar{B}(1/Z)B(Z) = S(Z). \quad (2.3)$$

Since $S(Z)$ is a power spectrum, it is non-negative by definition for all ω ; however, the

Kolmogorov process places the additional requirement that $S(Z)$ contains no zeros. If this is the case, then we can safely take its logarithm,

$$U(\omega) = \ln [S(\omega)]. \quad (2.4)$$

Since $U(\omega)$ is real and even, its time domain representation is also real and even. We can therefore isolate its causal part, $C(Z)$, and its anti-causal part, $\bar{C}(1/Z)$:

$$U(Z) = \bar{C}(1/Z) + C(Z). \quad (2.5)$$

Once we have $C(Z)$, we can easily obtain $B(Z)$ through

$$B(\omega) = e^{C(\omega)}. \quad (2.6)$$

To verify that $B(Z)$ of this form does indeed satisfy equation (2.3), consider

$$\bar{B}(1/Z)B(Z) = e^{\bar{C}(1/Z)}e^{C(Z)} \quad (2.7)$$

$$= e^{\bar{C}(1/Z)+C(Z)} \quad (2.8)$$

$$= e^{U(Z)} \quad (2.9)$$

$$= S(Z). \quad (2.10)$$

$B(Z)$ will be causal since $C(Z)$ was causal, and a power series expansion proves that the exponential of a causal function is also causal. It is also clear that $1/B(Z) = e^{-C(Z)}$ will also be causal in the time domain. Therefore, $B(Z)$ will be causal, and will have a causal inverse. Hence $B(Z)$ satisfies the definition of minimum phase given above.

2.2.2 Multi-dimensional factorization

Kolmogorov spectral factorization, as described above, is a purely one-dimensional theory. By applying helical boundary conditions (Claerbout, 1998b), a three-dimensional stochastic dopplergram can be mapped into an equivalent one-dimensional dataset, and the entire cube

can be factored with Kolmogorov.

The concept of helical boundary conditions is reviewed in Chapter 1 and demonstrated in Figure 1.1, which shows the mapping of small five-point two-dimensional filter into one dimension. For this application, however, rather than map a two-dimensional function, I map the entire three-dimensional MDI dataset into one dimension, and apply Kolmogorov spectral factorization on the entire super-trace.

The spatial axes need to be padded to reduce wrap-around effects. This spatial wrap-around is not an artifact of the Fourier transform, but rather it is an artifact of the helical boundary conditions. In this respect, there would be little advantage to choosing a time-domain spectral factorization algorithm (Wilson, 1969) over Kolmogorov.

To summarize, I perform the spectral factorization in three steps. Firstly, I transform the cube of data to an equivalent one-dimensional super-trace via helical boundary conditions. Secondly, I perform one-dimensional spectral factorization with Kolmogorov's frequency domain method. Finally, I remap the impulse response back to three-dimensional space.

2.2.3 Theoretical comparison between time-distance functions

The Kolmogorov impulse response is essentially a large impulse at zero lag (in time and space) with a small amplitude signal corresponding to the diving waves. Both components are band-limited, so we can write

$$B(Z) = W(Z)[1 + \epsilon F(Z)], \quad (2.11)$$

where $F(Z)$ is the causal function of interest, ϵ is simply a scalar indicating the small amplitude of that term, and $W(Z)$ is a minimum phase band-limited seismic wavelet.

The crosscorrelation process produces the autocorrelation of equation (2.11):

$$\bar{B}B = \bar{W}W[1 + \epsilon F + \epsilon \bar{F} + \epsilon^2 \bar{F}F]. \quad (2.12)$$

This function contains ϵF , the function we are interested in studying; however there are two major differences.

Firstly, $\bar{B}B$ also contains the additional terms $\epsilon\bar{F}$ and $\epsilon^2\bar{F}F$. We can discard the first of these terms, $\epsilon\bar{F}$ since it is anti-causal, and the second term contains ϵ^2 so will be much smaller than the signal of interest.

The second difference between equations (2.11) and (2.12) is the wavelet. The Kolmogorov wavelet is minimum phase, whereas the crosscorrelation wavelet $\bar{W}W$ is zero-phase. The amplitude spectrum of the crosscorrelation wavelet will also be the square of the Kolmogorov wavelet.

Thus the principal advantage of the Kolmogorov result is that it has a broader bandwidth than the crosscorrelation. Whereas the Kolmogorov result has the same amplitude spectrum as the original data, the amplitude spectrum of the crosscorrelation impulse response is equal to the *power* spectrum of the original data.

2.2.4 On the assumption of translational invariance

The justification for spectral factorization rests upon an assumption of translational invariance. This assumption runs counter to many applications of time-distance helioseismology, where the interest comes in three-dimensional structure. The assumption of translation invariance may be partly overcome by working with patches of data with small spatial extent.

Time-distance measurements by crosscorrelation may not seem to have this perceived disadvantage; however, studies (Kosovichev and Duvall, 1997) have shown that significant amounts of averaging are required to produce signal-to-noise levels high enough to make reliable measurements. There is an implied assumption of invariance in this averaging procedure.

2.3 Application to SOHO/MDI dataset

Figure 2.2 shows a cube of raw velocity data from the MDI instrument. The data have been transformed to Cartesian coordinates by projecting high-resolution data from an area approximately 18° square onto a tangent plane. The object in the center of the time-slice (top of cube) is a sun-spot (irrelevant to this study). The sample spacing is 1 minute on the time-axis and

approximately 825 km on the two spatial axes.

Time-variable features of Figure 2.2 fall into two distinct spectral windows. The low temporal frequency events (<1.5 mHz) are related to solar convection, while the higher frequency events are related to acoustic wave propagation. We were interested in studying acoustic wave phenomena; so as a preprocessing step, we removed the lower frequency spectral window by applying a low-cut filter to the data.

Figure 2.3 shows the impulse response derived by multi-dimensional spectral factorization. Due to the lack of coherent reflectors within the solar interior, no reflection events are visible. The strong vertical velocity gradient causes the first arrival to be a diving wave. Later arrivals are the multiple events of increasing order (PP, PPP etc.) illustrated in Figure 2.4.

2.3.1 Time-distance functions compared

Figure 2.5 shows a comparison between the impulse response derived from Kolmogorov spectral factorization, and the impulse response derived by crosscorrelation.

The raw MDI data has a narrow temporal bandwidth with most of its energy having a period of about five minutes: squaring the amplitude spectrum reduces this bandwidth even more resulting in the monochromatic appearance of the left panel in Figure 2.5. Moreover, it is not just the temporal bandwidth that is decreased by crosscorrelating traces; but the spatial bandwidth is reduced as well. The steep dips associated with the l - and low p -modes are clearly visible near the origin in the right panel of Figure 2.5 are very heavily attenuated in the crosscorrelation impulse response (left panel).

This difference in spatial bandwidth can be also be seen in the amplitude spectra of Figure 2.6. The amplitude of l - and low p -modes are much lower in the autocorrelation result than in the Kolmogorov result.

Whereas, the temporal bandwidth may be broadened relatively simply by conventional deconvolution, recovering the full spatial bandwidth that is present in the original data would be more difficult.

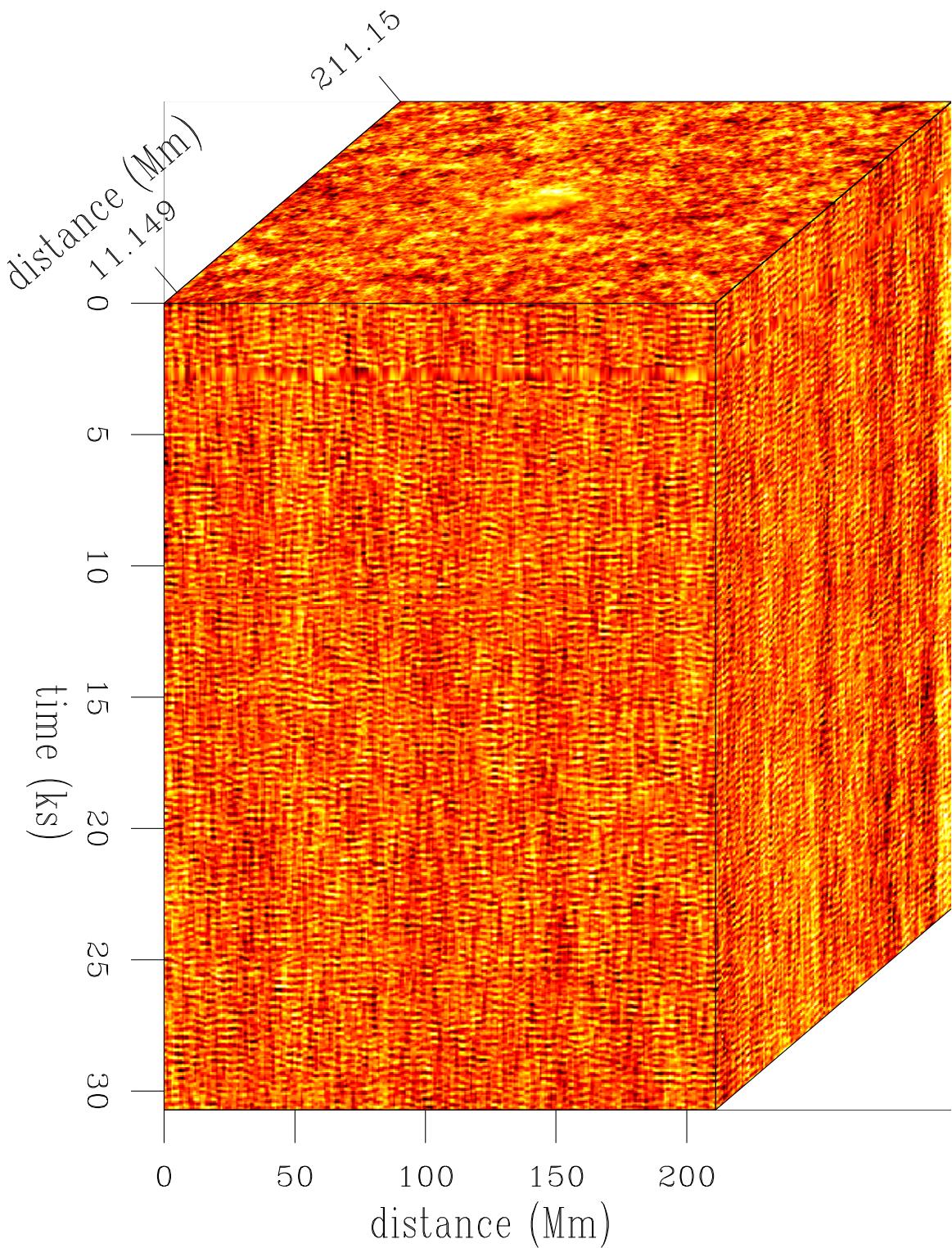


Figure 2.2: Cube of MDI data. `solar-tallcube` [ER]

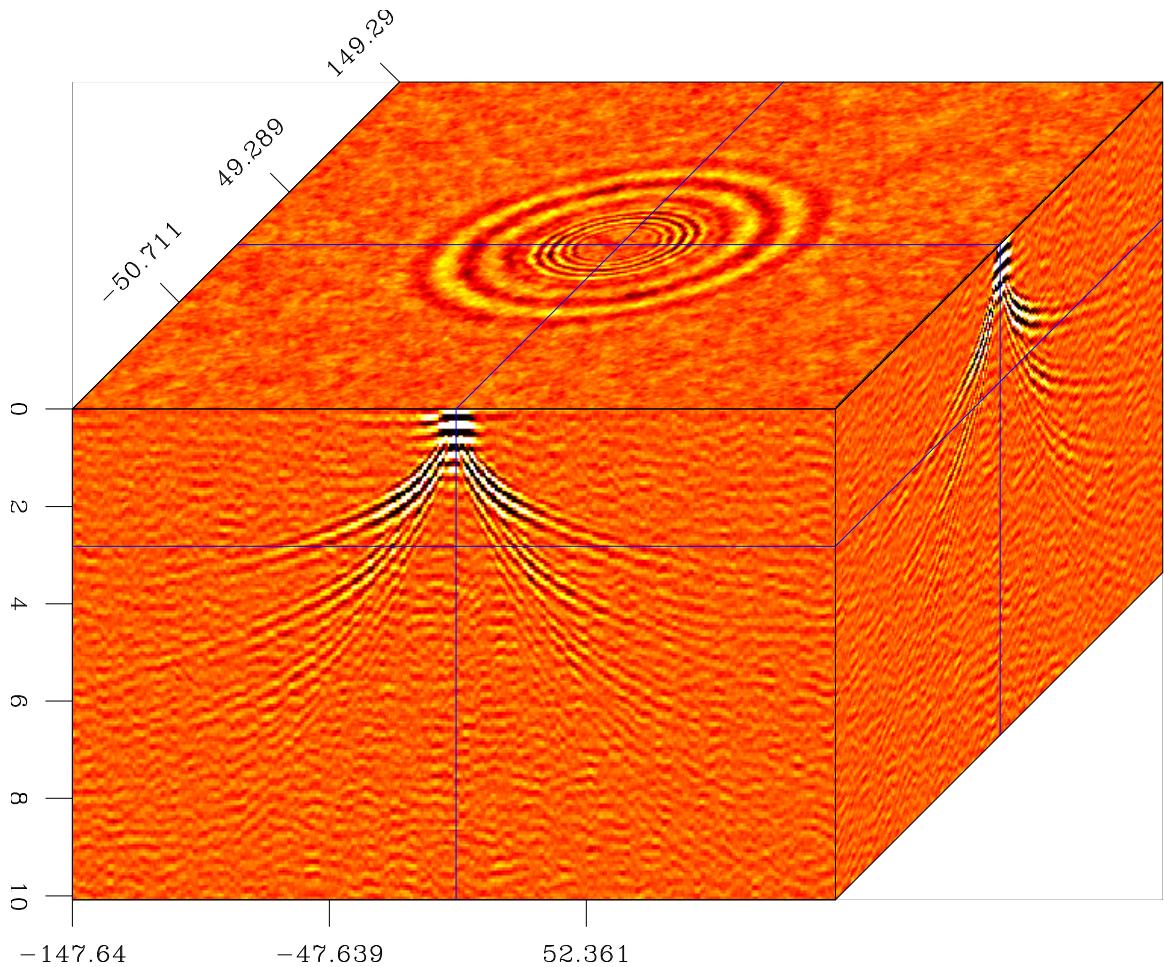


Figure 2.3: Three-dimensional minimum phase impulse response. solar-kolcube [CR,M]

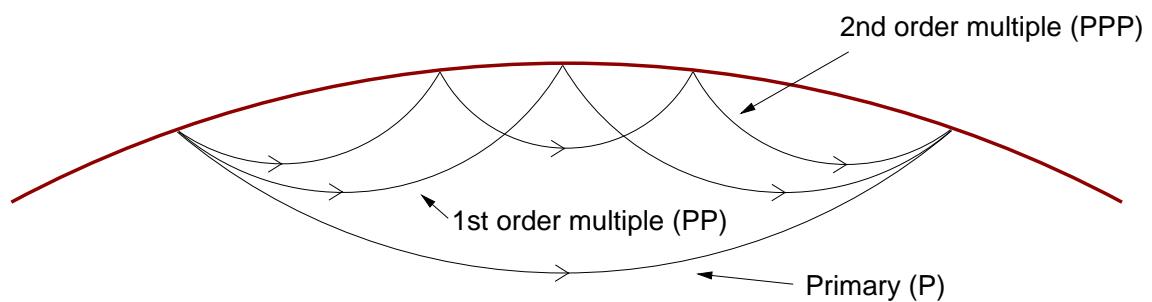


Figure 2.4: Schematic showing some of the raypaths visible in helioseismic time-distance plots. solar-multiples [NR]

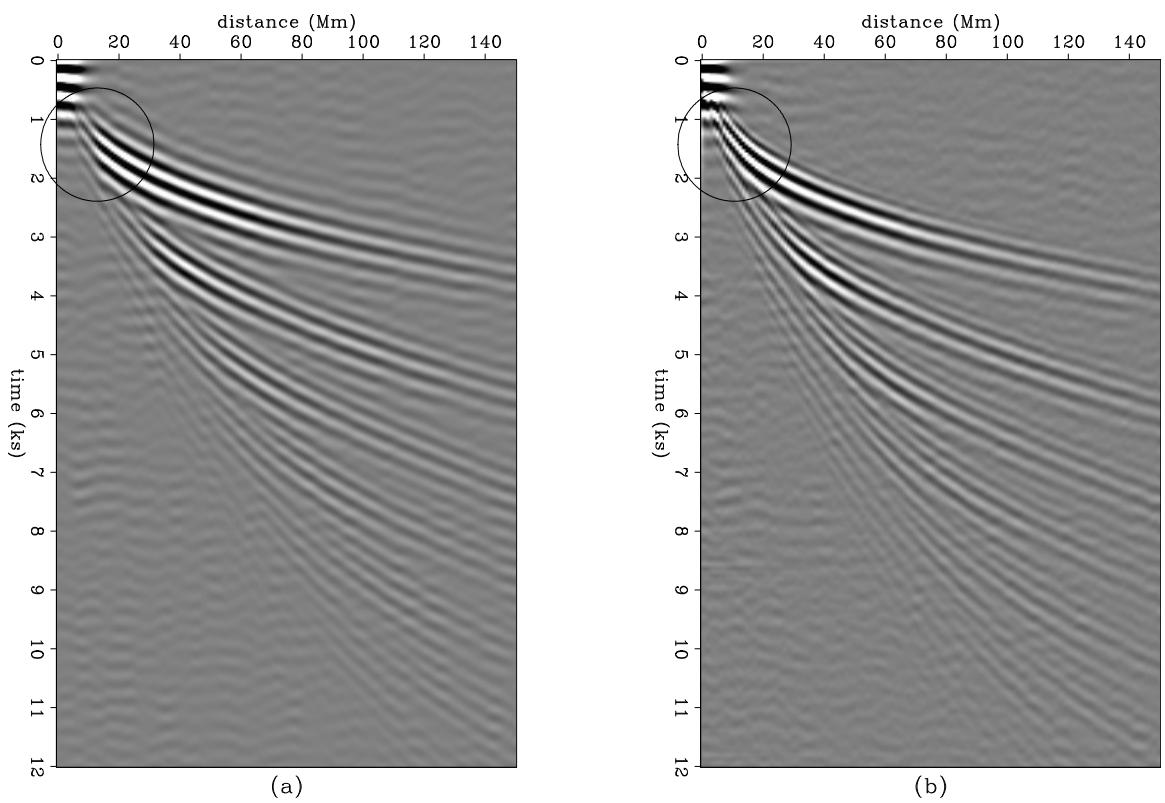


Figure 2.5: Time-distance impulse responses computed by (a) three-dimensional autocorrelation, and (b) three-dimensional Kolmogorov spectral factorization. Traces have been binned as a function of radius from the origin, averaging the impulse response over azimuth.

solar-stacks [CR,M]

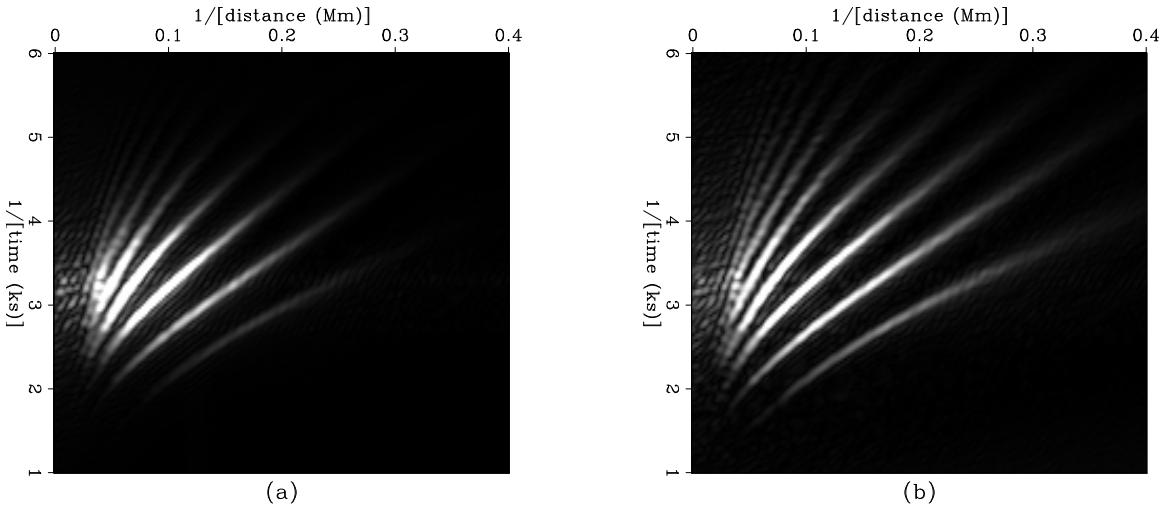


Figure 2.6: Two-dimensional amplitude spectra of impulse responses shown in Figure 2.5 above. The impulse responses were computed by (a) three-dimensional autocorrelation, and (b) three-dimensional Kolmogorov spectral factorization. solar-fk [CR,M]

2.4 Application to a terrestrial passive seismic dataset

Since Claerbout (1968) described the process of creating reflection seismograms by autocorrelating transmission seismograms, terrestrial geophysicists have been trying similar methods to obtain active-source seismograms from passive seismic data, but without the success of the helioseismologists.

Baskir and Weller (1975) attempted to use passive seismic energy to image subsurface reflectivity. They briefly describe crosscorrelating long seismic records to produce correlograms that could be processed, stacked and displayed as conventional seismic data. Unfortunately their field tests seem to have been inconclusive.

Cole (1995) also attempted to create impulsive reflection seismograms from passive data with data collected using a 4000 channel 2-D field array on Stanford campus. Unfortunately, again, possibly due to the short (20 minute) records or bad coupling between the geophones and the dry California soil, his results were inconclusive.

2.4.1 Long Beach experimental setup and preprocessing

In August 2000, Bryan Kerr conducted an experiment (Kerr and Rickett, 2001) with 60 single channel geophones laid out in an 10×6 array with 5 m receiver spacing on some unused land in Long Beach, CA. The immediate environs of the experiment could be classified as mixed light industrial/low-density residential, and possible sources of ambient seismic noise included a pair of freeways about 500 m away, some overhead power cables running above the experiment, and a drill truck about 50 m away. Kerr collected 1020 (10 second) records of passive data totaling 170 minutes.

As a preprocessing step, I notch-filtered the data to remove monochromatic spikes (e.g. 60 Hz noise). Although I later low-pass filtered the data to 30 Hz, high amplitude spikes need to be removed before crosscorrelation/factorization to avoid numerical underflow. Figure 2.7 shows 5 seconds of passive data after notch-filtering. Two types of event are clearly visible in Figure 2.7: slow linear events that appear to be traveling in the direction of decreasing channel number, and hyperbolic events with a higher apparent velocity. I speculate that these events were generated by the drill truck and power lines, respectively.

Following the processing steps applied to the helioseismic data, I crosscorrelated (or spectrally factorized) the data, stacked over the record axis, and then stacked over radius (distance from zero spatial lag). I then applied a (30 Hz) low-pass filter to the data to remove aliased ground-roll energy, and finally applied an offset dependent gain correction `xpow=1` for display purposes.

2.4.2 Time-distance functions compared

Figure 2.8 shows the impulse responses derived by autocorrelation and spectral factorization of about 95 minutes of passive data. The rest of the passive data contained significant coherent noise below 30 Hz, which caused degraded results.

From the results presented here, it is not clear that any reflection events are present in the data. However, both the autocorrelation and spectral factorization result do resemble active source seismograms. The slightly dispersive ground-roll cone is clearly visible with a velocity

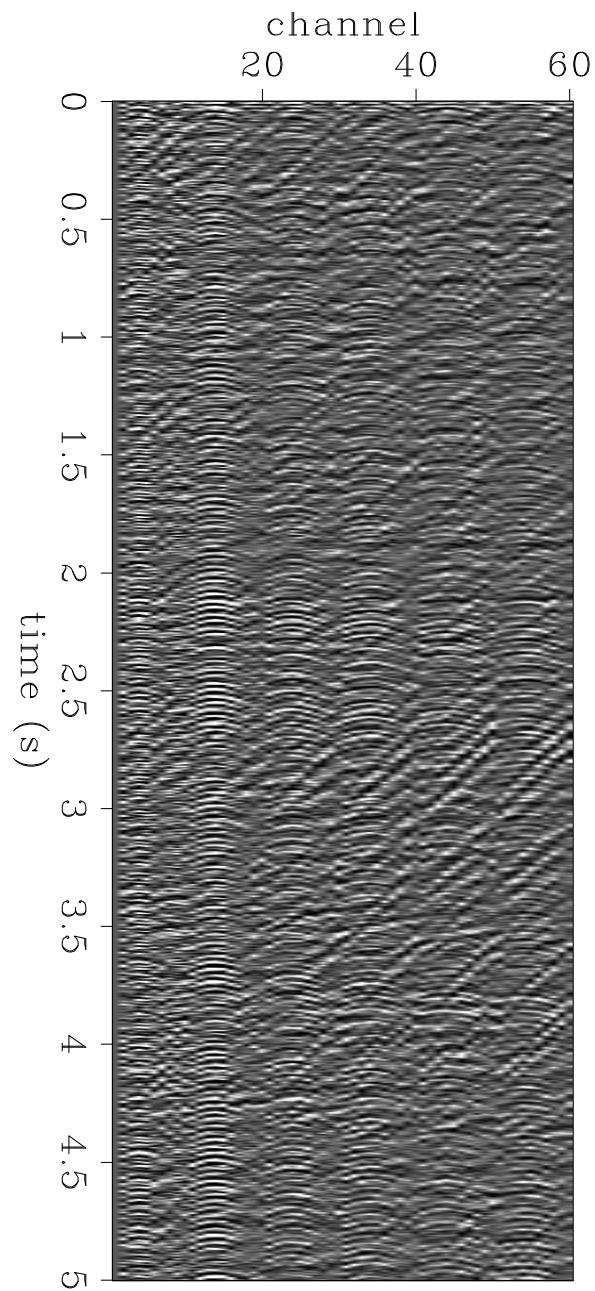


Figure 2.7: Five seconds of passive seismic data from the Long Beach experiment after notch-filtering.
solar-strip [CR]

of about 150 m/s.

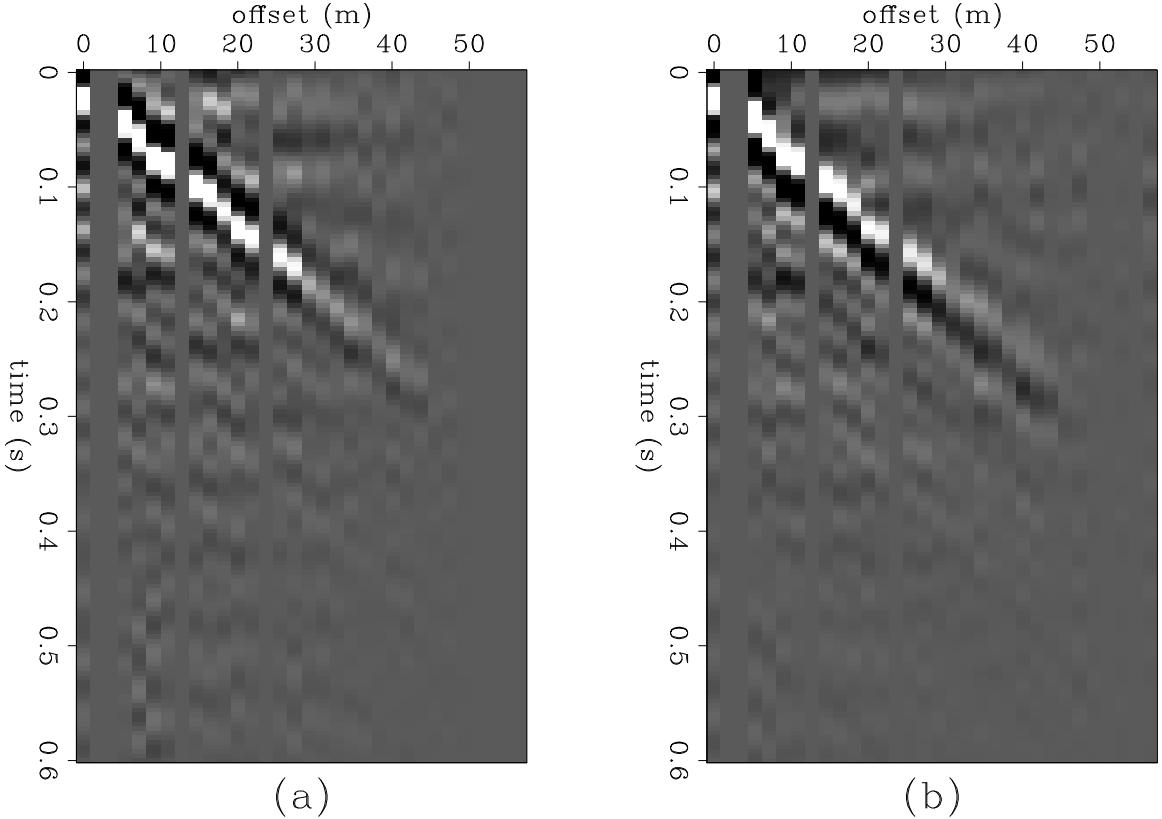


Figure 2.8: Impulse response calculated by (a) 3-D autocorrelation, and (b) spectral factorization. Compare with Figure 2.5. solar-longimp [CR,M]

2.5 Conclusions

Under assumptions of translational invariance, a white source function, and a minimum-phase impulse response, I have shown that estimation of the sun's acoustic time-distance impulse response amounts to multi-dimensional spectral factorization.

I performed the spectral factorization by transforming the entire multi-dimensional helioseismic wavefield into a one-dimensional super-trace, which could be factored efficiently with Kolmogorov's frequency domain method. I showed that time-distance curves obtained

by spectral factorization have a broader temporal and spatial bandwidth equivalent curves calculated by crosscorrelation.

In the final section, I calculated the 3-D autocorrelation and spectral factorization of a small passive dataset collected in Long Beach, CA. The seismograms resemble active-source seismograms with a clearly visible ground-roll event.

Chapter 3

Helical factorization of the Helmholtz equation

In this chapter, I demonstrate how multi-dimensional spectral factorization allows the efficient solution of constant coefficient partial differential equations (PDE's). I begin with the simple example of Poisson's equation: following Claerbout (1998b), I show how the helical coordinate system allows the factorization of the finite-difference stencil into a pair of minimum phase filters that can be inverted rapidly by back-substitution.

Moving to a wave propagation example, I construct a finite-difference approximation to the Helmholtz operator in the (ω, x) domain that describes propagation of a single frequency wave. This filter can also be factored into a pair of filters, and I show these factors can act as recursive one-way wave propagators that are accurate up to 90° .

3.1 Introduction

Depth migration algorithms are important for imaging in areas with strong lateral velocity gradients. Wavefield extrapolation algorithms in the (ω, x) domain have the advantage over Kirchhoff depth migration methods that they are based on finite bandwidth solutions to the wave-equation, rather than asymptotic approximations. Additionally, they have the advantage

over (ω, k) methods that they can incorporate lateral velocity variations in a single migration.

One of the aims of this thesis is to facilitate (ω, x) implicit finite-difference depth migrations, by providing a fast and efficient solution to wave-equation PDE's. To illustrate the underlying concepts, I begin with a review of the spectral factorization of Poisson's equation discussed by Claerbout (1998b). I then show how similar techniques can be used to solve the full Helmholtz wave-equation.

Traditional (ω, x) wavefield extrapolation algorithms are based on solutions to one-way wave equations, which can be derived by making paraxial approximations to the Helmholtz equation. For example, Claerbout (1985) describes implicit 2-D wavefield extrapolation based on the Crank-Nicolson formulation. Later, Hale (1990a; 1990b) popularized explicit extrapolation in 3-D by improving its stability and efficiency.

Rather than making a paraxial approximation, in this chapter, I construct a finite-difference stencil that approximates the *two-way* Helmholtz operator in the (ω, x) domain. The helical coordinate system then allows me to remap the multi-dimensional operator into one-dimensional space, where I can find two minimum-phase factors using a conventional spectral factorization algorithm.

The factorization provides a pair of filters: one causal minimum-phase and one anti-causal maximum-phase. I show that recursive application of the former propagates waves vertically downwards, and the latter propagates waves vertically upwards.

3.2 Direct solution of Poisson's equation

As a simple illustration of how helical boundary conditions can lead to recursive solutions to partial differential equations, consider Poisson's equation, which in the constant coefficient case relates potential, u , to source density, f , through the Laplacian operator:

$$\nabla^2 u = -f(x, y, z) \quad (3.1)$$

Poisson's equation crops up in many different branches of physics: for example, in electrostatics, gravity, fluid dynamics (where the fluids are incompressible and irrotational), and steady-state temperature studies. It also serves as a simple analogue to the wave-propagation equations which provide the main interest of this thesis.

To solve Poisson's equation with constant coefficients on a regular grid (Claerbout, 1997), we can approximate the differential operator, $-\nabla^2$, by a convolution with a small finite-difference filter, d . Taking the operator, \mathbf{D} , to represent convolution with filter, d , Poisson's equation becomes

$$\mathbf{D} \mathbf{u} = \mathbf{f}. \quad (3.2)$$

Although \mathbf{D} itself is a multi-dimensional convolution operator that is not easily invertible, helical boundary conditions (Claerbout, 1997) allow us to reduce the dimensionality of the convolution to an equivalent one-dimensional filter, which we can factor into the product of a lower-triangular matrix, \mathbf{L} , and its transpose, \mathbf{L}^T . These triangular matrices represent causal and anti-causal convolution with a minimum-phase filter, in the form

$$\mathbf{D} \mathbf{u} = \mathbf{L} \mathbf{L}^T \mathbf{u} = \mathbf{f}. \quad (3.3)$$

The operator, \mathbf{L} , is the *helical derivative*, discussed in more detail by Claerbout (1998a). We can calculate \mathbf{u} directly from equation (3.3) since \mathbf{L} and its transpose are easily invertible by recursive polynomial division:

$$\mathbf{u} = (\mathbf{L}^T)^{-1} \mathbf{L}^{-1} \mathbf{f}. \quad (3.4)$$

The right panel of Figure 3.1 shows the solution of the Poisson's equation with the single source and sink shown in the left panel. The center panel shows the intermediate result, $\mathbf{L}^{-1} \mathbf{f}$.

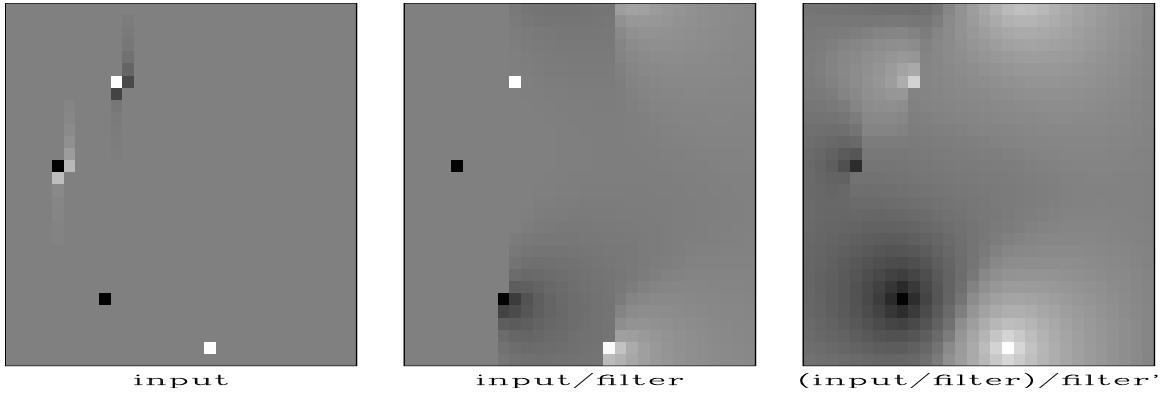


Figure 3.1: Deconvolution by a filter whose autocorrelation is the two-dimensional Laplacian operator. This amounts to solving the Poisson equation. After Claerbout (1997). `helmholtz-lapfac [ER]`

3.3 The Helmholtz equation

Starting from the acoustic wave equation in three-dimensions,

$$-\nabla^2 p + \frac{1}{v^2} \frac{\partial^2 p}{\partial t^2} = 0, \quad (3.5)$$

we can Fourier transform the time axis, and look for (ω, \mathbf{x}) solutions of the form:

$$p(\omega, \mathbf{x}) = q(\mathbf{x}) e^{-i\omega t}. \quad (3.6)$$

For a single frequency, the wave equation therefore reduces to the Helmholtz (time-independent diffusion) equation

$$\left(-\nabla^2 - \frac{\omega^2}{v^2} \right) q(\mathbf{x}) = 0. \quad (3.7)$$

We can solve the Helmholtz equation on a regular grid by approximating the differential operator with a finite-difference stencil. A simple (all-zero) convolutional approximation to the Laplacian, $-\nabla^2 \approx \mathbf{D}$, produces the matrix equation:

$$\left(\mathbf{D} - \frac{\omega^2}{v^2} \mathbf{I} \right) \mathbf{q} = \mathbf{0} \quad (3.8)$$

Unfortunately the direct solution of equation (3.8) requires the inversion of a multi-dimensional convolutional matrix. As with Poisson's equation above, the application of helical boundary conditions allows me to factor the convolutional filter into a minimum-phase causal and anti-causal pair that can be inverted rapidly by polynomial division.

Factoring the Helmholtz operator is not as simple as factoring the Poisson operator, since its spectrum is not positive definite. The spectrum of the differential Helmholtz operator can be obtained by taking the spatial Fourier transform of equation (3.7), to give

$$\left(|\mathbf{k}|^2 - \frac{\omega^2}{v^2} \right) \hat{q}(\mathbf{k}) = S(\mathbf{k}) \hat{q}(\mathbf{k}) = 0, \quad (3.9)$$

where $\hat{q}(\mathbf{k})$ represents the spatial Fourier transform of $q(\mathbf{x})$, and $S(\mathbf{k}) = |\mathbf{k}|^2 - \frac{\omega^2}{v^2}$ is the Fourier representation of the Helmholtz operator. $S(\mathbf{k})$ clearly becomes negative real for small values of $|\mathbf{k}|$; so as it stands, the Helmholtz operator does not represent an autocorrelation, and is not factorable. This problem also exists for discrete operators [e.g. equation (3.8)].

Fortunately replacing ω by $\hat{\omega} = \omega + \epsilon i$, where ϵ is a small positive number, successfully stabilizes the spectrum, by pushing the function off the negative-real axis. Equation (3.8), therefore becomes

$$\left(\mathbf{D} - \frac{\hat{\omega}^2}{v^2} \mathbf{I} \right) \mathbf{q} = \mathbf{0}. \quad (3.10)$$

The physical effect of ϵ is to provide damping as the wave propagates, differentiating between the forward and backward extrapolation directions. With the complex $\hat{\omega}$, the Helmholtz operator is symmetric but not Hermitian, and so represents a crosscorrelation rather than an autocorrelation.

Generalizing the concept of spectral factorization to cross-spectral factorization, Claerbout (1998c) showed that any function whose Fourier transform does not wrap around the origin in the complex plane can be factored into the crosscorrelation of two minimum-phase factors. Since for this class of function, the phase of the Fourier component at the positive Nyquist equals the phase at the negative Nyquist (with no bulk phase shift), he termed this class of function ‘level-phase’. Although the Helmholtz operator is not strictly an autocorrelation, it

does satisfy the ‘level-phase’ criterion, and so it can still be factored into a pair of minimum-phase factors.

Rather than considering a simple convolutional approximation to the Laplacian, we can form a rational approximation,

$$-\nabla^2 \approx \frac{\mathbf{D}}{\mathbf{I} + \beta\mathbf{D}} \quad (3.11)$$

where β ($\approx 1/6$) is Claerbout’s (1985) adjustable ‘one-sixth’ parameter, and \mathbf{D} again represents convolution with a simple finite-difference filter, d . It is safe to write equation (3.11) in this rational form since the numerator and denominator commute with each other.

Inserting equation (3.11) into equation (3.10) yields a matrix equation of similar form, but with increased accuracy at high spatial wavenumbers:

$$\left(\frac{\mathbf{D}}{\mathbf{I} + \beta\mathbf{D}} - \frac{\hat{\omega}^2}{v^2} \mathbf{I} \right) \mathbf{q} = \mathbf{0} \quad (3.12)$$

$$\left[\left(\frac{v^2}{\hat{\omega}^2} - \beta \right) \mathbf{D} - \mathbf{I} \right] \mathbf{q} = \mathbf{H} \mathbf{q} = \mathbf{0} \quad (3.13)$$

The operator on the left-hand-side of equation (3.13) represents a multi-dimensional convolution matrix, that can be mapped to an equivalent one-dimensional convolution by applying helical boundary conditions. Although the complex coefficients on the main diagonal cause the matrix not to be Hermitian, the spectrum of the matrix is of level-phase. Therefore, for constant v , it can be factored into causal and anti-causal (triangular) components with any spectral factorization algorithm that has been adapted for cross-spectra (Claerbout, 1998c).

$$\mathbf{H} \mathbf{q} = \mathbf{U} \mathbf{L} \mathbf{q} = \mathbf{0} \quad (3.14)$$

The challenge of extrapolation is to find \mathbf{q} that satisfies both the above equation and our initial conditions, $\mathbf{q}_{z=0}$. Starting from $\mathbf{q}_{z=0}$, we can invert \mathbf{L} recursively to obtain a function that satisfies both the initial conditions, and

$$\mathbf{L} \mathbf{q} = \mathbf{0}. \quad (3.15)$$

Gazdag:	$R(\mathbf{k}_x) = e^{i\sqrt{\omega^2/v^2 - \mathbf{k}_x ^2}}$
Implicit:	$R(\mathbf{k}_x) = e^{i\omega/v} \frac{A(\mathbf{k}_x)}{B(\mathbf{k}_x)}$
Explicit:	$R(\mathbf{k}_x) = e^{i\omega/v} C(\mathbf{k}_x)$
Helmholtz factorization:	$R(\mathbf{k}_x) \simeq \frac{1}{L(\mathbf{k})}$

Table 3.1: Comparison of the mathematical form of various wavefield extrapolators.

Hence \mathbf{q} will also satisfy equation (3.14).

3.3.1 Wave extrapolation

The basis for wavefield extrapolation is an operator, R , that marches the wavefield q , at depth z , down to depth $z + \Delta z$.

$$q_{z+\Delta z} = R q_z. \quad (3.16)$$

In constant velocity, R will be a function of horizontal wavenumber, $\mathbf{k}_x = (k_x \ k_y)^T$, and ideally, $R(\mathbf{k}_x)$ will have the form of the phase-shift operator (Gazdag, 1978),

$$R(\mathbf{k}_x) = e^{i\sqrt{\frac{\omega^2}{v^2} - |\mathbf{k}_x|^2}}. \quad (3.17)$$

Due to lateral velocity variations, and the desire to avoid spatial Fourier transforms, approximations to R are often applied in the (ω, x) domain. Typically R is split into a ‘thin-lens’ term that propagates the wave vertically, and a ‘diffraction’ term that models more complex wave phenomena. In the (ω, x) domain, the thin-lens term can be applied as a simple phase-shift, while the diffraction term is approximated by a small finite-difference filter. The method of extrapolation determines the nature of the finite-difference filter. The mathematical forms of different extrapolators are summarized in Table 1, and discussed below.

Implicit extrapolation (discussed in more detail in following chapters) approximates $R(\mathbf{k}_x)$

with a rational form, consisting of a convolutional filter, and an inverse filter,

$$R(\mathbf{k}_x) = e^{\frac{i\omega}{v}} \frac{A(\mathbf{k}_x)}{B(\mathbf{k}_x)}. \quad (3.18)$$

In constant velocity, the traditional Crank-Nicolson implicit formulation ensures the pair of convolutional operators, A and B , are complex conjugates, and so the resulting extrapolator is unitary.

Practical 3-D extrapolation is often done with an explicit operator using McClellan transforms. This approach amounts to approximating $R(\mathbf{k}_x)$ by with a simple convolutional filter, $C(\mathbf{k}_x)$. Explicit extrapolators, therefore, have the form

$$R(\mathbf{k}_x) = e^{\frac{i\omega}{v}} C(\mathbf{k}_x). \quad (3.19)$$

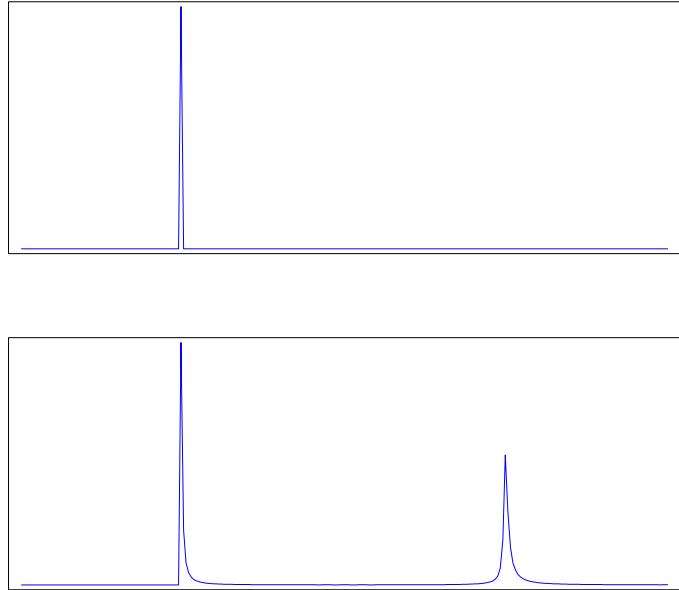
In contrast to these methods, the minimum-phase factorization of the Helmholtz operator provides a recursive depth extrapolator of a different form:

$$R(\mathbf{k}_x) \simeq \frac{1}{L(\mathbf{k})}, \quad (3.20)$$

where $L(\mathbf{k})$ is a minimum-phase filter. Because L is a function of $\mathbf{k} = (k_x \ k_y \ k_z)^T$, rather than \mathbf{k}_x , extrapolation with the Helmholtz factorization does not fit exactly with equation (3.16). In practice the wavefield needs to be zero-padded in depth before extrapolation, and so equation (3.20) is not written as an equality.

The apparent contradiction that we are approximating the unitary (delay) operator in equation (3.17) with the minimum-phase extrapolator in equation (3.20) is resolved by examining the impulse response of the operator $\frac{1}{L(\mathbf{k})}$ shown in Figure 3.2. The impulse response consists of two ‘bumps’. The first bump is the response of the impulse at the same depth step as the impulse. Because it looks like a delta function, it leaves that depth step essentially unchanged. The second bump, on the other hand, is the response to the impulse at the following depth step — it describes the wave propagation in depth. When taken together, the first and second bumps are indeed minimum phase; however, the second bump controls wave propagation in depth and is almost pure delay.

Figure 3.2: Amplitude of impulse response of polynomial division with minimum-phase factorization of the Helmholtz equation. The top panel shows the location of the impulse. The bottom panel shows the impulse response. Helical boundary conditions mean the second bump in the impulse response corresponds to energy propagating to the next depth step. [\[helmholtz-impresp\]](#) [ER]



3.3.2 Propagating waves with the Wavemovie program

The following pseudo-code provides an algorithm for propagating waves into the Earth with the new factorization of the wave equation.

```

Fourier transform input data over time-axis
Loop over frequency {
    Initialize wave at z=0
    Factor wave equation for this w/v
    Recursively divide input data by factor
    Fourier Transform back to time-domain
    Sum into output
}

```

Incorporating this code into the Wavemovie program (Claerbout, 1985) provides a laboratory for testing the new algorithm.

Figure 3.3 compares the results of the new extrapolation procedure with the conventional Crank-Nicolson solution to the 45° equation. The new approach has little dispersion since I use

a rational approximation (the ‘one-sixth trick’) to the Laplacian on the vertical and horizontal axes. In addition, the new factorization retains accuracy up to 90° . The high dip, evanescent energy in the 45° movie, propagates correctly in the new approach.

Figure 3.3: Comparison of the 45° wave equation (left) with the helical factorization of the Helmholtz equation (right). [helmholtz-vs45] [ER,M]

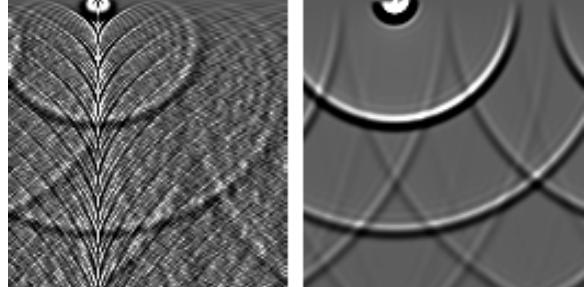


Figure 3.4 compares different value of the ‘one-sixth’ parameter, β . For this application, the optimal value seems to be $\beta = 1/12$.

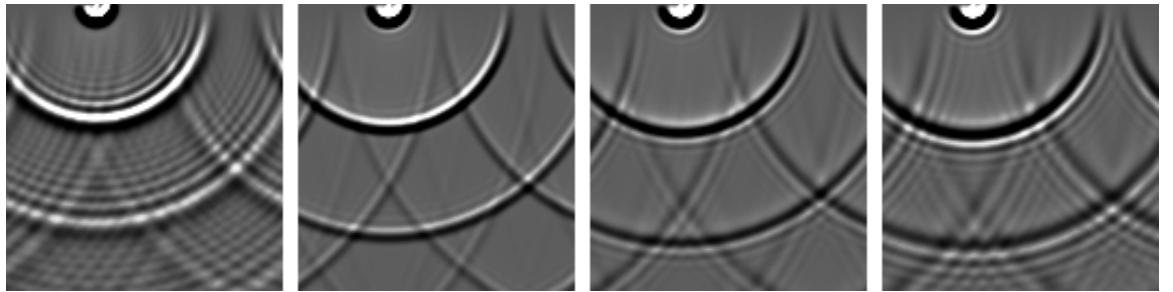


Figure 3.4: Helmholtz equation factorization with different values for the ‘one-sixth’ parameter, β . From left, $\beta = 0, 1/12, 1/8$ and $1/6$. [helmholtz-sixth] [ER,M]

Figure 3.5 compares different finite-difference Laplacian operators. In all cases the finite-difference Laplacian was given by the linear sum of 5-point filters,

$$(1-\gamma) \begin{pmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{pmatrix} + \gamma \begin{pmatrix} 1/2 & & 1/2 \\ & -2 & \\ 1/2 & & 1/2 \end{pmatrix}, \quad (3.21)$$

where γ is an adjustable parameter between 0 and 1. Best results were obtained with $\gamma = 2/3$. The impulse response with $\gamma = 0$ only contains energy on every second grid point, since the

rotated filter only propagates energy diagonally: as in the game of a chess, if a bishop starts on a white square, it always stays on white.

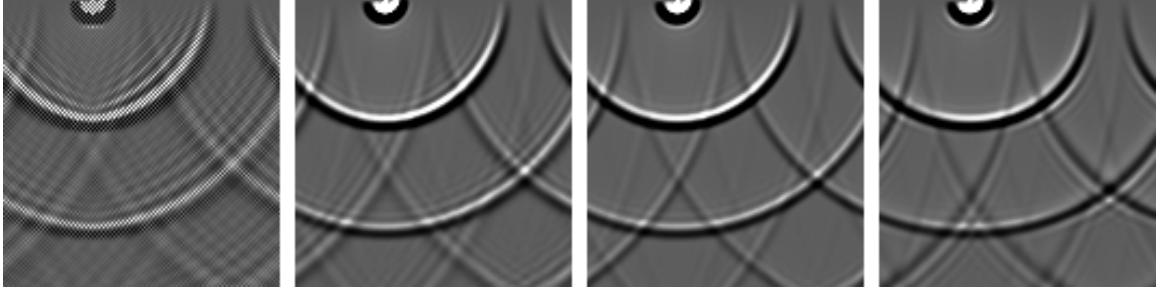


Figure 3.5: Helmholtz equation factorization with different 3×3 finite-difference representations of the Laplacian. From left, $\gamma = 0, 1/2, 2/3$ and 1 . `helmholtz-laplac` [ER,M]

3.3.3 Reducing the filter length

Although mostly zeros, the 9-point filter I factorize in 2-D is actually $2N_x + 3$ points long on a helix, and so its two factors each contain $N_x + 2$ points. For Figures 3.3, 3.4 and 3.5 above, I did not discard any of these filter coefficients: our recursive filters contained the full 128 coefficients. For this extrapolator to compete effectively with other methods (especially in 3-D problems), the number of filter coefficients has to be reduced significantly.

The factor coefficients themselves should be independent of the diameter of the helix. This leads us to expect many zero value coefficients on the helix ‘backside’. Fortunately this is observed in practice, and the filter coefficient amplitude drops rapidly from either end.

Unfortunately, however, the operator that I factor has roots very close to the unit circle in the complex plane. The Laplacian already has a pair of roots at $Z = 0$, and the effect of the extra term on the main diagonal is to destabilize the operator further. The damping rescues us in theory, but in practice I still encounter numerical problems depending on the factorization algorithm.

The results in this thesis have been generated with the Fourier-domain Kolmogorov method (Kolmogorov, 1939; Claerbout, 1998c) to factor the Helmholtz equation. The Kolmogorov

method has two main problems, both due to the proximity of the roots to the unit circle. Firstly, circular boundary conditions require us to pad the cross-correlation function before transforming it to the Fourier domain. With the roots close to the unit circle, extreme amounts of padding are needed: in the 2-D examples above, I need to pad the filters to over 4,000 times their original length. Secondly, the Kolmogorov method simultaneously computes all the filter coefficients. With roots so close to the unit circle, truncating filter coefficients, even in a reasonable manner, often leads to non-minimum-phase filters and divergent results.

The Wilson-Burg algorithm (Wilson, 1969; Sava et al., 1998) does not suffer from the same difficulties. By working in the time domain, the algorithm avoids circular boundary conditions, and the number of filter coefficients can be defined at each iteration, providing a best-fit filter with a given number of coefficients. However, the Wilson-Burg algorithm also encounters problems with roots close to the unit circle. Specifically, numerical problems cause filters to lose their minimum-phase nature, causing the algorithm to diverge. With roots very close to the unit circle, this can happen within the first couple of iterations.

3.4 Conclusions

I have shown that the minimum-phase helical factorization of the Helmholtz equation can be applied recursively to extrapolate waves up to 90° , demonstrating the power of helical boundary conditions to solve wave-equation PDE's. For the constant coefficient case, however, pure Fourier domain methods are still more efficient than the helical factorization, and do not suffer from the technical problems related to the factoring of functions with roots near the unit circle. This means that the methodology in this chapter does not have wide practical value, but rather serves as an conceptual introduction to the following chapters.

Chapter 4

Helical factorization of paraxial extrapolators

Implicit wavefield extrapolation has several potential advantages over other depth migration algorithms. For example, as an (ω, x) domain method, it naturally handles both finite-frequency effects and lateral velocity variations. Also as implicit method, it has the potential for unconditional stability. However, the simple 3-D extension of conventional 2-D wavefield extrapolation by implicit finite-differencing requires the inversion of a 2-D convolution matrix which is computationally difficult. In this chapter, I solve the 45° paraxial wave equation with helical boundary conditions on one of the spatial axes, and study the impulse response of the corresponding migration operators.

4.1 Introduction

Implicit 2-D finite-difference wavefield extrapolation (Claerbout, 1985) has proved itself as a robust, accurate migration method. It naturally and efficiently deals with lateral variations in velocity without the need for asymptotic approximations, such as ray-tracing. The implicit formulation can also be bullet-proofed to ensure unconditional stability (Godfrey et al., 1979).

Unfortunately, despite the widespread popularity of 3-D seismology, implicit 3-D wavefield extrapolation has yet to find wide-spread application. Whereas 2-D extrapolation requires the inversion of a tridiagonal system, the simple extension from 2-D to 3-D leads to a blocked

tridiagonal system, which is prohibitively expensive to solve.

Typically, the matrix inversion problem is avoided by an explicit finite-difference approach (Holberg, 1988). Explicit extrapolation has proved itself effective for practical 3-D problems; since stable explicit filters can be designed (Hale, 1990b), and McClellan filters provide an efficient implementation (Hale, 1990a). However, unlike implicit methods, stability can never be guaranteed if there are lateral variations in velocity (Etgen, 1994). Additionally, accuracy at steep dips requires long explicit filters, which can conflict with rapid lateral velocity variations, and can be expensive to apply.

The problem can also be avoided by splitting the operator to act sequentially along the x and y axes. Unfortunately this leads to azimuthal operator anisotropy, and requires an additional phase correction operator (Graves and Clayton, 1990; Li, 1991). Zhou and McMechan (1997) have presented an alternative to the traditional 45° equation, with form similar to the 15° equation plus an additional correction term. Although splitting their equations results in less azimuthal anisotropy than with the standard 45° equation, the splitting approximation is still needed to solve the equations.

Mixed domain methods such as Fourier finite-differences (Ristow and Ruhl, 1994), which shuttle between the (ω, x) and (ω, k) domains, often depend on an implicit extrapolation step in the (ω, x) domain. Although these implicit operators have only a residual effect, splitting errors may still cause unacceptable reflector mispositioning. The Fourier finite-difference plus interpolation method (Biondi, 2000) reduces operator anisotropy by extrapolating once from a reference velocity field above the medium velocity, once from a reference velocity below the medium velocity, and interpolating between the two. Although very accurate, this algorithm is also significantly more expensive than conventional (ω, x) migration algorithms, since it requires two (ω, x) extrapolations as well as multiple (ω, k) reference fields.

In this chapter, I apply helical boundary conditions to the implicit operators at the heart of (ω, x) migration, showing how this can lead to azimuthally isotropic migration impulse responses without the need for either additional phase-correction operators or multiple passes of the finite-difference operator.

4.2 Implicit extrapolation theory

As described in Chapter 3, wavefield extrapolation algorithms depend on an operator, R , that marches the wavefield q , at depth z , down to depth $z + \Delta z$.

$$q_{z+\Delta z} = R q_z. \quad (4.1)$$

An implicit finite-difference formulation approximates R with a convolution followed by an inverse convolution. For example, a rational approximation to equation (3.17) that corresponds to the Crank-Nicolson scheme for the 45° one-way wave equation (Fomel and Claerbout, 1997), is given by

$$R_{\text{implicit}}(\mathbf{k}_x) = e^{i\omega s} \frac{1 - 4\omega^2 s^2 + i\omega s |\mathbf{k}_x|^2}{1 - 4\omega^2 s^2 - i\omega s |\mathbf{k}_x|^2}, \quad (4.2)$$

where $s = 1/v$ and $\Delta z = 1$.

This operator can be implemented numerically (without Fourier transforms) by replacing $|\mathbf{k}_x|^2$ with a finite-difference equivalent whose amplitude spectrum, D , in the constant velocity case will also be a simple (non-negative) function of \mathbf{k}_x . Irrespective of the choice of $D(\mathbf{k}_x)$, this operator can be written as a pure phase-shift operator,

$$R_{\text{implicit}}(\mathbf{k}_x) = e^{i\phi(\mathbf{k}_x)}, \quad (4.3)$$

where $\phi(\mathbf{k}_x) = \omega s + 2 \arctan \frac{\omega s D(\mathbf{k}_x)}{1 - 4\omega^2 s^2}$. Consequently, in the constant velocity case, this formulation is unconditionally stable for all values of ωs .

An explicit approach approximates R directly with a single convolutional filter. For example, a three-term expansion of equation (3.16) yields

$$R_{\text{explicit}}(k) = e^{i\omega s} (1 + \gamma_1 k^2 + \gamma_2 k^4) \quad (4.4)$$

where complex coefficients γ_1 and γ_2 can be calculated using a Taylor series, for example, and $k^2 = |\mathbf{k}_x|^2$. Although in practice stable explicit operators can be constructed for constant

velocity media (Hale, 1990b), they can never represent a pure phase-shift. Their stability is conditional, and cannot be guaranteed for media with lateral velocity variations.

Also in order to preserve high angular accuracy for steep dips, explicit filters need to be longer than their implicit counterparts. The advantage of finite-difference methods over Fourier methods is that the effect of the finite-difference convolution filters is localized, leading to accurate results for rapidly varying velocity models. This is less of an advantage for long filters.

4.2.1 Matrix representation of implicit operators

The diffraction term of the 45° equation (Claerbout, 1985) can be rewritten as the following matrix equation, by inserting the rational part of the implicit extrapolator (4.2) into equation (3.16):

$$(\mathbf{I} + \alpha \mathbf{D}) \mathbf{q}_{z+\Delta z} = (\mathbf{I} + \alpha' \mathbf{D}) \mathbf{q}_z \quad (4.5)$$

$$\mathbf{A} \mathbf{q}_{z+\Delta z} = \mathbf{A}^* \mathbf{q}_z \quad (4.6)$$

where the complex coefficient, α can be calculated, and \mathbf{D} is a finite-difference representation of the Laplacian, ∇^2 .

The right-hand-side of equation (4.6) is known. The challenge is to find the vector $\mathbf{q}_{z+\Delta z}$ by inverting the matrix, \mathbf{A} . Given the wavefield on the surface, this equation provides a way to downward-continue in depth.

The matrices in equation (4.6) represent convolution with a scaled finite-difference Laplacian, with its main diagonal stabilized. In the two-dimensional problem, the ∇^2 operator acts only in the x -direction, and can be represented by the three-point convolutional filter, $d = (1, -2, 1)$. The matrix, \mathbf{A} , therefore, has a tridiagonal structure, which can be inverted efficiently with a recursive solver.

In three-dimensional wavefield extrapolation, the ∇^2 operator acts in both the x and y -directions. \mathbf{A} therefore represents a 2-D convolution, and d can be represented by the a simple

5-point filter,

$$d = \begin{bmatrix} & & 1 \\ & 1 & -4 & 1 \\ & & 1 \end{bmatrix} \quad (4.7)$$

or a more isotropic 9-point filter (Iserles, 1996),

$$d = \begin{bmatrix} 1/6 & 2/3 & 1/6 \\ 2/3 & -10/3 & 2/3 \\ 1/6 & 2/3 & 1/6 \end{bmatrix} \quad (4.8)$$

The vectors \mathbf{q}_z and $\mathbf{q}_{z+\Delta z}$ contain the wavefield at every point in the (x,y) -plane. Therefore, the convolution matrices that operate on them are square with dimensions $N_x N_y \times N_x N_y$. As an illustration, for a 4×2 spatial plane, the structure of matrix \mathbf{D} with the five-point approximation and transient boundary conditions, will be the blocked-tridiagonal matrix

$$\mathbf{D} = \left[\begin{array}{cccc|cccc} -4 & 1 & . & . & 1 & . & . & . \\ 1 & -4 & 1 & . & . & 1 & . & . \\ . & 1 & -4 & 1 & . & . & 1 & . \\ . & . & 1 & -4 & . & . & . & 1 \\ \hline 1 & . & . & . & -4 & 1 & . & . \\ . & 1 & . & . & 1 & -4 & 1 & . \\ . & . & 1 & . & . & 1 & -4 & 1 \\ . & . & . & 1 & . & . & 1 & -4 \end{array} \right] \quad (4.9)$$

This blocked system cannot be easily inverted, even for the case of constant velocity, since the missing coefficients on the second diagonals break the Toeplitz structure.

4.2.2 Helical boundary conditions

As discussed previously, the helix transform (Claerbout, 1998b) provides boundary conditions that map multi-dimensional convolution into one-dimension. In this case, the 2-D convolution operator, $(\mathbf{I} + \alpha \mathbf{D})$ can be recast as an equivalent 1-D filter. The 5-point 2-D filter becomes the

sparse 1-D filter of length $2N_x + 1$ that has the form,

$$a_1 = (\alpha, 0, \dots, 0, \alpha, 1 - 4\alpha, \alpha, 0, \dots, 0, \alpha).$$

The structure of the finite-difference Laplacian operator, \mathbf{D} , is simplified when compared to equation (4.9).

$$\mathbf{D} = \begin{bmatrix} -4 & 1 & . & . & 1 & . & . & . \\ 1 & -4 & 1 & . & . & 1 & . & . \\ . & 1 & -4 & 1 & . & . & 1 & . \\ . & . & 1 & -4 & 1 & . & . & 1 \\ 1 & . & . & 1 & -4 & 1 & . & . \\ . & 1 & . & . & 1 & -4 & 1 & . \\ . & . & 1 & . & . & 1 & -4 & 1 \\ . & . & . & 1 & . & . & 1 & -4 \end{bmatrix} \quad (4.10)$$

The 1-D filter can be factored into a causal and anti-causal parts, and the matrix inverse can be computed by recursive polynomial division (1-D deconvolution).

4.2.3 Cross-spectral factorization

Kolmogorov spectral factorization (reviewed in Chapter 2) provides an algorithm for finding a minimum-phase wavelet with a desired spectrum, or auto-correlation function. Since positive-definite Hermitian matrices with Toeplitz structure represent convolutions with autocorrelation functions, Kolmogorov provides a way to efficiently decompose them into the product of lower (causal) and upper (anti-causal) parts. Once this LU factorization has been achieved, the two parts can be inverted rapidly by recursion (polynomial division).

As with the Helmholtz equation, the complex scale-factor, α , means \mathbf{A} is symmetric, but

not Hermitian, so the standard Kolmogorov factorization will fail. As discussed in Chapter 3, however, the method can be extended to factor any cross-spectrum into a pair of minimum phase wavelets and a delay (Claerbout, 1998c). The algorithm follows the standard Kolmogorov factorization; however, negative lags are kept separately rather than being discarded.

The Kolmogorov factorization is not exact because the filters are factored in the frequency domain, assuming circular boundary conditions; while the polynomial division is performed in the time domain with transient boundary conditions. As a result the filters must be padded in the time-domain before spectral factorization. Padding does not significantly effect the overall cost of the migration, as the computational expense lies in the polynomial division, not in the factorization.

Alternative methods for cross-spectral factorization may avoid the circular boundary condition problem. For example, the Wilson-Burg algorithm (Wilson, 1969; Sava et al., 1998), based on Newton's recursive linearization, can efficiently factor polynomials, and is especially suited to the helical coordinate system.

4.2.4 Polynomial division

Kolmogorov cross-spectral factorization, therefore, provides a tool to factor the helical 1-D filter of length $2N_x + 1$ into minimum-phase causal (and maximum-phase anti-causal) filters of length $N_x + 1$. Deconvolution with minimum-phase filters is unconditionally stable. However, inverse-filtering with the entire filters would be an expensive operation. Fortunately, filter coefficients drop away rapidly from either end. In practice, small-valued coefficients can be safely discarded, without violating the minimum-phase requirement; so for a given grid-size, the cost of the matrix inversion scales linearly with the size of the grid.

The unitary form of equation (4.2) can be maintained by factoring the right-hand-side matrix, \mathbf{A} in equation (4.6), with Kolmogorov before applying it to \mathbf{q}_z .

$$\mathbf{L}\mathbf{L}^T \mathbf{q}_{z+\Delta z} = (\mathbf{L}\mathbf{L}^T)^* \mathbf{q}_z \quad (4.11)$$

$$\mathbf{q}_{z+\Delta z} = \frac{\mathbf{L}\mathbf{L}^T}{(\mathbf{L}\mathbf{L}^T)^*} \mathbf{q}_z \quad (4.12)$$

Chapter 5 and Appendix A extend the concept of recursive inverse filtering to handle non-stationarity. There are pitfalls associated with this process, however; consequently, in this Chapter I limit the examples to the constant velocity case.

4.3 Synthetic examples

Figure 4.1 shows a slice through the broad-band impulse response of the 45° equation. As with the 2-D 45° equation, evanescent energy at high dip appears as noise, and takes the form of a cardioid. This is never a problem on field data, and has been removed from the depth-slices shown in Figures 4.2 and 4.3.

Figure 4.2 compares the impulse responses of the 45° equation obtained with the x - y splitting approximation [panel (a)] and the helical factorization methodology described in this chapter [panel (b)]. Implicit migration with the full Laplacian, instead of a splitting approximation, produces an impulse response that is azimuthally isotropic without the need for any phase corrections.

Figure 4.3 shows the effects of the different boundary conditions on the two spatial axes. The fast spatial axis (top and bottom of Figure) have helical boundary conditions, and show wrap-around. The slow spatial axis (left and right of Figure) has a zero-value boundary condition, and hence is reflective.

For the examples in this chapter, we set the ‘one-sixth’ parameter (Claerbout, 1985), $\beta_{1/6} = 0.125$, and used the isotropic nine-point Laplacian from equation (4.8) that corresponds to $\gamma = 2/3$ in equation (3.21).

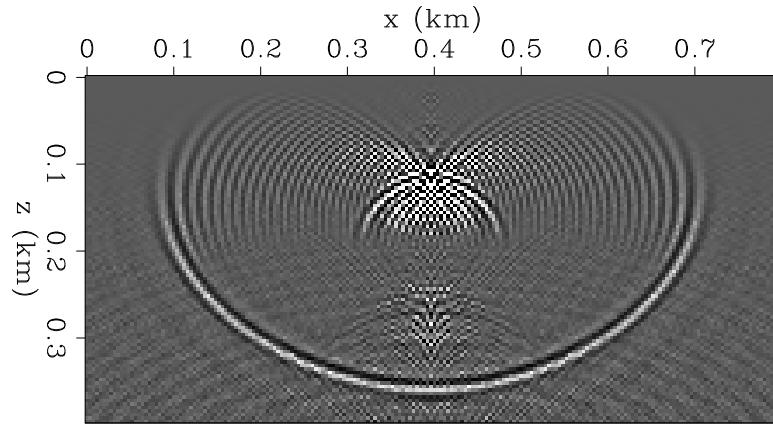


Figure 4.1: Vertical slice through broad-band impulse response of 45° wave equation, showing the distinctive cardioid. helimig-3Dcardioid [CR]

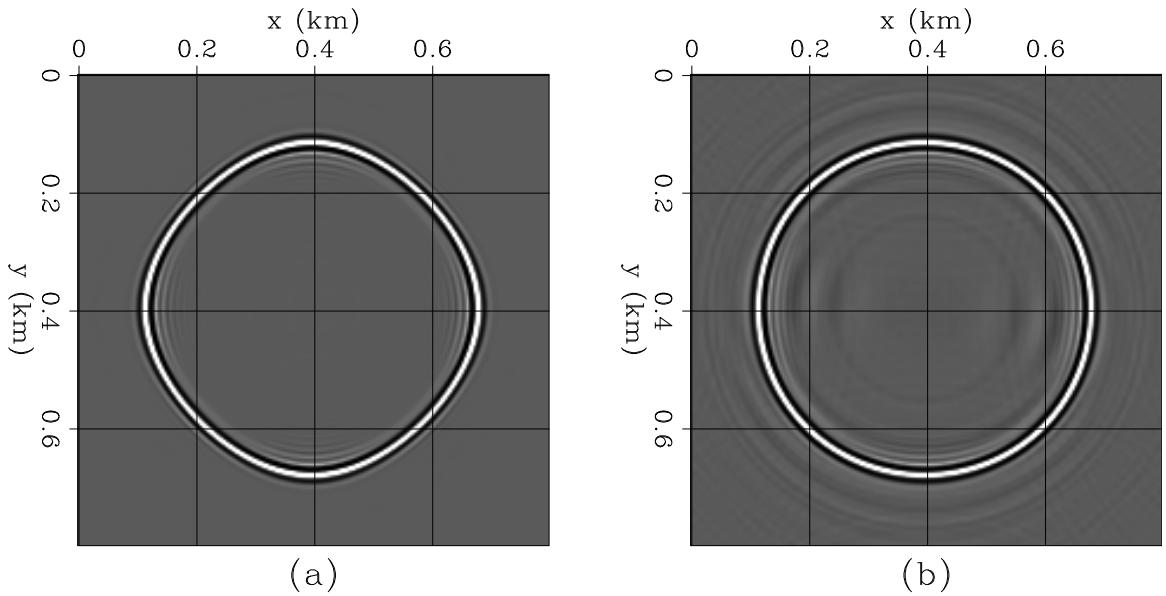
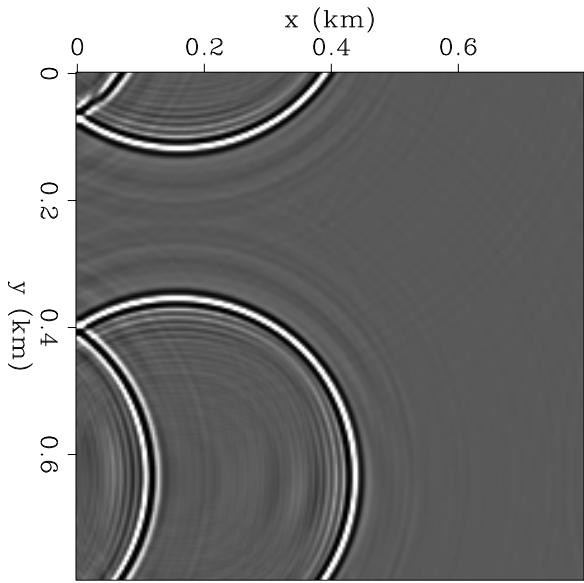


Figure 4.2: Depth-slice of centered impulse response corresponding to a dip of 50° . Panel (a) shows the result of employing an x - y splitting approximation, and panel (b) shows the result of the helical factorization. Note the azimuthally isotropic nature of panel (b). Evanescent energy has been removed by dip-filtering prior to migration. helimig-3Dtimeslice [CR,M]

Figure 4.3: Depth-slice of offset impulse response corresponding to a dip of 45° . Note the helical boundary conditions on the fast spatial axis.
 [helimig-3Dboundary] [CR]



4.4 Conclusions

Implicit extrapolations have several advantages over explicit methods: they have the potential for unconditional stability, and shorter filters are required to achieve higher accuracy. Through the helical coordinate system, I have recast the 2-D deconvolution at the heart of implicit 3-D wavefield extrapolation, into a one-dimensional problem that can be solved efficiently by recursion. I have demonstrated the algorithm by migrating simple constant velocity synthetic examples with the conventional 45° paraxial wave equation. The extension to lateral velocity variations is discussed in the next chapter.

Chapter 5

$V(x, y, z)$ and non-stationary inverse convolution

Implicit extrapolation with the helical factorization discussed in the previous chapter can be easily extended to smoothly-variable velocities. Stationary filtering and inverse filtering can be replaced by their non-stationary counterparts, and the spectral factorization becomes a problem of LU decomposition. I develop a solution the cost of which remains proportional to the number of grid nodes [$O(N)$]. This solution is also exactly equivalent to the constant velocity factorization in the smooth-velocity limit. Unfortunately, however, I will show that the Godfrey/Muir (1979) bulletproofing cannot be simply applied to the helical factorization, and so the method is susceptible to stability problems. Numerical examples confirm the accuracy of the method in models with smoothly-varying velocity; however, I observe instability in models with strong lateral velocity variations.

5.1 Introduction

In the previous chapter, I described the process of applying helical boundary conditions to facilitate the factorization of implicit extrapolators. However, I only covered the case where the velocity was constant within each depth layer, i.e. constant velocity and $v(z)$ earth models.

The advantage of working in the space domain, as opposed to the spatial-frequency domain, is that method can be adapted to handle operators changing laterally. Indeed, the strength of conventional implicit finite-difference methods comes in areas with strong lateral velocity variations, where the small filters can accurately model the rapid velocity changes, and the implicit formulation can guarantee unconditional stability (Godfrey et al., 1979).

In this chapter, I describe how recursive filtering can be extended to handle non-stationarity. This allows implicit depth migration with the helix factorization to be applied in areas with lateral velocity variations. Unfortunately, however, I am unable to formulate the helical factorization in such a way that maintains the unconditional stability of the conventional implicit schemes. Therefore, the stability of helical extrapolators in laterally variable media cannot be guaranteed.

5.2 Smooth lateral variations in velocity

The first section of this chapter describes how the method can be successfully applied in models where there are smooth lateral velocity variations.

5.2.1 Non-stationary recursive filtering

If the velocity varies smoothly in space, then we can extend the stationary theory to cover spatially-variable filtering. Rather than filtering with the stationary forward and inverse convolution equations,

$$y_k = x_k + \sum_i a_i x_{k-i}, \quad (5.1)$$

$$\text{and } x_k = y_k - \sum_i a_i x_{k-i}, \quad (5.2)$$

we can extend the concept of a filter to that of a filter-bank with one filter for every location in the input/output space (Claerbout, 1998a; Margrave, 1998). Now equations (5.1) and (5.2)

become

$$y_k = x_k + \sum_i a_{i,k-i} x_{k-i}, \quad (5.3)$$

$$\text{and } x_k = y_k - \sum_i a_{i,k-i} x_{k-i}. \quad (5.4)$$

Non-stationary convolution and recursive inverse convolution are indeed true inverse processes, but like stationary polynomial condition non-stationary inverse filtering has potential stability problems. Appendix A discusses this in more detail; it also demonstrates that, in general, the stability of recursive non-stationary inverse filtering is not guaranteed even if the individual filters that make up the filter-bank are all minimum phase.

5.2.2 Non-stationary factorization

In order to extrapolate waves, we must invert a non-stationary filtering operator. But before we can perform recursive inverse filtering with equation (5.4), we must factor the filter into causal and anticausal components.

Figure 5.1 illustrates a process for factorizing a non-stationary filtering matrix. This approach is accurate if the filters vary smoothly in space, and it is the approach that I follow for the examples in this chapter.

The first step is to build the extrapolation matrix. Then we can extract the impulse response associated with each spatial location, and factor it into causal and anticausal components. By inserting the causal components into the column of a lower triangular matrix, and the anticausal components into a row of an upper triangular matrix, we can begin to build the two invertible matrices. The process has to be repeated for each spatial location.

In practice, we can save time by factoring the filters in advance, and storing them in a table. For every value of ω/v , we precompute the factors of the 1-D helical filter, $a(\omega/v)$, and store filter coefficients in a look-up table. We then extrapolate the wavefield by non-stationary convolution [equation (5.3)], followed by non-stationary polynomial division with equation (5.4).

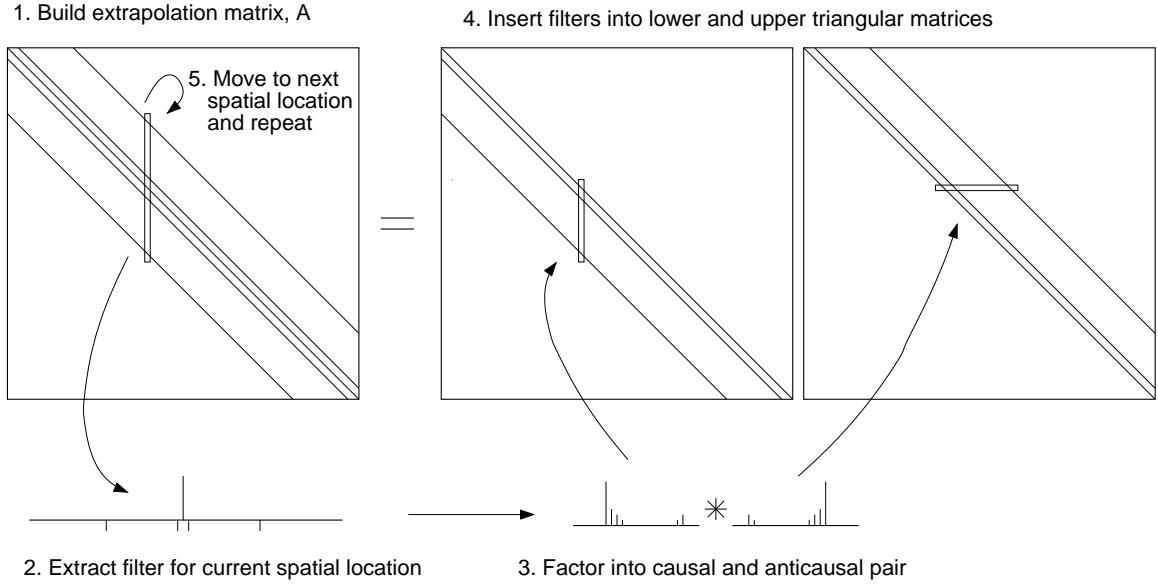


Figure 5.1: Schematic illustrating non-stationary factorization process. Step 1: Build the extrapolation matrix. Step 2: Extract the filter corresponding to the current spatial location. Step 3: Factor this filter. Step 4: Insert factors into lower and upper triangular matrices. Step 5: Move to next spatial location and repeat from step 2. **nstat-nsfactor** [NR]

Since we interpolate filters, not downward continued wavefields as in phase-shift plus interpolation migration (Gazdag and Sguazzero, 1984), the number of reference velocities used has minimal effect on the overall cost of the migration. Indeed, the cost of this process is proportional to the number of grid nodes times the number of filter coefficients; so for a fixed number of filter coefficients, the cost is linear in the number of grid nodes [$O(N)$].

5.2.3 Numerical example

The SEG-EAGE salt-dome synthetic dataset provides a good test-bed for 3-D migration algorithms since it contains both complex velocity and steeply-dipping reflectors (Vaillant, 1999). The images in this chapter are produced by migrating a zero-offset subvolume of the full 3-D dataset.

Figure 5.2 shows a slightly smoothed¹ version of the true SEG/EAGE velocity model, and Figure 5.3 shows the results of migrating the dataset with this model and the helical factorization of the conventional 45° equation (Claerbout, 1985). Both the top and bottom of the salt are well-imaged throughout most of the model. Unsurprisingly, however, problems exist around $x = 7000$ m, in the area where steep canyons disrupt the top salt. The growth in amplitude of the migrated image around this area is related to a mild instability in the algorithm that I discuss in the next section.

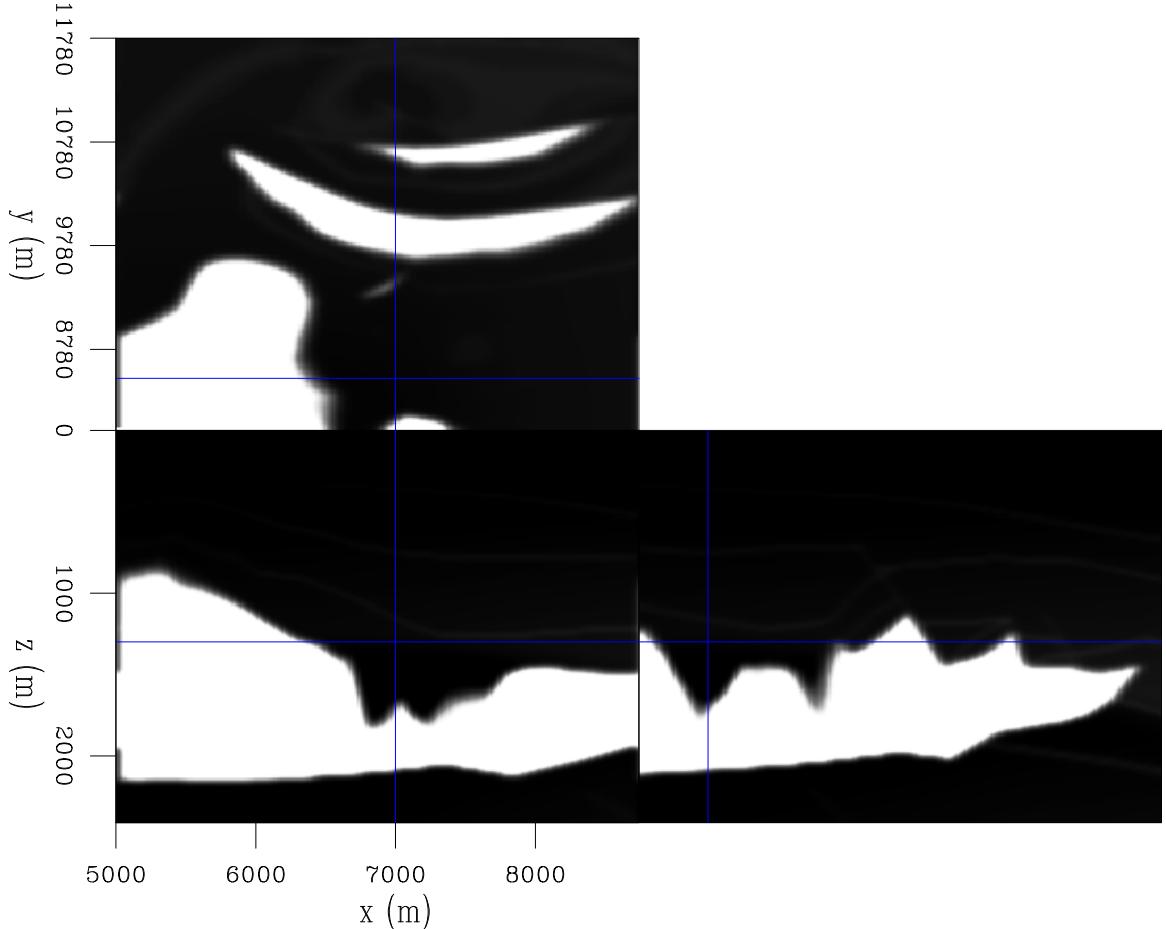


Figure 5.2: Velocity model with strong lateral velocity variations. Velocity varies between 1.5 km/s in the near surface sediments to 4.5 km/s in the salt. nstat-segvelunstabb [ER,M]

¹I applied triangle smoothing along the three axes. The half widths of the triangles were 60 m in the x and y directions, and 30 m in the z direction. I repeated this twice.

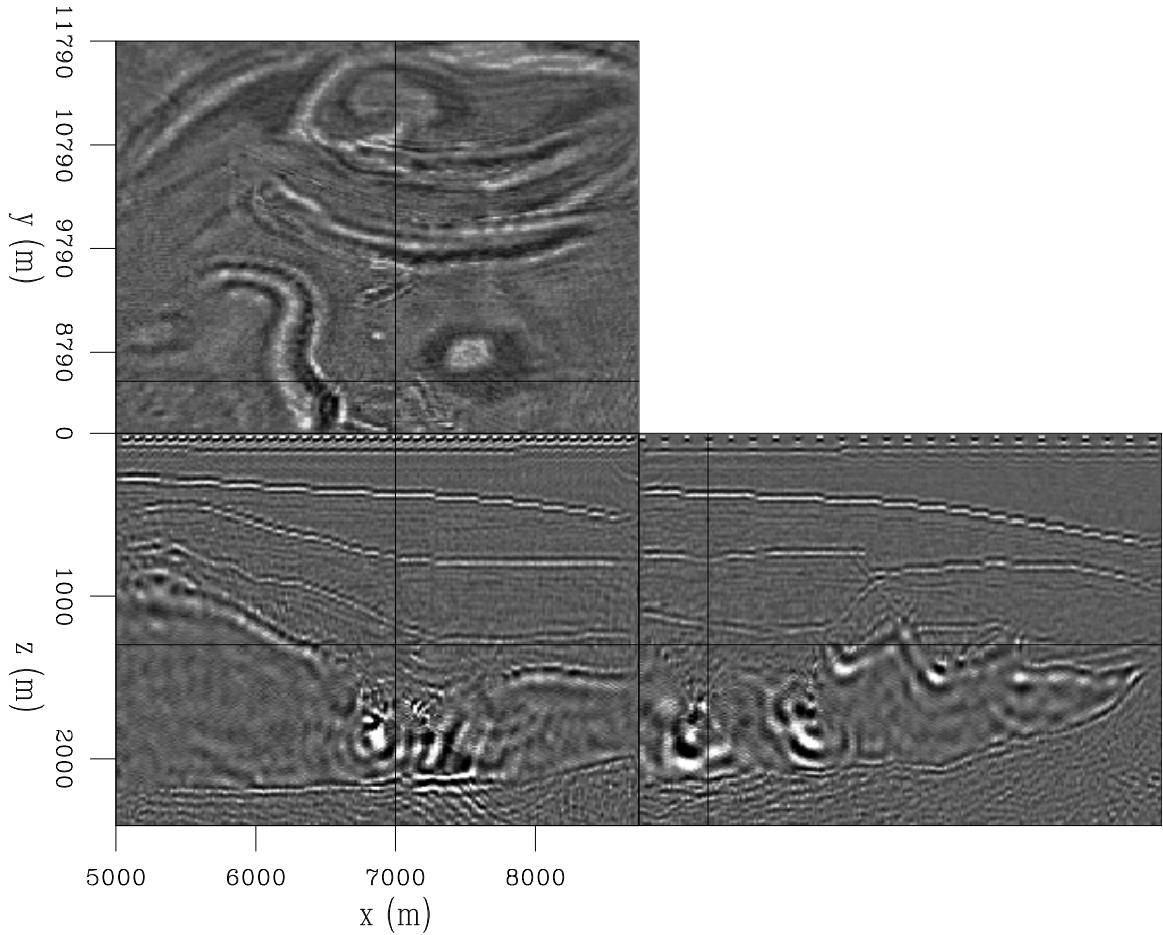


Figure 5.3: Result of migrating a subvolume from the zero-offset 3-D SEG/EAGE salt model with the non-stationary helical factorization. **nstat-segmig5** [CR,M]

5.3 Strong lateral variations in velocity

In order to extrapolate in areas with strong lateral velocity variations, care must be taken with the stability of the extrapolation algorithm. In this section, I analyze the stability of the helical factorization, making comparisons with conventional Crank-Nicolson extrapolation.

5.3.1 Stability analysis

The starting point for wavefield extrapolation algorithms is an equation that governs the evolution of the wavefield in depth,

$$\frac{dq}{dz} = -Rq. \quad (5.5)$$

The Crank-Nicolson finite-difference scheme makes a numerical approximation of the depth derivative, by equating the two sides of equation (5.5) at a point midway between depth steps n and $n + 1$:

$$\frac{1}{\Delta z} (\mathbf{q}_{n+1} - \mathbf{q}_n) = -\frac{1}{2} \mathbf{R} (\mathbf{q}_{n+1} + \mathbf{q}_n). \quad (5.6)$$

Rearranging terms gives the implicit system,

$$\left(\mathbf{I} + \frac{\Delta z}{2} \mathbf{R} \right) \mathbf{q}_{n+1} = \left(\mathbf{I} - \frac{\Delta z}{2} \mathbf{R} \right) \mathbf{q}_n. \quad (5.7)$$

An extrapolator in equation (5.5) will be stable if $q' q$ either decreases or remains constant with depth. Therefore for stable extrapolation,

$$\begin{aligned} \frac{d}{dz}(q' q) &\leq 0 \\ q' \frac{dq}{dz} + \frac{dq'}{dz} q &\leq 0 \\ q'(R + R')q &\geq 0. \end{aligned} \quad (5.8)$$

This implies that if $R + R'$ is symmetric non-negative definite, then extrapolation with equation (5.5) will be unconditionally stable. With a similar proof, Godfrey et al. (1979) showed the same condition applies to the extrapolation matrix \mathbf{R} under Crank-Nicolson extrapolation

with equation (5.7). If $\mathbf{R} + \mathbf{R}'$ is symmetric non-negative definite, then stable extrapolation is guaranteed.

While the helical factorization begins with equation (5.7), we really solve the implicit system,

$$\mathbf{L}\mathbf{L}^T \mathbf{q}_{n+1} = (\mathbf{L}\mathbf{L}^T)^* \mathbf{q}_n, \quad (5.9)$$

$$\mathbf{A} \mathbf{q}_{n+1} = \mathbf{A}^* \mathbf{q}_n, \quad (5.10)$$

where $\mathbf{A} = \mathbf{L}\mathbf{L}^T \approx \mathbf{I} + \frac{\Delta z}{2} \mathbf{R}$. To relate this to equation (5.7), we can premultiply that equation by an invertible matrix \mathbf{B} to provide a slightly more general Crank-Nicolson extrapolation system with the same stability requirements:

$$\mathbf{B} \left(\mathbf{I} + \frac{\Delta z}{2} \mathbf{R} \right) \mathbf{q}_{n+1} = \mathbf{B} \left(\mathbf{I} - \frac{\Delta z}{2} \mathbf{R} \right) \mathbf{q}_n. \quad (5.11)$$

Equating equations (5.10) and (5.11) produces a formula for \mathbf{R} in terms of the helical factorization, \mathbf{A} :

$$2\mathbf{B} = \mathbf{A} + \mathbf{A}^* \quad (5.12)$$

$$\Delta z \mathbf{B} \mathbf{R} = \mathbf{A} - \mathbf{A}^* \quad (5.13)$$

$$\mathbf{R} = \frac{2}{\Delta z} (\mathbf{A} + \mathbf{A}^*)^{-1} (\mathbf{A} - \mathbf{A}^*). \quad (5.14)$$

Bulletproof stability requires $\mathbf{R} + \mathbf{R}'$ to be symmetric non-negative definite. For the helical factorization this matrix is given by,

$$\frac{\Delta z}{2} (\mathbf{R} + \mathbf{R}') = (\mathbf{A} + \mathbf{A}^*)^{-1} (\mathbf{A} - \mathbf{A}^*) - (\mathbf{A} - \mathbf{A}^*) (\mathbf{A} + \mathbf{A}^*)^{-1}. \quad (5.15)$$

In the constant velocity case, the matrix \mathbf{A} represents a stationary filtering operation. Therefore the composite matrices, $\mathbf{A} + \mathbf{A}^*$ and $\mathbf{A} - \mathbf{A}^*$ commute with each other. Under this scenario, the matrix $\mathbf{R} + \mathbf{R}'$ becomes the zero matrix, which clearly satisfies the non-negative definite criterion required for stability. Constant velocity extrapolation with equation (5.10) is

therefore unconditionally stable.

Unfortunately, however, if the velocity varies laterally, \mathbf{A} and \mathbf{A}^* no longer commute with each other, and so the composite matrices $\mathbf{A} + \mathbf{A}^*$ and $\mathbf{A} - \mathbf{A}^*$ do not commute either. Consequently stable extrapolation cannot be guaranteed. Furthermore, there are no obvious steps we can take to ensure that equation (5.15) remains non-negative definite in areas of strong lateral velocity variations. In practice, equation (5.10) does indeed encounter stability problems in some areas. Section 5.3.2 illustrates this problem with some examples.

Essentially the problem revolves around the fact that I factor $\mathbf{I} + \Delta z/2 \mathbf{R}$ and $\mathbf{I} - \Delta z/2 \mathbf{R}$, rather than \mathbf{R} itself. We can ensure our factorization is symmetric non-negative definite, but not the extrapolator itself.

5.3.2 Numerical examples

Although the instability in Figure 5.3 is not catastrophic, the instability can cause amplitudes to explode and contaminate the entire model. For example, migration with the raw (unsmoothed) SEG/EAGE velocity model does become unstable, producing no useful output. Figure 5.4 shows another example: the results of imaging the zero-offset wavefield from the Marmousi dataset (Bourgeois et al., 1991) with a velocity model that contains a sharp discontinuity. This example is similar to the model that Etgen (1994) showed caused instability for explicit operators.

5.4 Conclusions

The helical factorization scheme outlined in Chapter 4 can be extended to deal with non-stationarity by replacing stationary convolution and inverse convolution with their non-stationary counterparts. I factorize non-stationary filters by making an assumption of local stationarity. Although this is only strictly valid for smoothly-varying media, it has the advantage that for a fixed number of filter coefficients the cost remains linear in the number of grid nodes.

Since I factorize the entire implicit system, it is difficult to ensure that the extrapolator

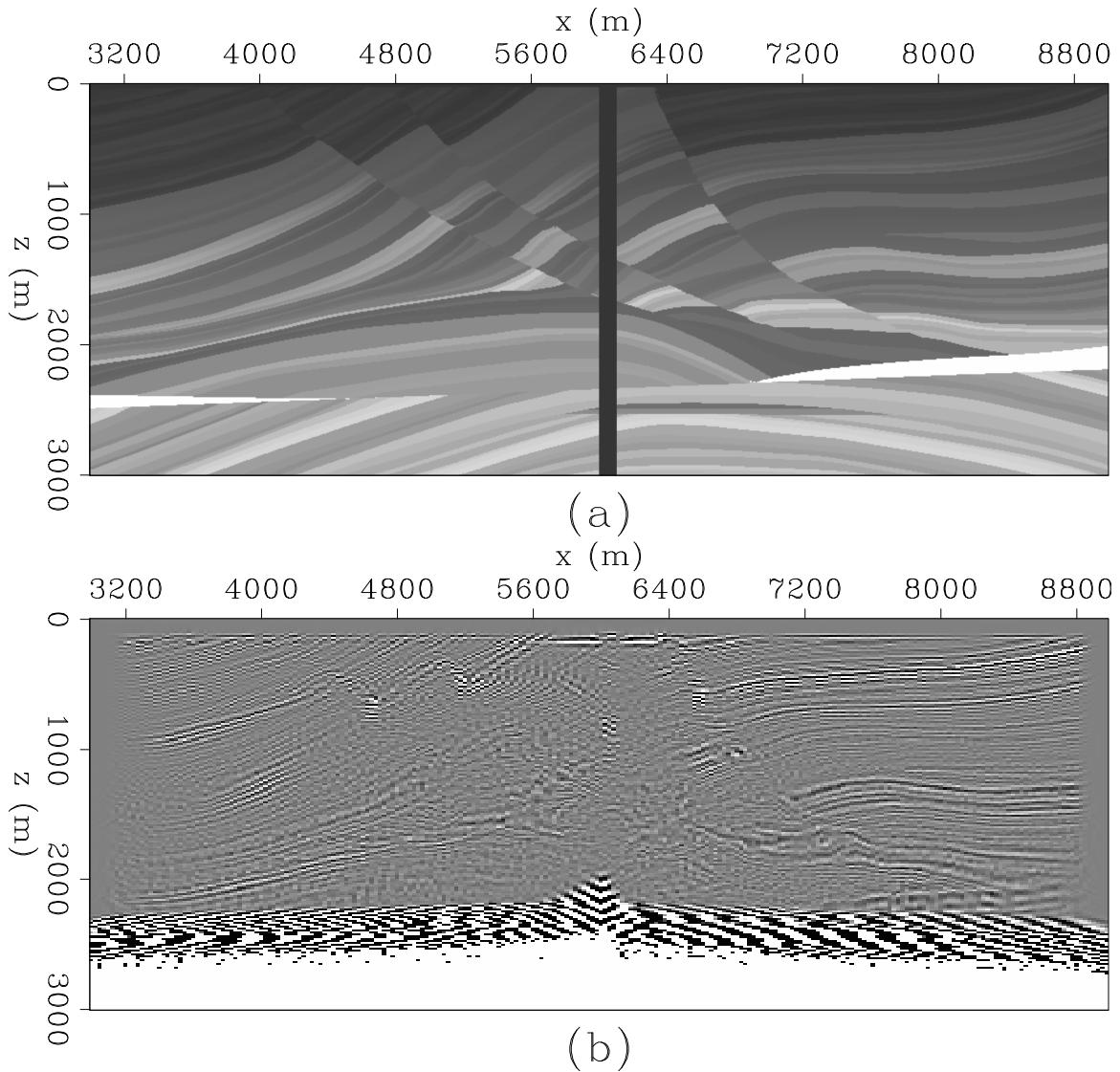


Figure 5.4: Instability in 2-D helical factorization of implicit 2-D system. [nstat-marmstab]
[ER]

embedded inside has the non-negative definite property necessary for unconditional stability. Unfortunately, this means the helical factorization is not robust to the presence of strong lateral velocity variations in the model, and the solution may diverge.

Chapter 6

Migration and shot illumination

In this chapter, I review modeling and migration by recursive wavefield extrapolation, contrasting the exploding-reflector, shot-profile and shot-geophone methods. I then show that for the special case of sparse-shot geometries with fully sampled receiver axes, the main difficulty with obtaining accurate migration amplitudes in vertical incidence reflections is the variability in shot illumination. I show that shot illumination problems can be compensated for by a model-space weighting function applied after migration. The weighting function is based on integrating the total downward-going energy at each point in the subsurface, which can be calculated at very little cost during the migration process.

6.1 Introduction to shot-profile migration

The two most widely-used downward-continuation algorithms for imaging prestack seismic data are shot-profile migration and shot-geophone migration.

Shot-profile migration is based on an imaging condition proposed by Claerbout (1971) as part of his theory of reflector mapping. He proposed obtaining reflector maps by crosscorrelating upgoing and downgoing wavefields in the earth. The upgoing and downgoing wavefields can be obtained by downward continuing into the earth the recorded wavefield and source function respectively.

An alternative to shot-profile migration is shot-geophone migration, so-called because the entire dataset, parameterized by shot and geophone (or equivalently by midpoint and offset) is downward continued one frequency at a time. This is achieved with a “survey sinking” operator derived from the double-square-root (DSR) equation (Claerbout, 1985).

Both algorithms may also be adapted to image offset-dependent (or angle-dependent) reflectivity. Appendix B addresses this subject in more detail.

For a given dataset any differences between migration images produced by the two methods depend on small implementation details. Therefore the choice of whether shot-geophone or shot-profile migration is more appropriate depends on the cost of the method, which in turn depends principally on the acquisition geometry of the dataset.

The cost of shot-geophone migration in the midpoint and offset domain is proportional to the number of midpoints times the number of offsets: in 2-D this is $N_x \times N_h$, and in 3-D this is $N_x \times N_{h_x} \times N_y \times N_{h_y}$. The cost of shot-profile migration proportional to the number of shots times the number of midpoints: $N_x \times N_y \times N_s$.

Therefore, for geometries with large numbers of shots, such as typical 3-D marine streamer geometries, shot-geophone migration is a more attractive choice, especially when the dimensionality of the problem can be reduced by a common-azimuth approximation (Biondi and Palacharla, 1996). However, for wide-azimuth geometries where the number of shots (or reciprocal receivers) is small compared to the number of offsets, shot-profile migration may be preferable. 3-D data collected with technologies such as vertical cables, borehole seismometers, and ocean bottom seismometers, may be efficiently migrated with shot-profile methods.

Like many other industrial strength geophysical processes, shot-profile migration is the adjoint of a linear forward modeling operator that mimics wave propagation in the earth. By breaking down the shot-profile modeling and migration operators into their constituent components, I am able to show that for some sparse-shot geometries, the least-squares inverse of the forward modeling operator is a chain of conventional migration operator followed by a model-space weighting function that can be calculated cheaply during the migration process.

6.2 Shot-profile migration and modeling

The application of multi-dimensional spectral factorization discussed in previous chapters of this thesis is migration by wavefield extrapolation in the (ω, \mathbf{x}) domain. These various methods discussed, however, are only a subset of a larger group of recursive downward-continuation imaging algorithms that work in (ω, \mathbf{x}) , (ω, \mathbf{k}) and mixed $(\omega, \mathbf{x}, \mathbf{k})$ domains. Many aspects of downward-continuation migration are independent of the choice of extrapolation algorithm itself.

6.2.1 Exploding-reflector migration by downward continuation

Recursive downward continuation migration algorithms start with an operator, R , that marches the wavefield down into the earth,

$$q(x, y, z + \Delta z, \omega) = R(z, \omega) q(x, y, z, \omega), \quad (6.1)$$

with initial conditions determined by the reflection data recorded on the surface,

$$q(x, y, z = 0, \omega) = d(x, y, \omega). \quad (6.2)$$

Taking $\Delta z = 1$, these equations can be rewritten in matrix form as

$$\mathbf{q}_{z+1, \omega} = \mathbf{R}_{z, \omega} \mathbf{q}_{z, \omega} \quad (6.3)$$

$$\mathbf{q}_{1, \omega} = \mathbf{d}_\omega, \quad (6.4)$$

where \mathbf{d}_ω is a single frequency component of data recorded at the surface, $\mathbf{q}_{z, \omega}$ is a vector containing a single frequency component of the wavefield at depth-step, z , and $\mathbf{R}_{z, \omega}$ is the corresponding extrapolation operator.

If the downward-continuation operator, $\mathbf{R}_{z, \omega}$, contains purely a phase-shift, then its adjoint, $\mathbf{R}'_{z, \omega}$ will fully describe the inverse process of upward-continuation. However, for the amplitudes to be treated accurately, $\mathbf{R}_{z, \omega}$ must respect the physics of wave propagation. Stolt

and Benson (1986) show that $v(z)$ extrapolators based on WKBJ Green's functions contain an amplitude term as well as a phase term, and for $v(x, y, z)$ earth models this effect is even more pronounced. So while kinematically-correct extrapolators are pseudo-unitary, true-amplitude depth extrapolators are not. In this introductory section, I follow conventional seismic processing methodology, and treat the depth extrapolator as a unitary operator; however, in section 6.2.4, I discuss how to model amplitudes correctly.

The recursion in equations (6.3) and (6.4) can be rewritten as:

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -\mathbf{R}_{1,\omega} & 1 & 0 & \dots & 0 \\ 0 & -\mathbf{R}_{2,\omega} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} \mathbf{q}_{1,\omega} \\ \mathbf{q}_{2,\omega} \\ \mathbf{q}_{3,\omega} \\ \vdots \\ \mathbf{q}_{N_z,\omega} \end{pmatrix} = \begin{pmatrix} \mathbf{d}_\omega \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad (6.5)$$

or more simply,

$$\mathbf{D}_\omega \mathbf{q}_\omega = \mathbf{Z} \mathbf{d}_\omega, \quad (6.6)$$

where \mathbf{q}_ω now contains $\mathbf{q}_{z,\omega}$ for all depth-steps, \mathbf{Z} is a zero-padding operator, and \mathbf{D}_ω is the extrapolation matrix in equation (6.5) that can be inverted rapidly by recursion.

Equation (6.6) encapsulates the idea that we can reconstruct the wavefield at every depth-step in the earth from the wavefield at the surface by inverting matrix, \mathbf{D}_ω .

As a final step, to produce a migrated image we need to invoke an imaging condition. For the case of exploding-reflector (zero-offset) migration [e.g. Claerbout (1995)], we need to extract the image, $\hat{\mathbf{m}}$, corresponding to $t = 0$. In the temporal frequency domain, we can do this by summing over frequency. This stacking process is described by the matrix equation,

$$\begin{pmatrix} \hat{\mathbf{m}}_1 \\ \hat{\mathbf{m}}_2 \\ \hat{\mathbf{m}}_3 \\ \vdots \\ \hat{\mathbf{m}}_{N_z} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_{N_{xy}} & 0 & 0 & \dots & 0 & \mathbf{I}_{N_{xy}} & 0 & 0 & \dots & 0 & \dots & \mathbf{I}_{N_{xy}} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{I}_{N_{xy}} & 0 & \dots & 0 & 0 & \mathbf{I}_{N_{xy}} & 0 & \dots & 0 & \dots & 0 & \mathbf{I}_{N_{xy}} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{I}_{N_{xy}} & \dots & 0 & 0 & 0 & \mathbf{I}_{N_{xy}} & \dots & 0 & \dots & 0 & 0 & \mathbf{I}_{N_{xy}} & \dots & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & \mathbf{I}_{N_{xy}} & 0 & 0 & 0 & \dots & \mathbf{I}_{N_{xy}} & \dots & 0 & 0 & 0 & \dots & \mathbf{I}_{N_{xy}} \end{pmatrix} \begin{pmatrix} \mathbf{q}_{\omega_1} \\ \mathbf{q}_{\omega_2} \\ \vdots \\ \mathbf{q}_{\omega_{N_\omega}} \end{pmatrix},$$

or again more simply,

$$\hat{\mathbf{m}} = (\mathbf{I}_{N_{xy}N_\omega} \quad \mathbf{I}_{N_{xy}N_\omega} \quad \dots \quad \mathbf{I}_{N_{xy}N_\omega}) \mathbf{q} = \Sigma_\omega \mathbf{q}, \quad (6.7)$$

where \mathbf{m} is the reflectivity model, \mathbf{u} is the total wavefield, \mathbf{I}_N is the identity matrix of rank N , and Σ_ω is the (exploding-reflector) imaging operator that sums over frequencies.

The process of imaging by exploding-reflector migration can then be summarized as the chain of composite operators:

$$\hat{\mathbf{m}} = \Sigma_\omega \mathbf{D}^{-1} \mathbf{Z}_{N_\omega} \mathbf{d}, \quad (6.8)$$

where

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{\omega_1} & 0 & \dots & 0 \\ 0 & \mathbf{D}_{\omega_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mathbf{D}_{\omega_{N_\omega}} \end{pmatrix}, \quad \mathbf{Z}_{N_\omega} = \begin{pmatrix} \mathbf{Z} \\ \mathbf{Z} \\ \dots \\ \mathbf{Z} \end{pmatrix}, \quad \text{and} \quad \mathbf{d} = \begin{pmatrix} \mathbf{d}_{\omega_1} \\ \mathbf{d}_{\omega_2} \\ \dots \\ \mathbf{d}_{\omega_{N_\omega}} \end{pmatrix}.$$

6.2.2 Exploding-reflector modeling by upward continuation

The adjoint¹ of exploding-reflector migration is exploding-reflector modeling: if we can migrate exploding-reflector data with

$$\begin{aligned} \hat{\mathbf{m}} &= \mathbf{A}'_{\text{ER}} \mathbf{d} \\ &= \Sigma_\omega \mathbf{D}^{-1} \mathbf{Z}_{N_\omega} \mathbf{d}, \end{aligned} \quad (6.9)$$

then we should be able to forward model synthetic data with

$$\begin{aligned} \hat{\mathbf{d}} &= \mathbf{A}_{\text{ER}} \mathbf{m} \\ &= \mathbf{Z}'_{N_\omega} (\mathbf{D}')^{-1} \Sigma'_\omega \mathbf{m}. \end{aligned} \quad (6.10)$$

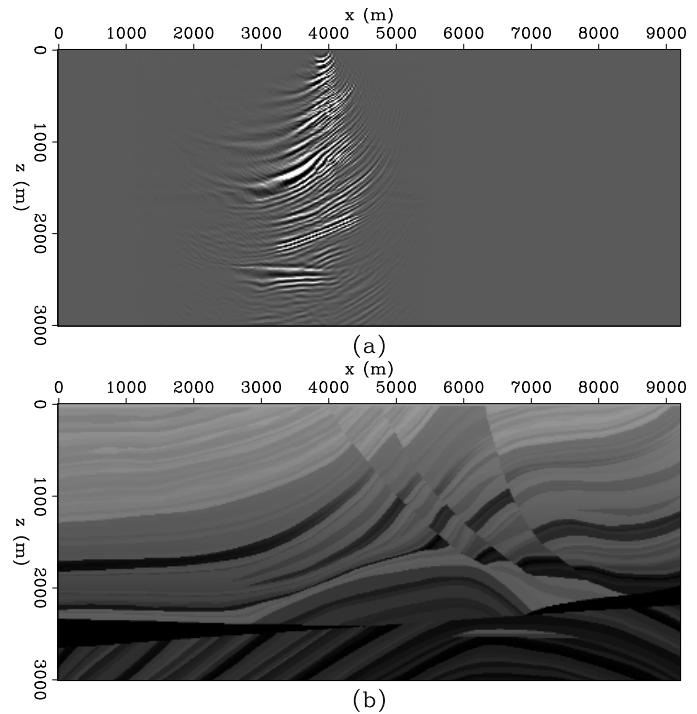
¹The adjoint operator here refers to the Hermitian conjugate operator, which I denote with the prime ('') notation.

A close look at the composite operator reveals that the first step is to apply Σ'_ω , which sprays the reflectivity model out to each frequency. The second step is to apply $(\mathbf{D}')^{-1}$ which recursively marches the wavefield up through the earth. The final step is to truncate the data at the surface with the \mathbf{Z}'_{N_ω} operator.

6.2.3 Shot-profile migration and modeling

Shot-profile migration is similar to exploding-reflector migration, except individual shot records are migrated independently. Single shot-record migrations (like Figure 6.1) are then stacked to provide a complete reflectivity image.

Figure 6.1: A single migrated shot record (a) over the Marmousi (Bourgeois et al., 1991) velocity model (b).
shots-singleshot [ER,M]



Rather than downward-continue a single wavefield into the earth, we downward-continue two wavefields simultaneously (but still one frequency at a time). Firstly, there is upgoing wavefield ($\mathbf{q}_{\omega,s}^+$), which was recorded on the surface as the shot-gather, $\mathbf{d}_{\omega,s}$, with a shot at s .

The equation for downward continuing this wavefield is:

$$\mathbf{D}_\omega \mathbf{q}_{\omega,s}^+ = \mathbf{Z} \mathbf{d}_{\omega,s}, \quad (6.11)$$

Secondly, there is the downgoing wavefield ($\mathbf{q}_{\omega,s}^-$),

$$\mathbf{D}_\omega \mathbf{q}_{\omega,s}^- = \mathbf{Z} \delta_{\omega,s}, \quad (6.12)$$

which we did not record on the surface but we can simulate as an impulse at the shot location, $\delta_{\omega,s}$.

To image an individual shot, we crosscorrelate the two wavefields and extract the image at zero-time, or equivalently multiply their Fourier coefficients and sum over frequencies. The matrix operation described by this process is,

$$\begin{pmatrix} \hat{\mathbf{m}}_{1,s} \\ \hat{\mathbf{m}}_{2,s} \\ \hat{\mathbf{m}}_{3,s} \\ \dots \\ \hat{\mathbf{m}}_{N_z,s} \end{pmatrix} = \Sigma_\omega \begin{pmatrix} \text{diag}(\mathbf{q}_{\omega_1}^-) & 0 & \dots & 0 \\ 0 & \text{diag}(\mathbf{q}_{\omega_2}^-) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \text{diag}(\mathbf{q}_{\omega_{N_\omega}}^-) \end{pmatrix} \begin{pmatrix} \mathbf{q}_{\omega_1}^+ \\ \mathbf{q}_{\omega_2}^+ \\ \dots \\ \mathbf{q}_{\omega_{N_\omega}}^+ \end{pmatrix}. \quad (6.13)$$

To image the entire survey we then stack over shot-location. Full prestack shot-profile migration is therefore described by the equation

$$\hat{\mathbf{m}} = \Sigma_s \Sigma_\omega \mathbf{Q}_- \mathbf{D}^{-1} \mathbf{Z}_{N_\omega N_s} \mathbf{d} = \mathbf{A}'_{SP} \mathbf{d}, \quad (6.14)$$

where $\mathbf{Q}_- = \text{diag}(\mathbf{q}^-)$, and \mathbf{A}'_{SP} is the composite shot-profile migration operator.

The adjoint of equation (6.14) describes a process of common-shot modeling:

$$\hat{\mathbf{d}} = \mathbf{Z}'_{N_\omega N_s} (\mathbf{D}')^{-1} \mathbf{Q}_- \Sigma'_\omega \Sigma'_s \mathbf{m} = \mathbf{A}_{SP} \mathbf{m}. \quad (6.15)$$

Implementing equation (6.15) is slightly more difficult than equation (6.14), since we cannot downward continue both wavefields from the surface simultaneously. However, we can still work with frequency slices; although for each frequency we must downward continue the shot

wavefield from the surface to the bottom of the model storing it as we go, and then upward continue the reflected wavefield to the surface.

6.2.4 True-amplitude shot-modeling

Under the single-scattering (first-order Born) approximation the scattered field at r due to a shot at s is given by

$$q_{\text{scat}}(r|s) \sim \omega^2 \int G(r'|s) G(r|r') \Delta(r') dV(r'), \quad (6.16)$$

where $G(r|s)$ is the Green's function response at r of the medium to an impulse at s , and $\Delta(r')$ represents perturbation in medium parameters.

Equation (6.15) describes the process of calculating the volume integral in equation (6.16) first by downward continuing the shot wavefield to give $G(r'|s)$, then by upward continuing the scattered field to the surface, accumulating contributions from each depth slice on the way back up. The ω^2 term can be applied as a data-space filter after modeling. However, for the amplitudes to be correct in equation (6.16), the extrapolation operators need to represent true-amplitude Green's functions.

For $v(z)$ media, the WKBJ Green's function for a smoothly-varying background velocity field [e.g. Aki and Richards (1980); Stolt and Benson (1986)] is given by

$$q(k_x, k_y, z_2, \omega) = q(k_x, k_y, z_1, \omega) \sqrt{\frac{k_z(z_1)}{k_z(z_2)}} \exp \left[i \int_{z_1}^{z_2} k_z(z') dz' \right], \quad (6.17)$$

$$\text{where } k_z = \frac{\omega^2}{v(z)^2} - k_x^2 - k_y^2. \quad (6.18)$$

The amplitude term in equation (6.17) can be incorporated into the shot-profile modeling procedure; however, different operators are required for upward versus downward continuation. Firstly, the true-amplitude downgoing WKBJ extrapolators should be substituted into equation (6.12), and secondly, the upgoing WKBJ extrapolators should be substituted for the adjoint of \mathbf{D} in equation (6.15).

6.3 Towards true-amplitude migration

If we know the subsurface reflectivity, simple linear wave propagation theory, such as that described in section 6.2.4, allows us to predict what seismic reflection data would be recorded in an experiment. This allows us to solve the forward problem of seismic imaging: given a reflectivity model, \mathbf{m} , we can construct a forward modeling operator, \mathbf{A} , such that the predicted data, $\mathbf{d} = \mathbf{A} \mathbf{m}$.

Unfortunately, in a geophysical experiment we record \mathbf{d} , and would like to find \mathbf{m} . This is the so-called inverse problem, and is much more difficult to solve. As discussed in the previous section, rather than actually trying to invert the operator, \mathbf{A} , seismic migration amounts simply to applying its adjoint,

$$\hat{\mathbf{m}} = \mathbf{A}' \mathbf{d}. \quad (6.19)$$

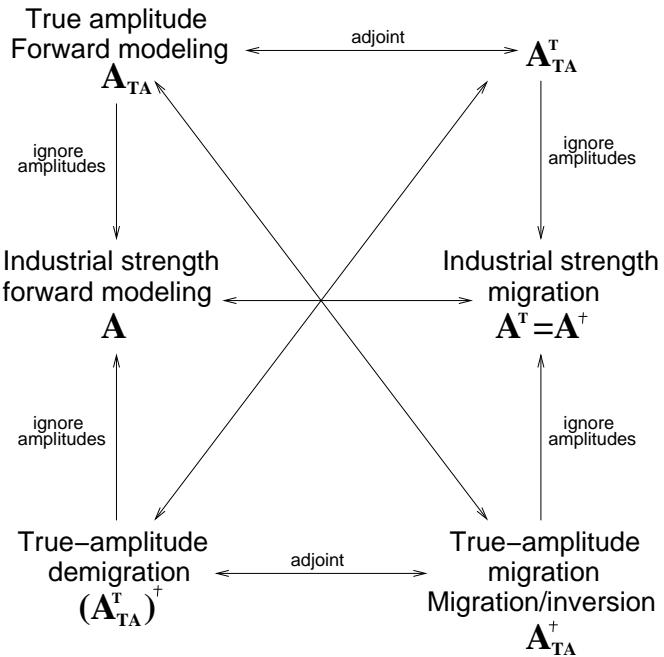
Due to the symmetry of wave-propagation with respect to time-reversal, it turns out that migrating with the adjoint operator treats event kinematics correctly, and produces structurally correct images of the subsurface. Adjoint processing is also robust to the presence of noise, and missing or inconsistent data. However, the major shortcoming of migrating with the adjoint is that it does not treat seismic amplitudes correctly. Also, because processing places the emphasis on kinematics not amplitudes, amplitude terms are often completely ignored, or artificially constructed so that $\mathbf{A}'\mathbf{A} \approx \mathbf{I}$.

Interpreters, however, often try to extract more from seismic reflection images than kinematics: for example, rock physics studies show how important rock parameters such as porosity, lithology and fluid saturations may influence seismic amplitudes. The failure of migrating with the adjoint to correctly handle amplitudes has lead to the search for “true-amplitude”, or “amplitude-preserving” migration operators. Rather than applying the adjoint of a loosely-defined forward-modeling operator, true-amplitude schemes rigorously formulate the forward-modeling operator \mathbf{A}_{TA} , and then approximate \mathbf{A}_{TA}^{-1} with a pseudo-inverse \mathbf{A}_{TA}^\dagger . For example, Bleistein (1987) describes a Kirchhoff operator that becomes the inverse of the modeling operator in the high-frequency asymptotic limit.

If true-amplitude migration is the pseudo-inverse of the physically correct forward-modeling

operator, then its adjoint, $(\mathbf{A}_{\text{TA}}')^\dagger$, is known as “demigration”. Demigration is receiving increasing attention as part of amplitude-preserving processing flows (Hubral et al., 1996). Figure 6.2 illustrates the relationships between forward modeling, migration, true-amplitude migration and demigration. Figure 6.2 also shows the relationship these operators have with what I refer to as “industrial-strength” migration - that is the pseudo-unitary operator, which is kinematically correct, but ignores the amplitude effects of wave-propagation in depth.

Figure 6.2: Relationship between industrial strength migration, modeling, their pseudo-inverses, and their (pseudo-unitary) industrial-strength counterparts.
 shots-miginv4 [NR]



6.3.1 Exploding-reflector modeling as a pseudo-unitary operator

In a constant velocity medium, the process of exploding-reflector modeling followed by exploding-reflector migration acts to preserve the amplitudes of flat-events. This can be proved by considering the zero-offset case for the true-amplitude migration weights derived by Sava et al. (2001). Figure 6.3 confirms this by displaying the frequency-domain response of cascaded modeling and migration ($\mathbf{A}'_{\text{ER}} \mathbf{A}_{\text{ER}}$). Panel (a) shows the input spectrum that has been band-passed and dip-limited to remove evanescent energy. Panel (b) shows the spectrum after exploding-reflector modeling and migration. Although high-spatial wavenumbers are attenuated slightly, the low-spatial wavenumbers associated with flat events remain essentially

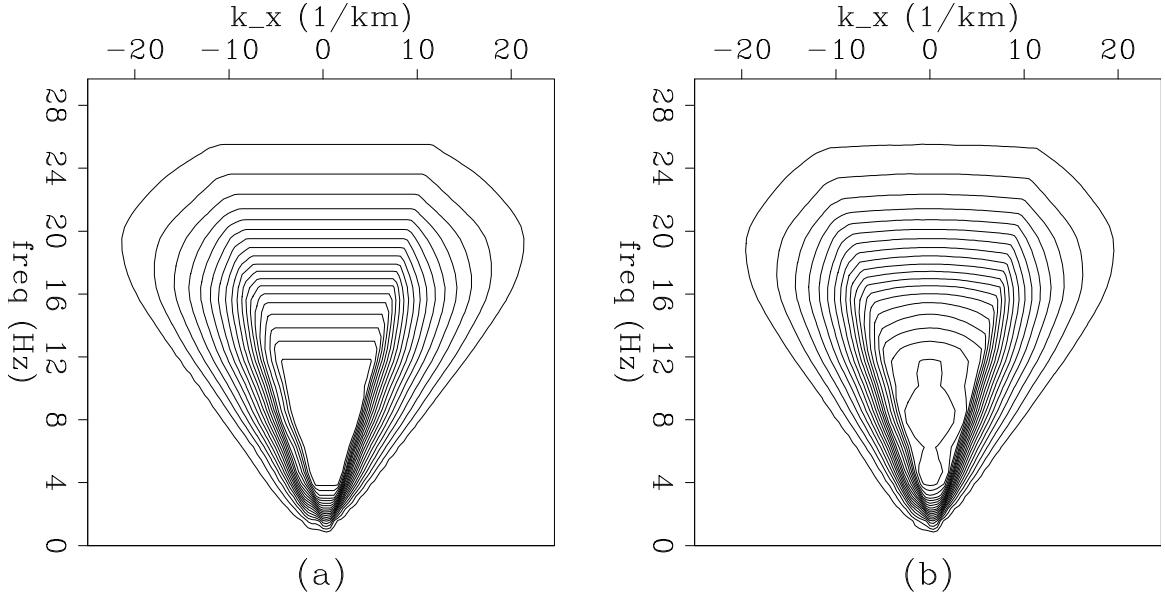


Figure 6.3: Frequency ($f - k_x$) response of cascaded modeling and migration in a constant velocity medium. Panel (a) shows amplitude spectrum of dip-limited impulsive input. Panel (b) shows spectrum after exploding-reflector modeling and migration. [shots-unitary] [ER,M]

unchanged. Therefore, if a model consists of mostly flat-events, exploding-reflector modeling acts as a pseudo-unitary operator: $\mathbf{A}'_{\text{ER}} \mathbf{A}_{\text{ER}} \approx \mathbf{I}$.

Equation (6.10) decomposes exploding-reflector modeling by wavefield extrapolation into a spraying operator, Σ'_{ω} , followed by an extrapolation/recording operator, \mathbf{B} , so that

$$\mathbf{d} = \mathbf{A}_{\text{ER}} \mathbf{m} \quad (6.20)$$

$$= \mathbf{B} \Sigma'_{\omega} \mathbf{m} \quad (6.21)$$

where, in the terminology of Chapter 1, $\mathbf{B} = \mathbf{Z}'_{N_{\omega}} (\mathbf{D}')^{-1}$. For a model consisting of mostly flat-events, this implies $\mathbf{B}' \mathbf{B} \approx \mathbf{I}$ subject to a multiplicative constant.

If the approximation that $\mathbf{B}' \mathbf{B} \approx \mathbf{I}$ is indeed valid, then amplitude problems on zero-offset migrations must be associated with the two-way wave propagation rather than the exploding-reflector migration. To test the validity of this conjecture, I can generate exploding-reflector and genuine zero-offset seismograms and compare the results of exploding-reflector migration

on the two datasets.

Figure 6.4 compares exploding-reflector data with true zero-offset data. I generated the two datasets by running the adjoint of exploding-reflector and shot-profile downward continuation migration algorithms respectively. Kinematically they are similar; however, vertical amplitude streaking is much more apparent in the true zero-offset section. Field datasets, such as the stacked section shown in Figure 6.5, often contain vertical amplitude streaks. Such streaks often pose a dilemma for a processing geophysicist as to how they should be correctly treated.

An even more interesting picture emerges when the two modeled datasets are migrated (Figure 6.6) with an exploding-reflector migration algorithm. The exploding-reflector data migrates nearly perfectly. Whereas the high amplitude streaks in the true zero-offset dataset migrate into the commonly seen “migration-smile” artifacts.

6.4 Compensating for irregular shot illumination

In the first section of this Chapter, I discussed how shot-profile migration is the adjoint of a linear first-order Born modeling operator. Given such a modeling operator, geophysical inverse theory [e.g. Tarantola (1987)] provides a rigorous framework which allows us to estimate a model of the subsurface that is unbiased by our recording geometry and the effects of wave propagation. Several authors [see Ronen and Liner (2000) for a full review] leverage this, and calculate a least-squares (L_2) pseudo-inverse to the forward modeling operator.

For a generic forward modeling operator, \mathbf{A} , and more data values than unknown model parameters, the model estimate, $\hat{\mathbf{m}}$ that best predicts the data in a least-squares sense is given by the solution to the normal equations,

$$\mathbf{A}' \mathbf{A} \hat{\mathbf{m}} = \mathbf{A}' \mathbf{d}. \quad (6.22)$$

Often for large geophysical problems such as migration, this system of equations is not

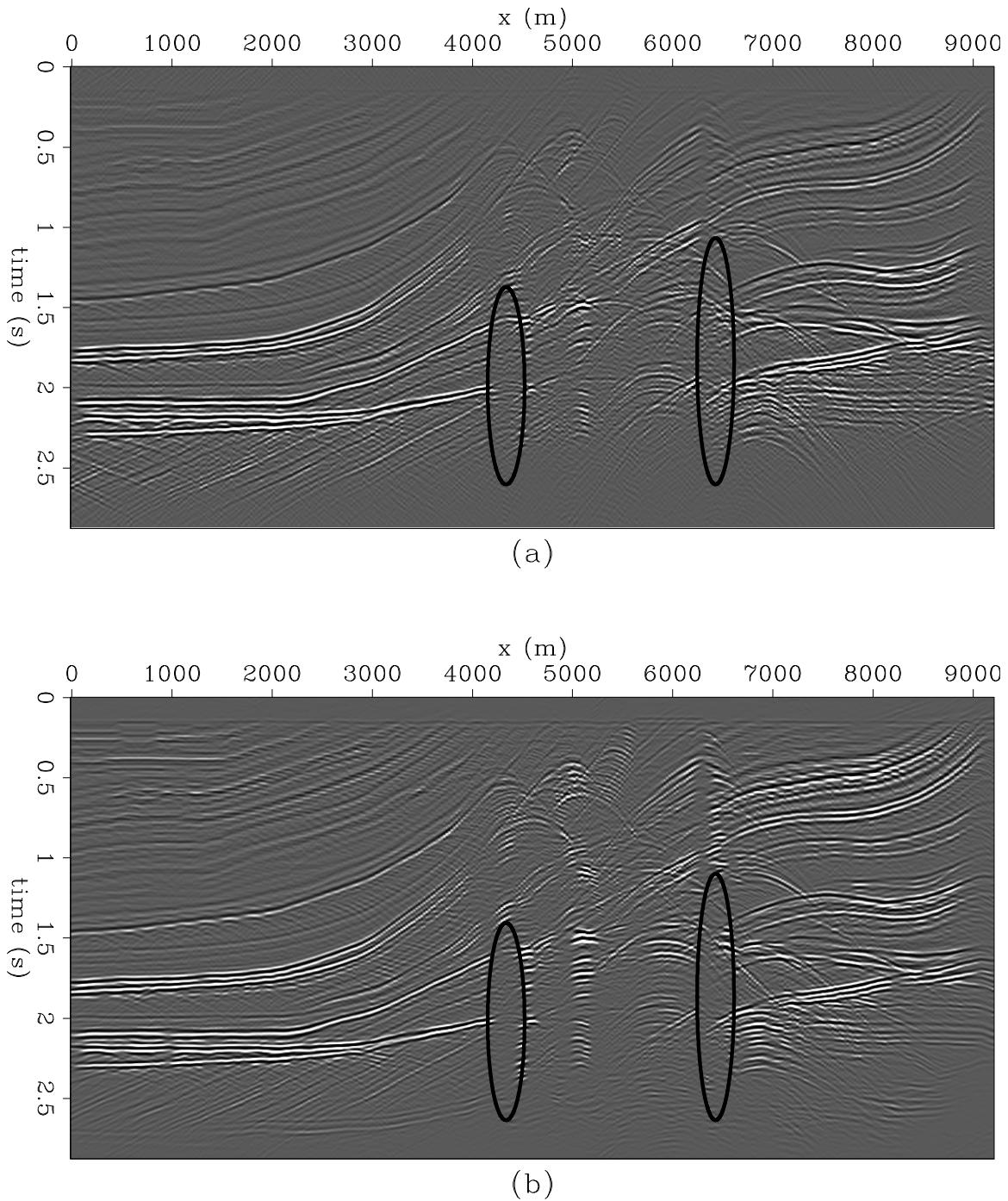
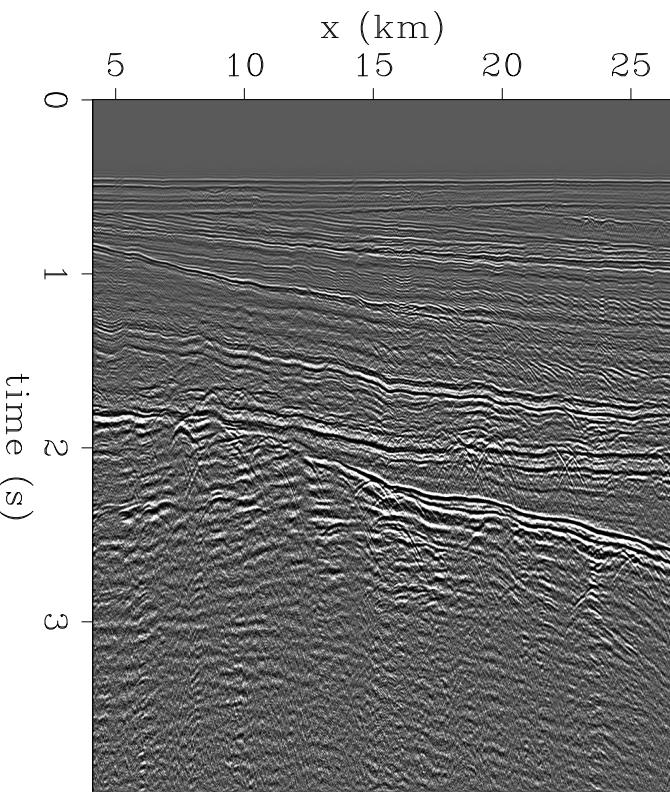


Figure 6.4: Synthetic Marmousi datasets generated with one-way wave modeling: panel (a) shows the exploding-reflector dataset, and panel (b) shows the zero-offset dataset. The ellipses highlight the increased vertical streaking in the true zero-offset dataset.
shots-marmERZ0data [CR,M]

Figure 6.5: Stacked section from the Mobil AVO dataset (Clapp, 1999). Vertical streaking is visible throughout the section.

shots-illumexamples [ER,M]



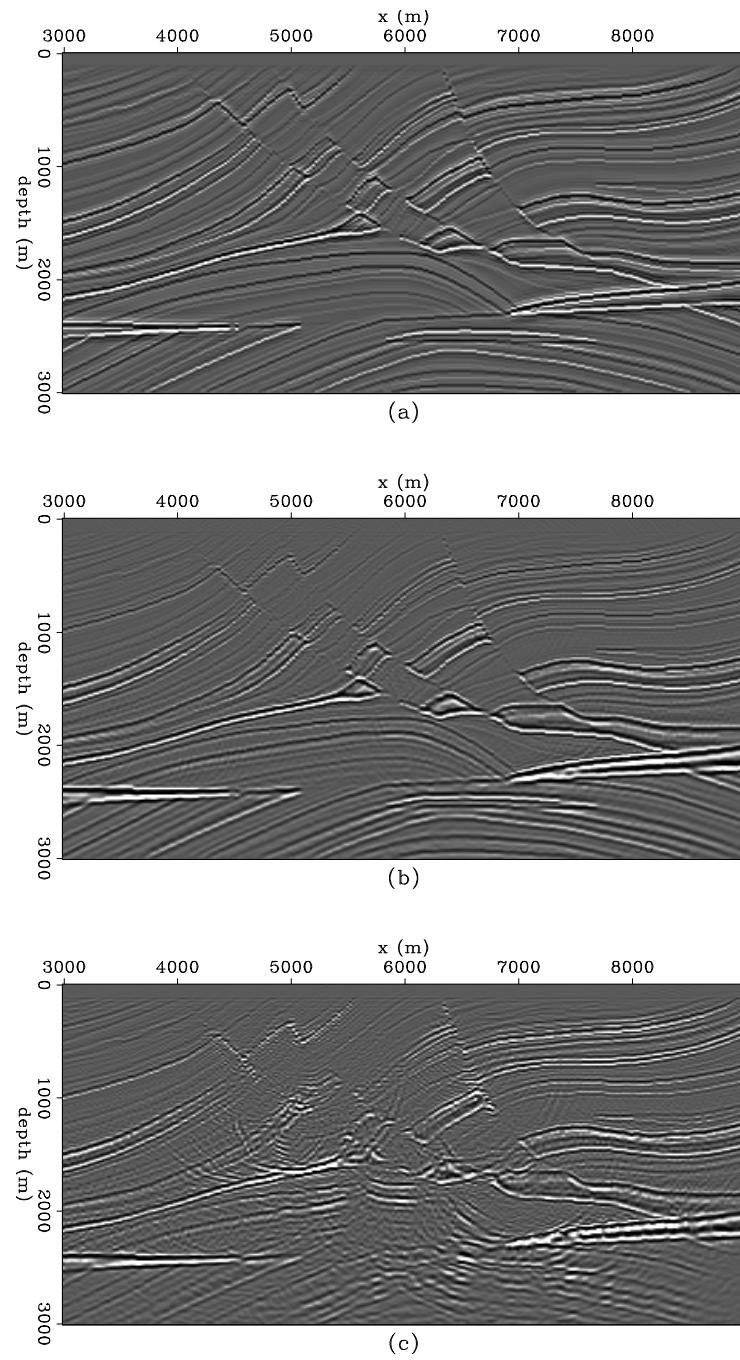


Figure 6.6: Marmousi reflectivity: panel (a) shows the initial model, panel (b) shows the reflectivity estimate by migrating the exploding-reflector data shown in Figure 6.4 (a), and panel (c) shows the reflectivity estimate obtained by migrating the zero-offset data from Figure 6.4 (b).

shots-marmERZO [CR,M]

solved directly, rather the solution is estimated by an iterative method such as conjugate gradients [e.g. Nemeth et al. (1999); Prucha et al. (2000)].

Obviously, however, if the operator, $\mathbf{A}'\mathbf{A}$ is diagonal, it is easy to calculate $(\mathbf{A}'\mathbf{A})^{-1}$, and obtain the L^2 solution directly from the adjoint (migrated) image. Rather than iterate with the entire migration operator, in this section I will consider the effect of explicitly inverting the chain of operators that make up wave-equation modeling, looking for situations in which $\mathbf{A}'\mathbf{A}$ is diagonal.

The shot-profile modeling operator, given in equation (6.15), can also be expressed in terms of the extrapolation operator, \mathbf{B} , as

$$\mathbf{d} = \mathbf{A}_{SP} \mathbf{m} \quad (6.23)$$

$$= \mathbf{B} \mathbf{Q}_- \Sigma'_\omega \Sigma'_s \mathbf{m}. \quad (6.24)$$

The classical L^2 estimate of \mathbf{m} is given by

$$\hat{\mathbf{m}} = (\mathbf{A}'_{SP} \mathbf{A}_{SP})^{-1} \mathbf{A}'_{SP} \mathbf{d} \quad (6.25)$$

$$= (\Sigma_s \Sigma_\omega \mathbf{Q}'_- \mathbf{B}' \mathbf{B} \mathbf{Q}_- \Sigma'_\omega \Sigma'_s)^{-1} \mathbf{A}'_{SP} \mathbf{d}. \quad (6.26)$$

Following the discussion in the previous section, I will assume $\mathbf{B}' \mathbf{B} \approx \mathbf{I}$, so that

$$\hat{\mathbf{m}} = (\Sigma_s \Sigma_\omega |\mathbf{Q}_-|^2 \Sigma'_\omega \Sigma'_s)^{-1} \mathbf{A}'_{SP} \mathbf{d} \quad (6.27)$$

$$= \mathbf{V}_{SP}^{-1} \mathbf{A}'_{SP} \mathbf{d}. \quad (6.28)$$

Close inspection of $\mathbf{V}_{SP}(=\Sigma_s \Sigma_\omega |\mathbf{Q}_-|^2 \Sigma'_\omega \Sigma'_s)$ reveals that it is a diagonal matrix that can be applied as a model-space weighting function after migration. In physical terms, the diagonal of \mathbf{V}_{SP} contains the total *shot illumination*: the integration over frequency of $|\mathbf{q}_-|^2$, the energy in the downgoing (shot) wavefield at each model point. Shot illumination can be computed cheaply during the migration process.

This weighting function is also equivalent to the upgoing/downgoing wavefield imaging condition originally proposed by Claerbout (1971). Field data results (Jacobs, 1982) show

this imaging condition is very susceptible to noise. However, dividing by the downgoing wavefield after migration, rather than directly as part of the imaging condition, has a significant advantage: the choice of appropriate local smoothing and stabilization parameters (e.g. ϵ) may be made after the migration is finished.

Duquet et al. (2000) calculate the subsurface illumination for Kirchhoff migration by summing all contributions from a single scatterer that get modeling into dataspace. The shot illumination described here is a special case of Duquet's illumination that assumes all scattered energy is recorded. However, the extra cost of calculating shot illumination is negligible compared to the cost of a single migration, and Duquet's approach to calculating illumination is not appropriate for wave-equation migration schemes (see Appendix C).

6.4.1 Shot illumination examples

As described above, normalization by shot illumination only approximates an inverse operation if the illumination problems come from poor sampling on the shot axis (or reciprocal shot) rather than on both shot and receiver axes. As a best case scenario, to test the validity of this approach, I generated a dataset consisting of 20 split-spread shots over the Marmousi velocity model, using a paraxial Fourier finite-difference wavefield extrapolation algorithm (Ristow and Ruhl, 1994). The receiver axis was well sampled, as each shot-gather contained 200 traces.

The top panel in Figure 6.7 shows the result of migrating the 20 shots with the correct velocity model, and the center panel shows the shot illumination. Clearly areas of high illumination correspond to bright spots in the migrated image. Panel (c) shows the migrated image after normalization by the shot illumination. Although only 20 shots were migrated into this image, its clarity is comparable with the full migration of the entire dataset (about 240 shots). It is also encouraging that while the shot locations are visible on Figure 6.7 (a), they are hidden in Figure 6.7 (c): the normalized image appears to be unbiased by the recording geometry.

Compensation for shot illumination in Figure 6.7 worked so well partly because the modeling and migration procedures were true adjoints. To test how robust this normalization would

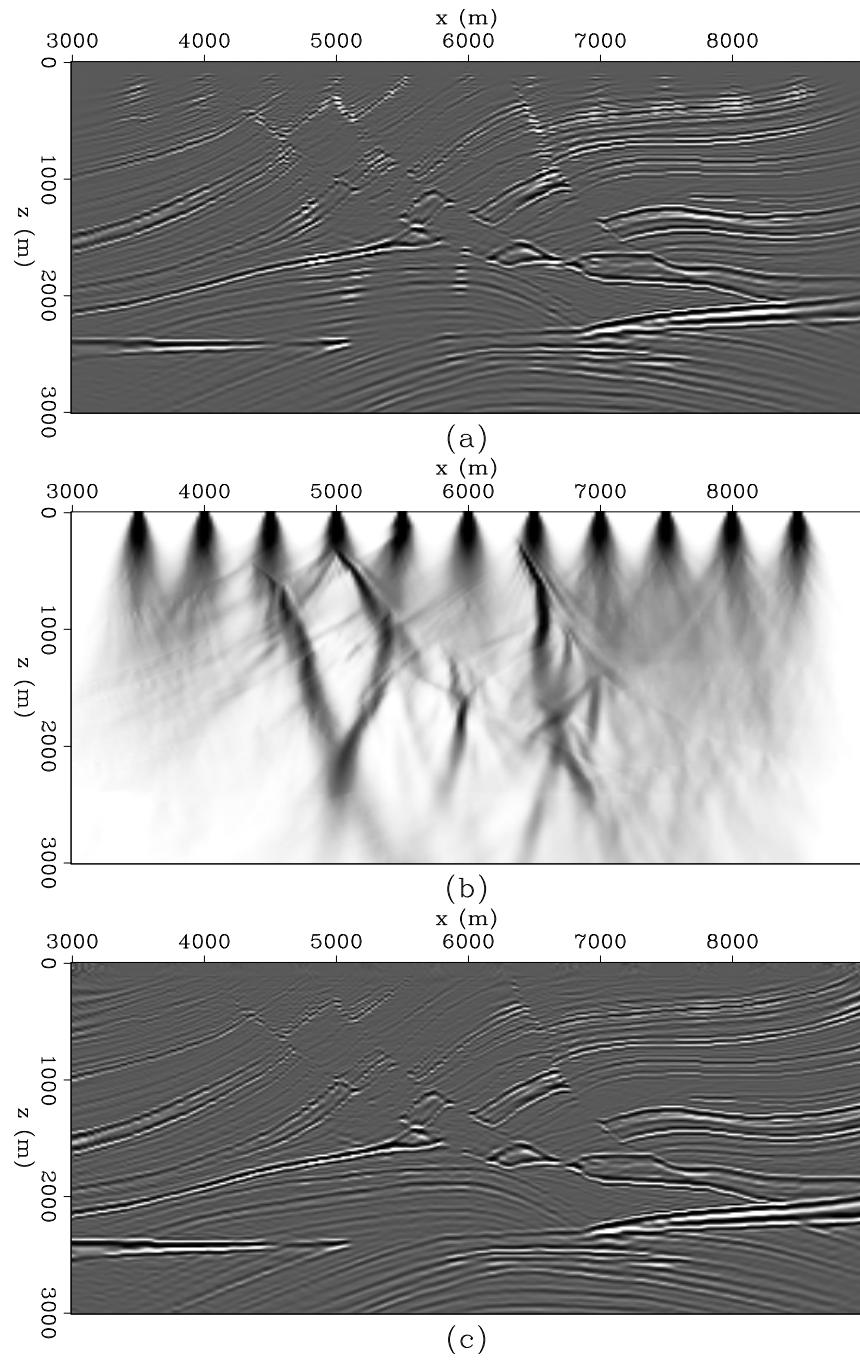


Figure 6.7: Normalization by shot illumination: the panel (a) shows shot profile migration with the correct velocity model of 20 split-spread shots over the Marmousi model, panel (b) shows shot illumination, and panel (c) shows the normalized image. [shots-marmsparseill1] [ER,M]

be in the more realistic situation where the modeling and migration procedures are not true adjoints, I migrated the same 20 shots with an incorrect velocity model. Panels (a) and (b) of Figure 6.8 show the migrated image, and the corresponding shot illumination respectively. Panel (c) shows the migrated image after normalization. Despite the incorrect velocity, the amplitude artifact visible in Figure 6.8 (a) are largely gone from Figure 6.8 (c). For example in the top 500 m of the section, the footprint of the acquisition geometry is removed from Figure 6.8 (c). The vertical amplitude streaks lower in the section are also reduced — illumination of the target reflectors at 2600 m depth is more uniform after normalization, especially in the highlighted area.

6.4.2 Application to true-amplitude migration

The analysis above ignores the $v(z)$ true-amplitude WKBJ Green's functions (Stolt and Benson, 1986) discussed in section 6.2.4. It also ignores the effect the imaging condition has on dipping events. Sava et al. (2001) discuss this in more detail; but the Jacobian associated with the transformation from temporal frequency to vertical wavenumber causes the slight attenuation of dipping events visible in Figure 6.3.

Both of these effects, however, are well-understood, and they can be compensated for in a true-amplitude migration operator, $\mathbf{A}_{\text{TA}}^\dagger$. Such a true-amplitude migration operator will no longer be the true adjoint of the forward modeling operator; but we can still build and solve the system of normal equations,

$$\mathbf{A}_{\text{TA}}^\dagger \mathbf{A}_{\text{TA}} \tilde{\mathbf{m}} = \mathbf{A}_{\text{TA}}^\dagger \mathbf{d}. \quad (6.29)$$

Since $\mathbf{A}_{\text{TA}}^\dagger$ replaces \mathbf{A}_{TA}' , the solution to this system will not be equivalent to the solution that minimizes the residual error in fitting goal,

$$\mathbf{A}_{\text{TA}} \mathbf{m} \approx \mathbf{d}. \quad (6.30)$$

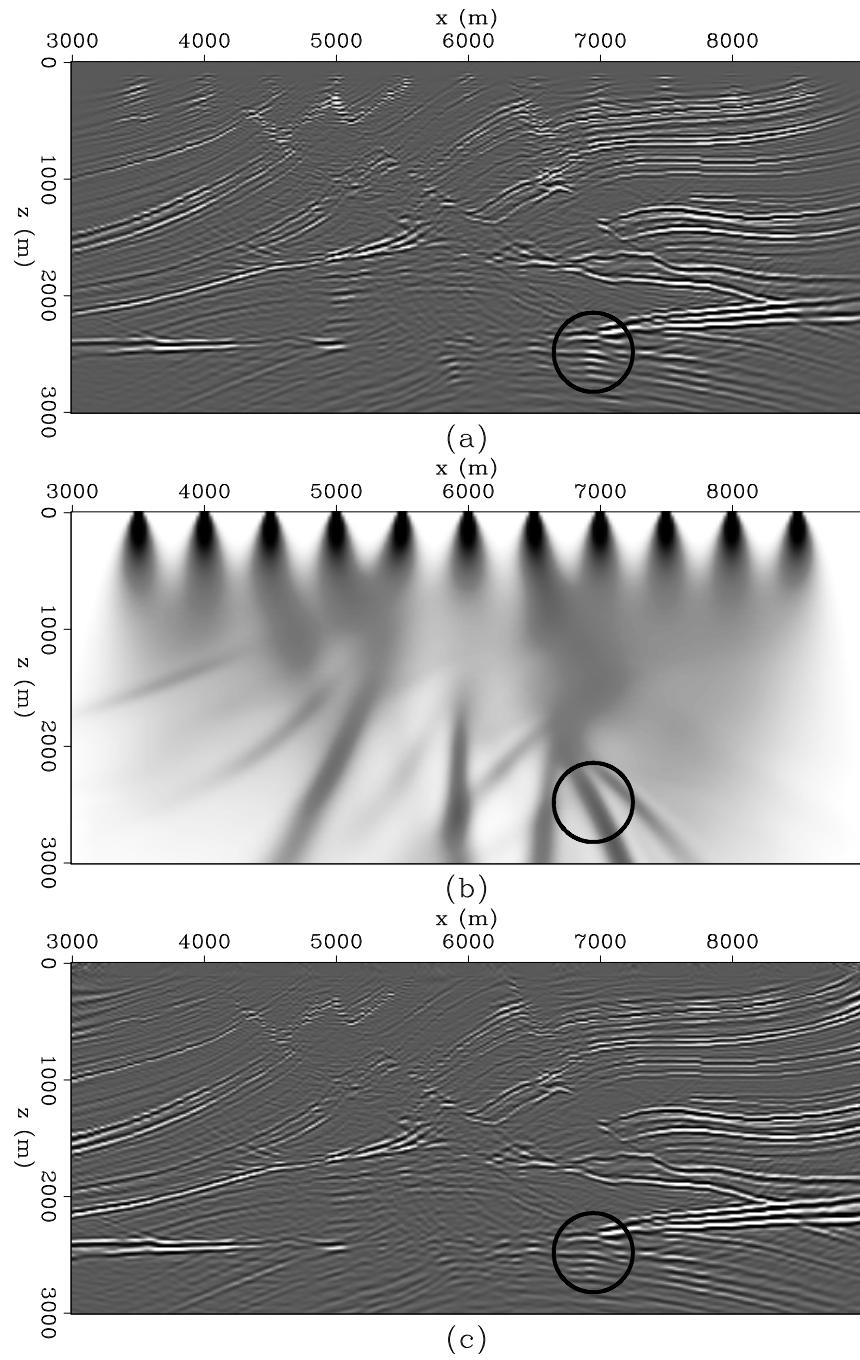


Figure 6.8: Normalization by shot illumination: the panel (a) shows shot profile migration with the *incorrect* velocity model of 20 split-spread shots over the Marmousi model, panel (b) shows shot illumination, and panel (c) shows the normalized image. [shots-marmsparseill2] [ER,M]

However, if a pair of invertible operators (\mathbf{F}_M and \mathbf{F}_D) exist, such that

$$\mathbf{A}_{\text{TA}}^\dagger = \mathbf{F}'_M \mathbf{F}_M \mathbf{A}'_{\text{TA}} \mathbf{F}'_D \mathbf{F}_D,$$

then the solution to equation (6.29) will be the same as the solution that minimizes the residual error in fitting goal,

$$\mathbf{F}_D \mathbf{A}_{\text{TA}} \mathbf{m} \approx \mathbf{F}_D \mathbf{d}. \quad (6.31)$$

The solution will also converge faster under an iterative inversion scheme since $\mathbf{A}_{\text{TA}}^\dagger \mathbf{A}_{\text{TA}} \approx \mathbf{I}$.

6.5 Conclusions

Shot-profile and shot-geophone migration algorithms provide two alternatives for finite-frequency migration of prestack seismic data. For marine-type geometries shot-geophone migration will be cheaper; however, this situation is reversed for sparse-shot and sparse-receiver geometries.

For sparse-shot geometries with fully sampled receiver axes, I have shown that shot illumination plays a major role in observed seismic amplitudes. I show how this problem can be addressed by a simple model-space weighting function that can be calculated at very little cost during the migration process, and applied afterwards.

Chapter 7

Model versus data normalization

Illumination problems caused by finite-recording aperture and lateral velocity lensing can lead to amplitude fluctuations in migrated images. I calculate both model and data-space weighting functions that compensate for these illumination problems in finite-frequency depth migration results. These weighting functions can either be applied directly with migration to mitigate the effects of poor subsurface illumination, or used as preconditioning operators in iterative least-squares (L_2) migrations. Computational shortcuts allow the weighting functions to be computed at about the cost of a single migration. Results indicate that model-space normalization can significantly reduce amplitude fluctuations due to illumination problems. However, for the examples presented here, data-space normalization proved susceptible to coherent noise contamination.

7.1 Introduction

The generic geophysical inverse problem (Tarantola, 1987; Claerbout, 1998a) can be summarized as follows - given a linear forward modeling operator \mathbf{A} , and some recorded data \mathbf{d} , estimate a model \mathbf{m} such that $\mathbf{A}\mathbf{m} \approx \mathbf{d}$. If the system is over-determined, the model that minimizes the expected (L_2) error in predicted data is given by the solution to the normal

equations (6.22):

$$\mathbf{m}_{L2} = (\mathbf{A}' \mathbf{A})^{-1} \mathbf{A}' \mathbf{d}. \quad (7.1)$$

Typically the matrices involved in industrial-scale geophysical inverse problems are too large to invert directly, and we depend on iterative gradient-based linear solvers to estimate solutions. However, operators such as prestack depth migration are so expensive to apply that we can only afford to iterate a handful of times, at best.

To attempt to speed convergence, we can always change model-space variables from \mathbf{m} to \mathbf{x} through a linear operator \mathbf{P} , and solve the following new system for \mathbf{x} ,

$$\mathbf{d} = \mathbf{A} \mathbf{P} \mathbf{x} = \mathbf{B} \mathbf{x}. \quad (7.2)$$

When we find a solution, we can then recover the model estimate, $\mathbf{m}_{L2} = \mathbf{P} \mathbf{x}$.

If we choose the operator \mathbf{P} such that $\mathbf{B}' \mathbf{B} \approx \mathbf{I}$, then even simply applying the adjoint (\mathbf{B}') will yield a good model estimate; furthermore, gradient-based solvers should converge to a solution of the new system rapidly in only a few iterations.

7.1.1 Preconditioning and spectral factorization

In Chapter 1, I discussed how spectral factorization amounts to the problem of finding an invertible square root of an autocorrelation function. The problem of finding an appropriate preconditioning operator for a linear inverse problem can be considered a generalization of this.

In general, any Hessian operator $\mathbf{A}' \mathbf{A}$ can be described as a non-stationary autocorrelation filter. If we could find a pair of invertible non-stationary factors, they would be the perfect preconditioning operators. Unfortunately for large problems, however, we cannot actually form the Hessian matrix explicitly, let alone factor it directly.

7.1.2 Diagonal weighting functions for recursive prestack depth migration

In this Chapter, I compute diagonal preconditioning operators (weighting functions) appropriate for least-squares recursive depth migration. Only considering the main diagonal of $\mathbf{A}'\mathbf{A}$ simplifies matters significantly: the factorization itself reduces to taking a square root, and inverting the factors reduces to simple division. The only potential pitfall comes with division by zero.

As well as looking at model-space weights, I also consider data-space weighting functions derived from the operator $\mathbf{A}\mathbf{A}'$, and develop a framework for computing and applying both model and data-space weights simultaneously.

7.2 Model-space weighting functions

Although the matrix $\mathbf{A}'\mathbf{A}$ is not strictly diagonal, a diagonal model-space weighting function may still reduce illumination-related amplitude artifacts caused by wave-propagation in the overburden or incomplete recording geometry.

Rather than trying to solve the full inverse problem given by equation (7.1), in this section I will look for a diagonal operator \mathbf{W}_m such that

$$\mathbf{W}_m^2 \mathbf{A}' \mathbf{d} \approx \mathbf{m}_{L2}. \quad (7.3)$$

\mathbf{W}_m can be applied to the migrated (adjoint) image with equation (7.3); however, in their review of L_2 migration, Ronen and Liner (2000) observe that normalized migration is only a good substitute for full (iterative) L_2 migration in areas of high signal-to-noise. In areas of low signal-to-noise, \mathbf{W}_m can be used as a model-space preconditioner to the full L_2 problem, as described in the previous section.

Claerbout and Nichols (1994) noticed that if we model and remigrate a reference image, the ratio between the reference image and the modeled/remigrated image will be a weighting

function with the correct physical units. For example, the weighting function, \mathbf{W}_m , given by

$$\mathbf{W}_m^2 = \frac{\text{diag}(\mathbf{m}_{\text{ref}})}{\text{diag}(\mathbf{A}' \mathbf{A} \mathbf{m}_{\text{ref}})} \approx \frac{1}{\mathbf{A}' \mathbf{A}}, \quad (7.4)$$

will have the same units as \mathbf{A}^{-1} . Furthermore, \mathbf{W}_m^2 will be the *ideal* weighting function if the reference model equals the true model and we have the correct modeling operator.

Equation (7.4) forms the basis for the first part of this chapter. However, when following this approach, there are two important practical considerations to take into account: firstly, the choice of reference image, and secondly, the problem of dealing with zeros in the denominator.

Similar normalization schemes have been proposed for Kirchhoff migration operators [e.g. Slawson et al. (1995); Biondi (1997); Chemingui (1999); Duquet et al. (2000)]. In fact, both Nemeth et al. (1999) and Duquet et al. (2000) report success with using diagonal model-space weighting functions as preconditioners for Kirchhoff L_2 migrations. Appendix C explains how Kirchhoff normalization schemes fit into the framework of equation (7.4).

7.2.1 Three choices of reference image

The ideal reference image would be the true subsurface model. However, since we do not know what that is, we have to substitute an alternative model. I experiment with three practical alternatives, which I will denote \mathbf{m}_1 , \mathbf{m}_2 , and \mathbf{m}_3 .

Claerbout and Nichols (1994) attribute to Bill Symes the idea of using the adjoint (migrated) image as the reference model. The rationale for this is that migration provides a robust estimate of the true model. As the first alternative I take Symes' suggestion, so that $\mathbf{m}_1 = \mathbf{A}' \mathbf{d}$. A potential problem with this choice is that it may depend too much on the data: the weighting function may be poorly determined in areas with little or no signal, and it will be difficult to separate data problems from operator problems.

The second alternative is to try a reference image of purely random numbers: $\mathbf{m}_2 = \mathbf{r}$, where \mathbf{r} is a random vector. This has the advantage of not being influenced by the data, but has the disadvantage that different realizations of \mathbf{r} may produce different weighting functions.

The third alternative (denoted \mathbf{m}_2) that I consider is using a monochromatic reference image consisting of purely flat events: literally flat-event calibration as proposed by Black and Schleicher (1989) and discussed further in Appendix C.

7.2.2 Stabilizing the denominator

To avoid division by zero, Claerbout and Nichols (1994) suggest multiplying both the numerator and denominator in equation (7.4) by $\text{diag}(\mathbf{A}' \mathbf{A} \mathbf{m}_{\text{ref}})$, and stabilizing the division by adding a small positive number to the denominator. Although this does solve the problem of division by zero, the numerator and denominator will still oscillate rapidly in amplitude with the phase of the image.

Illumination, however, should be independent of the wavefield's phase. Therefore, I calculate weighting functions from the ratio of the smoothed analytic signal envelopes (denoted by $\langle \rangle$) of the model-space images:

$$\mathbf{W}_m^2 = \frac{\text{diag}(\langle \mathbf{m}_{\text{ref}} \rangle)}{\text{diag}(\langle \mathbf{A}' \mathbf{A} \mathbf{m}_{\text{ref}} \rangle) + \epsilon^2 \mathbf{I}}, \quad (7.5)$$

where ϵ is damping parameter that is related to the signal-to-noise level.

7.2.3 Numerical comparison

The Amoco 2.5-D synthetic dataset (Etgen and Regone, 1998; Dellinger et al., 2000) provides an excellent test for the weighting functions discussed above.

The velocity model (Figure 7.1) contains significant structural complexity in the upper 3.8 km, and a flat reflector of uniform amplitude at about 3.9 km depth. Since the entire velocity model ("Canadian foothills overthrusting onto the North Sea") is somewhat pathological, I restricted my experiments to the North Sea section of the dataset ($x > 10$ km). The data were generated by 3-D acoustic finite-difference modeling of the 2.5-D velocity model. However, making the test more difficult is the fact that the FFD one-way recursive extrapolators (Ristow

and Ruhl, 1994) that I use for modeling and migration do not accurately predict the 3-D geometric spreading and multiple reflections that are present in this dataset. Figure 7.2 illustrates this by comparing a gather produced by full two-way 3-D acoustic finite-differences with a gather modeled with the one-way depth extrapolation algorithm.

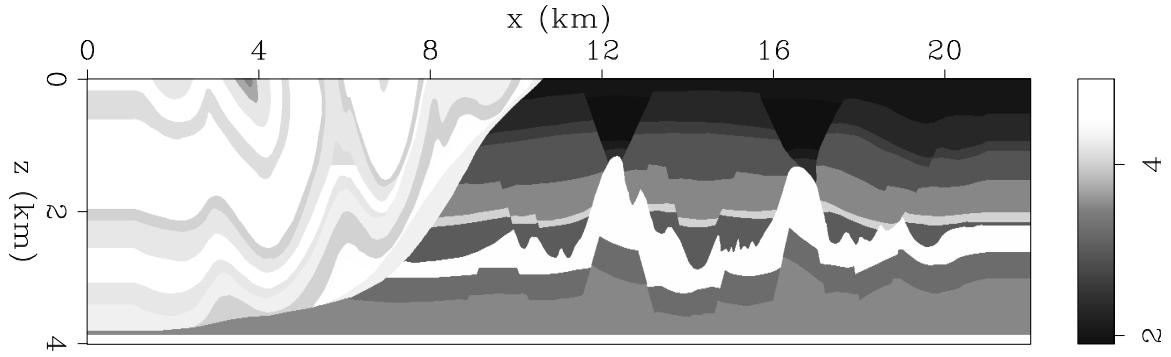


Figure 7.1: Velocity (in km/s) model used to generate the synthetic Amoco 2.5-D dataset.
wghts-amocovel [ER]

Figure 7.3 compares the migrated image (\mathbf{m}_1) with the results of remodeling and remigrating the three reference images described above. The imprint of the recording geometry is clearly visible on the three remigrations in Figures 7.3 (b-d).

Figure 7.4 compares the illumination calculated from the three reference images with the shot illumination from section 6.4. Noticeably, the shot-only weighting function [panel (a)] does not take into account the off-end (as opposed to split-spread) receiver geometry. Panel (b), the weighting function derived from model \mathbf{m}_1 , appears slightly noisy. However, in well-imaged areas (e.g. along the target reflector), the weighting function is well-behaved. Panel (c) shows the weighting function derived from the random reference image (\mathbf{m}_2). Despite the smoothing, this weighting function clearly bears the stamp of the random number field. A feature of white noise is that no amount of smoothing will be able to completely remove the effect of the random numbers. The final panel (d) shows the flat-event illumination weighting function, derived from \mathbf{m}_3 . This is noise-free and very well-behaved since it depends only on the velocity model and recording geometry, not the data.

For a quantitative comparison, I picked the maximum amplitude of the 3.9 s reflection

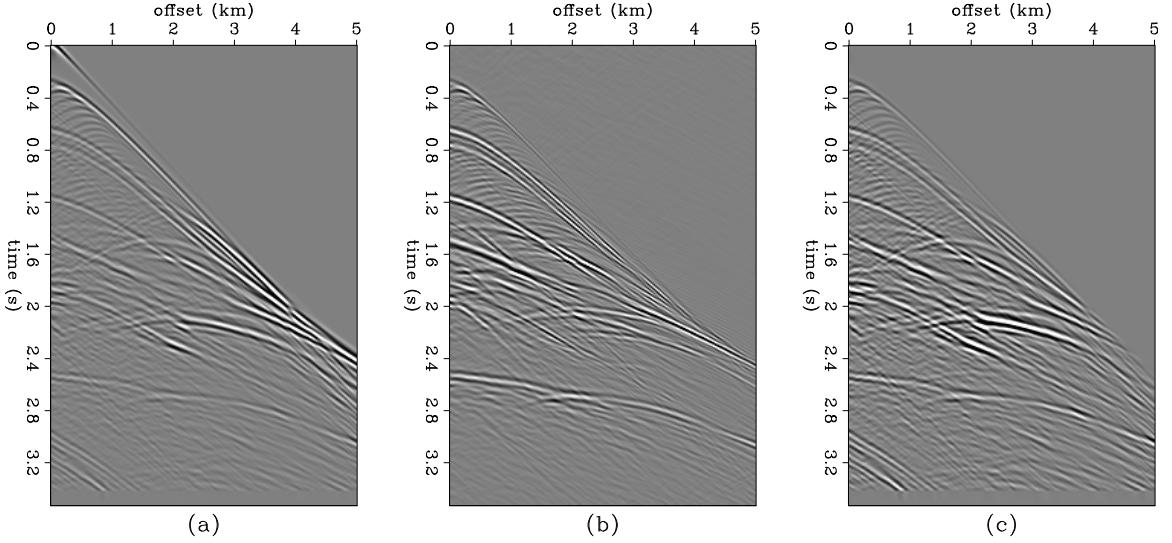


Figure 7.2: Synthetic shot-gathers from the Amoco 2.5-D dataset ($s_x = 18.2$ km). Panel (a) shows the gather generated by full two-way finite-difference modeling. Panel (b) shows the gather generated by linear one-way modeling. Panel (c) shows the same two-way equation gather as panel (a) but with a top-mute to enable easier comparison with panel (b).

wghts-amocogather18 [CR,M]

event on the calibrated images. The normalized standard deviation (NSD) of these amplitudes is shown in Table 7.1, where

$$\text{NSD} = \sqrt{\sum_{i_x} \left(\frac{a_{i_x}}{\bar{a}} - 1 \right)^2}. \quad (7.6)$$

Table 7.1, therefore, provides a measure of how well the various weighting function compensate for illumination difficulties. The amplitudes of the raw migration, and the migration after flat-event normalization are shown in Figure 7.5. This illustrates the numerical results from Table 7.1: for this model the normalization procedure improves amplitude reliability by almost a factor of two.

To compare the results of a well-scaled adjoint with full L^2 Fourier finite-difference migration, I ran 10 iterations of full conjugate gradients, using Paul Sava's out-of-core optimization library (Sava, 2001). Figure 7.6 shows images after four and ten iterations. I did not impose an explicit regularization ("model-styling") term during the inversion, so as the solution evolves

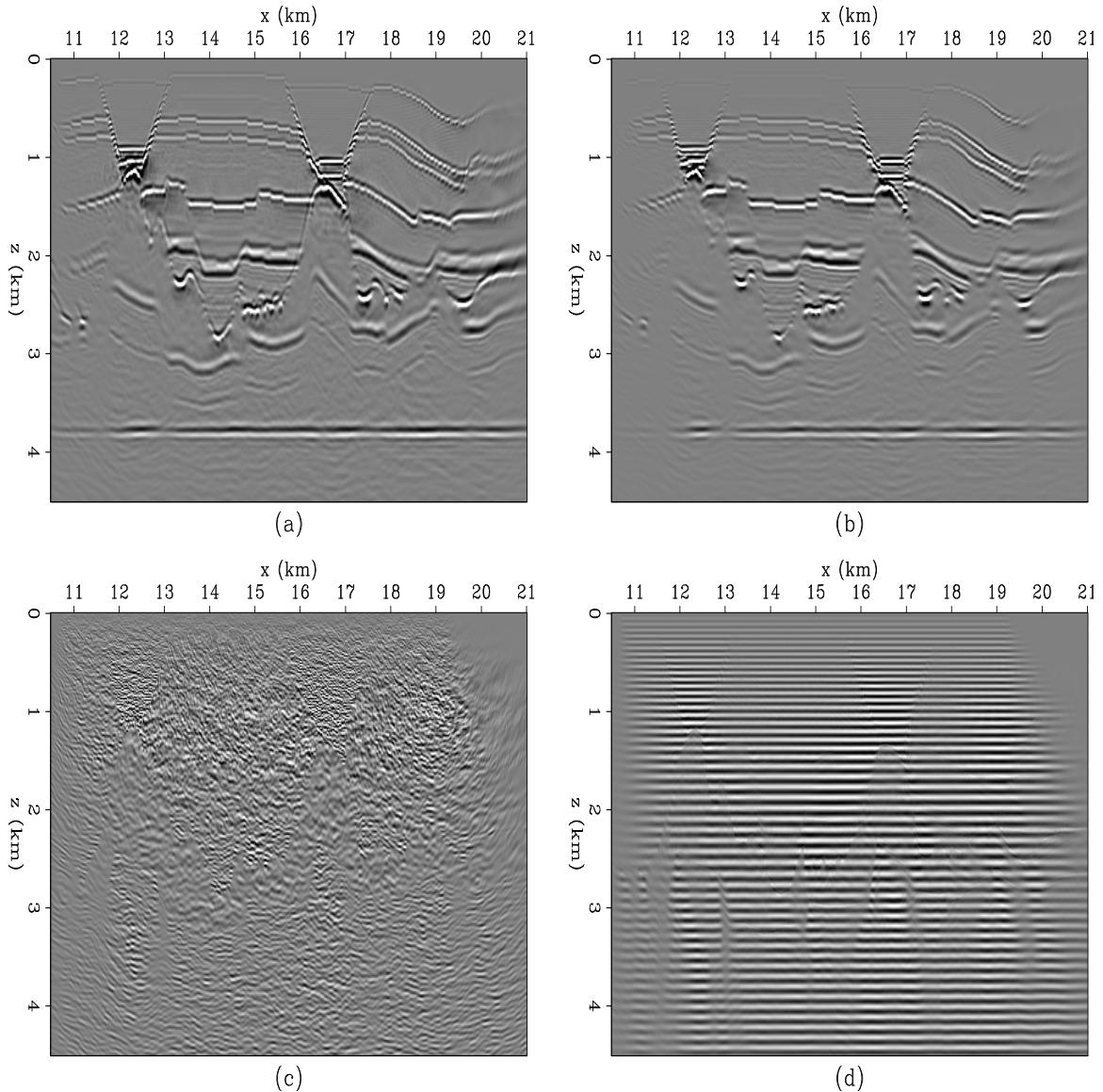


Figure 7.3: Comparison of calibration images: (a) original migration, (b) original migration after modeling and migration, (c) random image after modeling and migration, and (d) flat event image after modeling and migration. wghts-amocomigs [CR,M]

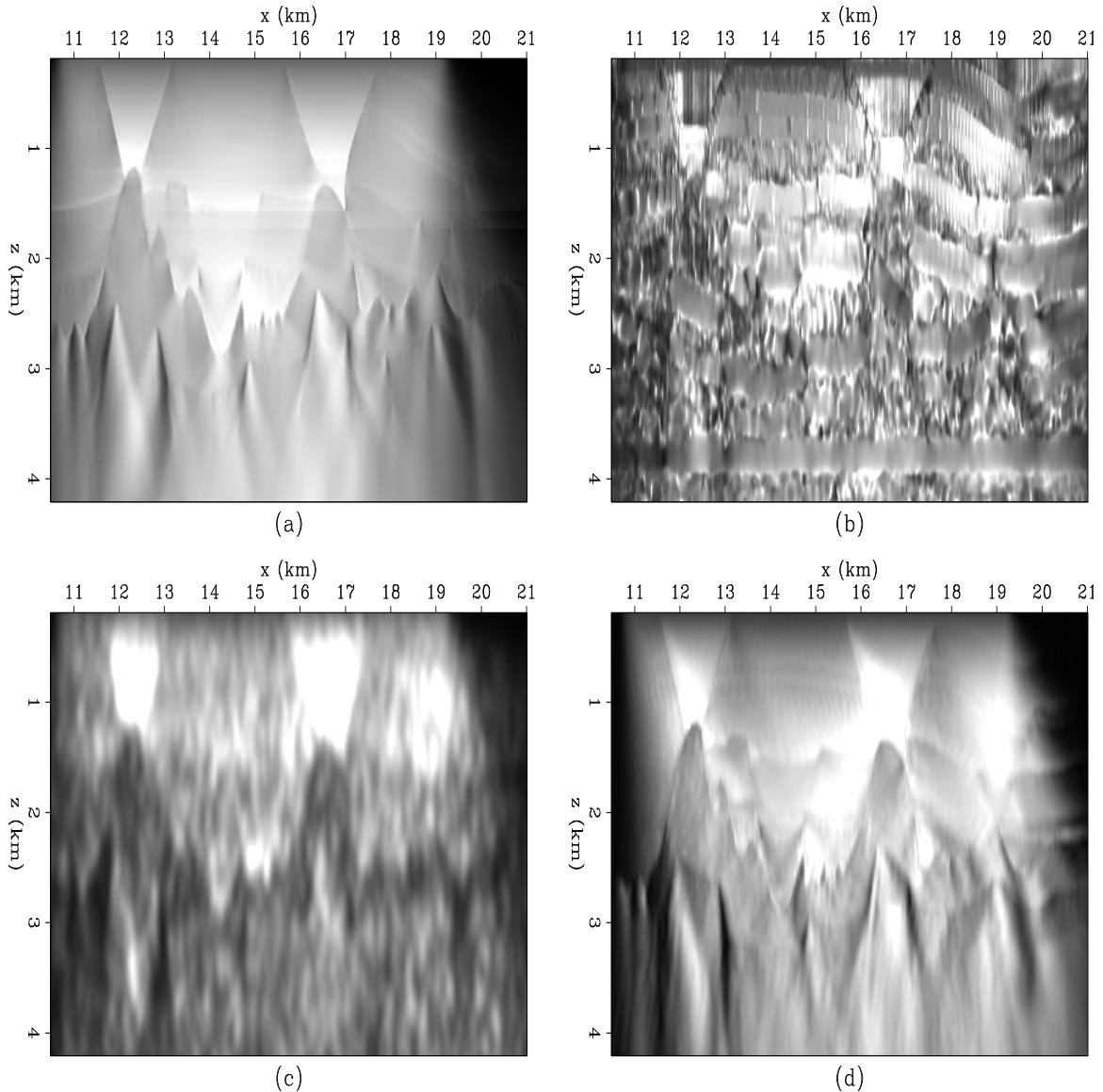
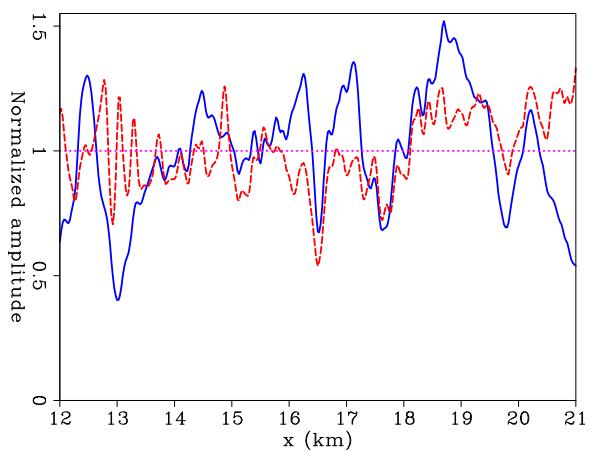


Figure 7.4: Comparison of weighting functions: (a) shot illumination as described in Chapter 6, (b) reference model was migrated image, (c) reference model was random image, and (d) reference model consisted of purely flat events. wgths-amocowght [CR,M]

Weighting function:	Normalized standard deviation:
No weighting function	0.229
Shot illumination	0.251
$\mathbf{m}_{\text{ref}} = \mathbf{m}_1$ (migrated image)	0.145
$\mathbf{m}_{\text{ref}} = \mathbf{m}_2$ (random image)	0.195
$\mathbf{m}_{\text{ref}} = \mathbf{m}_3$ (flat events)	0.140
Four iterations of CG	0.157

Table 7.1: Comparison of the reflector strength for different choices of illumination-based weighting function.

Figure 7.5: Normalized peak amplitude of 3.9 km reflector after migration (solid line), and then normalization by flat-event illumination (dashed-line). The ideal result would be a constant amplitude of 1.
 wghts-eventampnm5 [CR]



less well-constrained components of the model-space start to appear in the solution, including both low and high frequency noise and steeply-dipping energy. This causes the NSD to actually begin to increase after the fourth iteration (see Figure 7.7).

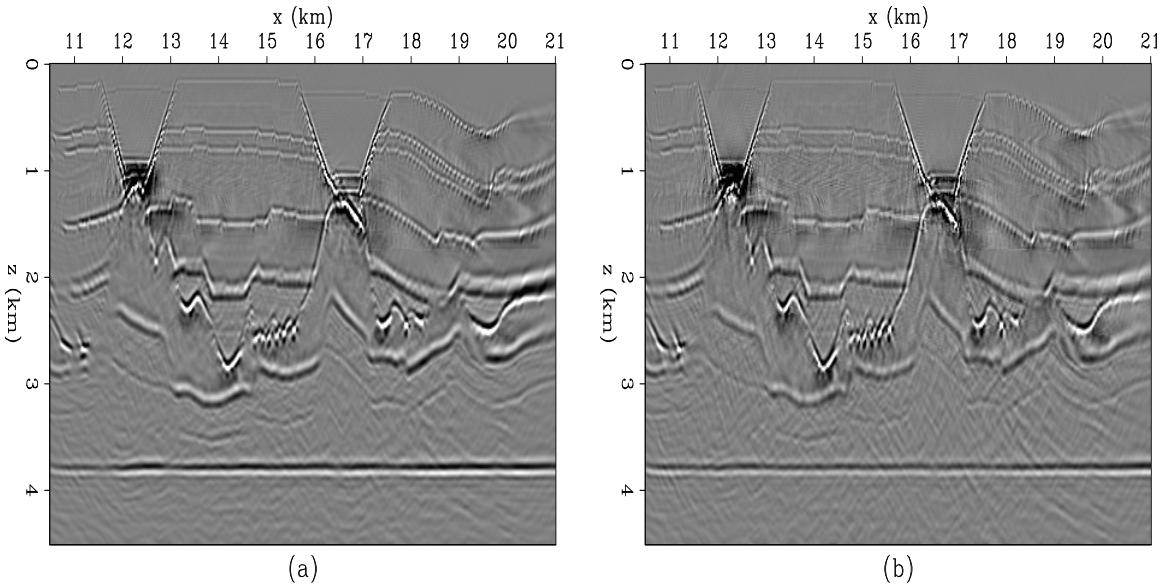


Figure 7.6: Results of full L^2 inversion of the Amoco 2.5-D dataset with FFD modeling/migration. Panel (a) shows results after four iterations, and panel (b) shows results after ten iterations. wghts-amocoinv [CR,M]

7.2.4 Computational cost

As it stands, the cost of computing a weighting function of this kind is twice the cost of a single migration. Add the cost of the migration itself, and this approach is 25% cheaper than running two iterations of conjugate gradients, which costs two migrations per iteration.

However, the bandwidth of the weighting functions is much lower than that of the migrated images. This allows considerable computational savings, as modeling and remigrating a narrow frequency band around the central frequency produces similar weighting functions to the full bandwidth. Repeating the first experiment ($\mathbf{m}_{\text{ref}} = \mathbf{m}_1$) with half the frequencies gives a $\text{NSD} = 0.147$ - the same as before within the noise-level of the experiment.

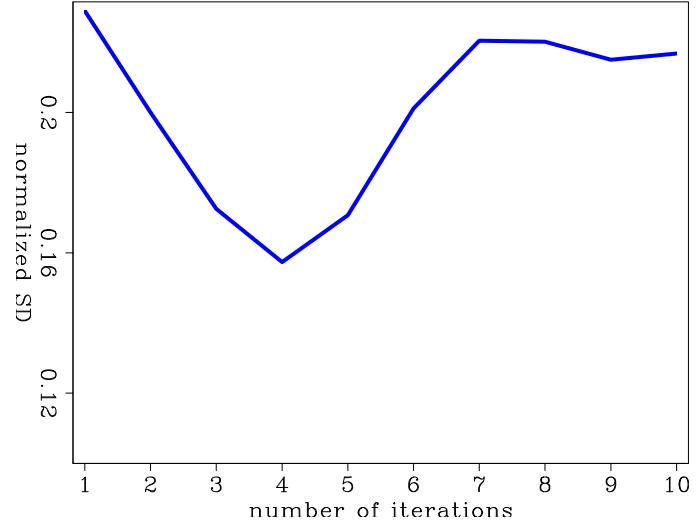


Figure 7.7: Normalized standard deviation of flat reflector versus iteration number. After four iterations the noise-level causes degradation of amplitude reliability. wghts-invnorm [CR]

7.3 Data-space weighting functions

If the estimation problem, $\mathbf{d} \approx \mathbf{A}\mathbf{m}$, is underdetermined, then a standard approach is to find the solution with the minimum norm. For the $L2$ norm, this is the solution to

$$\hat{\mathbf{m}}_{L2} = \mathbf{A}' (\mathbf{A}\mathbf{A}')^{-1} \mathbf{d}. \quad (7.7)$$

As a corollary to the methodology outlined above for creating model-space weighting functions, Claerbout (1998a) suggests constructing diagonal approximations to $\mathbf{A}\mathbf{A}'$ by probing the operator with a reference data vector, \mathbf{d}_{ref} . This gives data-space weighting functions of the form,

$$\mathbf{W}_d^2 = \frac{\text{diag}(\mathbf{d}_{\text{ref}})}{\text{diag}(\mathbf{A}\mathbf{A}' \mathbf{d}_{\text{ref}})} \approx \frac{1}{\mathbf{A}\mathbf{A}'}, \quad (7.8)$$

which can be used to provide a direct approximation to the solution in equation (7.7),

$$\hat{\mathbf{m}}_{L2} \approx \mathbf{A}' \mathbf{W}_d^2 \mathbf{d}. \quad (7.9)$$

Alternatively, we could use \mathbf{W}_d as a data-space preconditioning operator to help speed up

the convergence of an iterative solver:

$$\mathbf{W}_d \mathbf{d} = \mathbf{W}_d \mathbf{A} \mathbf{m}. \quad (7.10)$$

7.3.1 Combining model-space and data-space weighting functions

With two possible preconditioning operators, \mathbf{W}_m and \mathbf{W}_d , the question remains, what is the best strategy for combining them?

Chemingui (1999) calculated both \mathbf{W}_m and \mathbf{W}_d from operator fold and data fold respectively, but observed that because they both contain the inverse units to the operator, \mathbf{A} , he should not apply them both at once. Instead he combined the two preconditioning operators, and solved the system

$$\mathbf{W}_d^n \mathbf{d} = \mathbf{W}_d^n \mathbf{A} \mathbf{W}_m^{n-1} \mathbf{x} \quad (7.11)$$

$$\mathbf{m} = \mathbf{W}_m^{n-1} \mathbf{x}, \quad (7.12)$$

where $0 \leq n \leq 1$ is an adjustable parameter. Chemingui (1999) provided no advice on the choice of n , but for the problem he was solving, he observed that applying both \mathbf{W}_m and \mathbf{W}_d with $n = 1/2$ converged to a solution more rapidly than either end member ($n = 0$ or 1).

The first alternative strategy that I propose is to calculate a model-space weighting function, \mathbf{W}_m , and use it to create a new preconditioned system with the form of

$$\mathbf{d} = \mathbf{A} \mathbf{W}_m \mathbf{x} = \mathbf{B} \mathbf{x}.$$

Now probe the composite operator, \mathbf{B} , for a data-space weighting function for the new system,

$$\tilde{\mathbf{W}}_d^2 = \frac{< \mathbf{diag}(\mathbf{d}_{\text{ref}}) >}{< \mathbf{diag}(\mathbf{B} \mathbf{B}' \mathbf{d}_{\text{ref}}) > + \epsilon_d \mathbf{I}} \approx \frac{1}{\mathbf{B} \mathbf{B}'}. \quad (7.13)$$

The new data-space weighting function is dimensionless, and can be applied in consort with

the model-space operator. This leads to a new system of equations,

$$\tilde{\mathbf{W}}_d \mathbf{d} = \tilde{\mathbf{W}}_d \mathbf{A} \mathbf{W}_m \mathbf{x} \quad (7.14)$$

$$\text{with } \mathbf{m} = \mathbf{W}_m \mathbf{x}, \quad (7.15)$$

with appropriate model-space and data-space preconditioning operators. The adjoint solution to this system is given by

$$\mathbf{m} = \mathbf{W}_m^2 \mathbf{A}' \tilde{\mathbf{W}}_d^2 \mathbf{d}. \quad (7.16)$$

A second alternative strategy is the corollary of this: create a new system that is preconditioned by an appropriate data-space weighting function, and then calculate a model-space weighing function based on the new system. The first step is to calculate \mathbf{W}_d with equation (7.8), and set up a new system of equations,

$$\mathbf{W}_d \mathbf{d} = \mathbf{W}_d \mathbf{A} \mathbf{m} = \mathbf{C} \mathbf{m}. \quad (7.17)$$

The second step is to calculate a model-space weighting function based on this new operator,

$$\tilde{\mathbf{W}}_m^2 = \frac{< \mathbf{diag}(\mathbf{m}_{\text{ref}}) >}{< \mathbf{diag}(\mathbf{C}' \mathbf{C} \mathbf{m}_{\text{ref}}) > + \epsilon_m \mathbf{I}}. \quad (7.18)$$

The preconditioned composite system of equations is now

$$\mathbf{W}_d \mathbf{d} = \mathbf{W}_d \mathbf{A} \tilde{\mathbf{W}}_m \mathbf{x} \quad (7.19)$$

$$\text{with } \mathbf{m} = \tilde{\mathbf{W}}_m \mathbf{x}. \quad (7.20)$$

7.3.2 Numerical comparisons

Again, the Amoco 2.5-D dataset provides an excellent test dataset for comparing flavors of weighting function. Figure 7.8 shows the data-space weighting function for the gather shown in Figure 7.2 (a). Similarly, Figure 7.9 shows the weighting function for a common-offset section corresponding to a half-offset of 500 m.

Figure 7.8: Data-space weighting function for the gather shown in Figure 7.2 (a). Darker shades of grey correspond to higher amplitude.
 wghts-weightgather18 [CR,M]

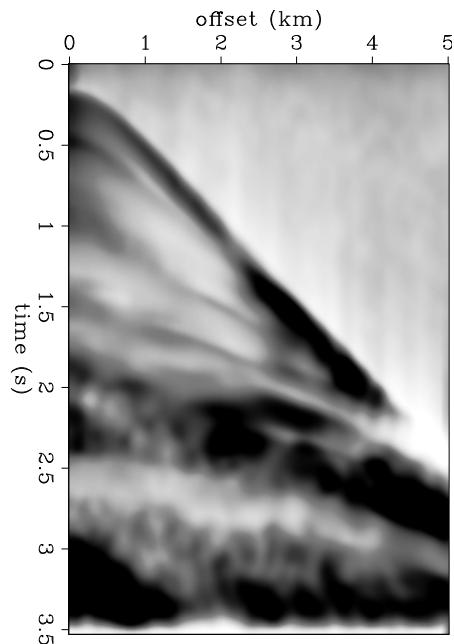
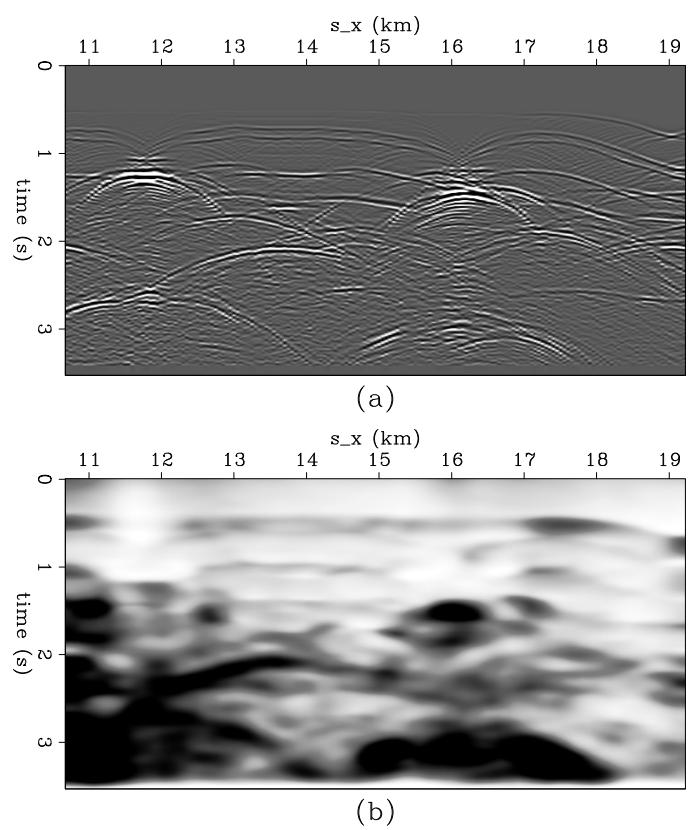
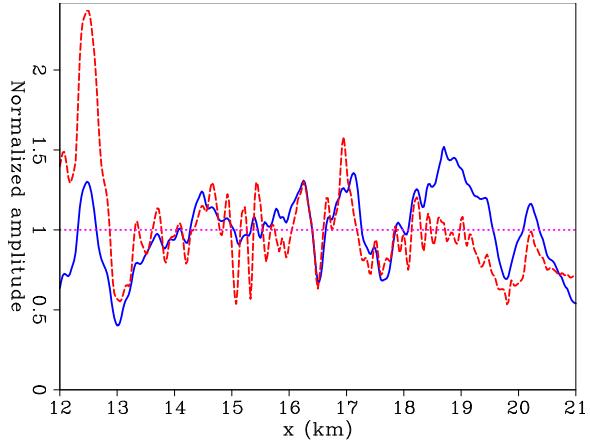


Figure 7.9: Common offset-section corresponding to $h = 500$ m (a), and its data-space weighting function (b). Darker shades of grey correspond to higher amplitude.
 wghts-weightsection1 [CR]



Unfortunately the data-space weights proved susceptible to coherent noise in the form of multiples not predicted by the modeling operator. While data-space weights did improve the signal in poorly illuminated areas, they also boosted up the noise level causing an *increase* in NSD. Figure 7.10 illustrates this with a plot to compare with Figure 7.5. While the reliability of the amplitudes are increases in some areas (notably around $x = 19$ km), in other areas, especially $x < 13$ km, there are obvious artifacts. Because of the large (non-Gaussian) outliers visible in Figure 7.10 the normalized standard deviation is no longer a meaningful measure of reliability.

Figure 7.10: Normalized peak amplitude of 3.9 km reflector after migration of the raw data (solid line), and after migration of a dataset that has been normalized by an appropriate data-space weighting function (dashed-line). Compare with Figure 7.5. [wghts-eventampnd5] [CR]



7.4 Discussion

Model-space weighting functions have shown to be robust in the presence of noise. However, for the examples in this chapter data-space weighting functions have tended to amplify poorly-modeled coherent noise. The question remains: why is this the case? This is especially unclear since Figure 6.4 suggests that a data-space weighting function is exactly what would be required to compensate for the illumination problems in that example.

A possible answer to this question comes from considering the relative sizes of model and data spaces for the example in this chapter and the example in Chapter 6. For the prestack

Amoco dataset, the data space is many times larger than the model space, leaving an over-determined problem. However, the formulation for the data-space weighting functions [equation (7.8)] is based on the underdetermined problem [equation (7.7)]. In contrast, model spaces and data spaces for the zero-offset example in Chapter 6 are approximately of equal size. Therefore the data-space weights may be more appropriate for that example.

7.4.1 Limitations

The methodologies described in this Chapter have several fundamental limitations. Most importantly, they require an accurate forward modeling operator: both the physics of wave propagation and the true earth velocity must be accurately modeled. While the physics of wave-propagation is well-understood, earth velocity models are never completely true-to-life.

Another important caveat is that the wave-equation methods outlined here require the data and models to be represented on a regular grid. While we can choose our model-space, prestack seismic data is never recorded on a perfectly regular grid. Before we can apply the techniques described here, we need to regularize the data. Chemingui (1999) and Fomel (2000) provide two ways of achieving this goal.

7.5 Conclusion

With appropriate choice of reference model, model-space weighting functions based on equation (7.4) provide a robust way to compensate for illumination problems during recursive (finite-frequency) depth migration. Data-space weights can also be calculated either to work alone, or in consort with model-space weights. However, for the overdetermined example described here, they are less robust to errors caused by inadequate forward modeling.

Appendix A

Non-stationary inverse convolution

In this Appendix, I show that non-stationary filtering and inverse filtering are true inverse processes. I also show that the stationary stability theory is not valid for non-stationary recursive inverse filtering: in general, the stability of non-stationary inverse filtering is not guaranteed, even if all possible filter-impulse responses are minimum phase.

A.1 Introduction

Previously, applications of non-stationary inverse filtering by recursion have been limited to problems in 1-D, such as time-varying deconvolution (Claerbout, 1998a). Theory presented no way of extending polynomial division to higher dimensions.

With the development of the helical coordinate system (Claerbout, 1998b), recursive inverse filtering is now practical in multi-dimensional space. Non-stationary, or adaptive (Widrow and Stearns, 1985), recursive filtering is now becoming an important tool for preconditioning a range of geophysical estimation (inversion) problems (Fomel et al., 1997; Clapp et al., 1997; Crawley, 1999), and enabling 3-D depth migration with a new breed of wavefield extrapolation algorithms discussed in Chapters 4 and 5.

With these applications in mind, it is important to understand fully the properties of non-stationary filtering and inverse-filtering. Of particular concern is the stability of the non-stationary operators.

I formulate causal non-stationary convolution and combination and their adjoints in such a way that it is apparent that the corresponding non-stationary recursive filters are true inverse processes. Stationary recursive inverse-filtering is stable if, and only if, the filter is minimum-phase. I show that recursive inverse-filtering with a filter-bank consisting of minimum-phase two-point filters is also unconditionally stable. However, I demonstrate that, for a more general set of minimum-phase filters, stability of non-stationary recursive inverse filtering is not guaranteed, even though stationary filtering with each individual filter is stable.

A.2 Theory

A.2.1 Stationary convolution and inverse convolution

The convolution of a vector, $\mathbf{x} = (x_0 \ x_1 \ x_2 \ \dots \ x_{N-1})^T$, with a causal filter, \mathbf{a} , whose first element, $a_0 = 1$, and whose length, $N_a < N$, onto an output vector, \mathbf{y} , can be defined by the set of equations:

$$y_k = x_k + \sum_{i=1}^{\min(N_a-1, k-1)} a_i x_{k-i}. \quad (\text{A.1})$$

This can be rewritten in linear operator notation, as $\mathbf{y} = \mathbf{Ax}$, where \mathbf{A} is a lower-triangular Toeplitz matrix representing convolution with the filter, \mathbf{a} .

The adjoint operator, \mathbf{A}' , which describes time-reversed filtering with filter \mathbf{a} , can similarly be expressed by considering the rows of the matrix-vector equation, $\mathbf{x} = \mathbf{A}'\mathbf{y}$, as follows,

$$x_k = y_k + \sum_{i=1}^{\min(N_a-1, N-1-k)} a'_i y_{k+i}. \quad (\text{A.2})$$

The `helicon` Fortran90 module (Claerbout, 1998a) exactly implements the linear operator (and adjoint) pair described by equations (A.1) and (A.2).

Equation (A.1) explicitly prescribes internal boundary conditions near $k = 0$; however, since \mathbf{a} is causal, no particular care is needed near $k = N - 1$. On the other hand, equation (A.2) explicitly imposes internal boundary conditions near $k = N - 1$, and no care is needed near $k = 0$. It is possible to rewrite equations (A.1) and (A.2) in a more symmetric form; however, as written, the equations lead naturally to recursive inverses for operators, \mathbf{A} and \mathbf{A}' .

Rearranging equation (A.1), we obtain

$$x_k = y_k - \sum_{i=1}^{\min(N_a-1, k-1)} a_i x_{k-i}. \quad (\text{A.3})$$

This provides a recursive algorithm, starting from $x_0 = y_0$ for solving the system of equations, $\mathbf{y} = \mathbf{Ax}$. Equation (A.3) describes the exact, analytic inverse of causal filtering with equation (A.1). In principle, given a filter, \mathbf{a} , and a filtered trace, \mathbf{y} , the above equation can recover the unfiltered trace, \mathbf{x} exactly; although in practice, with numerical errors, the division may become unstable if \mathbf{a} is not minimum phase. Similarly, if we inverse filter a trace with equation (A.3), we can recover the original by causal filtering with equation (A.1) subject to the stability of the inverse filtering process.

Equation (A.3) appears very similar to polynomial division. However, the output of polynomial division is an infinite series, while equation (A.3) is defined only in the range, $0 \leq k \leq N - 1$. As such, equation (A.3) describes polynomial division followed by truncation.

Equation (A.2) can also be rewritten as

$$y_k = x_k - \sum_{i=1}^{\min(N_a-1, N-1-k)} a'_i y_{k+i}, \quad (\text{A.4})$$

which provides an exact recursive inverse to adjoint operator, \mathbf{A}' , that can be computed starting from $y_{N-1} = x_{N-1}$, and decrementing k .

A.2.2 Non-stationary convolution and combination

There are several possible approaches to generalizing convolution described by equation (A.1) to deal with non-stationarity. The simplest approach (Yilmaz, 1987) is to apply multiple stationary filters and interpolate the results. This approach, however, gives incorrect spectral response in the interpolated areas (Pann and Shin, 1976).

Following Claerbout (1998a) and Margrave (1998), I extend the concept of a filter to that of a filter-bank, which is a set of N filters: one filter for every point in the input/output space. I identify \mathbf{a}_j with the filter corresponding to the j^{th} location in the input/output vector, and the coefficient, $a_{i,j}$, with the i^{th} coefficient of the filter, \mathbf{a}_j .

Margrave (1998) describes two closely related alternatives which both reduce to normal convolution in the limit of stationarity. The first approach is to place the filters in the columns of the matrix, \mathbf{A} . This associates a single filter with a single point in the output space, and defines *non-stationary convolution*:

$$y_k = x_k + \sum_{i=1}^{\min(N_a-1, k-1)} a_{i,(k-i)} x_{k-i}. \quad (\text{A.5})$$

In contrast, the second approach is to place individual filters in the rows of the matrix, \mathbf{A} , associating a single filter with a single point in the input space. This defines what Margrave (1998) refers to as *non-stationary combination*:

$$y_k = x_k + \sum_{i=1}^{\min(N_a-1, k-1)} a_{i,k} x_{k-i}. \quad (\text{A.6})$$

The advantage of non-stationary convolution over non-stationary combination is that the response of equation (A.5) to an impulse at the j^{th} element of \mathbf{x} , is \mathbf{a}_j . A more general output is then a scaled superposition of filter-bank filters, which fits with Green's function theory for linear, constant coefficient, partial differential equations. Again, in contrast, the response of equation (A.6) to an impulse at the j^{th} element of \mathbf{x} , is the j^{th} column of non-stationary combination matrix, which bears no direct relationship to the filter, \mathbf{a}_j , or any other individual filter for that matter.

As an illustration, consider the differences between matrices, \mathbf{F}_{conv} and \mathbf{F}_{comb} below, which represent, respectively, non-stationary convolution and combination with a causal three-point ($N_f = 3$) filter-bank, \mathbf{f} , to vectors of length, $N = 5$. The two are equivalent in the stationary limit; however, while the columns of \mathbf{F}_{conv} contain filters, \mathbf{f}_j , the columns of \mathbf{F}_{comb} do not.

$$\mathbf{F}_{\text{conv}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ f_{10} & 1 & 0 & 0 & 0 & 0 \\ f_{20} & f_{11} & 1 & 0 & 0 & 0 \\ 0 & f_{21} & f_{12} & 1 & 0 & 0 \\ 0 & 0 & f_{22} & f_{13} & 1 & 0 \\ 0 & 0 & 0 & f_{23} & f_{14} & 1 \end{bmatrix}, \quad \mathbf{F}_{\text{comb}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ f_{11} & 1 & 0 & 0 & 0 & 0 \\ f_{22} & f_{12} & 1 & 0 & 0 & 0 \\ 0 & f_{23} & f_{13} & 1 & 0 & 0 \\ 0 & 0 & f_{24} & f_{14} & 1 & 0 \\ 0 & 0 & 0 & f_{25} & f_{15} & 1 \end{bmatrix}$$

It is also clear that while \mathbf{F}_{conv} and \mathbf{F}_{comb} are related, they are not simply adjoint to each other.

A.2.3 Adjoint non-stationary convolution and combination

The adjoint of non-stationary convolution can be written as

$$x_k = y_k + \sum_{i=1}^{\min(N_a-1, N-k-1)} a'_{i,k} y_{k+i}, \quad (\text{A.7})$$

and the adjoint of non-stationary combination can be written as

$$x_k = y_k + \sum_{i=1}^{\min(N_a-1, N-k-1)} a'_{i,(k+i)} y_{k+i}. \quad (\text{A.8})$$

For many applications, the adjoint of a linear operator is the same operator applied in a (conjugate) time-reversed sense. For example, causal and anti-causal filtering, integration, differentiation, upward and downward continuation, finite-difference modeling and reverse-time migration etc.

Inverse NS convolution:	$x_k = y_k - \sum_{i=1}^{\min(N_a-1, k-1)} a_{i,(k-i)} x_{k-i}$	(A.9)
Inverse NS combination:	$x_k = y_k - \sum_{i=1}^{\min(N_a-1, k-1)} a_{i,k} x_{k-i}$	(A.10)
Adjoint inverse NS convolution:	$y_k = x_k - \sum_{i=1}^{\min(N_a-1, N-1-k)} a'_{i,k} y_{k+i}$	(A.11)
Adjoint inverse NS combination:	$y_k = x_k - \sum_{i=1}^{\min(N_a-1, N-1-k)} a'_{i,(k+i)} y_{k+i}$	(A.12)

Table A.1: Recursive formulae for non-stationary (NS) inverse operators.

For non-stationary filtering, it is important to realize this is *not* the case: the adjoint of non-stationary convolution is time-reversed non-stationary combination, and vice-versa. Therefore, the output of adjoint combination is a superposition of scaled time-reversed filters, \mathbf{a}'_j . So for anti-causal non-stationary filtering, it may be advantageous to apply adjoint combination, as opposed to adjoint convolution.

A.2.4 Inverse non-stationary convolution and combination

As with the stationary convolution described above, formulae for non-stationary recursive inverse convolution and combination follow simply by rearranging the equations in (A.5) and (A.6). Similarly, their adjoints can be obtained by rearranging the equations in (A.7) and (A.8). The recursive formulae describing these inverse processes are given in Table 1.

As with the stationary inverse convolution described above, it is apparent that subject to numerical errors, non-stationary inverse filtering with these equations in Table 1 is the exact, analytic inverse of non-stationary filtering with the corresponding forward operator described in equations (A.5) through (A.8): they are true inverse processes. If operator **A** represents filtering with a non-stationary causal-filter, and **B** represents recursive inverse filtering with

the same filter then

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I} \quad \text{and} \quad \mathbf{A}'\mathbf{B}' = \mathbf{B}'\mathbf{A}' = \mathbf{I}.$$

The `nhelicon` module (Claerbout, 1998a) implements the non-stationary combination operator/adjoint pair, described by equations (A.6) and (A.8), while `npolydiv` implements the corresponding inverse operators, described by equations (A.10) and (A.12).

A.2.5 The stability of non-stationary inverse filtering

A filter is stable if any bounded input produces a bounded output (Robinson and Treitel, 1980). Therefore, to prove that inverse filtering with a class of filters is stable, we have to demonstrate that all possible members of the class have bounded outputs for all bounded inputs. On the other hand to show that a class of filters is not stable, we just need to find a single example where a bounded input produces an unbounded output.

The stability of stationary recursive inverse filtering depends on the phase of the causal filter: if (and only if) the filter is minimum phase, then its inverse filter is stable. This raises the question: is non-stationary inverse filtering stable if all filters contained in the filter-bank are minimum-phase?

For the case of inverse filtering with a two-point filter ($N_a = 2$), equation (9) reduces to $x_0 = y_0$, and the following formula for $k > 0$:

$$x_k = y_k - a_{1,(k-1)}x_{k-1}. \quad (\text{A.13})$$

Recursive substitution then produces an explicit formula for elements of \mathbf{x} in terms of elements of \mathbf{y} :

$$x_N = y_N - a_{1,(N-1)}y_{N-1} + a_{1,(N-1)}a_{1,(N-2)}y_{N-2} \dots + (-1)^N \left(\prod_{i=0}^{N-1} a_{1,i} \right) y_0. \quad (\text{A.14})$$

For stability analysis, we need to understand how the above series behaves as $N \rightarrow \infty$. If the

filters, \mathbf{a}_i , are all minimum phase, and there exists a real number, κ , such that $|a_{1,i}| \leq \kappa < 1$ for all i , then

$$\prod_{i=0}^{N-1} a_{1,i} \leq \kappa^N. \quad (\text{A.15})$$

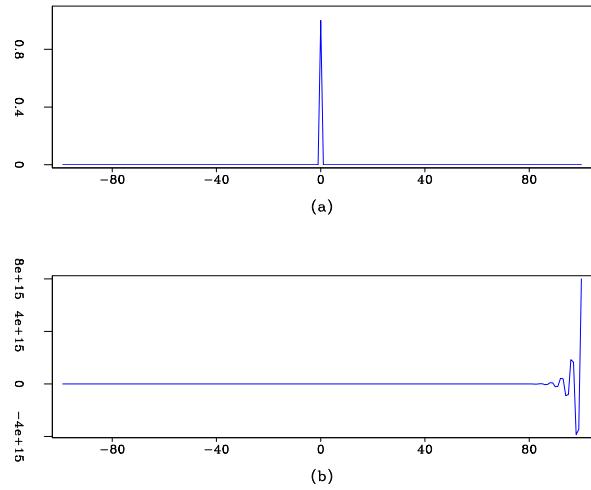
The above series will therefore converge, and stability is guaranteed. Furthermore, this proof can easily be extended to gapped two-point minimum-phase non-stationary filters, which correspond to matrices with ones on the main diagonal, and variable coefficients whose magnitude is less than one on a secondary diagonal.

There is a larger class ($N_a \geq 3$) of stable non-stationary recursive filters that can be obtained by repeatedly multiplying stable bidiagonal matrices. However, given a general non-stationary filter matrix, there is no straightforward way to determine whether it is a member of this stable class. In fact, it is relatively easy to find an example filter-bank consisting of minimum-phase individual filters whose recursive output is unbounded for finite input. Consider the filter-bank, \mathbf{f} , consisting of minimum-phase filters,

$$\begin{aligned} \mathbf{f}_{0,2,4\dots} &= (1 \ -0.9 \ 0), \text{ and} \\ \mathbf{f}_{1,3,5\dots} &= (1 \ 0.8)*(1 \ 0.8) = (1 \ 1.6 \ 0.64). \end{aligned} \quad (\text{A.16})$$

Figure A.1 shows the impulse response of non-stationary inverse filtering with this filter: clearly an unstable process.

Figure A.1: Impulsive input (a) and response (b) to non-stationary filtering with filter-bank defined in equation (A.16). **tvdecon-one [ER]**



The instability stems from the fact that as N increases, so does the number of boundaries between different filters. Such rapid non-stationary filter variations, as in the example above, are pathological in the context of seismic applications, where filters are typically quasi-stationary. For these applications instability is rarely observed; however, we must be aware that we are not dealing with an unconditionally stable operator, and instability may rear its ugly head from time-to-time.

A.3 Conclusions

There are three important points to draw from this Appendix. Firstly, I have formulated causal non-stationary convolution and combination and their adjoints in such a way that it is apparent that the corresponding non-stationary recursive filters are true inverse processes. If you think of causal non-stationary filtering as a lower triangular matrix, then recursive inverse filtering applies the inverse matrix.

The second important point is that recursive inverse-filtering with a filter-bank consisting of minimum-phase two-point filters is *unconditionally stable*, and as such it is totally safe to apply in any circumstance.

However, the final point is that for a more general set of minimum-phase filters, stability of non-stationary recursive inverse-filtering is *not* guaranteed: use with care.

Appendix B

Common-image gathers for shot-profile migration

In order to estimate elastic parameters of the subsurface, geophysicists need reliable information about angle dependent reflectivity. In this Appendix, I discuss how imaging non-zero offsets amounts to estimating off-diagonal elements of Berkhouit's reflectivity matrix. The information in these offset domain common-image gathers (CIGs) can then be transformed to the angle domain, and potentially used for lithologic inversion. CIGs also contain information about how well focused events are at depth, and so provide a natural domain for migration-focusing velocity analysis.

B.1 Introduction

For simple velocity models (e.g. constant v) a one-to-one mapping exists between recording offset ($|\vec{s} - \vec{g}|$) on the surface and incidence angle. Early amplitude versus offset (AVO) studies took advantage of this fact: authors looked at amplitude variation of events in common-midpoint gathers after normal moveout (NMO). It soon became clear, however, that NMO does not adequately correct for the focusing aspects of wave propagation, and that amplitude studies should be conducted after migration (Resnick et al., 1987).

Kirchhoff migration algorithms can easily migrate data recorded at different offsets separately, and produce common-image point (CIP) gathers with migrated amplitude versus offset. Alternatively, interpreters often compare amplitude on near-offset migration images with far-offset images to look for anomalies linked to the presence of hydrocarbons. However, offset on the surface is a *data-space* parameter, whereas incidence angle is a *model-space* parameter; even for many relatively simple $v(z)$ velocity models, no simple relationship exists between surface offset and incidence angle at the reflector.

For Kirchhoff amplitude versus angle (AVA) studies, all offsets need to be migrated, and the incidence angle determined at the imaging step by ray-tracing. Two disadvantages of this process are that it is sensitive to the stability of the ray-paths, and complicates the algorithm's I/O requirements.

de Bruin et al. (1990) first described how to calculate angle-dependent reflectivity with a wave-equation migration process. Their algorithm is appropriate for shot-profile algorithms, and involves a local slant-stack as part of the image-forming step. Similarly, Prucha et al. (1999) describe how to extract angle domain common-image-point gathers during shot-geophone migration in the offset-midpoint domain. Since these two migration methods are based on finite-frequency solutions to the wave-equation, there is no sensitivity to ray-traced angles. There is also no significantly increased I/O burden over zero-offset imaging with the same algorithm other than the need to output another model-space axis.

After wave equation migration, offset no longer refers to the recording offset at the surface, but rather refers to the subsurface offset between upgoing and downgoing wavefields. Offset after migration is therefore a model-space rather than a data-space parameter. Figure B.1 graphically illustrates the meaning of offset before, during, and after migration.

For shot-geophone migration in the offset-midpoint domain, Sava and Fomel (2000) provide an algorithm that converts CIGs in the offset domain to angle gathers. The authors offer this as an alternative to Prucha et al.'s approach which requires extracting offset ray-parameter gathers as part of the migration imaging condition.

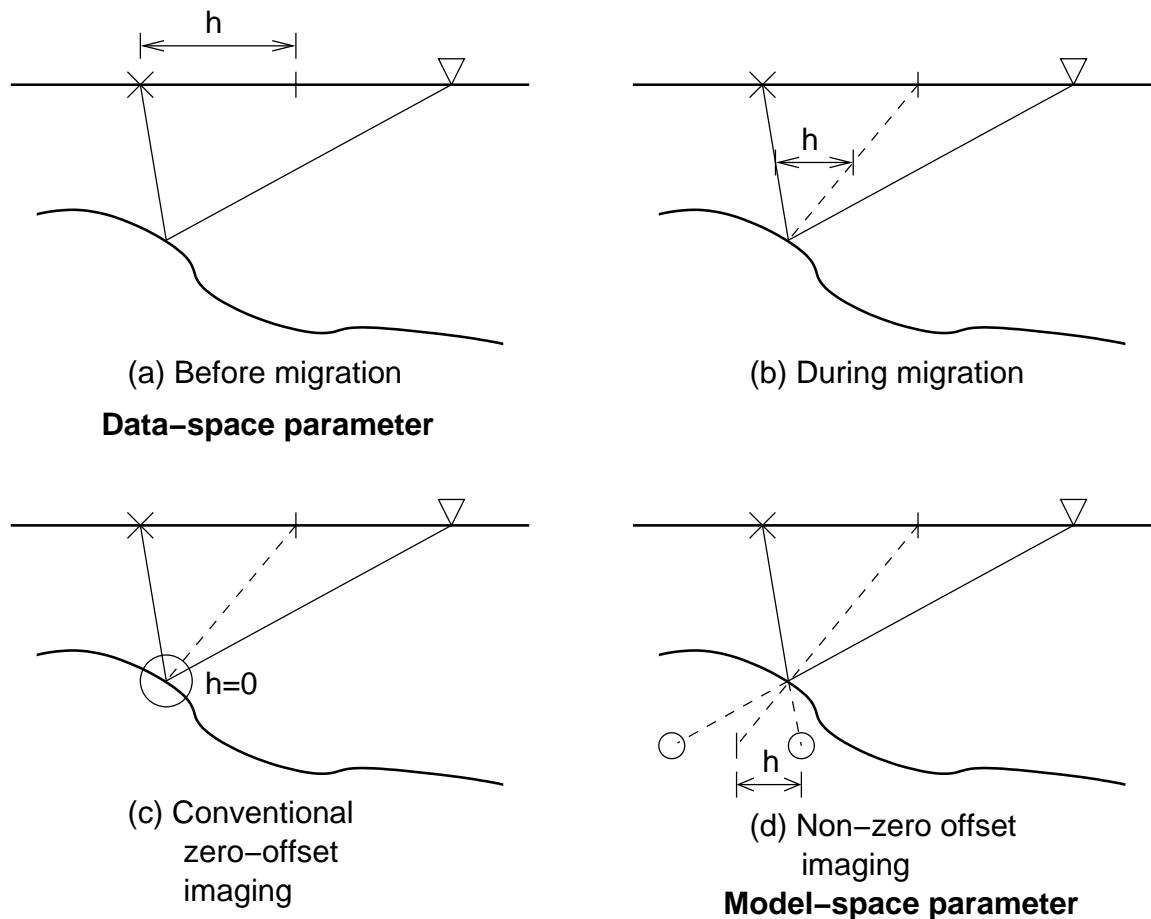


Figure B.1: Illustration of offset before (a), during (b), and after (c,d) migration.
 cigs-offsetexp [NR]

B.2 Multi-offset imaging for shot-profile migration

I produce angle gathers for shot-profile migrations by combining de Bruin et al.'s (1990) approach with that of Sava and Fomel (2000). Rather than extracting a single zero-offset/zero-time reflectivity image, I extract multiple zero-time images with a range of offsets. In 2-D this can be performed with the following sum over frequency:

$$I(x, h, z) = \sum_{\omega} q_{-}(x - h, z, \omega) q_{+}(x + h, z, \omega)^{*}. \quad (\text{B.1})$$

Gathers produced this way contain off-diagonal elements of Berkhout's reflectivity matrix (1985), and are equivalent to those produced by imaging multiple non-zero offsets in an offset-midpoint shot-geophone migration. Consequently, the offset axes can be mapped to angle with Sava and Fomel's (2000) transformation, which is based on the relationship

$$\tan \gamma = - \left. \frac{\partial z}{\partial h} \right|_{t,x} = - \frac{|\vec{k}_h|}{k_z}, \quad (\text{B.2})$$

where γ is the half-opening angle. The imaging condition ($t = 0$) provides the constant t , and the common midpoint nature of CIGs provides the constant x , allowing γ to be obtained directly from CIGs.

B.2.1 Shot to midpoint transformations

Implicit in the multi-offset imaging scheme outlined above is a transformation from shot-receiver to midpoint-offset space. Although this transformation is buried within the migration process, the subtleties associated with the conversion remain.

If the wavefields are sampled with spacing Δx , then equation (B.1) will image with half-offset spacing $\Delta h = \Delta x$, as shown in Figure B.2. Sampling in offset can be refined further by considering Figure B.3; however, to do so requires imaging onto midpoints which do not lie on the propagation grid.

Figure B.2: Imaging offsets with $\Delta h = \Delta x$ based on equation (B.1) alone. **cigs-cmpeven** [NR]

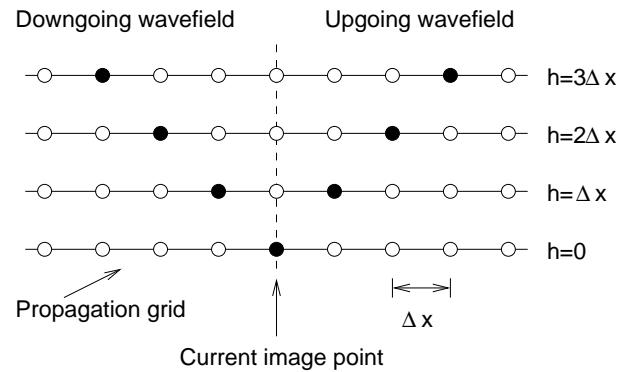
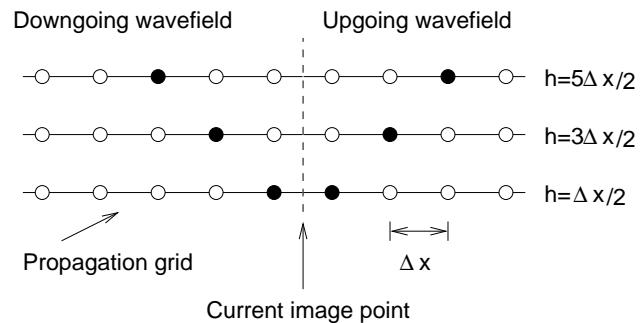


Figure B.3: Imaging offsets with $\Delta h = \Delta x/2$. The midpoint lies between propagation grid nodes. **cigs-cmpodd** [NR]



This problem is experienced whenever data are transformed from shot-geophone to mid-point-offset space, and no perfect solution exists. A typical workaround is to refine the mid-point grid, and fill empty bins with zeros; however, this doubles the data-volume and hence also doubles the cost of migration. Another alternative is to process even and odd offset separately; the disadvantage of this approach is that each half of the dataset may be undersampled.

Since shot-profile migration works in the shot-geophone domain, these problems may be avoided until after the migration is complete. Migration decreases the data-volume, increases the signal-to-noise ratio, and resolves locally conflicting dips. Therefore, it is easier to resample the data on whatever grid suits the interpreter after migration.

B.2.2 Computational considerations

As discussed in Chapter 6, the cost of downward continuing a 3-D wavefield one depth step in the Fourier domain is proportional to $N_{xy} \log(N_{xy})$. On the other hand, the cost of imaging with equation (B.1) is proportional to $N_{xy} N_{h_{xy}}$. Clearly, if the number of offsets is large, then the cost of the imaging step may be comparable (or even greater) than the cost of the downward continuation.

This increase in computational cost is not a problem with shot-geophone migration. In that instance, the cost of the imaging step (proportional to $N_{xy} N_{h_{xy}}$), will always be less than the cost of the downward continuation [proportional to $N_{xy} N_{h_{xy}} \log(N_{xy} N_{h_{xy}})$].

B.3 Interpretation of common-image gathers

As with AVO studies, the rock-physics interpretation of angle-domain CIGs after shot-profile migration is fraught with pitfalls: two of the most important being caused by velocity problems and sampling problems.

The first major pitfall is common to all angle-dependent reflectivity studies: if events are not correctly imaged, they will not be flat in the angle domain, and artifacts will appear in AVA interpretations. For these methods to be successful, velocities must be very well understood.

While the second major pitfall may also cause problems for other migration algorithms, geometries suitable for shot-profile migration are particularly susceptible to sampling problems on the shot axis.

B.3.1 Effect of velocity on common-image gathers

Figure B.5 illustrates the effect velocity plays on offset-domain common-image gathers. The three panels show CIGs produced by migrating the Marmousi synthetic dataset with three different velocity models: the correct velocity [panel (a)], a velocity that is too low [panel (b)], and a velocity model that is too high [panel (c)]. Interpreting patterns in the offset-domain CIGs is difficult; however, after transformation to the angle domain (Figure B.6, standard residual-moveout patterns indicates the sign of the velocity error: events curving up meaning too low, and events curving down meaning too high.

B.3.2 Effect of shot sampling on common-image gathers

As discussed previously in Chapter 6, datasets collected with sparse-shot geometries are most suitable for shot-profile migration. Figure B.4 compares the migrated (zero-offset) images of the two prestack datasets. The dataset that produced panel (a) had a fully-sampled shot axis, while the shot axis on the dataset that produced panel (b) contained only every twentieth shot. Even with the very sparsely-sampled shot axis, the geologic structure is clearly interpretable in Figure B.4 (b).

To illustrate the problems associated with sampling for shot-profile migrations, Figure B.7 shows the same image gathers as Figure B.5, but after migrating only every twentieth shot. Energy is no longer concentrated around zero-offset.

Figure B.8 (a) shows the equivalent picture in the angle domain. Even if though the velocity is correct and the zero-offset image [Figure B.4 (b)] seems reasonable, coverage in the angle-domain is very irregular. When the velocities are incorrect, the angle gathers remain chaotic: shot aliasing has effectively rendered the angle-gathers uninterpretable in terms of velocities.

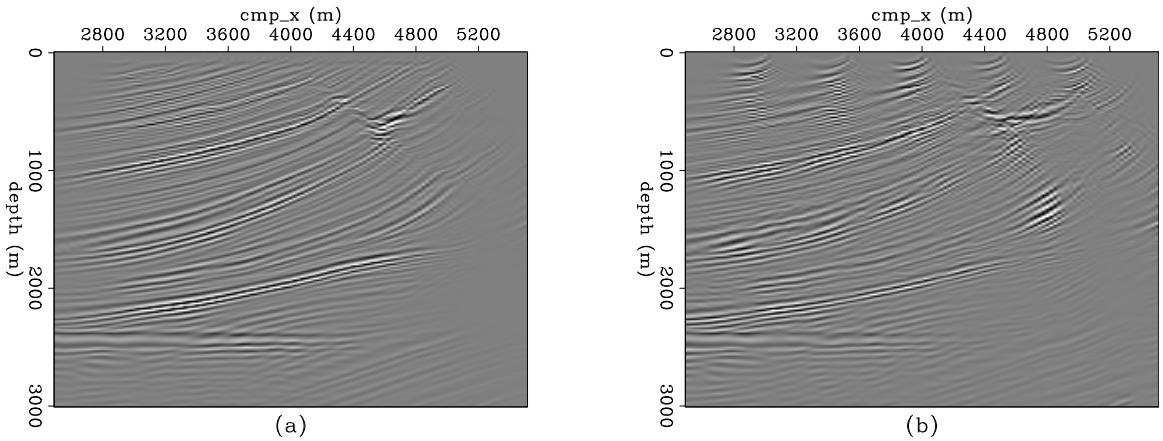


Figure B.4: Migrated images produced with data having a fully-sampled shot axis (a), and a very sparsely-sampled shot axis (b). cigs-zoff [ER,M]

Although both de Bruin's (1990) original methodology and the approach described here provide means of obtaining common-image gathers from shot-profile migration, the problem of shot aliasing remains important for the geometries that are best suited to shot-profile migration.

B.4 Conclusions

I construct offset-domain common-image gathers after shot-profile migration. The offset-domain gathers then can be transformed to the angle domain by Sava and Fomel's (2000) transformation where residual moveout can be used for velocity analysis or image enhancement. The angle domain also potentially contains information about reflectivity as a function of angle that can be used to infer rock properties across the reflecting interface.

For sparse-shot geometries suitable for shot-profile migration, however, the problem of shot aliasing remains. Shot aliasing can cause corruption of the angle domain gather, rendering simple moveout-based velocity analysis very difficult.

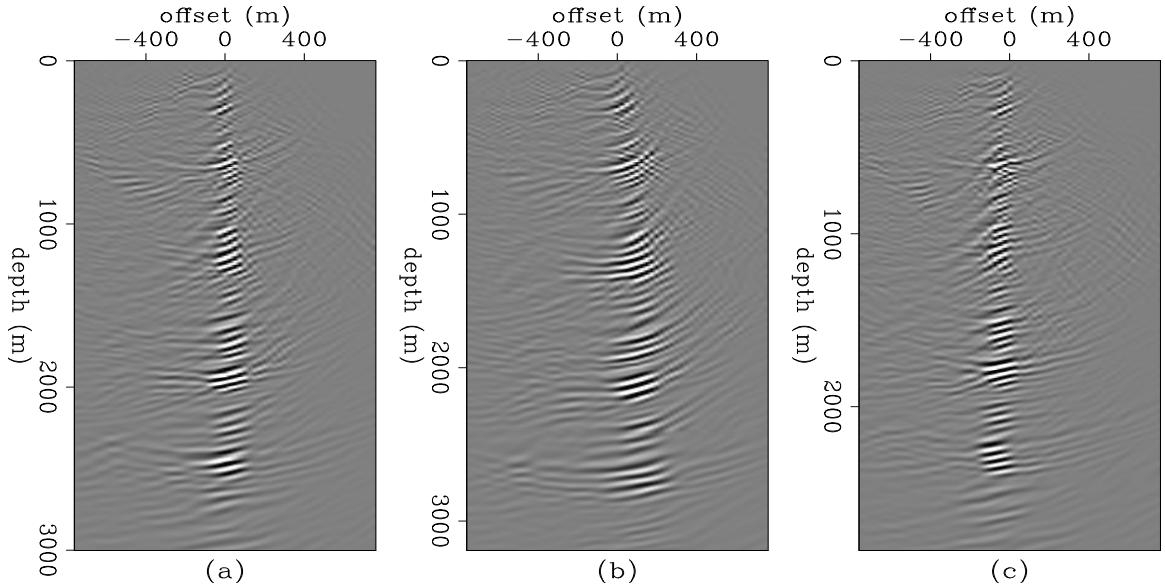


Figure B.5: Offset-domain common-image gathers corresponding to $x = 4000$ m. Panels (a), (b), and (c) were migrated with velocity models that were correct, 6% too low, and 6% too high, respectively. `cigs-offvel4000 [CR,M]`

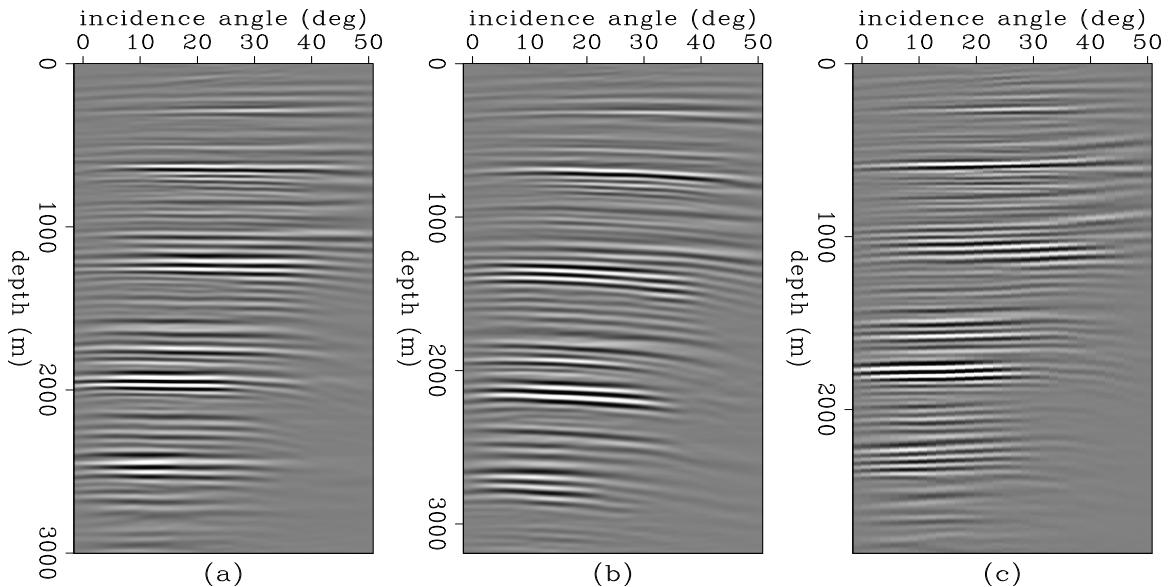


Figure B.6: Angle-domain common-image gathers corresponding to $x = 4000$ m. Panels (a), (b) and (c) were migrated with velocity models that were correct, too low, and too high, respectively. `cigs-angvel4000 [CR,M]`

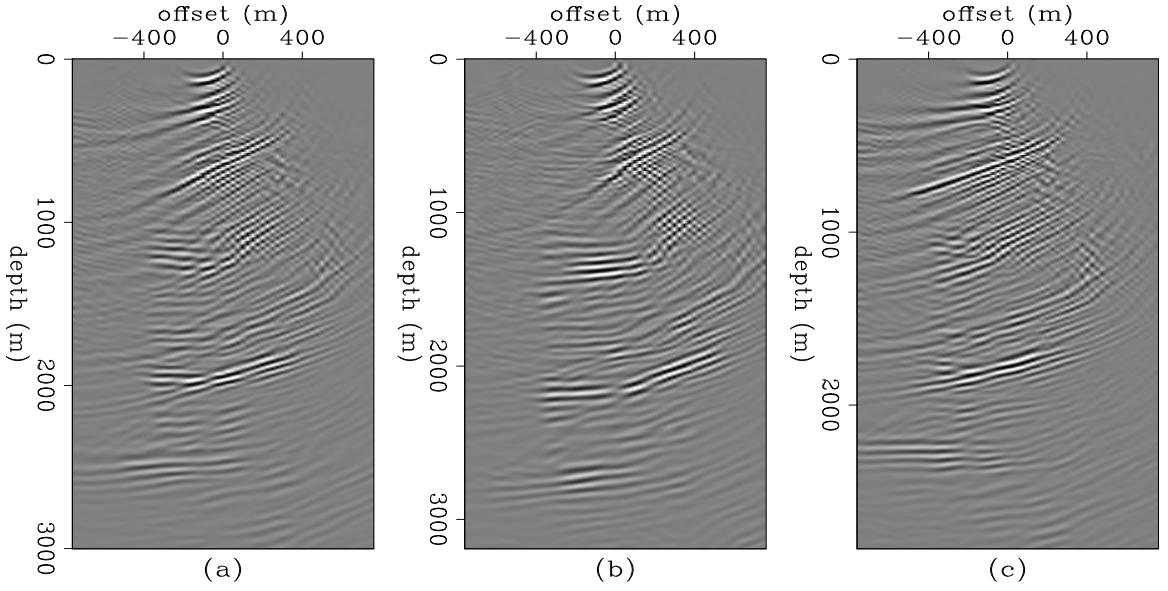


Figure B.7: Poorly illuminated offset-domain common-image gathers corresponding to $x = 4000$ m. Panels (a), (b), and (c) were migrated with velocity models that were correct, 6% too low, and 6% too high, respectively. Shot spacing was 500 m, instead of 25 m for Figure B.5.

cigs-offcov4000 [ER,M]

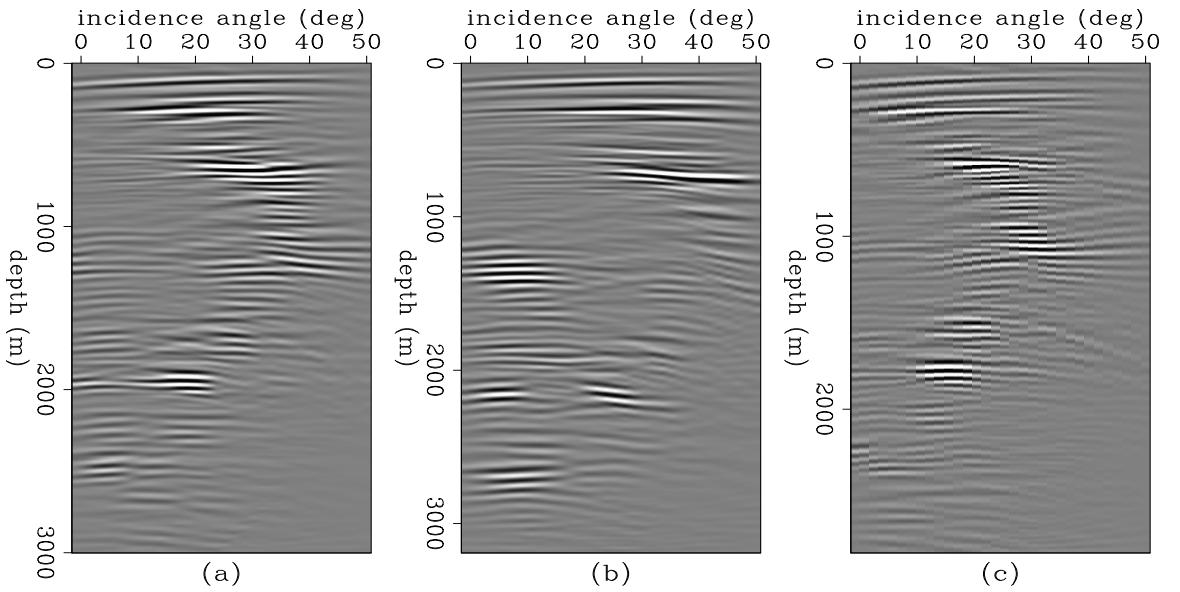


Figure B.8: Poorly illuminated angle-domain common-image gathers corresponding to $x = 4000$ m. Panels (a), (b) and (c) were migrated with velocity models that were correct, 6% too low, and 6% too high, respectively.

cigs-angcov4000 [ER,M]

Appendix C

Normalization by operator fold

C.1 Introduction

Various authors have discussed approaches for normalizing Kirchhoff-style operators based on principles such as operator fold [e.g. Beasley and Klotz (1992); Slawson et al. (1995); Chemingui (1999)]. While these concepts are based on reasonable heuristic ideas, it is difficult to identify exactly when these approximations are valid in the context of depth migration. Equation (7.4) provides a methodology for calculating weighting functions that are appropriate for normalizing any generic linear operators (Claerbout and Nichols, 1994). However, this approach has the advantage that we know it will provide the *ideal* weighting function if we have the correct forward modeling operator, and $\mathbf{m}_{\text{ref}} = \mathbf{m}_{\text{true}}$.

Operator fold is not well-defined for finite-frequency recursive migration operators; however, it is closely connected to the ideas developed in Chapter 7. In this Appendix, I identify explicitly how Kirchhoff fold normalization relates to equation (7.4).

C.2 Normalization of interpolation operators

Operator fold is defined as the impulse response of the operator to an input vector full of ones.

To understand this concept more clearly, it is best to begin with a very simple operator: nearest neighbour interpolation (\mathbf{A}_{NNI}) and its adjoint (binning). For this operator, the Hessian matrix, $\mathbf{A}'_{\text{NNI}} \mathbf{A}_{\text{NNI}}$, is exactly diagonal, and it would be possible to calculate the diagonal values accurately with *any* reasonable choice of reference model in equation (7.4). A particularly simple choice of reference model is to fill the model-space with ones. Since the operator is nearest-neighbour interpolation, this results in a data-space full of ones too. Binning this data vector gives the fold of the operator in model-space (Claerbout, 1998a); and its inverse can be used directly as a weighting function for inverse nearest neighbour interpolation,

$$\mathbf{diag}(\mathbf{A}'_{\text{NNI}} \mathbf{1})^{-1} \equiv (\mathbf{A}'_{\text{NNI}} \mathbf{A}_{\text{NNI}})^{-1}, \quad (\text{C.1})$$

where $\mathbf{1} = (1 \ 1 \ 1 \ \dots \ 1)^T$.

For an example that is slightly more complex than nearest neighbour interpolation, consider linear interpolation. The Hessian matrix for linear interpolation, $\mathbf{A}'_{\text{LI}} \mathbf{A}_{\text{LI}}$ is tridiagonal rather than exactly diagonal; therefore, unless we know the true solution, any diagonal operator that we produce will be an approximation. This leads to a conundrum: is it better to find the diagonal of $\mathbf{A}'_{\text{LI}} \mathbf{A}_{\text{LI}}$, or another approximation that incorporates information about the off-diagonal elements?

Following the same approach as above, a vector full of ones seems a reasonable choice for a reference model; and as before, this generates a data vector full of ones. Applying the adjoint of linear interpolation to this data vector produces the model-space fold, $\mathbf{A}'_{\text{LI}} \mathbf{1}$, that can be used in the approximation,

$$\mathbf{diag}(\mathbf{A}'_{\text{LI}} \mathbf{1})^{-1} \approx (\mathbf{A}'_{\text{LI}} \mathbf{A}_{\text{LI}})^{-1}. \quad (\text{C.2})$$

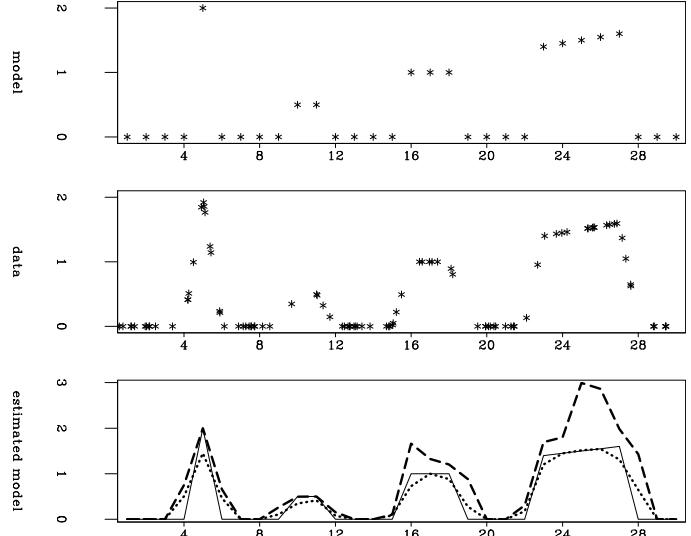
The fold of the linear interpolation operator is not the diagonal of $\mathbf{A}'_{\text{LI}} \mathbf{A}_{\text{LI}}$, but rather the sum along the rows of \mathbf{A}'_{LI} (Chemingui, 1999), which for the case of any interpolator is equivalent to the row-sums of $\mathbf{A}'_{\text{LI}} \mathbf{A}_{\text{LI}}$ (Biondi, 1997). The distinction between taking the diagonal and the row-based sum is an important one because equation (C.2) is exact if the true model is a constant function, and will be a good approximation if the true model is smooth compared to the size of an impulse response of $\mathbf{A}'_{\text{LI}} \mathbf{A}_{\text{LI}}$ (two sample intervals). As long as this smoothness

requirement is satisfied, equation (C.2) implies that a row-based normalization function is better than the diagonal itself.

Figures C.1 and C.2 strengthens this conclusion. The top panels show a simple regularly-sampled model consisting of a four events. The middle panels show data-points obtained by linear interpolation from this model, and the lower panels show the reestimated models after normalized binning with the adjoint of the linear interpolator. The thin solid-lines, the dashed-lines and the dotted-lines show the true model and the results of normalization by the matrix diagonal and operator fold, respectively.

For Figure C.1, I sampled one hundred data points. Only where the model consists of an isolated single spike (at m_5) does the diagonal normalization out-perform operator fold. Elsewhere the amplitude of the original function is recovered more accurately by operator fold. For Figure C.2, I only sampled 30 data points, leading to a less well-conditioned system than Figure C.1. The row-based fold normalization scheme shows itself to be more robust in areas of poor coverage than the scheme that considers only the diagonal elements of the $\mathbf{A}_{\text{LI}}' \mathbf{A}_{\text{LI}}$ matrix.

Figure C.1: Comparison of normalizing adjoint linear interpolation with the row-sum of $\mathbf{A}'\mathbf{A}$ as opposed to its diagonal. Top panel shows the model. Center panel shows 100 interpolated data points. Lower panel shows reestimated data: dashed-line after diagonal normalization, dotted-line after column-based normalization. The solid-line shows the true model. **[kirch-linterp100 [ER,M]]**



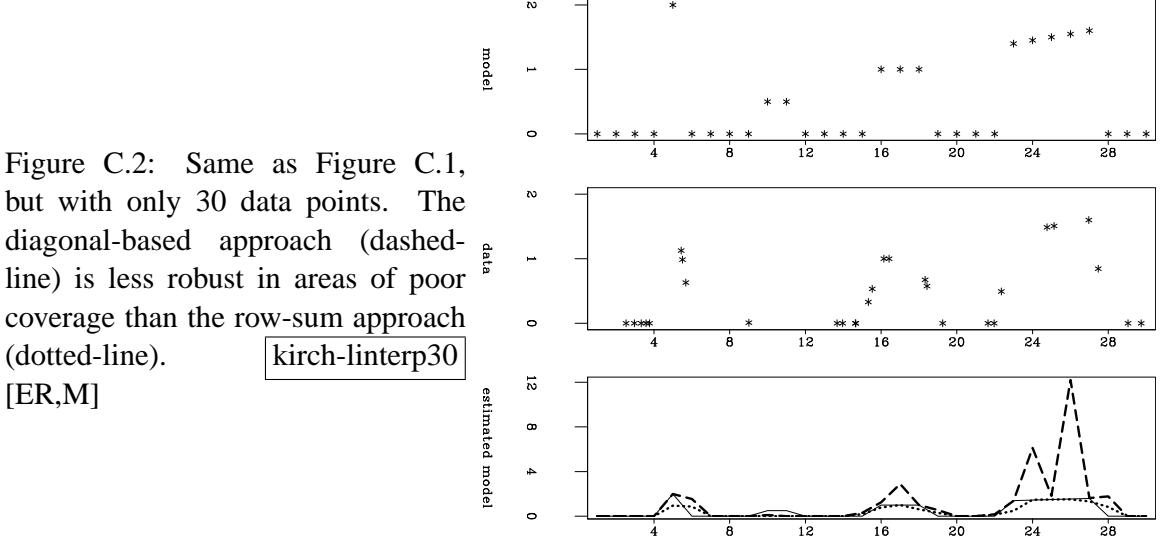


Figure C.2: Same as Figure C.1, but with only 30 data points. The diagonal-based approach (dashed-line) is less robust in areas of poor coverage than the row-sum approach (dotted-line).

kirch-interp30

[ER,M]

C.3 Normalization of Kirchhoff operators

Kirchhoff operators are more complex than interpolators: each output is a weighted-sum of an entire summation surface, rather than just a few nearby points. The process of calculating rows of $\mathbf{A}'_{\text{Kirch}} \mathbf{A}_{\text{Kirch}}$ by modeling and migrating a reference model full of ones will give the ideal weighting function if the true model is smooth on the scale of an impulse response of $\mathbf{A}'_{\text{Kirch}} \mathbf{A}_{\text{Kirch}}$ (the familiar bow-tie shape). This is true for both time and depth migrations.

The validity of operator fold is less well-defined. Dimensionally, it has the units of $\mathbf{A}_{\text{Kirch}}^{-1}$ as opposed to $\mathbf{A}_{\text{Kirch}}^{-2}$ which one would expect. Furthermore, if we generate synthetic data over a model-space full of ones, we no longer get simply a data-space full of ones. Although for a $v(z)$ model, the data will be a constant function of midpoint, it may vary as a function of offset and time depending on the implementation of the modeling algorithm.

We can still ask, however, what reference model *would* generate a constant function in data-space? Restricting ourselves to $v(z)$ problems (time-migration, DMO, AMO etc.), it is clear that the reference model would have to be constant along any midpoint axes, but may potentially vary along the time and offset (or angle) axes. However, rapid variations along these axes are unlikely.

Starting with such a reference model, we can go ahead and define the row-based operator fold as the migration response to a data-space full of ones: $\mathbf{A}'_{\text{Kirch}} \mathbf{1}$. The correct fold [from equation (7.4)] will be the result of dividing this by the original model; but since the original model did not vary along the midpoint axis, and presumably varied slowly along the other axes, the operator fold should be close to the correct fold. In fact, because the original model was a constant value of midpoint, the relative fold of flat events at constant offset will remain unchanged. Hence calculating operator fold via a data-space full of ones is in some sense equivalent to “flat-event calibration” for time migration operators (Chemingui, 1999).

For Kirchhoff depth migration operators, it is not clear what reference model will produce a data-space full of ones. However, the operator fold calculated by migrating such a dataset will still captures the effects of an incomplete recording geometry; and so normalization by operator fold may still be useful (albeit approximate) technique.

Duquet et al. (2000) calculate illumination appropriate for Kirchhoff depth imaging by independently modeling and migrating point scatterers at every location in the model-space. This explicitly evaluates the diagonal elements of $\mathbf{A}'_{\text{Kirch}} \mathbf{A}_{\text{Kirch}}$. As for the case of linear interpolation, this approach is not equivalent to normalization by the operator fold, which includes a summation along the rows of $\mathbf{A}'_{\text{Kirch}}$. Considering only the diagonal of $\mathbf{A}'_{\text{Kirch}} \mathbf{A}_{\text{Kirch}}$ will lead to similar problems that are observed in Figures C.1 and C.2. The diagonal may approach the ideal weighting function if the true model consists of point scatterers that are isolated on the scale of the bow-tie ($\mathbf{A}'_{\text{Kirch}} \mathbf{A}_{\text{Kirch}}$) impulse responses; however, for any kind of *reflectors*, as opposed to diffractors, this situation is not realized.

C.4 Operator fold for finite-frequency migration operators

Calculating operator fold for finite-frequency migration operators is not as simple as modeling and migrating (or just migrating) a vector full of ones. A vector that is full of ones in the time domain has zero temporal frequency, and will not propagate in the same way as the frequencies of interest. Similarly, a vector that is full of ones in the frequency domain cause problems because it will be localized in time. Randomizing the phase may solve this particular problem, but it will cause others.

Duquet et al.'s approach (2000) is also not appropriate for normalizing wave-equation migrations because they are necessarily full-volume methods that cannot be implemented in a target-oriented manner. A naive implementation would require full modeling and migration for every point in model-space.

Fortunately the general formulation in Chapter 7 does not have these limitations, and is applicable to any linear operator provided an appropriate choice of reference model exists.

C.5 Conclusion

For nearest neighbour interpolation, the inverse of the fold produced by modeling and migrating a *model* vector full of ones will be the ideal normalization operator. Similarly, for linear interpolation it will be close to the ideal normalization operator as long as the true model varies slowly on the scale of the sampling interval. For these interpolation operators, operator fold calculated from a *data* vector full of ones is exactly equivalent to the weighting function that can be derived from modeling and migrating the a reference model full of ones.

For Kirchhoff operators, modeling and migrating a reference model full of ones only produces a good normalization fold map if the true model varies slowly on the scale of a bow-tie-shaped modeling/migration impulse response. In the special case of $v(z)$ Kirchhoff operators, normalization by the operator fold is incorrect dimensionally, but will approximate the correct weighting functions for flat-events.

Bibliography

- Aki, K., and Richards, P., 1980, Quantitative seismology: Freeman and Co.
- Baskir, E., and Weller, C. E., 1975, Sourceless reflection seismic exploration: *Geophysics*, **40**, no. 1, 158–159.
- Beasley, C. J., and Klotz, R., 1992, Equalization of DMO for irregular spatial sampling: 62nd Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 970–973.
- Berkhout, A. J., 1985, Seismic migration: Elsevier Science Publ. Co., Inc., Amsterdam.
- Biondi, B., and Palacharla, G., 1996, 3-D prestack migration of common-azimuth data: *Geophysics*, **61**, no. 6, 1822–1832.
- Biondi, B. L., 1997, 3-D seismic imaging: Stanford Exploration Project.
- Biondi, B., 2000, Stable wide-angle Fourier-finite difference downward extrapolation of 3-D wavefields: *SEP-103*, 79–103.
- Black, J. L., and Schleicher, K., 1989, Effect of irregular sampling on prestack DMO: 59th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1144–1147.
- Bleistein, N., 1987, On the imaging of reflectors in the earth: *Geophysics*, **52**, 931–942.
- Bourgeois, A., Bourget, M., Lailly, P., Poulet, M., Ricarte, P., and Versteeg, R., 1991, Marmousi, model and data Marmousi, model and data, The Marmousi experience: Proc. of 1990 EAEG workshop on practical aspects of seismic data inversion.

- Chemingui, N., 1999, Imaging irregularly sampled 3D prestack data: Ph.D. thesis, Stanford University.
- Claerbout, J., and Nichols, D., 1994, Spectral preconditioning: SEP-**82**, 183–186.
- Claerbout, J. F., 1968, Synthesis of a layered medium from its acoustic transmission response: Geophysics, **33**, no. 2, 264–269.
- Claerbout, J. F., 1971, Toward a unified theory of reflector mapping: Geophysics, **36**, no. 3, 467–481.
- Claerbout, J. F., 1985, Imaging the earth's interior: Blackwell Scientific Publications, Inc.
- Claerbout, J. F., 1992, Earth soundings analysis: Processing versus inversion: Blackwell Scientific Publications, Inc.
- Claerbout, J. F., 1995, Basic earth imaging: Stanford Exploration Project.
- Claerbout, J., 1997, Multidimensional recursive filters via a helix: SEP-**95**, 1–13.
- Claerbout, J. F., 1998a, Geophysical estimation by example: Stanford Exploration Project.
- Claerbout, J. F., 1998b, Multidimensional recursive filters via a helix: Geophysics, **63**, 1532–1541.
- Claerbout, J., 1998c, Factorization of cross spectra: SEP-**97**, 337–342.
- Clapp, R. G., Fomel, S., and Claerbout, J., 1997, Solution steering with space-variant filters: SEP-**95**, 27–42.
- Clapp, R. G., 1999, Mobil AVO: SEP Data Library.
- Cole, S., 1995, Passive seismic and drill-bit experiments using 2-D arrays: Ph.D. thesis, Stanford University.
- Crawley, S., 1999, Interpolation with smoothly nonstationary prediction-error filters: SEP-**100**, 181–196.

- de Bruin, C. G. M., Wapenaar, C. P. A., and Berkhouit, A. J., 1990, Angle-dependent reflectivity by means of prestack migration: *Geophysics*, **55**, no. 9, 1223–1234.
- Dellinger, J. A., Gray, S. H., Murphy, G. E., and Etgen, J. T., 2000, Efficient 2.5-D true-amplitude migration: *Geophysics*, **65**, no. 3, 943–950.
- Duquet, B., Marfurt, K. J., and Dellinger, J. A., 2000, Kirchhoff modeling, inversion for reflectivity, and subsurface illumination: *Geophysics*, **65**, no. 4, 1195–1209.
- Duvall, T. L., Jefferies, S. M., Harvey, J. W., and Pomerantz, M. A., 1993, Time-distance helioseismology: *Nature*, **362**, 430–432.
- Etgen, J., and Regone, C., 1998, Strike shooting, dip shooting, widepatch shooting - Does prestack depth migration care? A model study.: 68th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 66–69.
- Etgen, J. T., 1994, Stability of explicit depth extrapolation through laterally varying media: 64th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1266–1269.
- Fomel, S., and Claerbout, J. F., 1997, Exploring three-dimensional implicit wavefield extrapolation with the helix transform: *SEP-95*, 43–60.
- Fomel, S., Clapp, R., and Claerbout, J., 1997, Missing data interpolation by recursive filter preconditioning: *SEP-95*, 15–25.
- Fomel, S., 2000, Three-dimensional seismic data regularization: Ph.D. thesis, Stanford University.
- Gazdag, J., and Sguazzero, P., 1984, Migration of seismic data by phase-shift plus interpolation: *Geophysics*, **49**, no. 2, 124–131.
- Gazdag, J., 1978, Wave equation migration with the phase-shift method: *Geophysics*, **43**, no. 7, 1342–1351.
- Giles, P. M., Duvall, T. L., and Scherrer, P. H., 1997, A subsurface flow of material from the sun's equator to its poles: *Nature*, **390**, 52.

- Godfrey, R. J., Muir, F., and Claerbout, J. F., 1979, Stable extrapolation: *SEP-16*, 83–87.
- Gough, D., and Toomre, J., 1991, Seismic observations of the solar interior: *Annual Review of Astronomy and Astrophysics*, **29**, 627–685.
- Graves, R. W., and Clayton, R. W., 1990, Modeling acoustic waves with paraxial extrapolators: *Geophysics*, **55**, no. 3, 306–319.
- Hale, 1990a, 3-D Depth migration via McClellan transformations: *Geophysics*, **56**, no. 11, 1778–1785.
- Hale, 1990b, Stable explicit depth extrapolation of seismic wavefields: *Geophysics*, **56**, no. 11, 1770–1777.
- Harvey, J., 1995, Sounding out the sun: *Physics Today*, pages 32–38.
- Holberg, O., 1988, Towards optimum one-way wave propagation: *Geophys. Prosp.*, , no. 36, 99–114.
- Hubral, P., Schleicher, J., and Tygel, M., 1996, A unified approach to 3-D seismic reflection imaging, Part I: Basic concepts: *Geophysics*, **61**, no. 3, 742–758.
- Iserles, A., 1996, A first course in the numerical analysis of differential equations: Cambridge Univ. Press.
- Jacobs, A., 1982, The pre-stack migration of profiles: Ph.D. thesis, Stanford University.
- Kerr, B., and Rickett, J., 2001, Preliminary results from a small-scale 3-D passive seismic study in Long Beach, CA: *SEP-108*.
- Kolmogorov, A. N., 1939, Sur l'interpolation et l'extrapolation des suites stationnaires: *C.R. Acad.Sci.*, **208**, 2043–2045.
- Kosovichev, A. G., and Duvall, T. L., 1997, Acoustic tomography of solar convective flows and structures *Acoustic tomography of solar convective flows and structures*, Kluwer, Solar Convection and Oscillations and their Relationship, 241.

- Kosovichev, A. G., 1999, Inversion methods in helioseismology and solar tomography: submitted to Elsevier Preprint.
- Li, Z., 1991, Compensating finite-difference errors in 3-D migration and modeling: Geophysics, **56**, no. 10, 1650–1660.
- Margrave, G. F., 1998, Theory of nonstationary linear filtering in the Fourier domain with application to time-variant filtering: Geophysics, **63**, no. 1, 244–259.
- Nemeth, T., Wu, C., and Schuster, G. T., 1999, Least-squares migration of incomplete reflection data: Geophysics, **64**, no. 1, 208–221.
- Pann, K., and Shin, Y., 1976, A class of convolutional time-varying filters: Geophysics, **41**, no. 1, 28–43.
- Prucha, M., Biondi, B., and Symes, W., 1999, Angle-domain common image gathers by wave-equation migration: 69th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 824–827.
- Prucha, M. L., Clapp, R. G., and Biondi, B., 2000, Seismic image regularization in the reflection angle domain: SEP–**103**, 109–119.
- Resnick, J. R., Ng, P., and Larner, K., 1987, Amplitude versus offset analysis in the presence of dip: 57th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 617–620.
- Ristow, D., and Ruhl, T., 1994, Fourier finite-difference migration: Geophysics, **59**, no. 12, 1882–1893.
- Robinson, E. A., and Treitel, S., 1980, Geophysical signal analysis: Prentice-Hall, Inc.
- Ronen, S., and Liner, C. L., 2000, Least-squares DMO and migration: Geophysics, **65**, no. 5, 1364–1371.
- Sava, P., and Fomel, S., 2000, Migration angle-gathers by Fourier Transform: Geophysical Prospecting, submitted.

- Sava, P., Rickett, J., Fomel, S., and Claerbout, J., 1998, Wilson-Burg spectral factorization with application to helix filtering: **SEP-97**, 343–351.
- Sava, P., Biondi, B., and Fomel, S., 2001, Amplitude-preserved common-image gathers by wave equation migration: 71st Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts.
- Sava, P., 2001, An out-of-core optimization library: **SEP-108**.
- Scherrer, P. H., Bogart, R. S., Bush, R. I., Hoeksema, J. T., Kosovichev, A. G., Schou, J., Rosenberg, W., Springer, L., Tarbell, T. D., Title, A., Wolfson, C. J., Zayer, I., and the MDI Engineering Team, 1995, The Solar Oscillations Investigation - Michelson Doppler Imager: *Solar Physics*, **162**, no. 1/2, 129–188.
- Slawson, S. E., Monk, D. J., and Sudhakar, V., 1995, DMO implications for 3-D survey design: 65th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 935–938.
- Stolt, R. H., and Benson, A. K., 1986, Seismic migration: Geophysical Press, London.
- Tarantola, A., 1987, Inverse problem theory: Elsevier Science Publ. Co., Inc., Amsterdam.
- Vaillant, L., 1999, SEG-EAGE salt data: SEP Data Library.
- Widrow, B., and Stearns, S. D., 1985, Adaptive signal processing: Prentice Hall, Inc.
- Wilson, G., 1969, Factorization of the covariance generating function of a pure moving average function: *SIAM J. Numer. Anal.*, **6**, no. 1, 1–7.
- Yilmaz, O., 1987, Seismic data processing: Soc. Expl. Geophys.
- Zhou, H., and McMechan, G. A., 1997, One-pass 3-D seismic extrapolation with the 45° wave-equation: *Geophysics*, **62**, no. 6, 1817–1824.