

Important Notice

This copy may be used only for the purposes of research and private study, and any use of the copy for a purpose other than research or private study may require the authorization of the copyright owner of the work in question. Responsibility regarding questions of copyright that may arise in the use of this copy is assumed by the recipient.

UNIVERSITY OF CALGARY

High Resolution Seismic Imaging using Least Squares Migration

by

Abdolnaser Yousefzadeh

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY IN GEOPHYSICS

Department of Geoscience

CALGARY, ALBERTA

December, 2012

© Abdolnaser Yousefzadeh 2012

Abstract

High resolution seismic imaging methods are getting increasing attention in the exploration and development industry. Regularized Least Squares Prestack Migration (LSPSM) is one of these methods.

Of the various methods of seismic migration, due to its low cost and flexibility of handling acquisition and topography irregularities, Kirchhoff migration has been the most frequently used method of migration in the industry for decades. LSPSM based on Kirchhoff migration is an effective method to attenuate acquisition footprint that result from sparseness or irregularities of seismic data sampling.

LSPSM is a costly choice when compared to a conventional Kirchhoff migration. As shown in this study, the LSPSM equation cannot be solved efficiently using a standard multigrid method as it requires an explicit form of the corresponding Hessian matrix which needs to be diagonally dominant. It is shown that the Hessian of the LSPSM equation is a very large, dense, and diagonally non-dominant matrix.

The performance of LSPSM is highly sensitive to the accuracy of the velocity model. Without a reasonably accurate velocity model, LSPSM cannot improve the quality of the migration image or do proper data reconstruction. This property provides a method to quantify the accuracy of the velocity model. Velocity analysis that is based on the migration CIGs can be extended to LSPSM CIGs. A coherency spectrum measured from LSPSM CIGs provides a velocity model that is accurate enough to give a high resolution image using LSPSM, and good data reconstruction in a few iterations.

In a complementary study, the application of LSPSM in time lapse seismic analyses is investigated. Separate and joint LSPSM inversion of time lapse data gives a high resolution time lapse image which is less affected by different acquisition geometries of the baseline and monitor surveys. Data reconstruction of old and new surveys into a consistent new geometry will provide comparable prestack time lapse data sets.

In conclusion, LSPSM is an effective method for high resolution imaging, with its advantages possibly outweighing its high cost.

Acknowledgements

I express my deep appreciation to my academic advisor and mentor, Dr. John Bancroft, who has offered continuous help, advice, guidance, encouragement, and friendship during my four years of study at the University of Calgary. A course on migration that I had with John in 2007 and his view of practical seismic migration, encouraged me to choose the University of Calgary to continue my education with him. John has always tried his best to help me not only with my technical questions on migration algorithms, but also my many other problems that I encountered. John is a superior supervisor who always recommends students to put more weight on one thing of real importance, family, than research.

I would also like to thank the Department of Geoscience and the Consortium for Research in Elastic Wave Exploration Seismology (CREWES) and its sponsors, for granting admission, providing a great learning environment and opportunities, and support. I acknowledge Nexen Inc. for providing the NEBC seismic data and permitting publication of the results.

I appreciate the examining committee, Dr. Laurence R. Lines, Dr. Gary F. Margrave, Dr. Clayton V. Deutsch, Dr. J. A. Rodrigue Blais, and Dr. Jennifer Leslie-Panek for their valuable time that they spent to read my dissertation, catch my mistakes, and for their comments, questions, and suggestions.

Special thanks are given to Dr. Gary Margrave, for his excellent teaching and advice. I found Gary one of the best instructors in geophysics. My research and I have also greatly benefitted from courses offered by Dr. Edward Krebes, Dr. Kristopher Innanen, Dr. Michael Lamoureux, and Dr. Robert Ferguson. Special thanks to Dr. Rolf Maier and Kevin Hall at the CREWES project for babysitting our desktops and keeping up our printers. Thanks to Rolf in particular for his careful and attentive editing. I would like to thank all other faculty, staff, and friendly students in the CREWES project.

I have enjoyed the technical guidance, knowledge, experience of, or discussions with Larry Lines, Helen Isaac, Sam Gray, Phillip Bording, Michael Lamoureux, Mostafa Naghizadeh, Hugh Geiger, Peter Manning, David Henley, and Doug Phillips. I appreciate all of them for their kind help, valuable discussions, suggestions, and ideas.

Lastly, I wish to express my very deep and extreme appreciation to my wife, Saeedeh Abbaszadeh, for her patience, encouragement, and support, and her commitment to our family. Living with a PhD student who never has time for anything other than his school, and whose mind is always busy solving an equation or finding a bug in his code is not really convenient. Saeedeh always supported and understood the conditions. I thank my son, Parsa, whose existence brought happiness and hope to our family. Thank you Parsa for understanding why Daddy does not always have time to draw a dinosaur or to play.

Dedication

To Saeedeh and Parsa for their supports and patience.

Table of Contents

Abstract	ii
Acknowledgements	iii
Dedication.....	v
Table of Contents	vi
List of Tables.....	ix
List of Figures and Illustrations.....	x
List of Symbols, Abbreviations and Nomenclature	xvii
Epigraph	xix
CHAPTER ONE: INTRODUCTION.....	1
1.1 Seismic surveying.....	1
1.2 Migration	3
1.3 Least squares migration	5
1.4 Scope of this work	5
1.5 My contributions.....	6
1.6 Outline of this dissertation.....	7
CHAPTER TWO: KIRCHHOFF MIGRATION AND LEAST SQUARES INVERSION	8
2.1 Introduction.....	8
2.2 Kirchhoff migration for a constant velocity medium	8
2.3 Discrete inverse theory	14
2.3.1 The least squares method.....	16
2.3.2 Existence of the least squares solution	18
2.3.2.1 Evendetermined problems	18
2.3.2.2 Overdetermined problems.....	18
2.3.2.3 Underdetermined problems.....	19
2.3.2.4 Mixed-determined problems.....	20
2.4 Least squares Kirchhoff migration	22
2.5 Regularized least squares migration	26
2.6 LSPSM for the resolution of a synthetic example	29
2.7 LSPSM for data reconstruction.....	37
2.8 Real data considerations and example	41
2.9 Summary and Conclusions	46
CHAPTER THREE: SOLVING THE LSPSM EQUATION WITH MULTIGRID AND MULTILEVEL METHODS	48
3.1 Introduction.....	48
3.2 Kirchhoff prestack modeling and migration operators in matrix form.....	50
3.2.1 Explicit form of the wavelet matrix, a starting example	50
3.2.2 Construction of the modeling matrix, \mathbf{G} , in explicit form	54
3.2.3 Construction of the migration matrix in explicit form	62
3.2.4 Example of modeling and migration matrices.....	66
3.2.5 Feasibility of working with the matrix forms.....	73
3.3 Solving the LSPSM equation.....	74

3.4 Multigrid methods for solving the LSPSM equation.....	76
3.4.1 Methods of Jacobi and Gauss-Seidel.....	77
3.4.2 Jacobi convergence and the smoothing property.....	79
3.4.3 Elements of the multigrid method.....	83
3.4.4 Feasibility of solving the LSPSM equation with standard multigrid.....	90
3.5 Feasibility of solving LSPSM with CG and multilevel CG methods.....	91
3.5.1 Steepest descent method.....	92
3.5.2 CG and LSCG.....	93
3.5.3 LSCG for solving a regularized LSPSM equation.....	96
3.5.4 BiCG Stabilized method for solving the LSPSM equation.....	97
3.5.5 Multilevel CG methods versus CG methods.....	99
3.5.6 Multilevel LSCG with spatial down-sampling of data.....	103
3.6 Summary and Conclusions.....	118
CHAPTER FOUR: VELOCITY ANALYSIS OF INCOMPLETE DATA WITH LSPSM.....	119
4.1 Introduction.....	119
4.2 Migration velocity analyses.....	120
4.3 Effect of the velocity accuracy on the LSPSM.....	122
4.3.1 Effect on the resolution enhancement.....	122
4.3.2 Effect on data reconstruction.....	125
4.3.3 The convergence rate of LSCG.....	129
4.4 Velocity analysis on LSPSM offset domain CIGs.....	130
4.5 Velocity analysis on LSPSM shot domain CIGs.....	134
4.6 Summary and Conclusion.....	146
CHAPTER FIVE: LSPSM/INVERSION FOR PRE- AND POSTSTACK TIME LAPSE STUDIES.....	147
5.1 Introduction.....	147
5.2 Separate LSPSM inversion of time lapse data.....	148
5.2.1 Reducing the effect of different acquisition geometries.....	148
5.2.1.1 Case I: both surveys have dense data sampling.....	151
5.2.1.2 Case II: both surveys are 80% decimated.....	161
5.2.2 Reducing the effect of losing receivers.....	166
5.2.2.1 Comparing migration and LSPSM images.....	166
5.2.2.2 Comparing data and reconstructed data from two surveys.....	171
5.2.3 High resolution detection of reflectivity changes.....	177
5.3 Joint inversion of time lapse data by LSPSM.....	182
5.4 Real data examples.....	188
5.5 Summary and Conclusion.....	193
CHAPTER SIX: CONCLUSION.....	194
REFERENCES.....	196
APPENDIX A: LSCG ALGORITHM FOR SOLVING DAMPED LSPSM EQUATION.....	203

APPENDIX B: LSCG ALGORITHM FOR SOLVING SMOOTH LSPSM EQUATION.....	204
APPENDIX C: BICG ALGORITHM	205
APPENDIX D: BICGSTAB ALGORITHM FOR SOLVING LSPSM.....	206
APPENDIX E: DEMONSTRATING THE POSITIVE DEFINITENESS OF THE SECOND DIFFERENCE MATRIX.....	207

List of Tables

Table 3-1 Convolution algorithm.....	54
Table 3-2 Wavelet matrix algorithm.....	54
Table 3-3 A simplified Kirchhoff prestack modeling subroutine showing how to create data for the given reflectivity.	55
Table 3-4 Calculation of a prestack modeling matrix.....	56
Table 3-5 Calculation of prestack modeling matrix with linear interpolation.....	65
Table 3-6 LSCG algorithm to solve the Kirchhoff LSPSM equation.....	95
Table 3-7 Frequency range, trace spacing, and the number of traces for each frequency band for the shot number 5.	105
Table 5.1 LSCG algorithm for solving image difference joint inversion equation.	184

List of Figures and Illustrations

Figure 1.1 CMP seismic acquisition design on an Earth model with two horizontal layers.	2
Figure 1.2 The data from seismic experiment in Figure 1.1, CMP gather with a) before, b) after NMO correction, and c) the stack.	2
Figure 1.3 CMP seismic acquisition with dipping layers.	3
Figure 2.1 Schematic design of an inverse problem after Scales and Snieder (2000).	15
Figure 2.2 The acquisition geometry used in generating synthetic data. Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.	30
Figure 2.3 Interval velocity model used to generate synthetic data.	31
Figure 2.4 Reflectivity model corresponding to the velocity model of Figure 2.2.	31
Figure 2.5 Migration of synthetic data. Comparing to Figure 2.4, shallow reflectors are weak and sinusoidal reflectors are not very well retrieved.	32
Figure 2.6 LSPSM of the synthetic data. Better presentation of the shallow reflectors and improved focusing of deep folding layers compared to migration.	32
Figure 2.7 Acquisition geometry after decimating 80% of traces randomly. Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.	33
Figure 2.8 Comparison between the a) complete, and b) decimated acquisition geometry for the shot #7. blue: sources, red: receivers.	34
Figure 2.9 Migration of decimated synthetic data, 80% of traces are randomly deleted.	34
Figure 2.10 LSPSM of 80% randomly decimated data.	35
Figure 2.11 a) and b) model, c) and d) migration, and e) and f) LSPSM relative amplitudes of the complete (left) and decimated (right) synthetic data for the position of 2280 to 2400 m.	36
Figure 2.12 Convergence rate of LSPSM in 20 iterations with the complete data (solid line) and the decimated data (dotted line). Both cases converge to 4% in 20 iterations.	37

Figure 2.13 Data reconstruction by LSPSM. a) data, b) 80% decimated data, c) reconstructed data, d) residuals. The first 50 traces from shotgather # 7 are shown.	39
Figure 2.14 FK spectra of the data in Figure 2.13. Spectrum for a) original data, b) decimated data, c) reconstructed data, d) residuals.....	40
Figure 2.15 Images resulting from a) migration and b) LSPSM of NEBC data set.	43
Figure 2.16 a) migration and b) LSPSM of decimated synthetic data, 90% of traces are randomly deleted.....	44
Figure 2.17 Data reconstruction by LSPSM. a) data, b) 90% decimated data, c) reconstructed data, d) residuals.....	45
Figure 2.18 Regularized LSPSM of decimated data.....	46
Figure 2.19 a) LSPSM and b) Regularized LSPSM of the decimated data set.	47
Figure 3.1 A simple model including one source (red triangle) on the top of a scatterpoint (blue circle) and eight receivers (green flags) on the surface.....	56
Figure 3.2 Model includes one source (red triangle) and eight receivers (green flags), and two scatterpoints (blue circles) on the surface.	58
Figure 3.3 Acquisition geometry defined for construction of the synthetic modeling and migration matrices. Geometry includes two sources and 10 receivers per source. Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.....	66
Figure 3.4 Interval velocity model used to construct the synthetic modeling and migration matrices.	67
Figure 3.5 Non-zero elements of a) matrix \mathbf{g} , and c) matrix \mathbf{G} , close-up of the first 500 rows and 300 columns of b) \mathbf{g} , and d) \mathbf{G} matrices.....	68
Figure 3.6 a) Non-zero elements of matrix \mathbf{g}^T , b) Close-up of the first 500 rows and 300 columns.....	69
Figure 3.7 a) Non-zero elements of the Hessian matrix $\mathbf{g}^T \mathbf{g}$, b) Close-up of the first 300 rows and columns.	70
Figure 3.8 a) Non-zero elements of matrix $\mathbf{G}^T \mathbf{G}$, b) Close-up of the first 300 rows and columns. $\mathbf{G}^T \mathbf{G}$ is denser than the $\mathbf{g}^T \mathbf{g}$ matrix.	71
Figure 3.9 a) Non-zero elements of the first 500 rows and columns of matrix $\mathbf{W}^T \mathbf{W}$, b) Close-up of the first 100 rows and columns.	72

Figure 3.10 The ratio of absolute values of diagonal elements to the sum of absolute values of the non-diagonal elements for each row of $\mathbf{G}^T\mathbf{G}$. In a diagonally dominant matrix all values must be greater than one.....	74
Figure 3.11 a) The Euclidean error norm versus the number of iterations for the weighted Jacobi iterations with different initial values to solve equation 3.70, b) Solutions after 50 iterations.	82
Figure 3.12 a) Convergence rate of the weighted Jacobi method when the initial solution is a superposition of both low and high frequencies to solve equation 3.70, b) Solution after 50 iterations.....	83
Figure 3.13 Interpolation and restriction processes illustrated schematically on a 2D grid.	85
Figure 3.14 Schematic v-cycle multigrid. The main problem is solved on the fine grid, and low-frequency residuals are restricted to the coarse grid used to solve equation in the coarse grid. Solution in the coarse grid is interpolated to the fine grid and improved the initial value for iterations on the main grid.	86
Figure 3.15 A v-cycle multigrid concept is shown on a 2D grid.....	87
Figure 3.16 A comparison between v-cycle, V-cycle, W-cycle, and full multigrid methods.	88
Figure 3.17 V-cycle multigrid concept shown on a 2D grid.....	89
Figure 3.18 a) The residuals error norm versus the number of iterations is plotted for the last cycle of the full multigrid method. b) Comparison of the solution of the weighted Jacobi after 50 iteration with the solution of the full multigrid after 25 iterations on each grid.....	90
Figure 3.19 Convergence rate of LSPSM in 20 iterations in the LSCG (solid line) and BiCGSTAB (dashed line) methods.....	98
Figure 3.20 a) The Euclidean norm error versus the number of iterations is plotted for the CG iterations with different initial values to solve the linear equation 3.70 with \mathbf{b} a random vector. b) Solutions after n iterations.	100
Figure 3.21 The convergence rate of the CG method for the LSPSM example in Chapter Two for data with different dominant frequency wavelets.	100
Figure 3.22 The convergence rate of the LSCG method used to solve LSPSM for the filtered versions of the Marmousi data set. Data are band-limited with a low pass ($< 15Hz$), two band pass ($15Hz < f < 30Hz$, and $30Hz < f < 45Hz$), and a high pass ($45Hz < f$) filters.....	101

Figure 3.23 Comparison between a) multilevel LSCG and b) LSCG to solve regularized LSPSM of the Marmousi data set with 5 iterations on each grid.	102
Figure 3.24 Frequency spectrum of shot number 5 with 9 chunking areas.	106
Figure 3.25 Frequency spectrum (a, c, and e) and corresponding data (b, d, f) for shot number 5 in chunks 1 (a,b), 5 (c,d), and 9 (e,f). Continued at the next page.....	107
Figure 3.26 Comparison between an LSPSM image with a) a zero initial model, and b) an image resulting from a lower frequency chunk as the initial model, after same number of LSCG iterations.....	111
Figure 3.27 Comparison of convergence rates for the LSCG method when f) the starting model is zero, with the result from a higher chunk as the initial value for chunks a) 8, b) 6, c) 4, d) 2, and e) 1.	112
Figure 3.28 LSPSM images after three LSCG iterations, with the image resulting from a lower frequency chunk as the initial model. a), b), c), d), e), f), g), h), and i) correspond to chunk 9, 8, 7, 6, 5, 4, 3, 2, and 1, respectively.	113
Figure 3.29 LSPSM images after eight LSCG iterations with zero as the initial model.	115
Figure 3.30 LSPSM images after three LSCG iterations on 84% regularly decimated data, a) in chunk 1 with the image resulting from a lower frequency chunk as the initial model, b) on data with the image resulting from chunk 1 as the initial model.....	116
Figure 3.31 LSPSM images after three LSCG iterations on 84% decimated data with zero as the initial model.	117
Figure 4.1 Geometry of seismic data modeling, includes 16 sources and 200 receivers per source. Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.	123
Figure 4.2 Interval velocity model used for studying the effect of velocity errors on LSPSM.....	123
Figure 4.3 Migration of data with the exact velocity.....	124
Figure 4.4 LSPSM image of data with the true velocity.....	124
Figure 4.5 Migration of data with the 5% higher than the true velocity.....	125
Figure 4.6 LSPSM of data with the 5% higher than the true velocity.....	126
Figure 4.7 Reconstruction of data by LSPSM. Complete data are used and the velocity model is exact. a) original data, b) reconstructed data, and c) residuals...	127

Figure 4.8 Reconstruction of data by LSPSM. Complete data are used and the velocity is 5% higher than the true velocity. a) original data, b) reconstructed data, and c) residuals.....	128
Figure 4.9 Normalized convergence rate of the LSCG method in 10 iterations, with true, and a 5% higher velocity model.	130
Figure 4.10 Migration offset domain CIG at $x = 2422 m$ with the a) true velocity and b) with a 5% higher velocity.....	131
Figure 4.11 LSPSM offset domain LSPSM CIG at $x = 2422 m$, a) with the true velocity and b) with a 5% higher velocity.	132
Figure 4.12 Semblance spectrum from CMP gathers at a) $x = 2422 m$ and b) $x = 3922m$ using 10% of regularly selected data.....	133
Figure 4.13 Migration offset domain CIG semblance spectrum at a) $x = 2422 m$, and b) $x = 3922m$ with 10% of regularly selected data.....	133
Figure 4.14 LSPSM offset domain CIG semblance spectrum at a) $x = 2422 m$, and b) $x = 3922m$ with 10% of regularly selected data.....	134
Figure 4.15 Migration shot domain CIGs at position $x = 2422 m$, a) using the exact velocity and b) using a velocity 10% higher than the exact value.....	135
Figure 4.16 LSPSM shot domain CIGs at position $x = 2422 m$, a) using the exact velocity, and b) using a velocity 10% higher than the exact value.....	136
Figure 4.17 Migration shot domain CIG semblance spectrum at a) $x = 2422 m$, and b) $x = 3922m$, with 10% of the data.	137
Figure 4.18 LSPSM shot domain CIG semblance spectrum at a) $x = 2422 m$, and b) $x = 3922m$, with 10% of the data.....	137
Figure 4.19 Migration shot domain CIG XC spectrum at a) $x = 2422 m$, and b) $x = 3922m$, with 10% of the data.....	139
Figure 4.20 LSPSM shot domain CIG XC spectrum at a) $x = 2422 m$, and b) $x = 3922m$ with 10% of the data.....	139
Figure 4.21 Comparison between the true and the analyzed velocity function at 3 positions.	140
Figure 4.22 a) RMS velocity model used to generate synthetic data, and b) the extracted velocity using the XC method on the LSPSM shot domain CIGs.	141
Figure 4.23 Migration of the decimated data with the extracted velocity model.	143

Figure 4.24 LSPSM of the decimated data with the extracted velocity model.....	143
Figure 4.25 Reconstruction of data with the extracted velocity; a) original data, b) decimated data, c) reconstructed data, and d) residual.	144
Figure 4.26 Normalized convergence rate of the LSCG method in 10 iteration with true, higher, and extracted velocities.	145
Figure 5.1 Velocity model used for forward modeling and data generation.	151
Figure 5.2 Acquisition geometry for a) baseline, and b) monitor surveys Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.	152
Figure 5.3 Migration of data set from a) baseline, and b) monitor survey.	154
Figure 5.4 The difference between two migration images in Figure 5.3	155
Figure 5.5 LSPSM images of the data set from a) baseline, and b) monitor surveys.	156
Figure 5.6 The difference between the two LSPSM images in Figure 5.5.....	157
Figure 5.7 CMP gathers from a) baseline and b) monitor surveys and c) their difference.	159
Figure 5.8 Reconstructed CMP gathers from a) baseline and b) monitor surveys and c) their difference.....	160
Figure 5.9 Acquisition geometry for a) decimated baseline, and b) monitor surveys. Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.....	162
Figure 5.10 The difference between two a) migration, and b) LSPSM images.....	163
Figure 5.11 CMP-bin supergathers from a) baseline, and b) monitor surveys.	164
Figure 5.12 Reconstructed CMP gathers from a) baseline, and b) monitor surveys, and c) their difference.	165
Figure 5.13 Migration difference (a and c) and LSPSM difference (b and d) with using 50% (a and b) and 30% (c and d) of data.	168
Figure 5.14 CMP gathers from monitor surveys after 50%, 30%, 20%, 10%, 5%, and 2% in panels a), b), c), d), e), and f) respectively.	172
Figure 5.15 Reconstructed CMP gathers using LSPSM on the left and the difference from baseline data on the right, with (a,b) 50%, (c,d) 30%, (e,f) 20%, (g,h) 10%, (i,j) 5%, and (k,l) 2% of data.	174

Figure 5.16 Velocity model used in the monitor survey.....	177
Figure 5.17 a) Migration and b) LSPSM time lapse images resulting from subtraction of baseline and monitor images.	179
Figure 5.18 CMP gathers from a) baseline and b) monitor surveys, and c) their difference.	180
Figure 5.19 Reconstructed CMP gathers from a) baseline and b) monitor surveys, and c) their difference.....	181
Figure 5.20 Reflectivity of baseline survey achieved by MIJI.	185
Figure 5.21 Reflectivity of monitor survey achieved by MIJI.....	185
Figure 5.22 Difference between reflectivity of baseline and monitor surveys by MIJI.	186
Figure 5.23 Reflectivity of baseline survey achieved by IDJI.....	186
Figure 5.24 Time lapse image from reflectivity by IDJI.	187
Figure 5.25 Migration of data set from a) baseline, and b) monitor survey selected from NEBC data set.	189
Figure 5.26 The difference between two migration images in Figure 5.25.....	190
Figure 5.27 The baseline image obtained from the time lapse joint inversion (MIJI)....	190
Figure 5.28 The monitor image obtained from the time lapse joint inversion (MIJI)....	191
Figure 5.29 The difference between reflectivity of baseline and monitor surveys by MIJI.....	191
Figure 5.30 Image of a) the baseline survey and b) time lapse obtained by IDJI.....	192

List of Symbols, Abbreviations and Nomenclature

<i>Symbol</i>	<i>Definition</i>
G_i	i^{th} element in \mathbf{G} vector
G_{ij}	Element of \mathbf{G} matrix at i^{th} row and j^{th} column
J	Cost/objective function
M_{trc}	Number of traces in a 2D seismic line
N_t	Number of samples in each trace of seismic
N_w	Length of wavelet
N_x	Number of model grids in transverse direction
N_z	Number of model grids in vertical direction
t_m	Migration time
$\mathbf{A}^h, \mathbf{A}^{2h}, \mathbf{A}^{4h}, \dots$	Matrix \mathbf{A} in grid $h, 2h, 4h, \dots$
\mathbf{A}^k	Matrix \mathbf{A} in/after k iterations
\mathbf{D}_h	First derivative matrix or operator in the offset direction
\mathbf{G}^{-1}	Inverse of \mathbf{G}
\mathbf{G}^*	Adjoint of \mathbf{G}
\mathbf{G}^T	Transpose of \mathbf{G} , in matrix form or as an operator
\mathbf{G}_v	Vectorized form of \mathbf{G}
\mathbf{I}_{2h}^h	Interpolation matrix in multigrid method
\mathbf{R}_h^{2h}	Restriction matrix in multigrid method
\mathbf{m}_{l_1}	l_1 least squares solution
$\hat{\mathbf{m}}$	Migration image, blurred version of model \mathbf{m}
\mathbf{m}_{DSL}	Damped least squares solution
\mathbf{m}_{SLS}	Smooth least squares solution
$^\circ$	Degree (angle)
$*$	Convolution operator
2D	Two dimension
3D	Three dimension
AGC	Automatic gain control
BiCG	Bi Conjugate gradients
BiCGSTAB	Bi Conjugate gradients stabilized
CDP	Common depth point
CG	Conjugate Gradients
CMP	Common mid point
DSR	Double squares root
E	Total error, scalar
FK	Frequency-wavenumber

<i>Symbol</i>	<i>Definition</i>
FOCI	Forward operator and conjugate inverse
IDJI	Image difference joint inversion
LSCG	Least squares Conjugate Gradients
LSM	Least squares migration
LSPSM	(Kirchhoff) least squares prestack migration
MIJI	Multiple image joint inversion
NMO	Normal move out
OBC	Ocean bottom cable method
PDE	Partial differential equation
<i>s</i>	second
VSP	Vertical seismic profiling
\mathcal{R}	Regularization term
RJID	Regularized Joint inversion for Image Differences
RJMI	Regularized Joint inversion of Multiple Images
rms	Root mean squares
Δx	Trace spacing
Δm	Time lapse image
<i>Hz</i>	Hertz
<i>f</i>	Temporal frequency
<i>l</i>	Norm of a vector
<i>D</i>	Diagonal matrix
<i>G</i>	Forward modeling operator or matrix
<i>I</i>	Identity matrix
<i>L</i>	Forward modeling in general (Chapter two)
<i>L</i> and <i>U</i>	Strictly lower and upper triangular matrices (Chapter Three)
<i>S</i>	Sampling matrix
<i>W</i>	Wavelet matrix
<i>d</i>	Data, measurements, observations
<i>e</i>	Misfit error vector
<i>g</i>	Kirchhoff forward modeling operator, matrix
<i>m</i>	Reflectivity model
<i>w</i>	Convolution matrix
β	Geological dip
δ	Kronecker delta
λ	Regularization weight for LSPSM
μ	Regularization weight in general
ϵ	Threshold value

Epigraph

“Anyone who has never made a mistake has never tried anything new”:

Albert Einstein.

Chapter One: INTRODUCTION

Seismic reflection techniques are widely used in the exploration, evaluation and characterization of shallow to deep hydrocarbon reservoirs. A high resolution image of the Earth's subsurface provided by seismic methods enables us to better assess and recover these valuable energy resources. Migration of seismic data is an essential process that is performed to obtain an accurate image in areas with complex underground geology.

1.1 Seismic surveying

Seismic methods are developed based on the propagation of elastic waves through the Earth's materials. An array of seismic sources releases energy into the ground, and a series of receivers records the energy reflected back to the surface or directly received at the receiver stations. Ignoring any density variations, reflection happens when the velocity changes in the seismic ray paths. Receivers convert the arrived mechanical signal into electrical currents. Recording instruments digitize and save the data on an appropriate medium to transfer to the processing unit. The data are processed, and result in an image of the Earth's subsurface.

The common midpoint (CMP) is the main method that is currently used for organizing seismic data. The CMP is a point on the Earth located at the middle of source and receiver locations. The common depth point (CDP) is the reflection point on the reflector at some depth. The distance (or sometimes double distance) between the CMP and the corresponding source or receiver is called the offset. Source and receiver positions are designed to have several reflections from a CDP point where there is a change in velocity or density. Figure 1.1 shows a simple two layer Earth model where the CMP and CDP are at the same location. The signal from source S1 is reflected back and is recorded at receiver R1. As the distance between the source and receiver increases, the arrival time of the reflected wavelet increases as illustrated in Figure 1.2a.

Processing of the recorded seismic data starts with trace editing, attenuation of noise, correcting the near surface effects, and removal of unwanted signals. Then, normal moveout (NMO) correction is applied to align the wavelets with zero offset time as in Figure 1.2b. All traces corresponding to the same CMP are then stacked to increase the signal to noise ratio as in Figure 1.2c. All the stacked traces create an initial image of the Earth's subsurface. Additional

processes such as muting, deconvolution, filtering, and multiple attenuation may be required and used for enhancing the quality or accuracy of the stacked image.

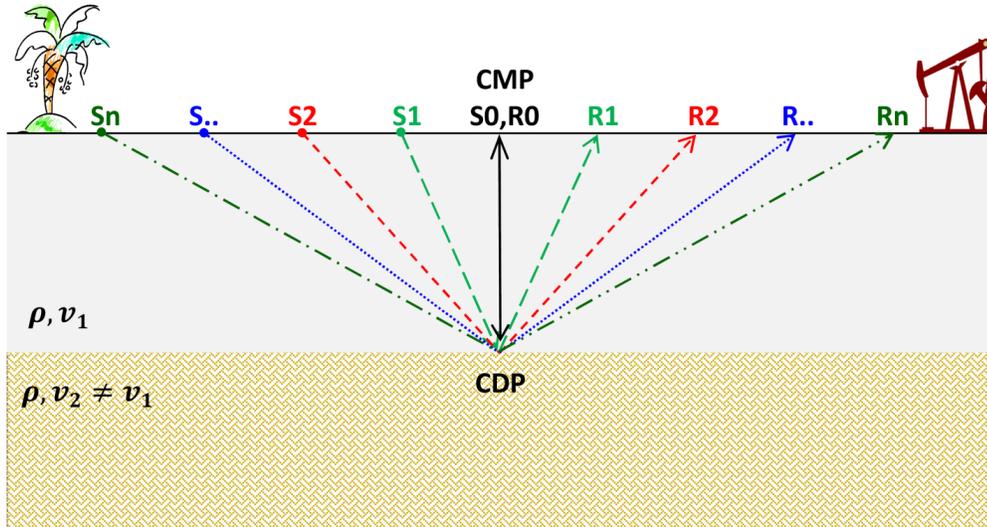


Figure 1.1 CMP seismic acquisition design on an Earth model with two horizontal layers.

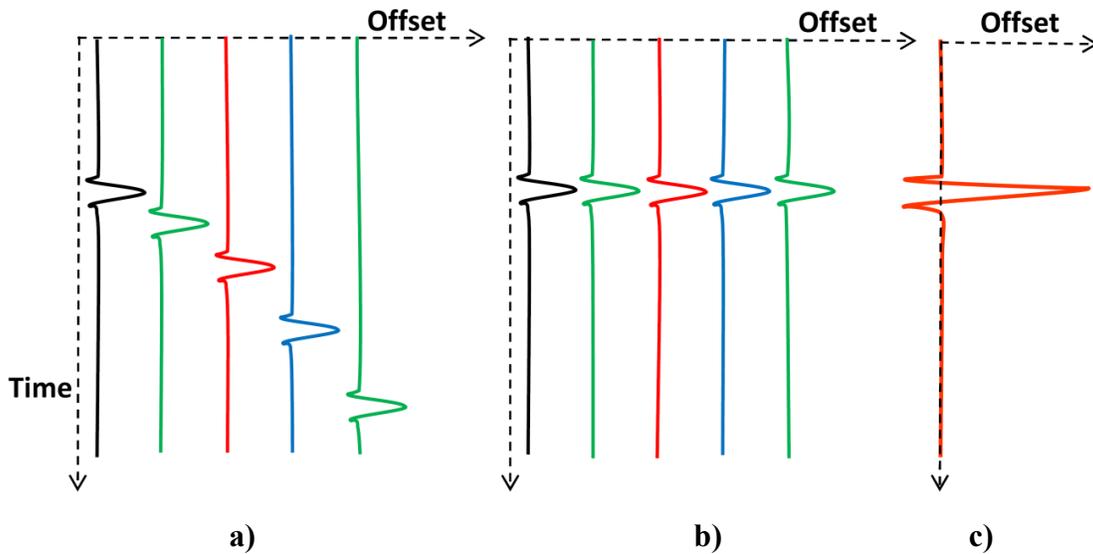


Figure 1.2 The data from seismic experiment in Figure 1.1, CMP gather with a) before, b) after NMO correction, and c) the stack.

In the CMP design of the seismic reflection method, all reflectors are assumed to be horizontal and horizontally homogenous. In the case of dipping layers, or a horizontal variation

in acoustic impedance (velocity multiplied by density at each point), reflection points are not beneath the CMP positions (Figure 1.3). With the regular processing procedure mentioned earlier, the dipping reflectors are not at the CMP location, and have a different dip in the stacked image. Another problem is the presence of diffracted energies that result from scattering from sharp edges (F1 in Figure 1.3, for example). These kinds of undesired signals remain in the final stacked section. The processing step of migration is used to relocate the energy to the correct location.

1.2 Migration

In seismic data processing, migration is a general term for the process that moves the dipping reflections to their true locations with correct amplitudes and dip. Migration not only moves the dipping events into the correct spatial position, but also collapses the diffracted energies into the scatterpoints. Prestack migration eliminates the requirement of NMO correction and stacking. Therefore, compared to the other steps in seismic processing, migration is a complex and time consuming procedure. The result of a migration is an image which ideally shows the Earth's reflectivity image as a function of time or depth.

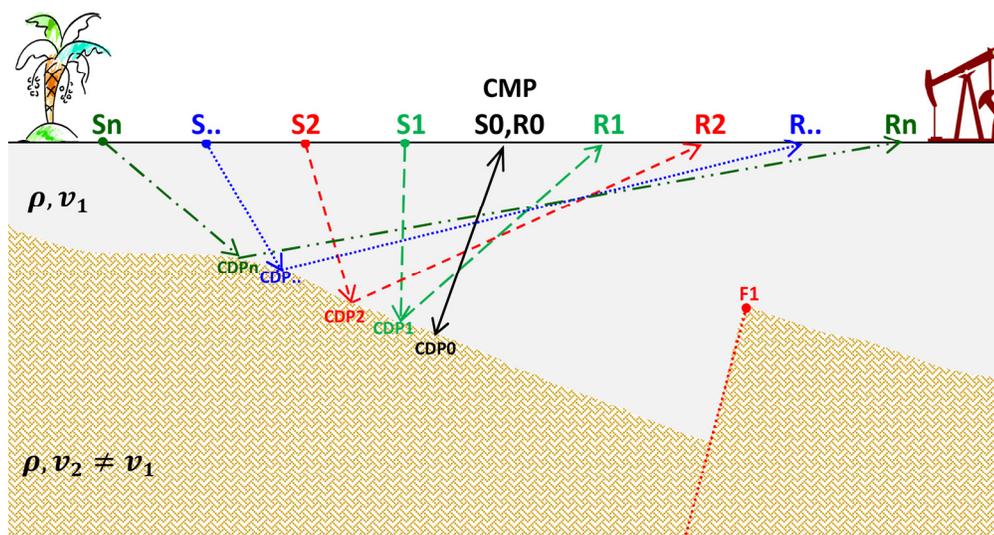


Figure 1.3 CMP seismic acquisition with dipping layers.

Migration methods have been greatly developed over time. First methods developed on the geometrical construction of reflectors using mechanical or optical devices, prior to implementing

computers in the seismic industry. Diffraction summation in the time domain, Kirchhoff migration, was introduced by Hagedoorn (1954). In the 1970's, Claerbout and Doherty (1972) showed how migration is an approximate solution to the wave equation, and demonstrated their work on digital computers. The integral formulation of Kirchhoff migration was derived by Schneider (1978). Gazdag (1978) and Stolt (1978) showed how to perform migration using properties of the Fourier transform. Reverse time migration, one of the most expensive wavefield continuation methods of migration was introduced later (Baysal et al., 1983; Whitmore, 1983; McMechan, 1983). Hill (1990) introduced the Gaussian beam migration based on the decomposition of data and source function in Gaussian beams.

Kirchhoff migration works by collapsing all diffracted energies back to the original scatterpoint. In the resulting migration image, each dipping reflector is moved to the correct location with the true dip and length. Kirchhoff prestack migration is one of the most frequently used methods of migration in the oil industry. This is due to its many advantages of being easy to program, implement, and computationally fast and cheap to run. It works with non constant background velocity, handles converted waves, has output in time or depth, and has the ability to work on a part of the image (targeting). It can be applied to a stacked seismic section or to prestack data, and works with irregularly sampled or incomplete data. The result is an image which ideally shows the Earth's reflectivity model. Kirchhoff migration is able to reveal the subsurface with complex geologic structures and strong lateral velocity variations. It is able to handle multipath rays, too.

In reality, and especially in 3D seismic surveys, it is difficult to have dense, regularly sampled seismic data. Human-made obstacles, mountains, rivers, land topography, acquisition layout and economical aspects, usually leave some gaps or cause an un-even distribution of azimuths (in 3D) and offsets in the acquired seismic data. Also, marine data lack a wide distribution of azimuths, and may have gaps due to marine construction (eg. drilling platforms) or streamer feathering. Kirchhoff migration is ideal for this type of irregularly sampled data. However, coarse data sampling produces acquisition footprint in the final image. Migration artifacts caused by the acquisition geometry may be referred to as acquisition footprint (Sheriff,

2006). Least squares migration¹ is an advanced form of Kirchhoff migration and is the best known cure for this type of problem.

1.3 Least squares migration

Kirchhoff prestack migration can handle incomplete or irregular data. However, incomplete data give a blurred image of the Earth's reflectivity (Ji, 1997; Nemeth et al., 1999). Where migration produces a blurred image of the Earth subsurface, least squares migration (LSM) helps to avoid this shortcoming. LSM attenuates acquisition footprint by minimizing the difference between observed and modeled data in a least squares sense (Ji, 1997; Nemeth et al., 1999; Duquet et al., 2000). Implementing LSM instead of conventional migration for achieving a high resolution seismic image is a costly replacement. Its success depends on the accuracy of the velocity model.

LSM is not limited to the Kirchhoff method. Wave equation migration and the other methods that utilize the Fourier domain can also be used in the least squares scheme. However, relatively low cost, and the ability of easily handling incomplete and irregular seismic data without the requirement of data interpolation, are some advantages that makes Kirchhoff migration, a more desirable method for LSM than other migration methods that require regularly sampled seismic data. Higher resolution images of the LSM can be used for data reconstruction, as a bypass product.

1.4 Scope of this work

In the research summarised in this dissertation, I undertook three separate tasks. The first task was to solve the LSPSM equation with multigrid and multilevel methods in order to speed up the process and reduce the computational cost, or to enhance the resolution of image. The second task was to benefit from the strong dependency of the method on the accuracy of the velocity field and use it as a tool for the evaluation of the velocity model. The final step consisted in using the LSPSM method for the inversion of time lapse data. I show how artifacts

¹ As called by geophysicists such as Ji (1997), Nemeth et al. (1999), and Duquet et al. (2000) in last two decades.

in the time lapse images, due to different acquisition geometries, can be attenuated using LSPSM/inversion of time lapse data.

1.5 My contributions

Different aspects of the implementation of Kirchhoff least squares prestack migration (LSPSM) for resolution enhancement and data reconstruction are investigated. My contributions to this area of research can be summarized as:

- Feasibility of using multigrid methods for solving LSPSM is investigated.
- A method for the construction of migration, modeling, and LSPSM's Hessian in matrix forms, is introduced. It is shown how the method can be extended to create the matrix form of the other linear operators.
- Feasibility and advantages of using Conjugate Gradients (CG) multigrid for solving LSPSM is investigated.
- Multilevel CG with spatial resampling to solve the LSPSM equation is shown.
- A BiConjugate Gradient Stabilized (BiCGSTAB) algorithm for solving LSPSM is derived.
- The effect of the velocity's accuracy on image resolution and data reconstruction by LSPSM is researched and demonstrated.
- Effect of velocity accuracy on the offset domain and shot domain Common Image Gather (CIG)s is shown.
- LSPSM resolution enhancement, data reconstruction, and the LSCG convergence rate as a method to evaluate the accuracy of the velocity model are introduced.
- The velocity analysis on migration CIGs is extended to the LSPSM offset domain and shot domain CIGs. The advantages and disadvantages of the method are discussed.
- The effect of regularization on resolution enhancement is shown.
- The effect of multiples on the convergence of LSPSM with real data is explained.
- The problem of different data acquisition geometries in prestack and poststack time lapse studies is explained and shows how separate and joint LSPSM/Inversion of data can attenuate the acquisition footprint.

- It is shown how data reconstruction by LSPSM can help with the prestack time lapse analysis.
- An LSCG algorithm for the Joint LSPSM/Inversion of time lapse data is demonstrated.

1.6 Outline of this dissertation

Chapter Two expresses the LSPSM in more detail. I review the derivation of Kirchhoff migration theory and equations. This chapter continues with an introduction to discrete inverse theory which is extended to the LSPSM method. Chapter Two ends with some examples of resolution enhancement and data reconstruction on synthetic and real data. I show how LSPSM can improve the migration image resolution and attenuate data acquisition artifacts.

In Chapter Three, the idea of using Kirchhoff modeling/migration operators in an explicit form is explained. The feasibility of using these matrix forms for solving the LSPSM with the standard multigrid methods is investigated. Also, I also inspected the advantages and disadvantages of multilevel CG method for solving the LSPSM equation.

Chapter Four focuses on migration and LSPSM velocity analysis for incompletely sampled data. Using synthetic data sets, I show the high sensitivity of LSPSM to the accuracy of the model. I suggest the improvement in image resolution and the ability of an acceptable reconstruction of data with LSPSM be used as tools to evaluate the accuracy of an imaging velocity model. The extension of migration velocity analysis from migration CIGs to LSPSM CIGs and its advantages are discussed.

In Chapter Five I show how separate and joint LSPSM inversion of time lapse data can be used to achieve a high resolution image for poststack time lapse investigations. It also shows the ability of the method for data reconstruction for use with reliable prestack time lapse studies.

Chapter Six contains comments and conclusions.

2.1 Introduction

Kirchhoff migration is one of the simplest migration algorithms to program and for performing quality migration. Kirchhoff migration considers all subsurface points to be scatterpoints that create diffractions, and collapses the diffraction energies back to the scatterpoints. The result is an image of reflectivity with true dip for all events. Kirchhoff prestack modeling, on the other hand, spreads the energy from the scatterpoint to the seismic data. Kirchhoff has been the favorite method of migration in the industry for decades, due to the low cost, its handling of irregular topography, and its ability to use irregularly sampled or spatially incomplete data.

In section 2.2 of this chapter I derive the Kirchhoff migration formula for a constant velocity medium. Discrete inverse theory and different types of the inverse problem are discussed in section 2.3. Discrete inverse theory is extended to least squares Kirchhoff migration in the section 2.4. The ability of least squares Kirchhoff migration for enhancing the image resolution and data reconstruction is shown in synthetic and real examples in sections 2.6 to section 2.8.

2.2 Kirchhoff migration for a constant velocity medium

In this section the derivation of the Kirchhoff migration equation for a constant velocity (and density) medium is reviewed. I follow the derivation method by Schneider (1978) and follow explanations by Scales (1997) and Margrave (2001).

The source free scalar wave equation for pressure in an inhomogeneous fluid is expressed by (Margrave, 2001)

$$\nabla^2 U(\mathbf{r}, t) - (\nabla \ln \rho(\mathbf{r}) \cdot \nabla U)(\mathbf{r}, t) = \frac{\rho(\mathbf{r})}{k(\mathbf{r})} \partial_t^2 U(\mathbf{r}, t), \quad 2.1$$

where U is the acoustic pressure, \mathbf{r} is the position vector, t is the time, $\nabla^2 (= \nabla \cdot \nabla)$ is the Laplacian, $k(\mathbf{r})$ is the bulk modulus, and $\rho(\mathbf{r})$ is the density of the medium. Assuming the medium has constant density, $\rho(\mathbf{r}) = \rho$, equation 2.1 can be further simplified to the scalar wave equation for pressure in a homogenous medium

$$\nabla^2 U(\mathbf{r}, t) = \frac{1}{c^2} \partial_t^2 U(\mathbf{r}, t), \quad 2.2$$

where

$$c = \sqrt{k/\rho},$$

is the speed of acoustic waves. In the presence of an impulse source at position \mathbf{r}_0 and time t_0 with amplitude A , equation 2.2 can be written as

$$\nabla^2 U - \frac{1}{c^2} \partial_t^2 U = -A \delta(\mathbf{r} - \mathbf{r}_0, t - t_0). \quad 2.3$$

where t and \mathbf{r} are the measurement time and positions of the receiver, respectively. We try to solve this equation using Greens' theorem. This procedure results in the Kirchoff migration equation given at the end of this section.

Assume a volume, Ω , enclosed with a smooth boundary, $\partial\Omega$. This volume is considered to be inside the Earth, and some of Earth's surface is a part of the boundary. By putting a seismic source on the Earth's surface, source will be part of the boundary. Hence, the medium is source free and homogenous and we can use partial differential equation (PDE) 2.2. To solve the PDE 2.2, knowledge about the initial values of U and its normal derivative $\partial U / \partial \mathbf{n}$ in Ω and one of U or $\partial U / \partial \mathbf{n}$ on the boundary $\partial\Omega$ must be specified. Neumann boundary conditions specify the value of the derivative of a solution, on the boundary of the domain. The problem of finding the solution is known as the Neumann problem (Morse and Feshbach, 1953). Dirichlet boundary conditions specify the value of the solution on the boundary of the domain. Dirichlet problem involves finding the solution to this kind of problems (Morse and Feshbach, 1953).

Green's function is defined as the solution of PDEs, such as the wave equation, to an impulse source. Therefore, if $A = 1$, the Green's function is the solution to equation 2.3. Without any assumption about the boundary condition, Green's function with the source and receiver coordinate and time, $\Gamma(\mathbf{r}, t; \mathbf{r}_0, t_0)$, is considered as the solution to the scalar wave equation 2.3 (Scales, 1997)

$$\nabla^2 \Gamma - \frac{1}{c^2} \partial_t^2 \Gamma = -4\pi \delta(\mathbf{r} - \mathbf{r}_0) \delta(t - t_0), \quad 2.4$$

where (\mathbf{r}_0, t_0) are source coordinate and time, and δ is the Dirac delta function. Multiplying this equation by U gives (Scales, 1997)

$$U \nabla^2 \Gamma - U \frac{1}{c^2} \partial_t^2 \Gamma = -4\pi \delta(\mathbf{r} - \mathbf{r}_0) \delta(t - t_0) U. \quad 2.5$$

Multiplication of the source free scalar wave equation 2.2 by Γ results in

$$\Gamma \nabla^2 U - \Gamma \frac{1}{c^2} \partial_t^2 U = 0. \quad 2.6$$

Subtracting equation 2.5 from equation 2.6 gives (Scales, 1997)

$$4\pi\delta(\mathbf{r} - \mathbf{r}_0)\delta(t - t_0)U = \Gamma\nabla^2 U - U\nabla^2\Gamma + \frac{(U \partial_t^2\Gamma - \Gamma\partial_t^2 U)}{c^2}, \quad 2.7$$

or

$$4\pi\delta(\mathbf{r} - \mathbf{r}_0)\delta(t - t_0)U = \nabla \cdot [\Gamma\nabla U - U\nabla\Gamma] + \frac{\partial_t[U \partial_t\Gamma - \Gamma\partial_t U]}{c^2}. \quad 2.8$$

Equation 2.8 can be integrated over Ω space and all times

$$\int_{\Omega} \int_{-\infty}^{\infty} 4\pi\delta(\mathbf{r} - \mathbf{r}_0)\delta(t - t_0)U dv' dt_0 = \int_{\Omega} \int_{-\infty}^{\infty} (\nabla \cdot [\Gamma\nabla U - U\nabla\Gamma] + \frac{\partial_t[U \partial_t\Gamma - \Gamma\partial_t U]}{c^2}) dv' dt_0. \quad 2.9$$

Using the properties of the Dirac delta we can write

$$4\pi \int_{-\infty}^{\infty} \int_{\Omega} \delta(\mathbf{r} - \mathbf{r}_0)\delta(t - t_0)U dv' dt_0 = \xi(\mathbf{r})U(\mathbf{r}, t), \quad 2.10$$

where

$$\xi(\mathbf{r}) = \begin{cases} 4\pi & \text{if } \mathbf{r} \text{ is inside } \Omega \\ 2\pi & \text{if } \mathbf{r} \text{ is on } \Omega \\ 0 & \text{otherwise} \end{cases}.$$

Then equation 2.9 can be expressed by (Scales, 1997)

$$\begin{aligned} \xi(\mathbf{r})U(\mathbf{r}, t) &= \int_{-\infty}^{\infty} \int_{\Omega} \nabla \cdot [\Gamma\nabla U - U\nabla\Gamma] dv' dt_0 \\ &+ \frac{1}{c^2} \int_{-\infty}^{\infty} \int_{\Omega} \partial_t [U \partial_t\Gamma - \Gamma\partial_t U] dv' dt_0, \quad \forall \mathbf{r} \in \mathcal{R}^3. \end{aligned} \quad 2.11$$

Gauss' theorem helps to convert the first integral on the right hand side of equation 2.11 to a surface integral over $\partial\Omega$

$$\int_{\Omega} \nabla \cdot [\Gamma\nabla U - U\nabla\Gamma] dv' dt_0 = \int_{\partial\Omega} [\Gamma\nabla U - U\nabla\Gamma] \cdot \mathbf{n} da' dt_0, \quad 2.12$$

where \mathbf{n} is the outward vector normal to the surface.

Since there is nothing happening before seismic source ignition, U and $\partial_t U$ are zero before $t_0 = 0$. Therefore, integration over $(-\infty, +\infty)$ in time can be reduced to integration over

$[0, +\infty)$. Since Green's functions are causal, the upper time limit for the second integral is zero, too. Therefore, the second integral on the right hand side of equation 2.11 can be ignored

$$\int_{-\infty}^{\infty} \int_{\Omega} \partial_t [U \partial_t \Gamma - \Gamma \partial_t U] dv' dt_0 = 0. \quad 2.13$$

This leaves us with the surface integral which is called the Kirchhoff integral theorem

$$\xi(\mathbf{r})U(\mathbf{r}, t) = \int_0^{\infty} \int_{\partial\Omega} [\Gamma \nabla U - U \nabla \Gamma] \cdot \mathbf{n} da' dt_0, \quad 2.14$$

which indicates that U is the solution to the wave equation everywhere in the space, by knowing U and its normal derivatives on the boundary. Since we do not have both simultaneously, we solve the equation for U in terms of its derivatives. Assuming $f = U|_{\partial\Omega}$ and $g = \frac{\partial U}{\partial n}|_{\partial\Omega}$, equation 2.14 will be a Neumann equation (Morse and Feshbach, 1953):

$$f(\mathbf{r}, t) = \frac{1}{2\pi} \int_0^{\infty} \int_{\partial\Omega} [\Gamma g - f \partial_n \Gamma] da' dt_0. \quad 2.15$$

By specifying g , the above equation is an integral equation for f . The solution to the wave equation can then be computed from the Kirchhoff integral with the knowledge of f .

An alternative view is to specify an assumed f and solve for g (Dirichlet problem). Then the term with the normal derivative in the Kirchhoff integral is cancelled:

$$U(\mathbf{r}, t) = \frac{1}{4\pi} \int_0^{\infty} \int_{\partial\Omega} f(\mathbf{r}_0, t_0) \partial_n \Gamma(\mathbf{r}, t; \mathbf{r}_0, t_0) da' dt_0. \quad 2.16$$

We are looking for a Green's function that vanishes on the boundary. Seismic data are measured on the Earth's surface, $z = 0$. With the assumption of constant velocity and the volume an infinite half space bounded by the plane at $z = 0$, which represents the Earth's surface, the total integral over the boundary $\partial\Omega$ reduces to that over the $z = 0$ plane. Using the method of images, two Green's functions consist a point source at \mathbf{r} and its negative image at \mathbf{r}_0 vanish on the $z = 0$ plane and are the spherically symmetric solutions to the wave equation (Scales, 1997)

$$\Gamma_r(\mathbf{r}, t; \mathbf{r}_0, t_0) = \frac{\delta(t - t_0 - R/c)}{R} - \frac{\delta(t - t_0 - R'/c)}{R'}, \quad 2.17$$

$$\Gamma_a(\mathbf{r}, t; \mathbf{r}_0, t_0) = \frac{\delta(t - t_0 + R/c)}{R} - \frac{\delta(t - t_0 + R'/c)}{R'}, \quad 2.18$$

where Γ_a and Γ_r are for the advanced and retarded Green's functions, respectively, and

$$R = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}, \quad 2.19$$

and

$$R' = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z + z_0)^2}, \quad 2.20$$

are the distances between sources and observer point. Retarded Green's function is causal and propagating outward with increasing t and advanced Green's function is anticausal, converging with increasing time. Since Kirchhoff migration works by collapsing the diffracted energies to the original scatterpoints, we are interested in using only the retarded Green's function, Γ_r (Scales, 1997).

By inserting Γ_r into equation 2.16 and considering $R'(z_0) = R(-z_0)$ we obtain

$$U(\mathbf{r}, t) = \frac{-1}{2\pi} \int_0^\infty \int_{z_0=0} f(\mathbf{r}_0, t_0) \partial_{z_0} \frac{\delta(t - t_0 + R/c)}{R} da' dt_0. \quad 2.21$$

Since z may appear only in a $(z - z_0)$ combination, equation 2.21 can be written by (Schneider, 1978)

$$U(\mathbf{r}, t) = \frac{1}{2\pi} \partial_z \int_0^\infty \int_{z_0=0} f(\mathbf{r}_0, t_0) \frac{\delta(t - t_0 + R/c)}{R} da' dt_0. \quad 2.22$$

Again, the properties of the Dirac delta help to eliminate the integration over time:

$$U(\mathbf{r}, t) = \frac{1}{2\pi} \partial_z \int_{z_0=0} \frac{f(\mathbf{r}_0, t + R/c)}{R} da'. \quad 2.23$$

Equation 2.23 is the Kirchhoff migration formula which is an integral relation that maps the boundary values into a solution for all points in Ω space.

In the simple case of poststack Kirchhoff migration, f is equal to CMP stacked data, $U_s(\mathbf{r}_0(= x, y, z), t)$, where U_s is acoustic pressure measured at time t , and x , y , and $z = 0$ are the coordinates at the surface. CMP stacked data represent the result of a single zero offset seismic experiment at each midpoint location (Scales, 1997). Poststack modeling approximates a single exploding reflector model. In an exploding reflector model all source points, which are considered everywhere on the reflector, explode simultaneously, and waves only propagate upward to the Earth's surface with the half speed. Therefore CMP stacked data is similar to the exploding reflector modeling ($c \rightarrow c/2$ and $t = 0$). The poststack migration can be expressed by integration of CMP stacked data over recording surface with the half speed (Scales, 1997)

$$U(\mathbf{r}) = \frac{1}{2\pi} \partial_z \int_{z_0=0} \frac{U_s(\mathbf{r}_0, 2R/c)}{R} da'. \quad 2.24$$

As equation 2.19 shows, R is a function of z . It can be shown that

$$\partial_z R = \frac{z}{R} = \cos \theta, \quad 2.25$$

where θ is the angle between the lines that connect the image point to the receiver and the z -axis (Margrave, 2001). Performing the z differentiation using the chain rule, the Kirchhoff migration formula is obtained as

$$U(\mathbf{r}) = \frac{1}{2\pi} \int_{z_0=0} \frac{\cos \theta}{Rc} \left[\partial_{t_0} U_s + \frac{c}{R} U_s \right]_{t_0=2R/c} da', \quad 2.26$$

The second term in the bracket is small and can be ignored (Scales, 1997)

$$U(\mathbf{r}) = \frac{1}{2\pi} \int_{z_0=0} \frac{\cos \theta}{Rc} \left[\partial_{t_0} U_s \right]_{t_0=2R/c} da'. \quad 2.27$$

In equation 2.27, $[\partial_{t_0} U_s]_{t_0=2R/c}$ means that expression in the bracket is evaluated at time

$$t_0 = \frac{2R}{c} = \frac{2 \sqrt{(x - x_0)^2 + (y - y_0)^2 + z_0^2}}{c}, \quad 2.28$$

which is the equation of a zero offset diffraction hyperbola (Margrave, 2001).

Equation 2.27 states that the poststack Kirchhoff migration in a constant velocity medium is the summation of data along the diffraction hyperbola after taken their time derivative and are multiplied by $\cos \theta / Rc$. Coefficient $\cos \theta$ is the obliquity or directivity factor and compensates for the angle dependence of the amplitudes at measuring point. The time derivative of data before summation is performed by applying a wavelet shaping factor which is a 45° constant phase shift plus an amplitude spectrum proportional to the square root of frequency for 2D migration and a 90° phase shift and amplitude spectrum proportional to frequency shift for 3D migration (Yilmaz, 2008).

Equation 2.27 is for a constant velocity medium. When velocity changes, the straight-ray traveltimes, R/c , will be replaced by the traveltimes calculated using the ray methods. In a medium with only horizontal homogenous layers (when velocity changes vertically), the traveltimes can be approximated by equation 2.28, if c is replaced by its root mean square (rms)

velocity, V_{rms} . In such case, the migration time, t_{mig} , for a prestack Kirchhoff migration is calculated by the double-squares-root (DSR) equation (Bancroft, 2007)

$$t_{mig} = \sqrt{\frac{t_0^2}{4} + \frac{(x+h)^2}{V_{rms}^2}} + \sqrt{\frac{t_0^2}{4} + \frac{(x-h)^2}{V_{rms}^2}}, \quad 2.29$$

where h is half-offset, the half of the horizontal distance between source and receiver, x is the horizontal distance between the image point and the scatterpoint, V_{rms} is rms velocity, and t_0 and t_{mig} are zero-offset and migration traveltimes, respectively. Methods of migration which handle only vertical or very smooth lateral variations in velocity are known as time migrations. Mentioned Kirchhoff migration using DSR equation is an example. In contrast, depth migration requires special computations to obtain the migration traveltimes, or wave propagation methods to properly migrated seismic data in the areas with a complex geology or variation in lateral velocity. In Kirchhoff depth migration, migration time can be calculated by raytracing methods which are relatively expensive.

In practical implementations there are other factors that must be considered such as proper weighting functions, filtering, interpolation, and dip and aliasing controls with migration of seismic data with Kirchhoff method.

2.3 Discrete inverse theory

In this section the discrete inverse theory and least squares solutions are briefly reviewed. The existence and different types of the solution to an inverse problem are discussed.

Since making quantitative measurements about physical properties of the underground materials is a major goal in geophysics, inverse problems are extremely important in this field of science. The history of applying inversion methods to geophysical problems such as inversion of global earthquake data, traveltime, gravity field, magnetic field, and remote sensing data returns date back to the late 60s and 70s (For a list of these attempts, refer to Lines and Treitel, 1984).

The Earth model with its physical properties can be considered in a mathematical space as the model space. The forward problem is the problem of computing the model response, seismic data for instance. The mathematical space containing the resulting data is called the data space. Inversion is the problem of finding the corresponding model for given (measured) data (Scales and Snieder, 2000). Geophysical data are discretely measured in both space and time domains.

Therefore, we use matrices and vectors instead of their corresponding continuous functions. Discrete inverse theory deals with time and space sampled data and models.

In all inversion problems, models have many properties or parameters which may be continuous functions in space and which may have many degrees of freedom (Scales and Snieder, 2000). However, recorded data are limited to a finite set. Mapping from an Earth model, \mathbf{m} , to the data space, \mathbf{d} , is unique. However, due to the limitation of measurements, the reverse procedure is not unique. Therefore, model estimation problems return the estimated model, $\hat{\mathbf{m}}$, which is a different form of the true model. In seismic imaging for instance, $\hat{\mathbf{m}}$ is a blurred version of \mathbf{m} (Nemeth, et al., 1999; Scales and Snieder, 2000). By solving the inverse problem, we are looking for the best reasonable $\hat{\mathbf{m}}$. An appraisal problem quantifies the uncertainty of the model, the difference between \mathbf{m} and $\hat{\mathbf{m}}$. Figure 2.1 depicts the relationship between forward, estimation, and appraisal problems for a typical inverse problem.

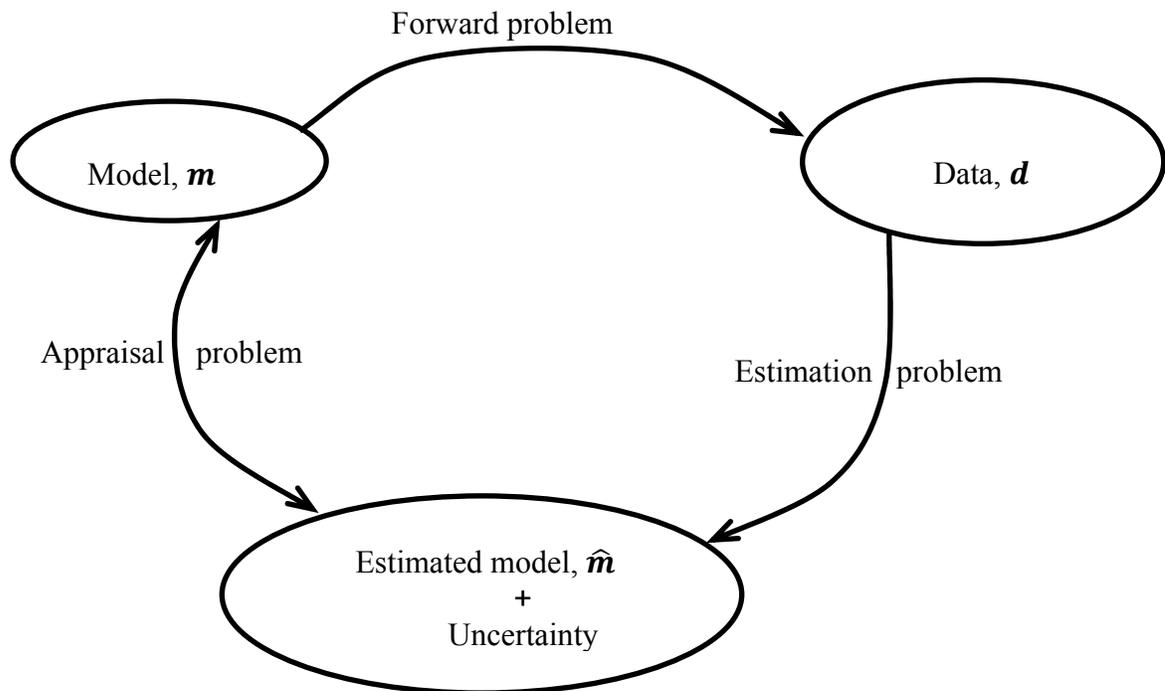


Figure 2.1 Schematic design of an inverse problem after Scales and Snieder (2000).

The theory of linear error propagation is developed to formulate the appraisal problems for the linear inverse problems. Bayesian inversion is the commonly used tool for the nonlinear cases (Scales and Snieder, 2000). Bayesian inversion uses some *a priori* information about the

model (information about the model before any data gathering) and combines them with information in the data to update the *a priori* information. The updated *a priori* model is the a posteriori model which constrains the resulting model more than the model from *a priori* information (Scales and Tenorio, 2001). Ulrych et al. (2001) provide a useful comprehensive tutorial about Bayes inversion.

2.3.1 The least squares method

The linear equation to be solved has the general form

$$\mathbf{G}\mathbf{m} = \mathbf{d}, \quad 2.30$$

where \mathbf{G} is forward modeling operator, called the data kernel matrix, \mathbf{m} , a column vector of length M , is the unknown model, and \mathbf{d} is (prestack) data, a column vector of length N . \mathbf{G} is an N by M matrix as in

$$\begin{bmatrix} G_{11} & G_{12} & \dots & G_{1M} \\ G_{21} & G_{22} & \dots & G_{2M} \\ \dots & \dots & \dots & \dots \\ G_{N1} & G_{N2} & \dots & G_{NM} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \dots \\ m_M \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \dots \\ d_N \end{bmatrix}. \quad 2.31$$

Least squares is the simplest effective method to solve equation 2.30 for an estimate of $\hat{\mathbf{m}}$. This method is based on measuring the size or length of the predicted data, $\hat{\mathbf{d}} = \mathbf{G}\hat{\mathbf{m}}$, from observed data, \mathbf{d} (Menke, 1984).

For simplicity, consider an example of fitting a straight line to a series of data points on the xy -plane. The slope of the desired line, m_l , and its intercept, h_l , are the only unknown parameters to be estimated. Data are N points with the pairs $(x_i, y_i)_{i=1:N}$ as the coordinates. The least squares method is well-known for finding the best-fitting line. The least squares method minimizes the misfit, $e_i = y_i - \hat{y}_i$, where $\hat{y}_i = m_l x_i + h_l$. The best approximate is a line with m_l and h_l such that the total error or Euclidean length of the misfit vector

$$E = \mathbf{e}^T \mathbf{e} = \sum_{i=1}^N e_i^2, \quad 2.32$$

be as small as possible (Menke, 1984).

Therefore, the least squares method finds a line that minimizes its Euclidean distance from the observations. The norm of a vector \mathbf{e} , $\|\mathbf{e}\|$, is a type of measurement of length and generally is defined as:

$$l_n \text{ norm:} \quad \|e\| = \left[\sum_i |e_i|^n \right]^{1/n} . \quad 2.33$$

The common norms of l_1 , l_2 , and l_∞ are frequently used in geophysical inverse problems. Higher order norms imply larger weight on the largest elements of the e vector. Therefore, outliers have significant influence in their solution. By using the l_2 norm, the least squares method gives an appropriate weight on the data with large prediction errors (Menke, 1984). The familiar least squares method for fitting a straight line to scattered data can be generalized to any linear inverse problem.

To minimize the total error, E , for the general linear equation 2.30, we set its derivative with respect to a model parameter, m_q for instance, equal to zero (Menke, 1984). Total error in the expanded formula can be computed from matrix equation 2.32 as:

$$\begin{aligned} E &= e^T e = (d - Gm)^T (d - Gm) \\ &= \sum_i^N \left[d_i - \sum_j^M G_{ij} m_j \right] \left[d_i - \sum_k^M G_{ik} m_k \right] \\ &= \sum_j^M \sum_k^M m_j m_k \sum_i^N G_{ij} G_{ik} - 2 \sum_j^M m_j \sum_i^N G_{ij} d_i + \sum_i^N d_i d_i. \end{aligned}$$

Since for any i and j ,

$$\frac{\partial m_i}{\partial m_j} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} = \delta_{ij},$$

(Kronecker delta), the derivative of the three terms on the right hand side of the previous equation with respect to m_q can be expressed by (Menke, 1984),

$$\begin{aligned} \frac{\partial}{\partial m_q} \left[\sum_j^M \sum_k^M m_j m_k \sum_i^N G_{ij} G_{ik} \right] &= \sum_j^M \sum_k^M [\delta_{jq} m_k + m_j \delta_{kq}] \sum_i^N G_{ij} G_{ik} \\ &= 2 \sum_k^M m_k \sum_i^N G_{iq} G_{ik}, \\ -2 \frac{\partial}{\partial m_q} \left[\sum_j^M m_j \sum_i^N G_{ij} d_i \right] &= -2 \sum_j^M \delta_{jq} \sum_i^N G_{ij} d_i = -2 \sum_i^N G_{iq} d_i, \end{aligned}$$

and

$$\frac{\partial}{\partial m_q} \left[\sum_i^N d_i d_i \right] = 0.$$

Setting the derivative of the total error equal to zero results in (Menke, 1984):

$$\frac{\partial E}{\partial m_q} = 0 = \sum_k^M m_k \sum_i^N G_{iq} G_{ik} - \sum_i^N G_{iq} d_i, \quad 2.34$$

or equivalently in matrix form as

$$\mathbf{G}^T \mathbf{G} \mathbf{m} - \mathbf{G}^T \mathbf{d} = \mathbf{0}, \quad 2.35$$

and \mathbf{m}_{ls} , the least squares solution, can be estimated by finding the inverse of $\mathbf{G}^T \mathbf{G}$:

$$\mathbf{m}_{ls} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{d}. \quad 2.36$$

This is the standard least squares solution, also known as l_2 , a Gauss-Newton solution, or an unconstrained solution (Lines and Treitel, 1984). Existence of the \mathbf{m}_{ls} depends on the existence of $(\mathbf{G}^T \mathbf{G})^{-1}$.

2.3.2 Existence of the least squares solution

The standard least squares problem may or may not have a unique or multiple solution(s). Least squares problems can be classified based on the existence of the solution(s) into the following four categories.

2.3.2.1 Evendetermined problems

When the number of measurements equals to the number of unknowns, $M = N$ in general, the problem is evendetermined (Menke, 1984) or strictly determined. In this case, the data determine all model parameters precisely. In the problem of fitting a straight line to data, as an example, having only two separate points in the data space provides an evendetermined problem and all model parameters (slope and intercept of the line) can be exactly determined.

2.3.2.2 Overdetermined problems

This happens when the number of rows, N , or measurements, are more than the number of columns, M , or unknowns, in data kernel matrix \mathbf{G} , ($M < N$), and all columns are linearly independent. Least squares returns the best (unique) solution. In the example of fitting a line to data, this happens when we have at least three points which are not placed exactly on the same

line. If we have more than two data points, ($M < N$), but all points are on the same line, they are linearly dependent and the problem is even determined.

It happens in many geophysical inverse problems that $M < N$, but due to a poorly structured data kernel, some model parameters are not connected with any data. For instance, a gap in seismic data acquisition may leave some areas of interest uncovered. These problems cannot count as overdetermined problems and may be considered in the underdetermined division. This kind of inverse problem will be discussed in more detail in the next two sections.

2.3.2.3 Underdetermined problems

When we do not have enough information to uniquely determine all model parameters, or the number of unknowns is more than the number of data, $M > N$, or the columns of the data kernel matrix are not linearly independent, there is no unique solution to the problem. In the example of line fitting this happens when we have only one data point. Obviously, there are infinite number of straight lines that pass through a single point in the xy -plan. The number of possible solutions to these kinds of problems may be limited by adding some constraint to them. For example, one may be looking for a line that passes through a given point and is parallel to the x -axis.

Due to the limited number of measuring instruments or limited access to land, geophysical acquisition projects may be spatially limited, and the geophysical inverse problem is underdetermined. However, as mentioned in the previous section, the definition of underdetermined problems based on the comparison between the number of data and unknown parameters is not always exact. The classification of inverse problems also strongly depends on the structure of the data kernel matrix and acquisition parameters, especially the density and regularity of the measurements.

A problem may be considered underdetermined due to too few available measurements. However, there may be some model parameters which do not affect the data, therefore, those model parameters are completely unconstrained (Menke, 1984). On the other hand, some model parameters may be overdetermined. Since there are not enough data to determine all model parameters, these kinds of problems are neither purely overdetermined nor purely underdetermined. An exploration seismic experiment is of this kind. Assuming a seismic wavefield travels as rays, more than one seismic ray may pass through some grids in the model

and a few grids may never be touched by any ray. These kinds of underdetermined problems are called mixed-determined or ill-posed problems (Menke, 1984).

The undetermined problem is called purely underdetermined when all model parameters are underdetermined (Menke, 1984). Purely underdetermined problems have an infinite number of solutions. Adding *a priori* information about the model helps to minimize the number of possible solutions. In the example of fitting a line to one data point, adding the condition that the desired line must be parallel to the x-axis provides a unique solution to this least squares problem.

Constraining purely underdetermined problems with *a priori* information is not an easy task. *A priori* information needs to be mathematically formulated, which is not always possible when combined with least squares problems. An easy and commonly used choice of *a priori* information is that the model be simple (Menke, 1984). A simple solution is a solution whose Euclidian length, $L = \mathbf{m}^T \mathbf{m} = \sum m_i^2$, is as small as possible. With this constraint, the corresponding inverse problem includes minimizing L subject to $\mathbf{d} = \mathbf{G}\mathbf{m}$. It can be shown that the solution to this problem is achieved by solving equation

$$\mathbf{m}_{simple} = \mathbf{G}^T (\mathbf{G}\mathbf{G}^T)^{-1} \mathbf{d}. \quad 2.37$$

Use of the Conjugate Gradients (CG) methods to solve an underdetermined least squares problem depends on the initial approximation. If a vector of zeros is used as the initial model, CG converges to the least squares solution of the smallest Euclidean norm.

2.3.2.4 Mixed-determined problems

Mixed-determined problems are a type of underdetermined problems that are not purely underdetermined. As mentioned in the previous section, seismic imaging problems are generally mixed-determined. Some underground model parameter grids are overdetermined by the transit of several seismic rays while others are underdetermined. There may also be some even-determined grids in the model.

One way to overcome this kind of inverse problems is by splitting the model parameters to separate overdetermined and purely underdetermined models and solve each individually. The purely underdetermined part of a problem is solvable by finding the minimum of the l_2 norm, and the overdetermined part of problem is solved by a least squares method. This procedure

requires decomposition of the data kernel by the method of singular value decomposition which is not effective for large problems such as least squares prestack Kirchhoff migration.

A second and more effective method of solving a mixed determined inversion problems is minimizing an objective function, J , which is some combination of both the prediction error (used for least squares solution of overdetermined problems), and the l_2 norm (used in solving purely underdetermined problems) in the following form (Menke, 1984):

$$J(\mathbf{m}) = E + \mu^2 L = \mathbf{e}^T \mathbf{e} + \mu^2 \mathbf{m}^T \mathbf{m}, \quad 2.38$$

where μ^2 is the tradeoff parameter or regularization weight. The trade-off parameter controls the priority given to the prediction error versus the solution length (simplicity). When $\mu = 0$, this is equivalent to ignoring the *a priori* information. Larger μ s make the *a priori* information and the underdetermined part of the solution more important. Finding the best value for μ depends on the nature of the problem, and a trial and error method is the simplest method to choose the right value.

Minimizing the $J(\mathbf{m})$ gives the damped (Menke, 1984; Lines and Treitel, 1984) or Marquardt-Levenberg (Levenberg, 1944; Marquardt, 1963) least squares solution:

$$\mathbf{m}_{DSL} = (\mathbf{G}^T \mathbf{G} + \mu^2 \mathbf{I})^{-1} \mathbf{G}^T \mathbf{d}. \quad 2.39$$

When measurements have different precisions, the quadratic norm can be generalized to have a weight matrix \mathbf{P} for correlated data of different qualities. Then, the generalized least squares solution will be calculated by (Blais, 2010)

$$\mathbf{m}_{LS} = (\mathbf{G}^T \mathbf{P} \mathbf{G})^{-1} \mathbf{G}^T \mathbf{P} \mathbf{d}. \quad 2.40$$

When the unknown model parameters have different relevance or importance, a weight matrix \mathbf{Q} can be added to generalize the regularized least squares solution, \mathbf{m}_{rLS} , to (Blais, 2010)

$$\mathbf{m}_{RLS} = (\mathbf{G}^T \mathbf{P} \mathbf{G} + \mu^2 \mathbf{Q})^{-1} \mathbf{G}^T \mathbf{P} \mathbf{d}. \quad 2.41$$

Since acquired seismic data are processed and traces are edited and equalized before any migration, we consider that all traces have an equal quality. The goal of the LSPSM is achieving a reflectivity image in which all model grids are equally important, too. Therefore, we always assume $\mathbf{P} = \mathbf{Q} = \mathbf{I}$ in this dissertation. These assumptions simplify the generalized least squares equation 2.41 into equation 2.39.

2.4 Least squares Kirchhoff migration

In this section the general discrete inverse theory is being specified for prestack migration and particularly for the prestack Kirchhoff least squares migration method. In forward modeling, L is an operator, such as Kirchhoff, finite-difference, or phase-shift, which approximates the continuous wave equation for a velocity or reflectivity model. Defining the real forward modeling operator in the Earth by L_r , and the continuous subsurface model by m , the recorded seismic data, d , is a spatially sampled type of the seismic wavefield (Ji, 1997):

$$d = SL_r m, \quad 2.42$$

where S is the seismic sampling operator simulating the discretization and aperture limitation of the seismic surveys.

An operator such as L is called unitary when $LL^* = L^*L = I$, where L^* is the adjoint of L , and I is an identity matrix. Migration is the adjoint of the forward modeling. Migration should be able to back project the recorded wavefield to the real Earth image if it was based on a unitary operator. Since our forward modeling operator, L , is always an approximation to the real forward modeling operator, L_r , and also due to the finite aperture and discrete sampling of the seismic wavefield, migration is not a unitary operator. By approximating the real forward operator by L , $SL = SL_r$, conventional migration can be expressed by

$$\hat{m} = L^* S^* d = L^* S^* SL_r m, \quad 2.43$$

where \hat{m} is the migration image (Ji, 1997). Assuming a dense and long seismic survey, which makes S a unitary operator, $S^*S = I$, then, due to the difference between L and L_r , the migration image is still different from the real subsurface image by migration artifacts. Accurate subsurface imaging requires both a spatially dense and regular sampling of the seismic wavefield and an accurate approximation to the wave equation (Ji, 1997).

Irregularities, coarseness, and inevitably limited aperture in seismic data sampling, produce artifacts in a slant stack image similar to those in the migration image. Kostov (1990) attenuated those artifacts and improved the image resolution by formulating a least squares slant stack inversion in the frequency domain. Kostov's least squares matrix was a Toeplitz (or diagonal-constant) matrix and he used the Levinson recursion method to solve his least squares equation for each frequency.

Cole and Karrenbach (1997) extended Kostov's least squares slant stack to a least squares poststack Kirchhoff migration for resolution enhancement. Since the summation trajectory of the Kirchhoff migration is time variant, they were not able to use a time invariant transform to directly perform their least squares migration in the frequency domain. They chose an expensive method by transforming 2D seismic data to a 3D cube where the third dimension was a panel corresponding to different time invariant moveout corrections where they could implement least squares in the frequency domain. They showed an improvement in the resolution of the resulting image, and better focusing of the migration. Cole and Karrenbach (1997) implemented a CG method for solving their equation and performed 20 CG iterations, equal to 40 migrations or modeling matrix-vector multiplication for each frequency to achieve the desired image.

Ji (1997) performed least squares optimization to attenuate poststack migration artifacts. He showed the improvement of image resolution by least squares optimization of zero-offset data for Kirchhoff migration in a constant velocity medium, Gazdag migration in $v(z)$ medium, and the finite difference method of migration in a $v(x, z)$ model.

The idea of using least squares migration of zero-offset data for attenuating migration artifacts and enhancing image resolution was then extended to prestack least squares migration/inversion. Kirchhoff migration was augmented by a generalized inverse as an approximation to the exact inverse by Tarantola (1984). This approach is called least squares migration/inversion (Nemeth et al., 1999; Duquet et al., 2000; Kuehl and Sacchi, 2001; Kuehl, 2002; Tang, 2007). My focus in this study is on prestack Kirchhoff Migration, therefore, I abbreviated my inversion to LSPSM which from here on stands for Kirchhoff Least-Squares Prestack Migration. In the rest of this chapter the mathematical structure of the LSPSM is reviewed, and using a synthetic and a real data set, the ability of LSPSM in resolution enhancement and data reconstruction is demonstrated.

Since Kirchhoff seismic modeling with diffractions can be formulated as an algebraic linear problem, it can be written in the general form of a linear modeling operator,

$$\mathbf{d} = \mathbf{G}\mathbf{m}, \tag{2.44}$$

where, \mathbf{d} are the observed prestack seismic data, \mathbf{m} is the real Earth reflectivity model, and \mathbf{G} is an operator acting on \mathbf{m} in order to produce \mathbf{d} . The inversion process,

$$\mathbf{m} = \mathbf{G}^{-1}\mathbf{d}, \tag{2.45}$$

should be able to recover the Earth model or reflectivity from the seismic data. Matrix \mathbf{G} is a large matrix which can have 10^{13} entries for a 20 km 2D seismic line with 100,000 traces and 1000 samples per trace, as an example. Due to the enormously large size and generally its non-square shape¹, inverting the \mathbf{G} matrix is generally impossible. As shown by Ji (1997) and discussed earlier in this chapter, due to sampling limitations, even if \mathbf{G} can be inverted, the resulting image is not the true Earth's reflectivity. Thus, approximations to the inversion are used. The first and most commonly performed approximation uses the mathematical transpose of \mathbf{G} ,

$$\hat{\mathbf{m}} = \mathbf{G}^T \mathbf{d}, \quad 2.46$$

which is the Kirchhoff migration that sums energy over diffractions (Duquet et al., 2000). In equation 2.46, $\hat{\mathbf{m}}$ is the migration image and \mathbf{G}^T is the migration operator. I assumed that the \mathbf{G} operator used in the previous three equations is real and exactly simulates the processes that are happening during the propagation of seismic waves in the Earth.

By defining Kirchhoff modeling as the forward process and Kirchhoff migration as its adjoint (mathematical transpose) operator, seismic imaging becomes an inversion problem. Substitution of \mathbf{d} from equation 2.44 into equation 2.46 gives

$$\hat{\mathbf{m}} = \mathbf{G}^T \mathbf{G} \mathbf{m}. \quad 2.47$$

For the same reason that Ji (1997) showed for zero-offset data, if the Hessian matrix², $\mathbf{G}^T \mathbf{G}$, were equal to the unity matrix ($\mathbf{G}^T \mathbf{G} = \mathbf{I}$), then Kirchhoff migration would be able to reconstruct the true model of the subsurface reflectivity from prestack seismic data. However, due to the finite seismic sampling aperture, or irregularity and incompleteness of the sampled seismic data, and the existence of noise, the Hessian matrix, $\mathbf{G}^T \mathbf{G}$ is different from the identity matrix, \mathbf{I} (Nemeth et al., 1999). In such cases, Kirchhoff prestack migration produces some artifacts in the migrated image. These migration artifacts can be attenuated by least squares migration with the same thought that Ji (1997) and Cole and Karrenbach (1997) used for removing migration artifacts in zero-offset data.

¹ A detailed explanation on the size and shape of \mathbf{G} matrix is brought in Chapter Three.

² As it is called by for instance: Lines and Treitel, 1984; Lambare et al., 1992; Nemeth et al., 1999; and Ayeni and Biondi, 2010.

Nemeth et al. (1999) implemented the method on ground penetrating radar data, vertical seismic profiling (VSP) data, and 2D seismic data with limited aperture, coarsely sampled data and data with gaps in the receiver line. They showed more focusing of geological events and better detection of faults in the seismic sections when replacing conventional Kirchhoff migration with the least squares Kirchhoff migration. Duquet et al. (2000) performed regularized least squares Kirchhoff migration for resolution enhancement of incomplete data with the similarity of adjacent common offset image gathers as the *a priori* information. They also created an illumination map to show the density of rays for each image point, which is a useful tool for balancing seismic amplitudes, for further AVO analyses. Kuehl and Sacchi (2001) and Tang (2007) performed regularized least squares wave equation migration on incomplete 2D seismic data. Tang (2007) used sparseness in the subsurface-offset domain as the regularization to improve the migration image resolution.

Least squares prestack migration practically minimizes the difference between observed prestack seismic data, \mathbf{d} , and modeled data, \mathbf{Gm} , as expressed by the l_2 norm: $\|\mathbf{Gm} - \mathbf{d}\|$.

In reality, data include noise, and the problem is mixed-determined. Therefore, an attempt to find a model to fit the data perfectly using standard least squares,

$$\mathbf{m}_{ls} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{d}, \quad 2.48$$

is replaced by an attempt to find the minimum or the Euclidian norm which includes searching for a model that minimizes the cost function

$$J_1(\mathbf{m}) = \|\mathbf{Gm} - \mathbf{d}\|^2, \quad 2.49$$

subject to the model constraint

$$J_2(\mathbf{m}) = \|\mathbf{m}\|^2. \quad 2.50$$

Minimization of these two cost functions can be combined to the minimization of a general cost function in the form of

$$J(\mathbf{m}) = J_1(\mathbf{m}) + \mu^2 J_2(\mathbf{m}) = \|\mathbf{Gm} - \mathbf{d}\|^2 + \mu^2 \|\mathbf{m}\|^2. \quad 2.51$$

which results in the damped least squares solution as we saw for the solution to the general mixed-determined problems

$$\mathbf{m}_{DSL} = (\mathbf{G}^T \mathbf{G} + \mu^2 \mathbf{I})^{-1} \mathbf{G}^T \mathbf{d}. \quad 2.52$$

It is important to mention since all seismic traces and model parameters are equally important, equations 2.48 and 2.52 are the specified form of the generalized least squares equations 2.40 and 2.41, respectively.

2.5 Regularized least squares migration

Numerical optimization methods provide the possibility of minimizing complicated objective functions. These methods allow easier integration of some model space regularization into the least squares solution. Regularization also improves the stability of the least squares solution by removing possible singularities in the $\mathbf{G}^T \mathbf{G}$ matrix in mixed-determined problems. With \mathbf{G} as the Kirchhoff forward modeling operator, different types of *a priori* information other than simplicity of the model are available to be implemented as a part of the least squares migration. A general cost function for least squares migration can be expressed by (Nemeth et al., 1999):

$$J(\mathbf{m}) = \|\mathbf{G}\mathbf{m} - \mathbf{d}\|^2 + \mu^2 \mathcal{R}(\mathbf{m}), \quad 2.53$$

where \mathbf{d} is the observed data which may be spatially incomplete or irregularly recorded, and $\mathcal{R}(\mathbf{m})$, the regularization term, is a linear operator acting on \mathbf{m} , and is different for each application. Regularization terms use some *a priori* information about the Earth model and serve to emphasize or attenuate models with certain geological features. A damped least squares solution is an example of regularization, $\mathcal{R}(\mathbf{m}) = \|\mathbf{m}\|^2$, which enforces the simplicity of the model. The first term in the right hand side of equation 2.53 is called data misfit. The minimization of the cost function, $J(\mathbf{m})$, the data misfit term, recovers the model to fit the data as much as desired. The parameter μ is a constant regularization weight and controls the amount of data fitting versus the model constraint.

Smoothing in the offset direction (the direction of the hyperbolic trajectories in each shot gather) is another constraint which can be performed on the LSPSM. This is accomplished with $\mathcal{R}(\mathbf{m}) = \|\mathbf{D}_h \mathbf{m}\|_2^2$, a quadratic norm, where \mathbf{D}_h , is an un-smoothing operator, and is the first derivative in the offset direction. Its action is mathematically expressed by the multiplication of the following matrix with a model vector

$$\mathbf{D}_h = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ 0 & 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}. \quad 2.54$$

The adjoint of un-smoothing operator is

$$\mathbf{D}_h^T = \begin{bmatrix} -1 & 0 & 0 & \dots & 0 \\ 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}. \quad 2.55$$

A first derivative operator such as equation 2.54 acts as a high pass filter. Therefore, minimization of a cost function with this regularization term is equal to penalizing high frequency contents. The solution of an LSPSM problem with smoothing as the regularization is the smooth least squares solution, \mathbf{m}_{SLS} :

$$\mathbf{m}_{SLS} = (\mathbf{G}^T \mathbf{G} + \lambda^2 \mathbf{D}_h^T \mathbf{D}_h)^{-1} \mathbf{G}^T \mathbf{d}. \quad 2.56$$

The tradeoff parameter, λ , controls the amount of regularization or weight of smoothing in the model and may be chosen by a trial and error method. I choose a high value, $\lambda = 10000$ for instance, and look at the first three LSCG iterations. If the error norm stays at 100%, I divide λ by 10. I repeat the procedure until I get an acceptable convergence rate in three LSCG iterations which is between 85% and 95% with a real data set and around 70% with a synthetic data set.

Smoothing and damping regularizations are quadratic regularizations in which model parameters are normally distributed (Sacchi, et al., 2004). Least squares inverse problems with quadratic regularization are linear. There are other useful regularization functions other than damping and smoothing least squares solution. Cauchy and l_1 norms are two non-quadratic regularization examples. In these types of regularizations the *a priori* distributions of the model is not Gaussian (Sacchi, et al., 2004). The cost functions resulting from non-quadratic regularizations are non-linear. The l_1 norm leads to a sparse solution by defining (Wang, 2005):

$$\mathcal{R}(\mathbf{m}) = \sum_{i=1}^N |m_i|, \quad 2.57$$

where N is the model size. The solution of a constrained least squares equation with this regularization term is:

$$\mathbf{m}_{l_1} = (\mathbf{G}^T \mathbf{G} + \lambda^2 \mathbf{Q}(\mathbf{m}))^{-1} \mathbf{G}^T \mathbf{d}, \quad 2.58$$

where \mathbf{Q} is a diagonal matrix with Q_{ii} , the diagonal entries, as

$$Q_{ii} = \begin{cases} |m_i|^{-1}, & \text{for } |m_i| > \epsilon \\ \epsilon^{-1}, & \text{for } |m_i| < \epsilon \end{cases} \quad 2.59$$

and ϵ is a threshold value.

The Cauchy norm is defined by (Wang, 2005):

$$\mathcal{R}(\mathbf{m}) = \sum_{i=1}^N \ln(1 + m_i^2 / \sigma_m^2), \quad 2.60$$

where N is the model vector size, and σ is a scale parameter.

Regularized least squares migration provides a high resolution seismic section and attenuates migration artifacts and acquisition footprint. However, there are two issues associated with replacing the conventional migration with the LSPSM. The main problem is that the convergence of an LSPSM solver to the correct solution strongly depends on the accuracy of the background velocity information. LSPSM is more sensitive to the accuracy of the velocity information than migration itself (Yousefzadeh and Bancroft, 2012d).

In addition to the dependency of the method on accurate velocity information, LSPSM is a more computer time intensive and memory consuming procedure than the migration alone. LSPSM attempts to solve a large system of linear equations. A modified version of the conjugate gradient method (CG) (Hestenes and Stiefel, 1952) method, the least squares conjugate gradient (LSCG) (Scales, 1987) method, has been widely used as a solver for the LSPSM equation (Nemeth et al., 1999; Duquet et al., 2000; Kuehl and Sacchi, 2001; and Yousefzadeh and Bancroft, 2010a, and 2010b). However, this is still an expensive method. In solving the equation with the LSCG method, each iteration runs more than twice as long as a conventional migration.

Each Kirchhoff prestack time migration includes calculation and/or application of the double square root (DSR) equation, proper migration amplitude weight function, the wavelet shaping factor (ρ filter), antialiasing filter, cross-correlation, and interpolation. In order to have a modeling operator which is exactly the adjoint of the migration operator, the adjoint of all these steps must be considered in the modeling. This procedure causes the modeling operator to be several times more expensive than the migration operator.

The ability to apply Kirchhoff migration on a small part of the data is a major advantage. This is possible by limiting the loops over CMPs and (pseudo) depth to the loop on the desired part of the section. An example would be around hydrocarbon traps. This reduces the migration

cost significantly. However, this advantage is extendable to only least squares poststack migration as performed by Jiang and Schuster (2003). Conventional LSPSM uses the whole data set and complete model grids simultaneously. Therefore, it is not possible to overcome only a part of the LSPSM cost by target orienting the method to a desired part of the image.

It is worth mentioning that poststack migration is an overdetermined problem which requires only implementation of a standard least squares method.

At the end of this section it is necessary to distinguish the difference between LSPSM and migration deconvolution (Hu et al., 2001; Yu and Schuster, 2003). As shown in equation 2.47, migration is a filtered (blurred) version of the Earth's true reflectivity. By LSPSM we are looking to find the inverse of the Hessian as exactly as possible. Ignoring regularization terms, the LSPSM image is obtained by multiplication of the inverse with migration image. Defining $\mathbf{G}^T \mathbf{G}$ as a blurring filter, $(\mathbf{G}^T \mathbf{G})^{-1}$ is considered as a deblurring filter (Yu and Schuster, 2003) and is related to migration and the true model by,

$$\mathbf{m} = (\mathbf{G}^T \mathbf{G})^{-1} \hat{\mathbf{m}}. \quad 2.61$$

Using migration deconvolution, Hu et al. (2001) and Yu et al. (2006) reduced migration artifacts by finding an approximation (instead of finding the exact inverse in LSPSM) to $(\mathbf{G}^T \mathbf{G})^{-1}$ using approximate Green's functions, and applying that approximation to the migration image. In their approximation they assumed that the velocity is varying only with depth where they could reduce the Green's function computational cost significantly (Hu et al., 2001). They showed improvement in the migration image after both poststack and prestack migration deconvolution at less cost than the LSPSM. Aoki and Schuster (2009) used the deblurring filter idea to obtain a relatively accurate *a priori* model for the constrained least squares migration. Also they implemented a deblurring filter as a preconditioner to increase the least squares migration convergence rate and reduce the computational cost.

2.6 LSPSM for the resolution of a synthetic example

I used the LSPSM to enhance the resolution of synthetic seismic data. In order to show the advantage of using LSPSM instead of migration, consider acquisition geometry with 16 sources and 96 receivers per source. The source interval is 187.5 m, and the receiver interval is kept at 15.625 m, to keep the fold number at 4 (except at the beginning and end of the survey line), on a

3km 2D velocity model. This geometry is shown in Figure 2.2. The temporal sampling rate is 2 milliseconds. This geometry is considered on an Earth model with horizontal, faulted, dipping, and folded layers with the velocity shown in Figure 2.3. The background velocity starts at 2000 m/s at the top, and with some variations ends at 6000 m/s at the bottom. Figure 2.4 shows the reflectivity resulting from this velocity model.

The synthetic data are prestack migrated (Figure 2.5) and least squares prestack Kirchhoff migrated using 20 iterations in the LSCG (Figure 2.6). The LSPSM produced a higher resolution image than the migration itself. Note especially the improved resolution of the fault plane.

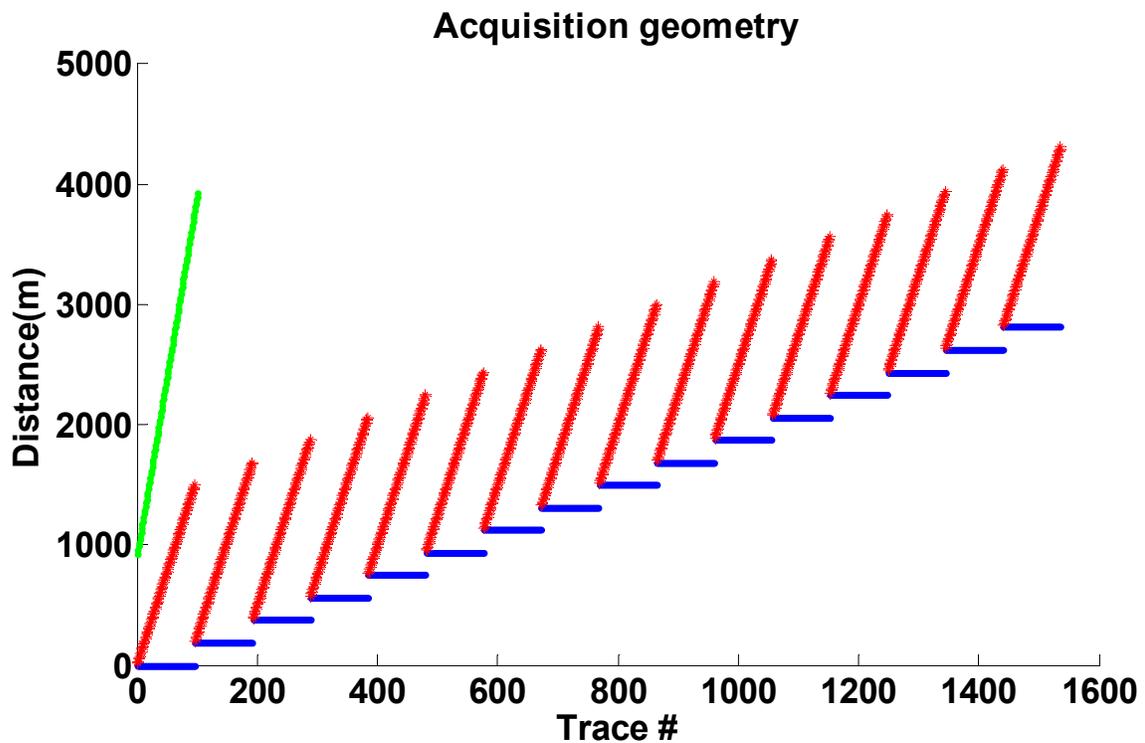


Figure 2.2 The acquisition geometry used in generating synthetic data. Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.

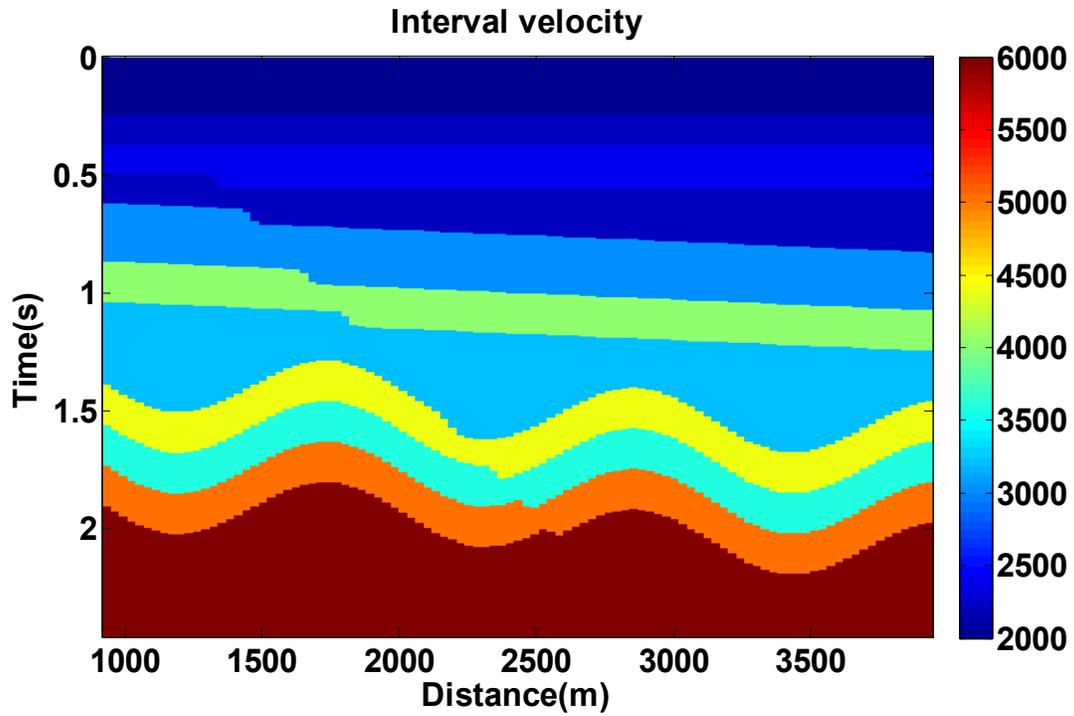


Figure 2.3 Interval velocity model used to generate synthetic data.

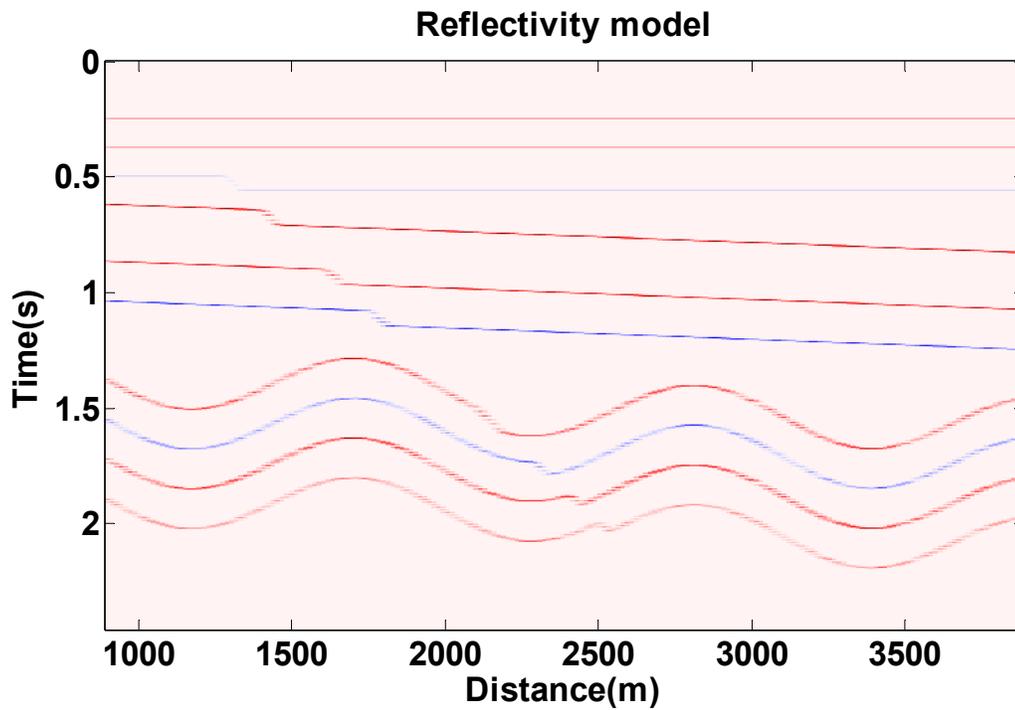


Figure 2.4 Reflectivity model corresponding to the velocity model of Figure 2.2.

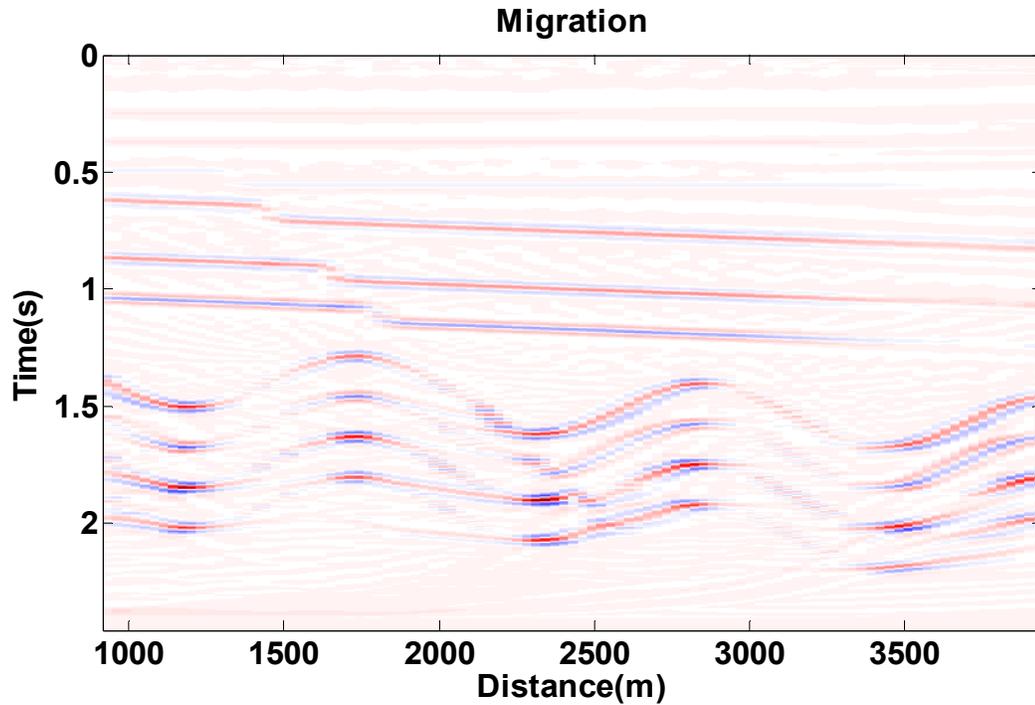


Figure 2.5 Migration of synthetic data. Comparing to Figure 2.4, shallow reflectors are weak and sinusoidal reflectors are not very well retrieved.

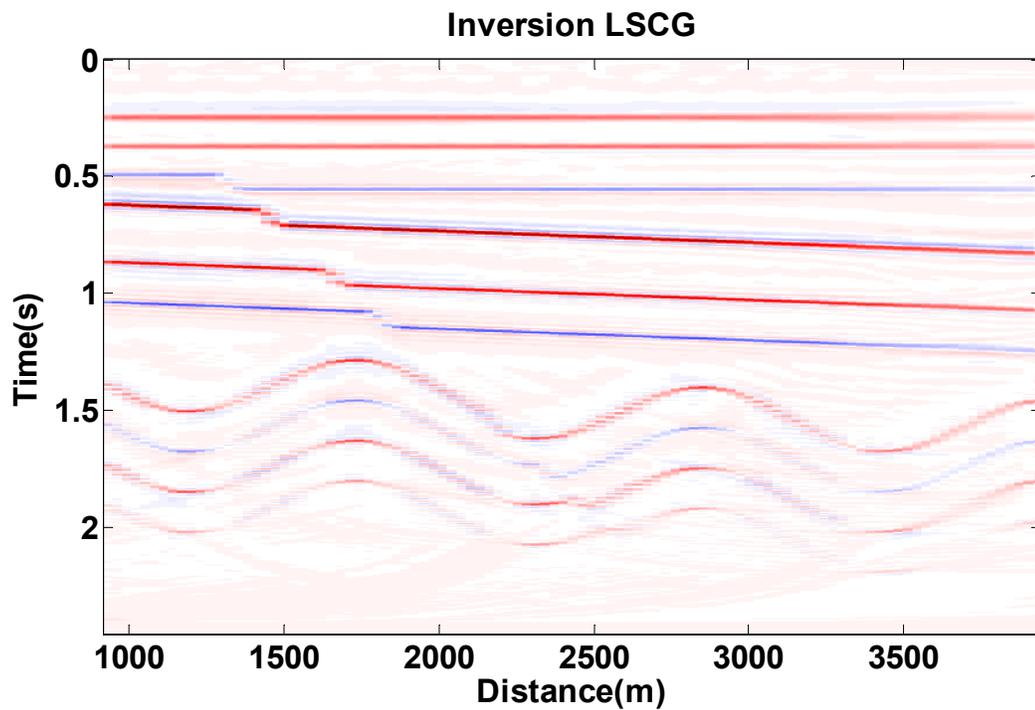


Figure 2.6 LSPSM of the synthetic data. Better presentation of the shallow reflectors and improved focusing of deep folding layers compared to migration.

The migration artifacts would be more attenuated if we used a regularized LSPSM with smoothing in the offset direction. The improvement is not obvious in the synthetic example, but it is shown on real data set at the end of this chapter.

Kirchhoff is probably the best migration method for irregularly sampled data. The Kirchhoff algorithm is able to handle irregularities in data sampling. However, sampling data more sparsely increases the migration artifacts. LSPSM is able to attenuate the artifacts related to data acquisition or spatially decimation of the seismic data. To show these, synthetic data are decimated by randomly eliminating 80% of traces. Figure 2.7 shows the geometry of decimated data. Figure 2.8 compares the acquisition geometry and geometry of the spatially decimated traces, in shot # 7. The migration of the decimated data set is shown in Figure 2.9, and the LSPSM in Figure 2.10, in which most of the migration artifacts are attenuated.

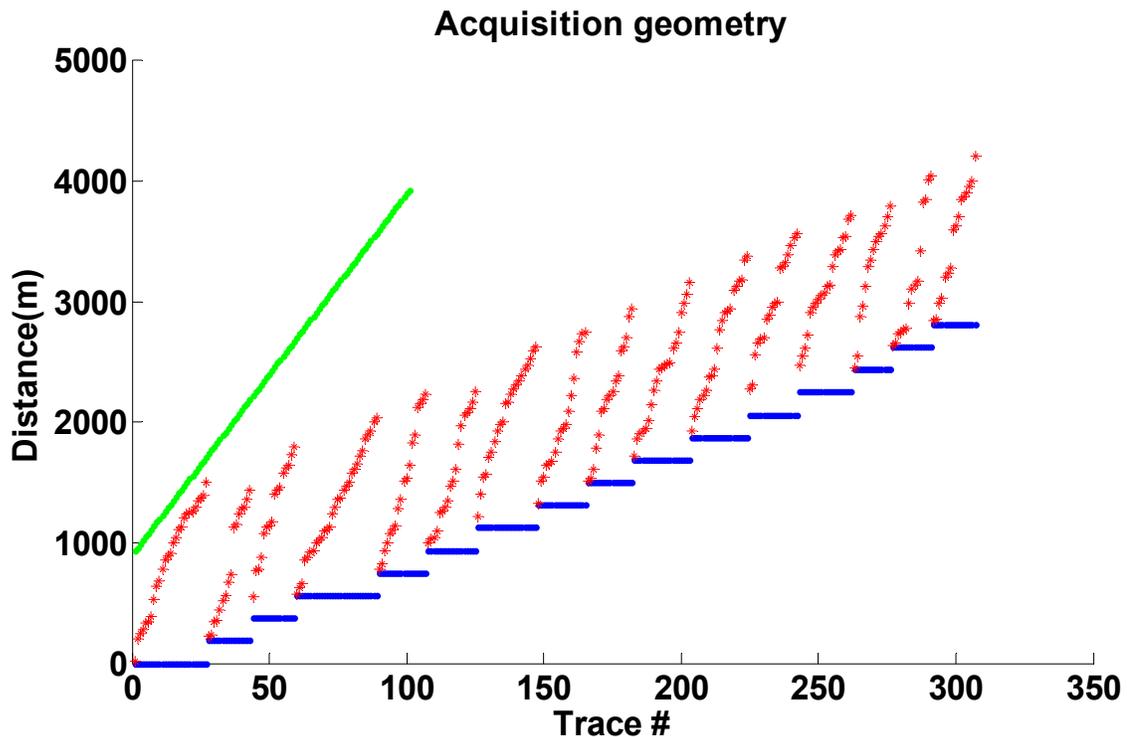


Figure 2.7 Acquisition geometry after decimating 80% of traces randomly. Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.

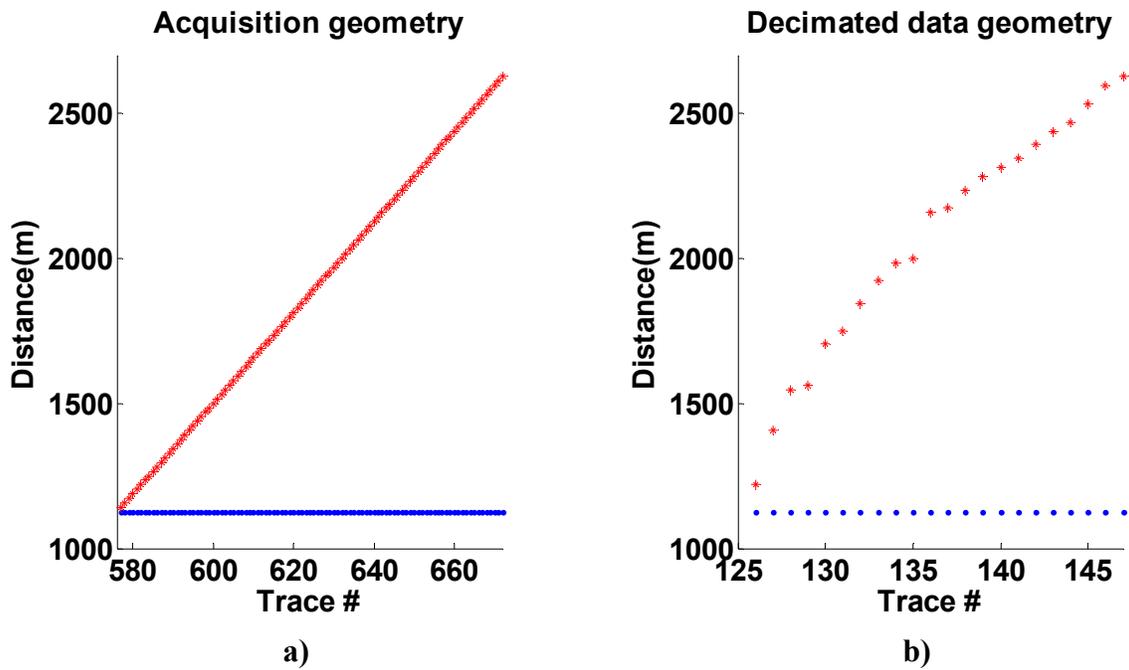


Figure 2.8 Comparison between the a) complete, and b) decimated acquisition geometry for the shot #7. blue: sources, red: receivers.

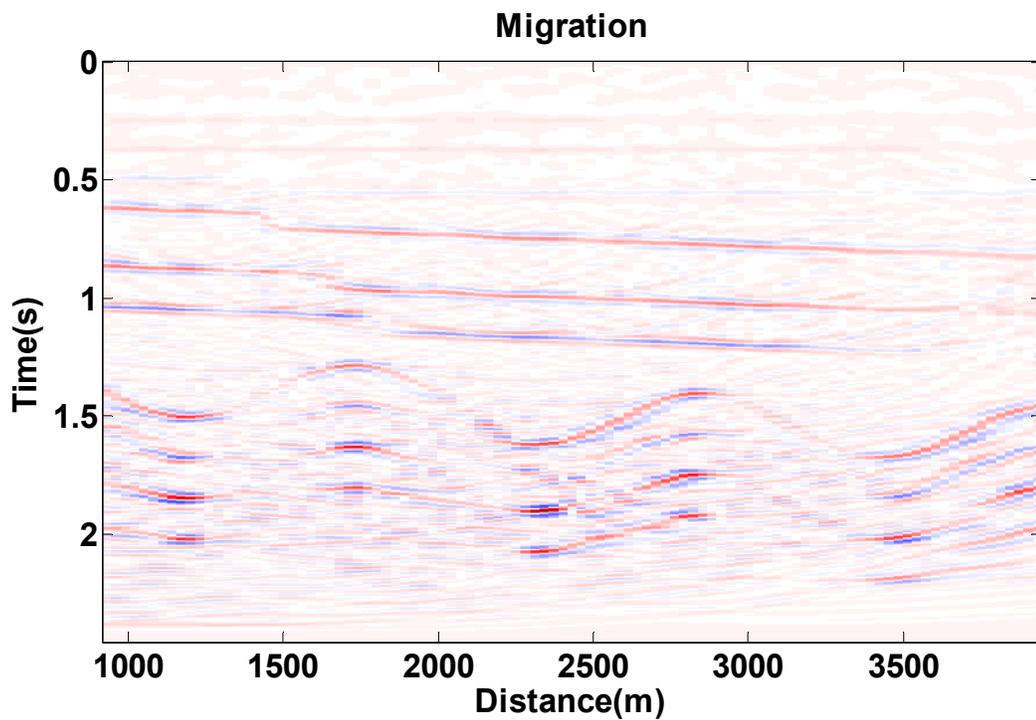


Figure 2.9 Migration of decimated synthetic data, 80% of traces are randomly deleted.

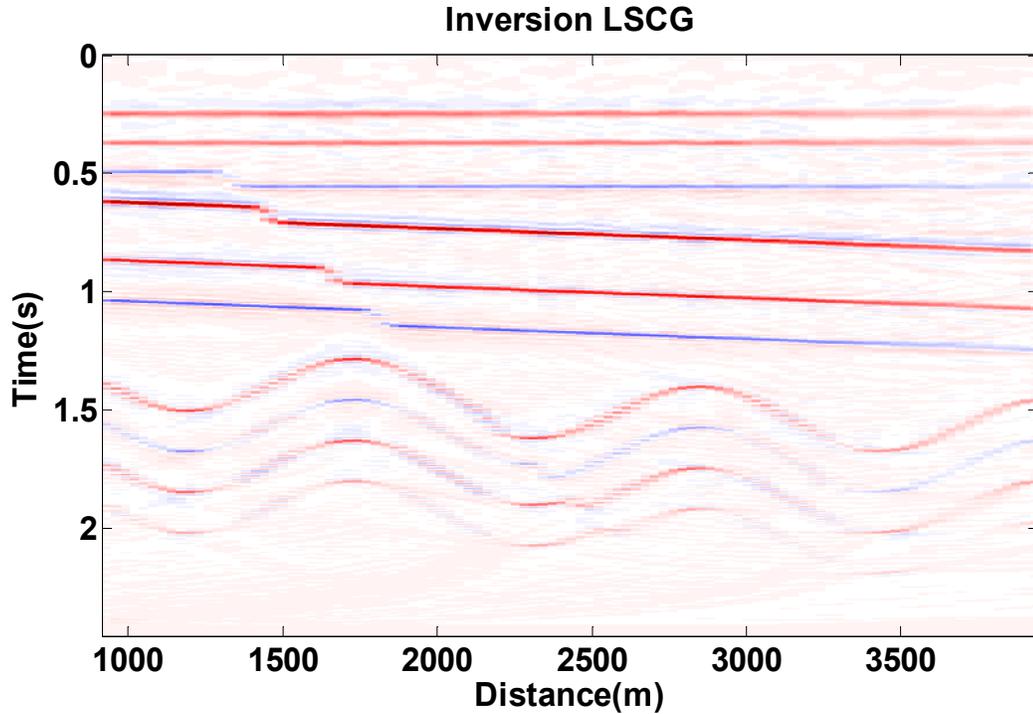


Figure 2.10 LSPSM of 80% randomly decimated data.

As seen in this synthetic example, LSPSM is able to attenuate migration artifacts which can be due to data incompleteness or irregularities.

The above mentioned comparison is based on the quality of the resulting image. Comparison between migration and LSPSM can also be quantified by measuring the relative amplitudes of the trace in the corresponding final images and comparing them with the true model. To do so, I created a stacked trace of the model, migration, and LSPSM by averaging five adjacent traces in the position of 2280 to 2400 m. The first 1.4 s of the resulting trace for the model, migration, and LSPSM of the complete and decimated data are shown in Figure 2.11. In both cases, migration shows higher amplitudes in the deeper part and lower amplitudes in the shallower part where in the model highest amplitude corresponds to the reflector at 0.8 s. In both scenarios, with the complete and decimated data, LSPSM amplitudes are closer to the model amplitudes.

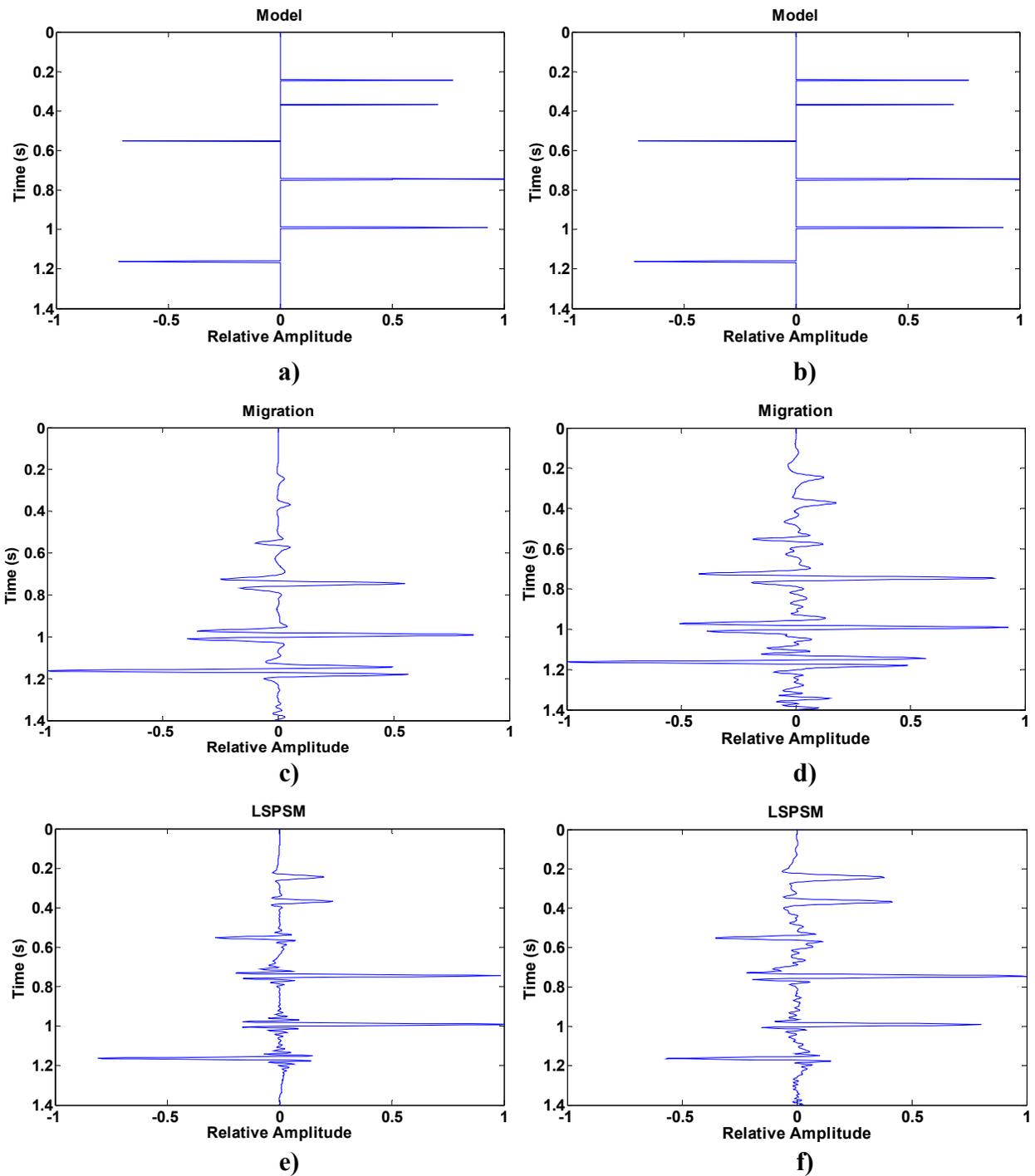


Figure 2.11 a) and b) model, c) and d) migration, and e) and f) LSPSM relative amplitudes of the complete (left) and decimated (right) synthetic data for the position of 2280 to 2400 m.

LSPSM images in Figure 2.6 and Figure 2.10 are calculated with 20 iterations in LSCG. The convergence rate in the LSCG method is an appropriate way of measuring the ability of the LSPSM to find the best reflectivity image. In this thesis, the convergence rate is the relative Euclidean norm of the difference between the original data and the data achieved by forward modeling of the LSPSM image after each iteration. Figure 2.12 shows the normalized convergence rate when using complete and decimated data. The residuals converge to less than 4% in 20 iterations in both cases.

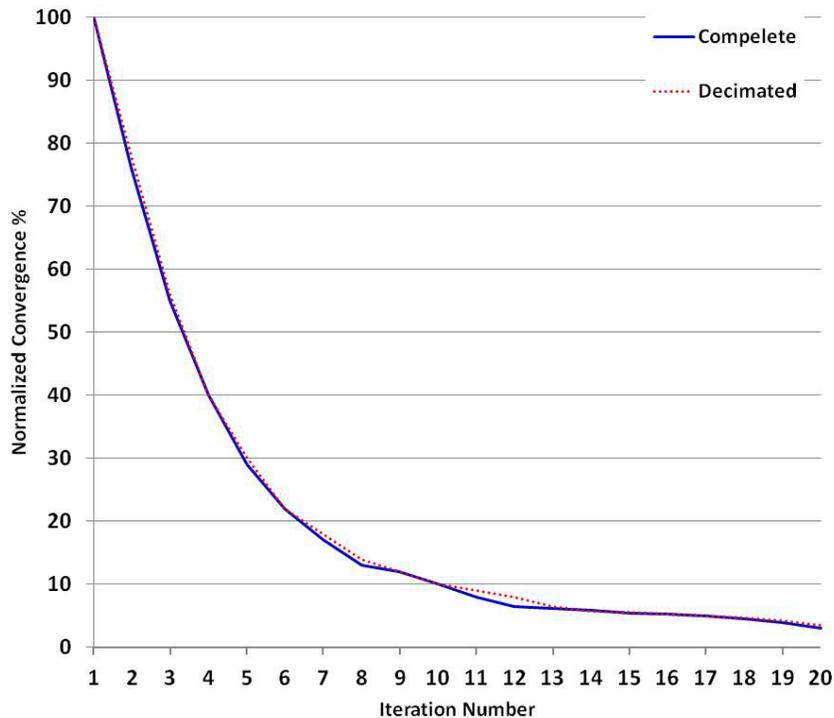


Figure 2.12 Convergence rate of LSPSM in 20 iterations with the complete data (solid line) and the decimated data (dotted line). Both cases converge to 4% in 20 iterations.

2.7 LSPSM for data reconstruction

Seismic data interpolation is another benefit of using LSPSM. There are many processes in seismic data analysis which require regularly sampled data. Some methods of multiple attenuation, and some powerful methods of migration that are mostly being implemented in the frequency domain, require regularly sampled data. Replacing missing traces with traces having a zero amplitude for all samples introduces some artifacts in the resulting images. Amplitude

Verses Offset (AVO) and Amplitude Verses Azimuth (AVAz) studies for the reservoir characterization are two more examples.

Nemeth et al. (1999) suggested data reconstruction and interpolation using LSPSM. Since the resulting image of LSPSM is a high resolution image, it can be used for data reconstruction. Suppose \mathbf{G}_i and \mathbf{d}_i show the forward modeling operator for incomplete data, and incomplete data itself, respectively. Then the damped least squares image may be calculated by,

$$\mathbf{m}_{DLS} = (\mathbf{G}_i^T \mathbf{G}_i + \mu^2 \mathbf{I})^{-1} \mathbf{G}_i^T \mathbf{d}_i. \quad 2.62$$

Then, complete data, \mathbf{d} , are computed by forward modeling of \mathbf{m}_{DLS} :

$$\mathbf{d} = \mathbf{G} \mathbf{m}_{DLS} = \mathbf{G} (\mathbf{G}_i^T \mathbf{G}_i + \mu^2 \mathbf{I})^{-1} \mathbf{G}_i^T \mathbf{d}_i. \quad 2.63$$

This method of data reconstruction is robust. However, this method is more expensive than the other methods of data reconstruction which usually use properties of the Fourier transform and works with a small part of data (one shotgather for instance) at each time step (for methods see: Spitz, 1991; Porsani, 1999; and Gulunay, 2003). Using LSPSM, the observed data must tolerate several migration and modeling runs in the LSCG scheme until the algorithm converges to a reasonable model, and then there is the additional cost of forward modeling with the new geometry of the sources and receivers.

Another disadvantage of this method of data reconstruction is its strong dependency on the accuracy of background velocity information. A small perturbation from the true velocity prevents this method of data reconstruction to work properly even if it gives back a high resolution LSPSM image. More details and examples can be found in Yousefzadeh et al. (2011, 2012).

To show the ability of data reconstruction with LSPSM, I used the image of LSPSM in Figure 2.10 for data reconstruction when 80% of data are decimated irregularly. Results in the first 50 traces from shotgather # 7 are shown in Figure 2.13. Panels a, b, c, and d show the original, decimated, reconstructed, and residual data, respectively. With having only 20% of the data, LSPSM proves itself a powerful method to reconstruct the missing data.

Figure 2.14 compares the frequency-wavenumber (FK) spectra of the original, decimated, and reconstructed data, and their residuals. The decimated data are highly aliased, yet reconstruction has removed the aliasing effects and residuals show that the data reconstruction procedure is able to retrieve all frequencies and wavenumbers.

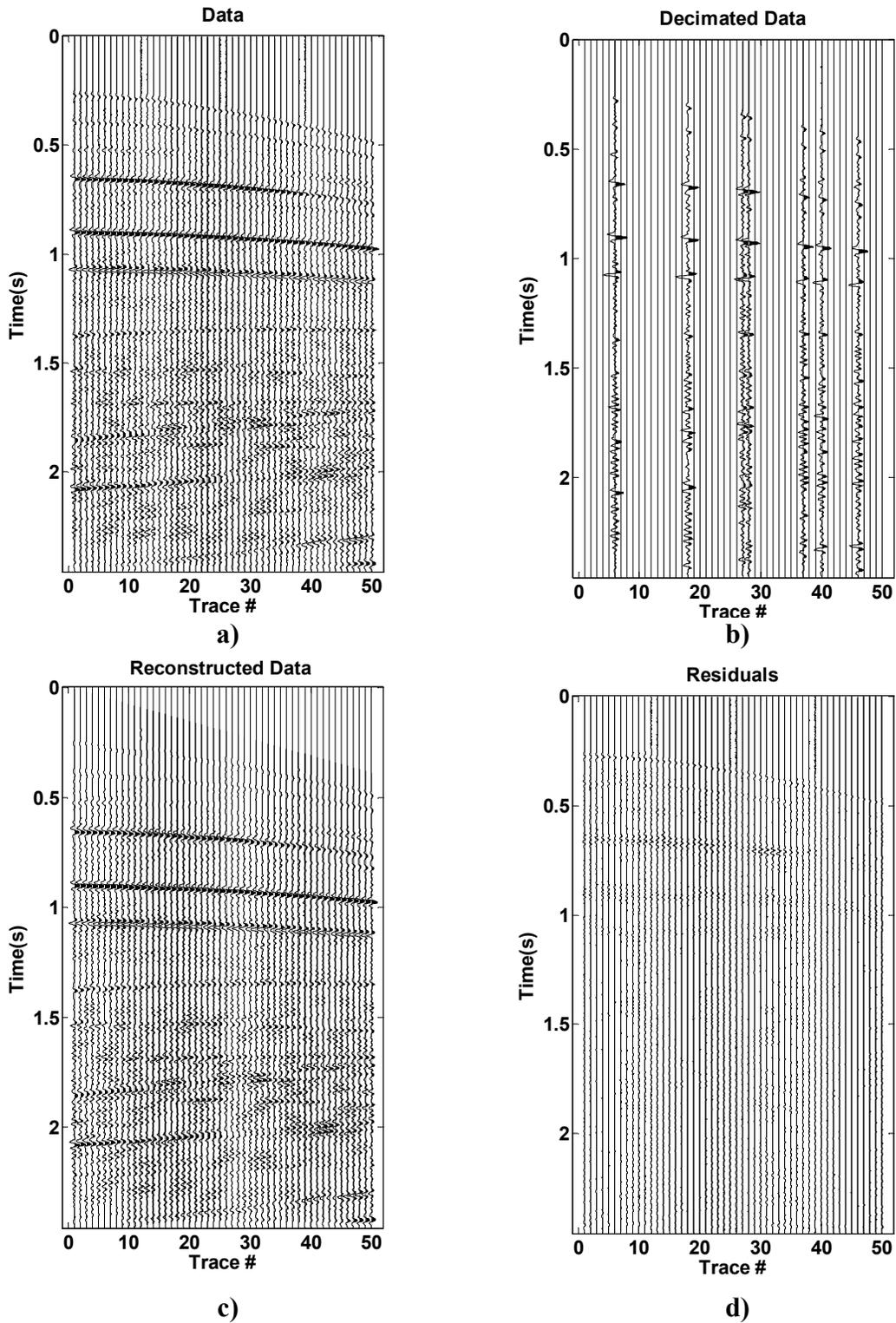


Figure 2.13 Data reconstruction by LSPSM. a) data, b) 80% decimated data, c) reconstructed data, d) residuals. The first 50 traces from shotgather # 7 are shown.

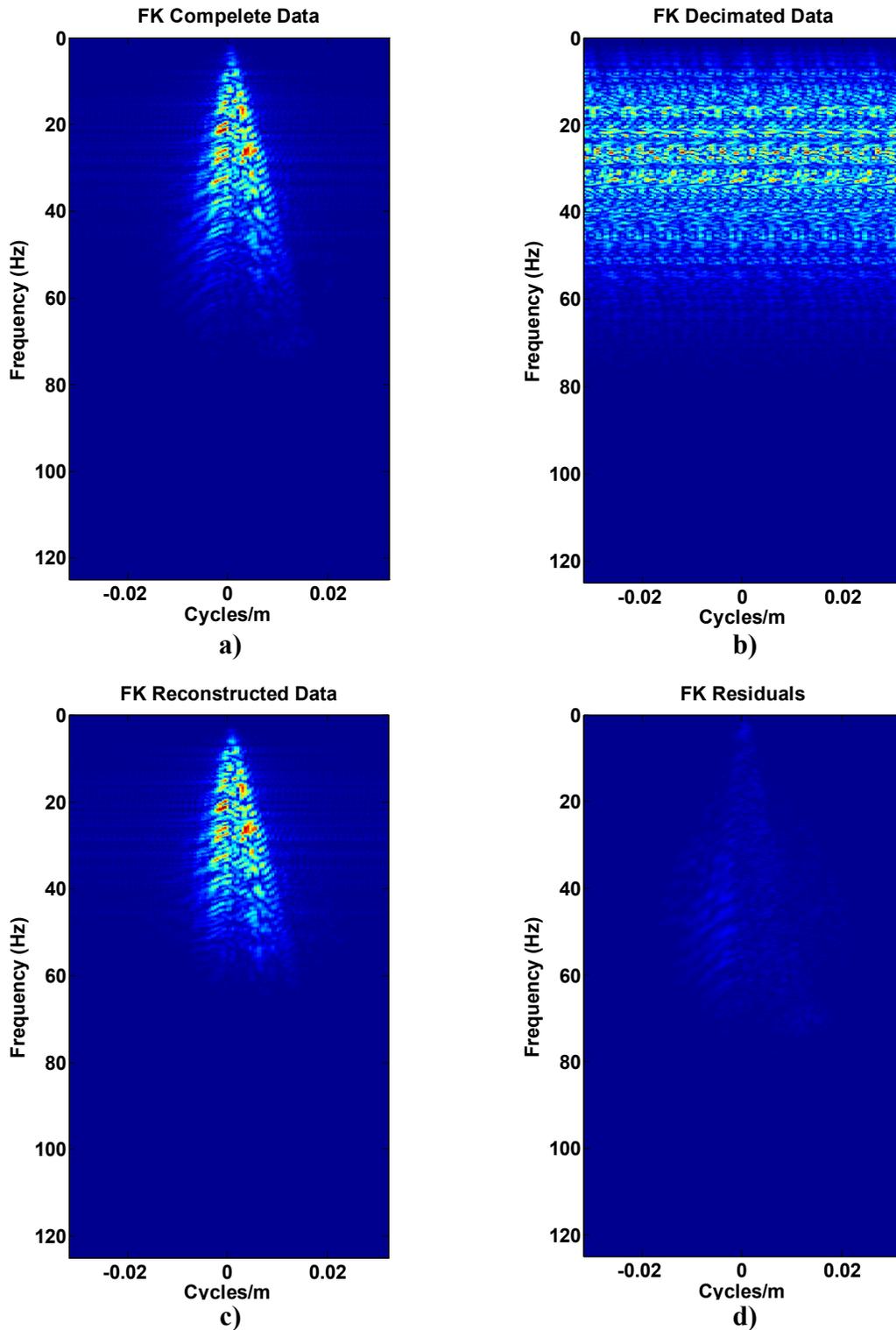


Figure 2.14 FK spectra of the data in Figure 2.13. Spectrum for a) original data, b) decimated data, c) reconstructed data, d) residuals.

2.8 Real data considerations and example

Resolution enhancement and data reconstruction of the synthetic data sets by LSPSM are shown in the previous sections. The results of applying the LSPSM method on a real data set may not be as good as on synthetic data. Real data must be edited, phase and amplitude corrected, static corrected, trace equalized, surface waves and multiple attenuated before using in many imaging algorithms including migration and LSPSM. Even with the best methods of processing, good quality data, and accurate information about source wavelet and velocity from different sources, it is impractical that LSPSM LSCG convergence goes as low as 30%. This is mostly due to the fact that migration and modeling algorithms do not use an exact representation of the real propagation of seismic waves through the Earth's layers.

LSPSM is performed on a real data set acquired in the NE of British Columbia (NEBC), Canada. A relatively large, long offset, and wide azimuth 3C/3D data volume was recorded for reservoir characterization and to suppress possible acquisition footprint. Irregularities in surface topography and the effects of thick organic material at the surface and shallow glacial tills of variable thickness make subsurface seismic imaging in this area difficult. However, the analysis of near surface velocity irregularities and statics corrections is not the main interest in this research. I used a high density P-wave 2D line, acquired for refraction analyses out of the 3D volume, which was preprocessed and was ready for velocity analysis, NMO correction and stacking or migration. The acquired data are relatively dense and regular. I intentionally decimated the data to produce a migration image that suffers from acquisition footprint. The data include more than 65000 traces in 220 shots with 10 m receiver spacing and 60 m source interval. The sampling rate is 2ms and the recording time is 3s.

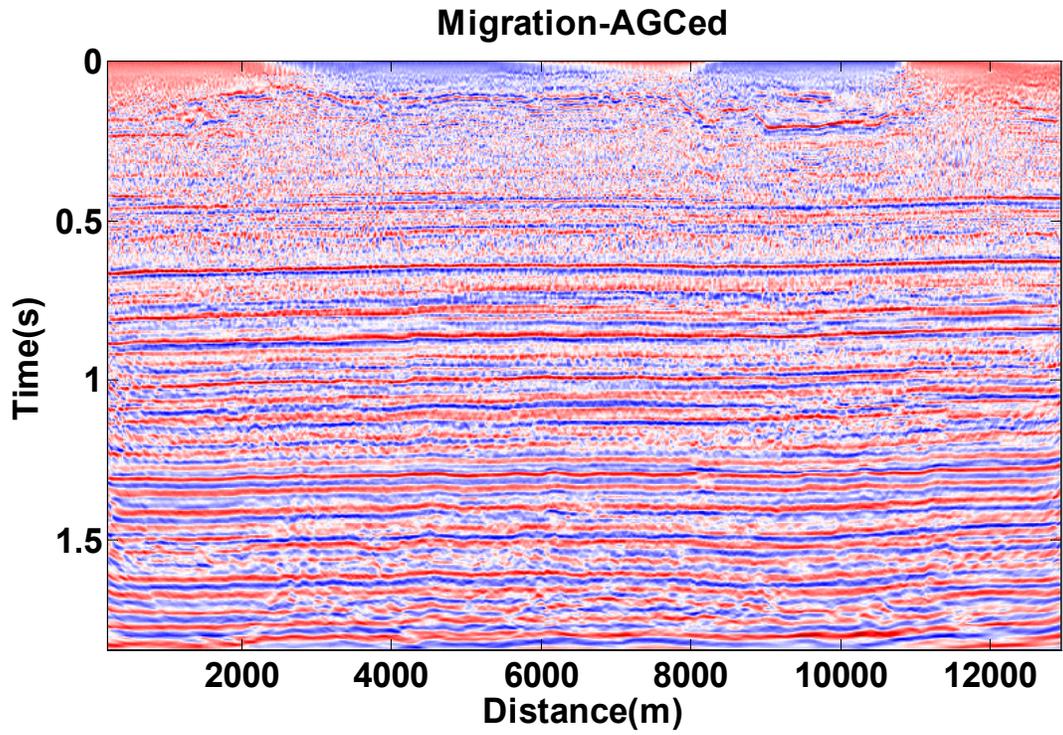
The NEBC data set has strong reflectors at relatively shallow depths which are responsible for the presence of some weak interbed multiples. Since conventional Kirchhoff time migration/modeling methods are not able to migrate or model multiples, it is necessary that multiple-free data be used for Kirchhoff migration and LSPSM. The NEBC data set was multiple attenuated using the Radon transform (Hampson, 1986). The residuals move out of the NMO corrected CMP gathers using a correct velocity for primary reflections can be approximated by parabolas. Radon transform is used to map data into (t, p) domain, where p is the curvature of the NMO corrected events. Multiples which are separated from primary reflections in the (τ, p)

domain can be easily muted. This method is able to remove multiples with the velocities that are significantly different from the velocity of primaries at the same depth.

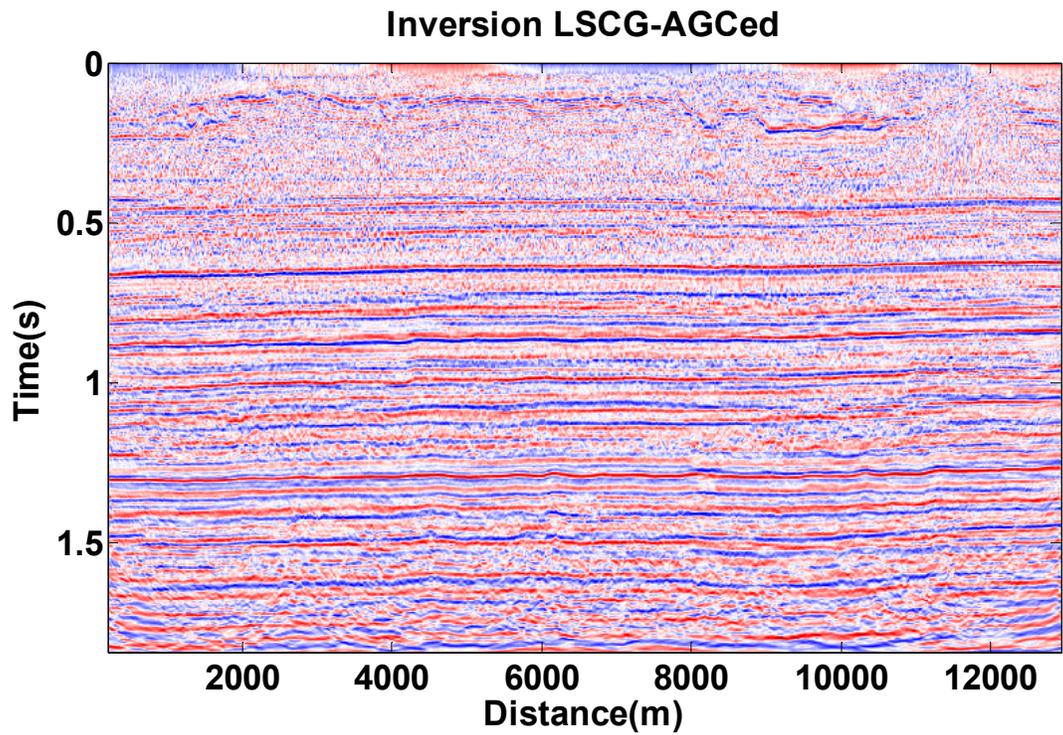
I used Seismic Unix software to multiple attenuate the NEBC data set using Radon transform. NEBC data have some high velocity interbed multiples that cannot be removed by the Radon transform multiple attenuation method. I performed migration and LSPSM on the multiple attenuated data. Results are shown in Figure 2.15. The horizontal axis in all figures in this section shows the distance from the first CMP position. There is a little resolution enhancement in the shallower part of the image by using LSPSM instead of the Kirchhoff migration. This is due to dense and regular sampling of the data and low level of acquisition footprint. There is no improvement in the LSPSM image below 1 s which can be related to the presence of the high velocity interbed multiples. I decimated the data by randomly removing 90% of the traces. Figure 2.16 shows the migration and LSPSM of the decimated data. Decimation of data produced many acquisition artifacts in the migration image as expected. LSPSM is able to remove these artifacts and return a high resolution image especially at the top 1 s of the image.

This image is used for the reconstruction of decimated data. Results of data reconstruction are shown in Figure 2.17. Figure 2.17d shows the residuals, the difference between original and reconstructed data, which shows the success of data reconstruction. Since the multiples remained in data after multiple attenuation using Radon transform have a velocity that is close to the velocity of primaries, I assume that Kirchhoff LSPSM was able to reconstruct these multiples as well as the primaries.

Regularized LSPSM with smoothing in the offset direction, with $\lambda = 1$, is performed on the decimated data, and the result is shown in Figure 2.18. Comparison between Figure 2.18 and Figure 2.16b shows slight improvement in the LSPSM image. A close-ups of the LSPSM image (Figure 2.16b) and regularized LSPSM image (Figure 2.18) are shown in Figure 2.19 for better comparison. The reflections seem to be more continuous in the lower part of the regularized LSPSM image.

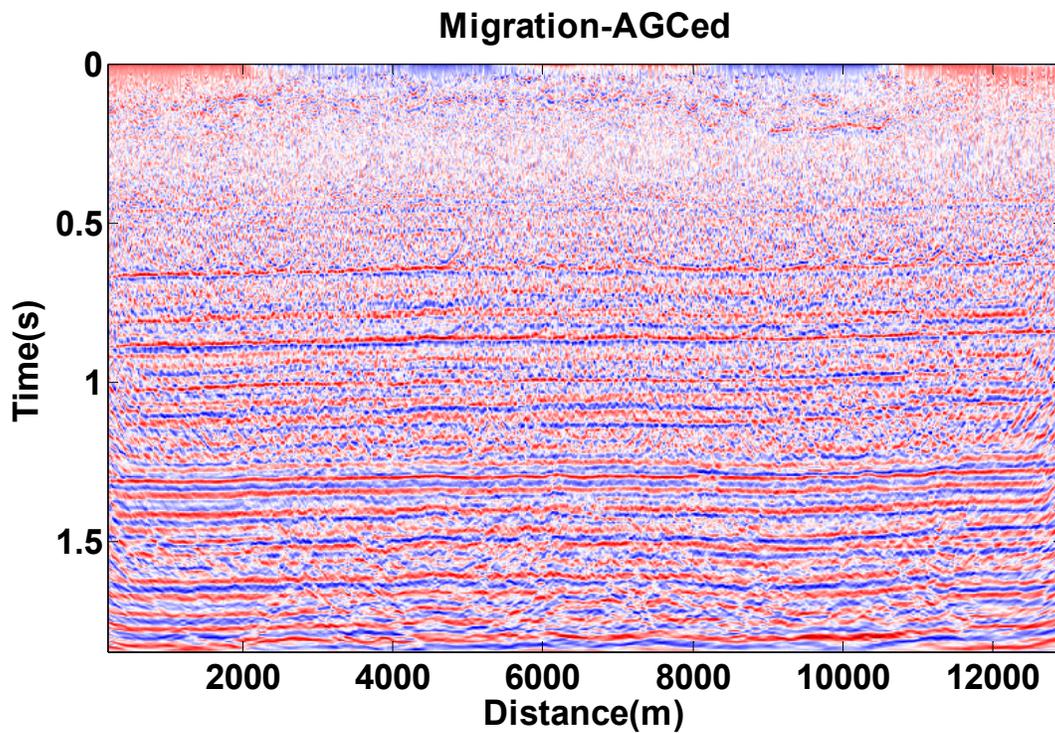


a)

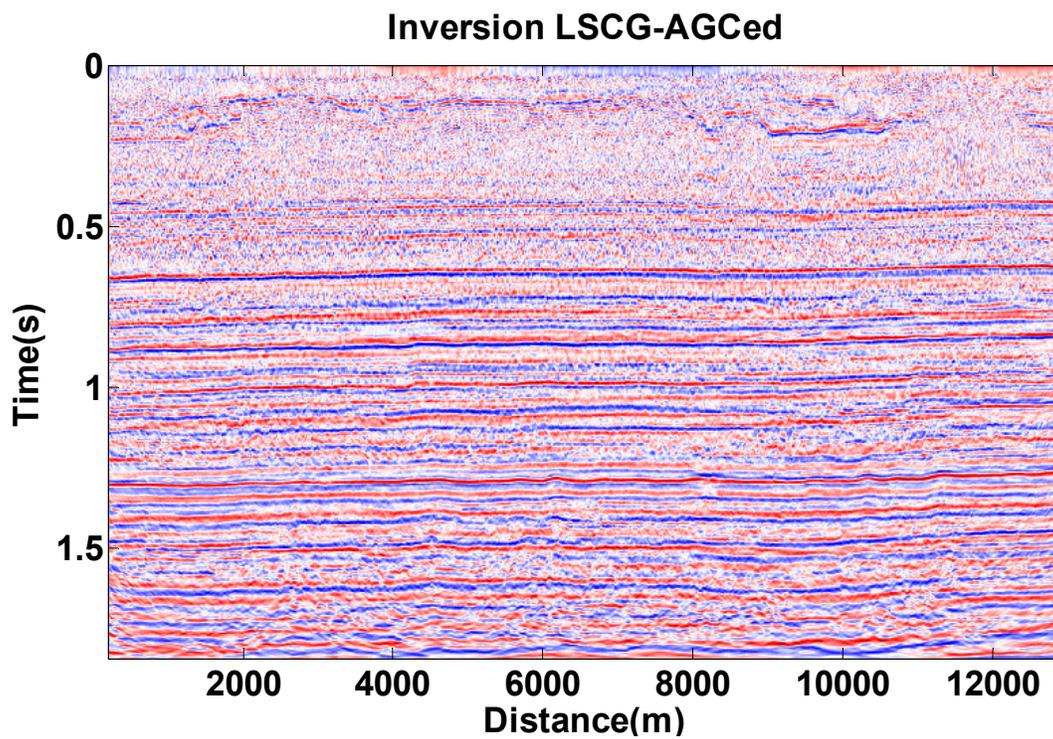


b)

Figure 2.15 Images resulting from a) migration and b) LSPSM of NEBC data set.



a)



b)

Figure 2.16 a) migration and b) LSPSM of decimated synthetic data, 90% of traces are randomly deleted.

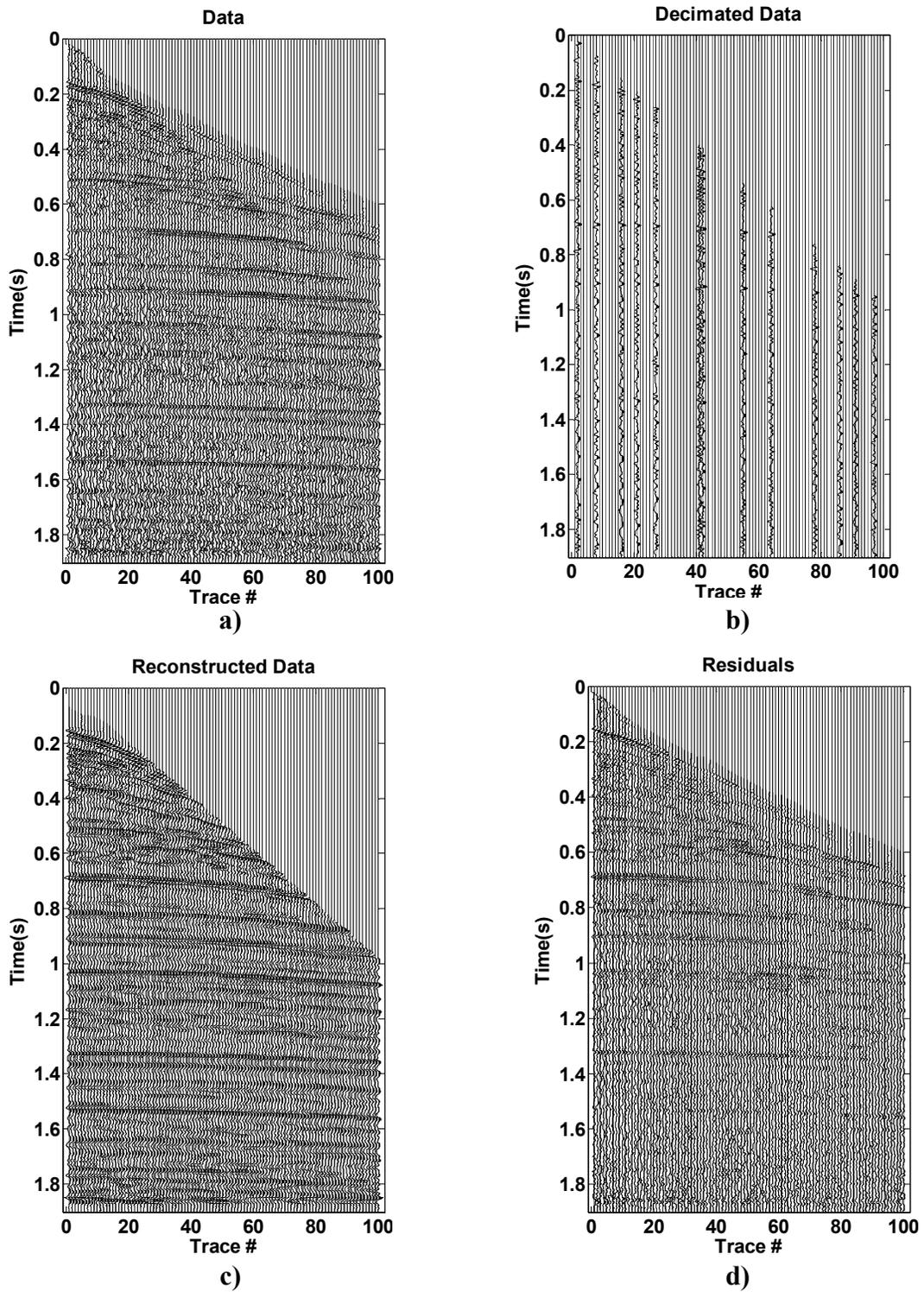


Figure 2.17 Data reconstruction by LSPSM. a) data, b) 90% decimated data, c) reconstructed data, d) residuals.

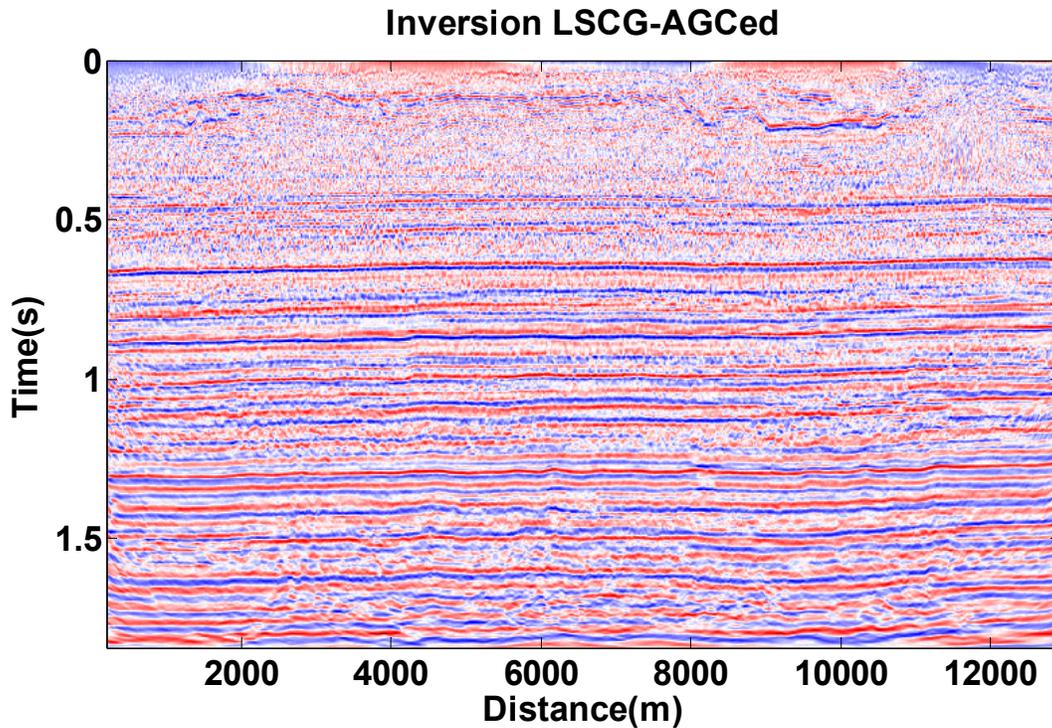


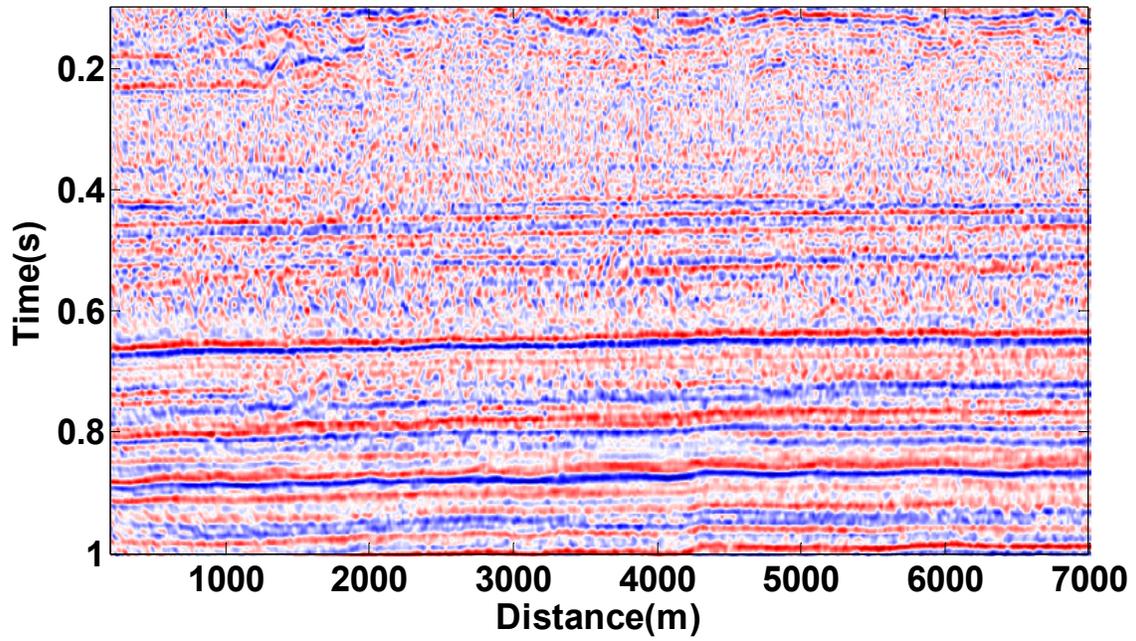
Figure 2.18 Regularized LSPSM of decimated data.

2.9 Summary and Conclusions

This chapter began with a review on the derivation of Kirchhoff migration. Discrete inverse theory has been explained in detail. LSPSM as an inverse problem has been described. With the help of synthetic and real data I showed how least squares Kirchhoff migration is able to attenuate the acquisition footprint and migration artifacts and produce a high resolution image that can be used for the construction of missing seismic traces.

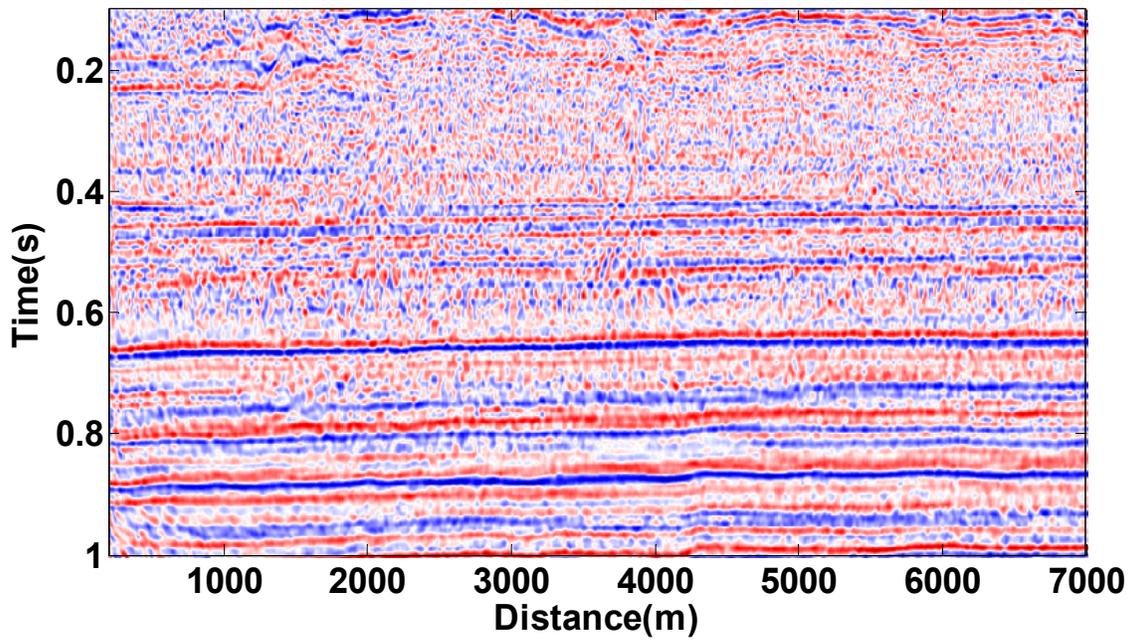
Real data used in this study are from a regularly and densely sampled data set. It is shown how LSPSM can be used for the reconstruction of 90% randomly decimated traces.

Inversion LSCG-AGCed



a)

Inversion LSCG-AGCed



b)

Figure 2.19 a) LSPSM and b) Regularized LSPSM of the decimated data set.

Chapter Three: SOLVING THE LSPSM EQUATION WITH MULTIGRID AND MULTILEVEL METHODS

3.1 Introduction

As explained in Chapter One, considering each subsurface point as a scatterpoint, Kirchhoff prestack migration collapses all diffracted energies to the scatterpoints. Kirchhoff prestack modeling, on the other hand, spreads the energy from the scatterpoints to the seismic traces. Kirchhoff prestack time modeling and migration use the double square root (DSR) equation to calculate the modeling and migration time for each sample in seismic traces. Proper migration amplitude weighting functions, filtering, interpolation, dip, and aliasing controls are also necessary considerations in Kirchhoff migration to obtain the best focused image.

Since Kirchhoff modeling and migration are linear operators, they can be expressed in a matrix-vector multiplication scheme. A 2D seismic data is a matrix with each trace as a column. Earth's reflectivity model is a matrix which shows the Earth's layers. These matrices may be simply converted to vectors. By defining seismic data by vector \mathbf{d} , Kirchhoff modeling operator by matrix \mathbf{G} , and the Earth's reflectivity model by vector \mathbf{m} , the forward modeling procedure is expressed by,

$$\mathbf{d} = \mathbf{G}\mathbf{m}. \quad 3.1$$

The modeling matrix, \mathbf{G} , is a multidimensional matrix. It consists of a diffraction hyperbola and a wavelet in a matrix for each grid point in the reflectivity model. Therefore, \mathbf{G} is a four dimensional matrix for a 2D seismic experiment (and reflectivity model). Using tensor notation the forward modeling for a 2D seismic survey is expressed by

$$d_{ij} = G_{ijkl}m_{kl}. \quad 3.2$$

Note that G is a 4th-order tensor and d and m are 2nd-order tensors. Multiplication in equation 3.1 is possible when \mathbf{G} is properly converted to a 2D matrix. The proper conversion of \mathbf{G} to a 2D matrix is explained in section 3.2. Then the modeling equation 3.1 with the tensor notation may be shown by

$$d_{\alpha} = G_{\alpha\beta}m_{\beta}. \quad 3.3$$

where G became a 2nd-order tensor and d and m became 1st-order tensors.

The inverse of calculation in equation 3.1 should return the reflectivity. However, calculation of the inversion is not always a possible or an easy task. The transpose (adjoint) of \mathbf{G} , \mathbf{G}^T , is easier

to calculate than the exact inverse. Migration methods are essentially transpose processes. In the case of Kirchhoff modeling, the transpose process is prestack Kirchhoff migration which is defined by,

$$\hat{\mathbf{m}} = \mathbf{G}^T \mathbf{d}, \quad 3.4$$

Where the matrix $\hat{\mathbf{m}}$ is the migrated image and \mathbf{G}^T is the migration operator.

In least squares migration, \mathbf{m} is estimated by minimizing the difference between the observed data, \mathbf{d} , and the modeled data, $\mathbf{G}\tilde{\mathbf{m}}$, where $\tilde{\mathbf{m}}$ is an approximation to \mathbf{m} . Solving standard least squares Kirchhoff migration, \mathbf{m}_{LS} , includes finding the inverse of matrix $\mathbf{G}^T \mathbf{G}$, and multiplication of the result with the migration image matrix, $\mathbf{G}^T \mathbf{d}$,

$$\mathbf{m}_{LS} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{d}. \quad 3.5$$

In this chapter the feasibility of using standard multigrid and multilevel methods for solving the linear system of LSPSM equations is investigated. I start in section 3.2 with introducing a geometric method for construction of the modeling operator, \mathbf{G} , and the migration operator, \mathbf{G}^T , in matrix form, that contain diffractions. I show how \mathbf{G}^T and \mathbf{G} matrices can be calculated inside a Kirchhoff migration algorithm. Once the matrices are constructed, modeling and migration can be repeated just by a matrix-vector multiplication. This may be useful in some applications that require several migration and modeling runs on data. Least squares migration is one of these applications. The shape of the least squares migration matrix, $\mathbf{G}^T \mathbf{G}$, is presented as well. Visualizing the $\mathbf{G}^T \mathbf{G}$ matrix is useful since we are looking for a faster method to calculate its inverse. By the end of section 3.2, I show that $\mathbf{G}^T \mathbf{G}$ is a dense and diagonally non-dominant matrix which explains why solving the least squares Kirchhoff migration equation is an expensive procedure. I then show that the matrix $\mathbf{G}^T \mathbf{G}$ is too large to work with.

In sections 3.3 and 3.4 the feasibility of using standard multigrid and multilevel LSCG methods for solving the LSPSM equation in order to speed up the process or enhance the resolution of the reflectivity image is investigated.

The discussion on the CG and multilevel CG methods to solve LSPSM is presented in section 3.5.

Section 3.6 is the summary and conclusion of this chapter.

3.2 Kirchhoff prestack modeling and migration operators in matrix form

To explain my method of creating the modeling and migration matrices, I start with constructing the convolution matrix using a simple example in subsection 3.2.1. I then extend my construction method to the Kirchhoff modeling and migration matrices in subsections 3.2.2 and 3.2.3. I show how to change a migration algorithm to create the corresponding migration matrix. Examples of modeling and migration matrices are used to show the difficulties of working with the matrix forms in subsection 3.2.4. I conclude that working with the explicit forms of the modeling and migration is currently impractical. The large size of modeling and migration matrices is a reason that the multigrid method is not a feasible solver for LSPSM equation. The second problem when using multigrid method to solve LSPSM equation is discussed in the sections 3.3 and 3.4.

3.2.1 *Explicit form of the wavelet matrix, a starting example*

In equation 3.1, matrix \mathbf{G} is the modeling operator, and \mathbf{m} is the reflectivity model. The seismic source does not produce a perfect impulse with zero duration. The real pulse or source wavelet always has time duration and is band limited in the frequency domain. Even if the source produces a perfect spike, the high frequency content of the pulse will be attenuated rapidly while it travels through the Earth's layers. Consequently, a relatively low frequency pulse penetrates down into the ground and reflects back. The seismic wavelet is non-stationary and its shape changes with time when traveling through the Earth's layers by losing more and more high frequency content. This effect must be considered in any seismic modeling method, such as Kirchhoff or finite difference modeling, by introducing a non-stationary wavelet to the model or data. I chose the latter approach. Implementing a non-stationary wavelet is not a very convenient procedure since it requires adequate knowledge about frequency dispersion rates of the underground layers. For simplicity, the wavelet is assumed to be stationary in this dissertation. A time-variant deconvolution before migration or LSPSM might help with the validity of this assumption. However, it is not investigated during this research. Therefore, I consider the

modeling operator in equation 3.1 as the multiplication of a stationary wavelet matrix¹, \mathbf{W} , with the Kirchhoff modeling operator without considering the wavelet, \mathbf{g} ,

$$\mathbf{G} = \mathbf{W}\mathbf{g}. \quad 3.6$$

Consider the Earth as a non-dispersive and time invariant linear system. The response of the Earth's reflectivity to an assumed stationary wavelet can be expressed as the convolution of the wavelet with the reflectivity series. Defining $h[n]$ as the impulse response of a discrete time invariant system, convolution, $*$, describes the system's response to the input signal $x[n]$:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[n-k]h[k], \quad 3.7$$

where $y[n]$ is the convolved signal. If x and h are signals with nx and nh samples, respectively, then y consists of $nx + nh - 1$ samples. Therefore, considering N_w as the length of the source wavelet and N_t as the number of seismic trace samples, the resulting convolved trace includes $N_t + N_w - 1$ samples. Consequently, if seismic data include M_{trc} traces with N_t sample per trace, the result will be an $N_t + N_w - 1$ by M_{trc} matrix.

Convolution of the seismic source wavelet with data can be performed by either using a convolution algorithm (Table 3-1) or multiplication of the convolution matrix, \mathbf{w} (MathWorks, 2012), or circulant matrix with a data matrix. All three methods are well-known. In order to multiply the circulant matrix with a data matrix of size N_t by M_{trc} , the data matrix must be zero padded to a be a $(N_t + N_w - 1)$ by M_{trc} matrix, where N_w is the number of wavelet samples. Circulant matrix is a $(N_t + N_w - 1)$ by $(N_t + N_w - 1)$ matrix. The size of the convolved data will be $(N_t + N_w - 1)$ by M_{trc} . To construct a circulant matrix, a zero matrix of appropriate size is chosen, then wavelet elements replace the first zeros in the first column. Other columns of the circulant matrix are each cyclic permutations of the wavelet with an offset equal to the index of that column (Wikipedia, 2012).

In order to multiply the convolution matrix with a data matrix of size N_t by M_{trc} , the convolution matrix should have a size of $(N_t + N_w - 1)$ by N_t . The size of the convolved data

¹ I use "wavelet matrix" term for \mathbf{W} to differentiate it from the convolution matrix, \mathbf{w} .

will be $(N_t + Nw - 1)$ by M_{trc} . To construct a convolution matrix, a zero matrix of size of $(N_t + Nw - 1)$ by N_t is chosen, then wavelet elements replace the first zeros in the first column. For the i^{th} column, the wavelet replaces the zero elements starting from the i^{th} row. This replacement continues for all columns. Multiplication of these matrices is equal to the convolution of each data trace with the wavelet. I chose the second approach, multiplication with a convolution matrix in this dissertation.

Consider \mathbf{d} as a 6 by 8 matrix of the form

$$\mathbf{d} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad 3.8$$

If the assumed wavelet vector is given by,

$$\mathbf{w}_v = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}, \quad 3.9$$

then the convolution matrix has the following form:

$$\mathbf{w} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 1 & 0 & 0 \\ 0 & 0 & -1 & 2 & 1 & 0 \\ 0 & 0 & 0 & -1 & 2 & 1 \\ 0 & 0 & 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}. \quad 3.10$$

Multiplication of \mathbf{w} and \mathbf{d} returns the convolved data:

$$\mathbf{wd} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 & -1 & 2 & 1 & 0 \\ 1 & 0 & 2 & -1 & 0 & -1 & 2 & 1 \\ 2 & 2 & -1 & 0 & 0 & 0 & -1 & 2 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & -1 \\ 2 & -2 & 2 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad 3.11$$

The convolution algorithm shown in Table 3-1 gives the same result as the multiplication of the convolution matrix with the data. However, convolution matrix multiplication is valid only for a 2D seismic line. For a 3D seismic survey, this multiplication needs to be performed inside a

loop over the third dimension. Instead, one can reshape an nD seismic data set as a column vector and create an appropriate “wavelet matrix” with which to multiply the data. If the data is vectorized to a single column by placing trace $j + 1$ under trace j , for all traces, then multiplication in equation 3.11 will be replaced by $\mathbf{W}\mathbf{d}_v$, where \mathbf{W} is the wavelet matrix and \mathbf{d}_v is the vectorized data.

$\mathbf{W}\mathbf{d}_v$ is a vector with $(N_t + N_w - 1) \times M_{trc}$ samples and \mathbf{d}_v is a $M_{trc} \times N_t$ vector. Multiplication in $\mathbf{W}\mathbf{d}_v$ is valid only when \mathbf{W} is a $(N_t + N_w - 1) \times M_{trc}$ by $M_{trc} \times N_t$ matrix. \mathbf{W} is a sparse matrix in which its semi-diagonal entries are \mathbf{w} matrices:

$$\mathbf{W} = \begin{bmatrix} \mathbf{w} & & & & \\ & \mathbf{w} & & & \\ & & \mathbf{w} & & \\ & & & \mathbf{w} & \\ & & & & \mathbf{w} \end{bmatrix}. \quad 3.12$$

Multiplication of \mathbf{W} with \mathbf{d}_v is possible and gives the convolved data:

$$\mathbf{w}_v * \mathbf{d} = \mathbf{W}\mathbf{d}_v = \begin{bmatrix} \mathbf{w} & & & & \\ & \mathbf{w} & & & \\ & & \mathbf{w} & & \\ & & & \mathbf{w} & \\ & & & & \mathbf{w} \end{bmatrix} \mathbf{d}_v. \quad 3.13$$

An algorithm to create \mathbf{W} is shown in Table 3-2. Comparing algorithms in Table 3-1 and Table 3-2 gives a general idea for the construction of a matrix whose multiplication with a vector is equivalent to performing a computer algorithm. A zero matrix with the appropriate size is created, and then the zero elements of the matrix are replaced by the multiplier of the input, in the most internal loop of the algorithm. Later in subsection 3.2.2, this idea is used to construct the Kirchhoff modeling and migration matrices.

Table 3-1 Convolution algorithm.

```

Subroutine:  $output = input * w$ 
For  $itrace = 1$  to  $M_{trace}$            % loop over traces
  For  $it = 1$  to  $N_t$                  % loop over data samples
    For  $iw = 1$  to  $N_w$              % loop over wavelet samples
       $output(it + iw - 1, itrace)$ 
        =  $output(it + iw - 1, itrace) + w_v(iw) \times input(it, itrace)$ 
    End
  End
End
End

```

Table 3-2 Wavelet matrix algorithm.

```

 $N_{tw1} = N_t + N_w - 1$ 
For  $itrace = 1$  to  $M_{trace}$            % loop over traces
  For  $it = 1$  to  $N_t$                  % loop over data samples
    For  $iw = 1$  to  $N_w$              % loop over wavelet samples
       $W((itrace - 1) \times N_{tw1} + it + iw - 1, (itrace - 1) \times N_t + it)$ 
        =  $W((itrace - 1) \times N_{tw1} + it + iw - 1, (itrace - 1) \times N_t + it) + w_v(iw)$ 
    End
  End
End
End

```

3.2.2 Construction of the modeling matrix, G , in explicit form

Yousefzadeh and Bancroft (2011, 2012a) explained how to construct a Kirchhoff modeling matrix using a Kirchhoff migration program. Including the wavelet, Kirchhoff modeling is expressed by:

$$d = Wgm. \quad 3.14$$

In order to create synthetic seismic data with the Kirchhoff modeling operator (equation 3.1 or 3.14), a form of \mathbf{g} in either an implicit or an explicit form is required. In implicit form, \mathbf{g} is a computer code, in the explicit form \mathbf{g} is a matrix (without considering the presence of the wavelet, or assuming a perfect impulse source). The matrix is multiplied directly to the reflectivity model, \mathbf{m} , to produce the data, \mathbf{d} . For example, the modeling matrix may be saved in memory and multiplied with the model several times when solving a least squares Kirchhoff migration by an iterative method.

In order to create the \mathbf{g} matrix, as I did for the wavelet matrix construction, a proper sized zero matrix is created and its zero elements are then replaced by the multiplier of the model in the most internal loop of the Kirchhoff modeling algorithm as shown in Table 3-3 and Table 3-4.

Table 3-3 A simplified Kirchhoff prestack modeling subroutine showing how to create data for the given reflectivity.

```

Subroutine:  $\mathbf{D} = \text{seismic modeling of } \mathbf{M}$ 
For  $itrace = 1$  to  $M_{trace}$     % loop over traces
  For  $iz = 1$  to  $N_z$         % loop over depth samples
    For  $ix = 1$  to  $N_x$       % loop over CMPs
       $time = \text{from DSR equation}$     % calculation of modeling time using DSR or ray
       $w_{mig} = \text{Migration Amplitude or Weight function}$ 
       $it = \text{floor}(time/dt)$         % to calculate the corresponding time sample
       $\mathbf{D}(it, itrace) = \mathbf{D}(it, itrace) + \mathbf{M}(iz, ix) * w_{mig}$ 
    End
  End
End

```

Table 3-4 Calculation of a prestack modeling matrix.

```

Subroutine:  $\mathbf{g}$  = seismic modeling matrix
For  $itrace = 1$  to  $M_{trace}$  % loop over traces
  For  $iz = 1$  to  $N_z$  % loop over depth samples
    For  $ix = 1$  to  $N_x$  % loop over CMPs
       $time =$  from DSR equation % calculation of modeling time using DSR or ray
       $w_{mig} =$  Migration Amplitude or Weight function
       $it = floor(time/dt)$  % to calculate the corresponding time sample
       $\mathbf{g}((itrace - 1) \times nt + it, (ix - 1) \times N_z + iz)$ 
        =  $\mathbf{g}((itrace - 1) \times nt + it, (ix - 1) \times N_z + iz) + w_{mig}$ 
    End
  End
End

```

To find the proper size and structure for \mathbf{g} , I start with a very simple reflectivity model. The construction of the \mathbf{g} matrix assumes a 2D Earth model with scatterpoints throughout the subsurface. Our example will consider only one scatterpoint with the amplitude of 1 located at the surface. Let's consider a seismic source right on the top of this scatterpoint and eight receivers on both sides of the source, as shown in Figure 3.1.



Figure 3.1 A simple model including one source (red triangle) on the top of a scatterpoint (blue circle) and eight receivers (green flags) on the surface.

This model can be presented by a matrix \mathbf{m} :

$$\mathbf{m} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad 3.15$$

The scatterpoint with the amplitude of 1 is located at row #1 and column #5. For simplicity I show this model as a 1 by 1 matrix as $\tilde{\mathbf{m}}_{1,5} = [1]$. Assuming a constant velocity background and ignoring any amplitude changes during wave propagation, the diffraction matrix for a scatterpoint at the #1 and column #5 of the model space, $\tilde{\mathbf{g}}_{1,5}$, will have a diffraction hyperbola in eight traces and six samples per trace with an assumed sampling rate:

$$\tilde{\mathbf{g}}_{1,5} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad 3.16$$

The data created with this geometry will produce a similar matrix, as illustrated in equation 3.17:

$$\mathbf{d}_{1,5} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad 3.17$$

Matrix $\mathbf{d}_{1,5}$ is the data resulting from only one scatterpoint at row #1 and column # 5. As was expected, the data resulting from this geometry have eight traces and six samples per trace. This is equal to spreading the amplitude of the scatterpoint along the flanks of a hyperbola. Therefore,

$$\tilde{\mathbf{g}}_{1,5} = \mathbf{d}_{1,5}. \quad 3.18$$

The \mathbf{g} matrix that created \mathbf{d} is a multidimensional matrix which requires one diffraction for each scatterpoint. Multiplication of the modeling operator with the model should be mathematically possible and creates the data,

$$\tilde{\mathbf{g}}_{1,5} \tilde{\mathbf{m}}_{1,5} = \mathbf{d}_{1,5}. \quad 3.19$$

However, $\tilde{\mathbf{g}}_{1,5}$ and $\mathbf{d}_{1,5}$ are matrices of size N_t by M_{trc} , where N_t is the number of samples per trace and M_{trc} is the number of traces, and $\tilde{\mathbf{m}}_{1,5}$ is a 1 by 1 matrix. The multiplication in equation 3.19 becomes possible by vectorizing the $\tilde{\mathbf{g}}_{1,5}$ and $\mathbf{d}_{1,5}$ matrices. In order to have $\tilde{\mathbf{g}}_{1,5}$ (or $\mathbf{d}_{1,5}$) as a vector, $\tilde{\mathbf{g}}_{v1,5}$ (or $\mathbf{d}_{v1,5}$), we put column $i + 1$ of $\tilde{\mathbf{g}}_{1,5}$ (or $\mathbf{d}_{1,5}$) matrix below column i of that matrix for all columns,

$$\tilde{\mathbf{g}}_{v1,5} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad 3.20$$

and

$$\mathbf{d}_{v1,5} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad 3.21$$

Now $\tilde{\mathbf{g}}_{v1,5}$ and $\mathbf{d}_{v1,5}$ are column vectors with $N_t \times M_{trc}$ elements and we can rewrite equation 3.19 as the following valid matrix-vector multiplication:

$$\tilde{\mathbf{g}}_{v1,5} \tilde{\mathbf{m}}_{1,5} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix} [1] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{d}_{v1,5}. \quad 3.22$$

Now suppose that our model, which has N_x Elements in each row, has other scatterpoints in the first row, right below the first receiver as shown in Figure 3.2.



Figure 3.2 Model includes one source (red triangle) and eight receivers (green flags), and two scatterpoints (blue circles) on the surface.

With the mentioned method of vectorization, the reflectivity for the second model with both scatterpoints looks like,

$$\tilde{\mathbf{m}}_{v1,:} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad 3.23$$

where $\tilde{\mathbf{m}}_{v1,:}$ stands for the first row and all columns of the reflectivity model, in column vectorized form. Ignoring the first scatterpoint, which is below the source, the seismic data modeled from only the second scatterpoint look like this:

$$\mathbf{d}_{1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad 3.24$$

Consequently, the modeling operator for the second scatterpoint in row # 1 and column # 1, $\tilde{\mathbf{g}}_{1,1}$, will be:

$$\tilde{\mathbf{g}}_{1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad 3.25$$

or in vector form,

$$\tilde{\mathbf{g}}_{v1,1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad 3.26$$

The data resulting from the presence of both scatterpoints in the same experiment is the sum of $\mathbf{d}_{1,5}$ (equation 3.17) and $\mathbf{d}_{1,1}$ (equation 3.24):

$$\mathbf{d}_{1,:} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad 3.27$$

where $\mathbf{d}_{1,:}$ represents data resulting from a seismic experiment on all scatterpoints in the first row of the model. To accomplish operator-matrix multiplication, as I did for the case of one scatterpoint, we need to do the following multiplication:

$$\tilde{\mathbf{g}}_{1,:} \tilde{\mathbf{m}}_{v1,:} = \mathbf{d}_{v1,:}, \quad 3.28$$

where $\tilde{\mathbf{g}}_{1,:}$ is the corresponding modeling operator for all scatterpoints in the first row of the model, $\tilde{\mathbf{m}}_{v1,:}$, and $\mathbf{d}_{v1,:}$ is the vectorized form of $\mathbf{d}_{1,:}$. This is possible by defining $\tilde{\mathbf{g}}_{1,:}$ an $N_t \times M_{trc}$ by N_x matrix by:

$$\tilde{\mathbf{g}}_{1,:} = [\tilde{\mathbf{g}}_{v1,1} \tilde{\mathbf{g}}_{v1,2} \quad \dots \quad \tilde{\mathbf{g}}_{v1,5} \quad \dots \quad \tilde{\mathbf{g}}_{v1,N_x}]. \quad 3.29$$

Then multiplication of $\tilde{\mathbf{g}}_{1,:}$ by $\tilde{\mathbf{m}}_{v1,:}$ gives the data resulting from modeling of all points in the first row of the reflection matrix in equation 3.28.

$$\tilde{\mathbf{g}}_{1,:} \tilde{\mathbf{m}}_{v1,:} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{d}_{v1,:}. \quad 3.30$$

This is the result of the seismic experiment for the first row of model. Results of modeling other rows in the image domain must be added to the current data. The mentioned procedure can be extended to the other rows of the model by defining the modeling matrix for all elements in each row, and placing them next to each other,

$$\mathbf{g} = [\tilde{\mathbf{g}}_{1,:} \tilde{\mathbf{g}}_{2,:} \quad \dots \quad \tilde{\mathbf{g}}_{N_z,:}] = [\tilde{\mathbf{g}}_{v1,1} \tilde{\mathbf{g}}_{v1,2} \quad \dots \quad \tilde{\mathbf{g}}_{v1,N_x} \tilde{\mathbf{g}}_{v2,1} \tilde{\mathbf{g}}_{v2,2} \quad \dots \quad \tilde{\mathbf{g}}_{vN_z,N_x}], \quad 3.31$$

where N_z is the number of rows in the model matrix.

The model needs to be vectorized by the same method,

$$\mathbf{m}_v = \begin{bmatrix} \tilde{\mathbf{m}}_{v1,:} \\ \tilde{\mathbf{m}}_{v2,:} \\ \vdots \\ \tilde{\mathbf{m}}_{vN_z,:} \end{bmatrix}, \quad 3.32$$

\mathbf{d}_v is the stack of the data resulting from performing a seismic experiment on each row of the model:

$$\mathbf{d} = \mathbf{d}_{v1,:} + \mathbf{d}_{v2,:} + \dots + \mathbf{d}_{vN_z,:}. \quad 3.33$$

Each column of the \mathbf{g} represents a point in the migrated image. Therefore, matrix \mathbf{g} includes $N_t \times M_{trc}$ rows and $N_x \times N_z$ columns:

$$\mathbf{g} = \begin{matrix} N_t \\ \times \\ M_{trc} \end{matrix} \begin{bmatrix} \dots & & & & & & & & & \\ & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ & & & & & \dots & & & & \\ & & & & & & & & & \end{bmatrix}. \quad 3.34$$

$\underbrace{\hspace{15em}}_{N_x \times N_z}$

The result of multiplication of \mathbf{g} , an $N_t \times M_{trc}$ by $N_x \times N_z$ matrix, with \mathbf{m}_v , an $N_x \times N_z$ column vector, is \mathbf{d}_v , an $N_t \times M_{trc}$ vector:

$$\mathbf{g}\mathbf{m}_v = \mathbf{d}_v \Rightarrow \begin{matrix} N_t \\ \times \\ M_{trc} \end{matrix} \underbrace{\left\{ \begin{bmatrix} \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \right\}}_{N_x \times N_z} \begin{matrix} N_x \\ \times \\ N_z \end{matrix} \begin{bmatrix} \dots \\ \dots \\ \dots \\ \dots \end{bmatrix} = \begin{matrix} N_t \\ \times \\ M_{trc} \end{matrix} \left\{ \begin{bmatrix} \dots \\ \dots \\ \dots \end{bmatrix} \right\}. \quad 3.35$$

The resulting \mathbf{d}_v vector must be reshaped to show the data in conventional form. In MATLAB software, vectorization is done by the “ $d = d(:);$ ” command and undone using the “reshape” function. In order to call a sample in the vectors \mathbf{d}_v and \mathbf{m}_v , it is useful to know that the sample number $(j - 1) \times N_i + i$ in a vector is equal to sample (i, j) in the corresponding matrix form, where N_i is the number of rows.

To calculate matrix \mathbf{g} we must include modeling amplitude weights (instead of having just ones) and other parameters. Amplitude weights can be implemented in the \mathbf{g} matrix by replacing the ones with proper amplitude weights. As seen in Table 3-3, weights are applied to the data in the most inner loop. Consequently, matrix \mathbf{g} can be calculated only by going to the most inner loop and replacing the entries of \mathbf{g} matrix by the calculated migration weights as seen in Table 3-4.

3.2.3 Construction of the migration matrix in explicit form

The transpose of the modeling matrix is the migration matrix, \mathbf{g}^T , and is explained as follows (Yousefzadeh and Bancroft, 2012b). Creating \mathbf{g}^T in matrix form follows the same rules that are used to create the \mathbf{g} matrix, but in reverse order. For simplicity, assume one shotgather with one hyperbola in eight traces, each with six samples, as:

$$\mathbf{d} = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad 3.36$$

In order to perform Kirchhoff migration, all samples on this hyperbola (elements with a value of 2) must be added, and the result ($8 \times 2 = 16$) placed on the apex of the hyperbola, in the position of row # 1 and column # 5 of the image domain. Let's $\tilde{\mathbf{g}}_{1,5}^T$ be a transposed version of \mathbf{d} , but with ones instead of the nonzero elements as seen in equation 3.37:

$$\tilde{\mathbf{g}}_{1,5}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad 3.37$$

Index 1,5 in $\tilde{\mathbf{g}}_{1,5}^T$ means that I am looking for a migration operator for the scatterpoint at row # 1 and column # 5 of the model. Then, multiplication of $\tilde{\mathbf{g}}_{1,5}^T$ by \mathbf{d} is possible and returns the following matrix:

$$\tilde{\mathbf{g}}_{1,5}^T \mathbf{d} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}. \quad 3.38$$

Summation of the diagonal elements of the $\tilde{\mathbf{g}}_{1,5}^T \mathbf{d}$ matrix is equal to the number that must be entered into row # 1 and column # 5 in the migration image:

$$\hat{\mathbf{m}}_{1,5} = \sum [diag(\tilde{\mathbf{g}}_{1,5}^T \mathbf{d})] = [16]. \quad 3.39$$

This procedure is valid only for one hyperbola in data and one point in the image domain, $\hat{\mathbf{m}}_{1,5}$. Practically, we need a matrix to multiply to the whole data to get the desired migration image:

$$\hat{\mathbf{m}}_{1,5} = \mathbf{g}_{1,5}^T \mathbf{d}, \quad 3.40$$

where $\hat{\mathbf{m}}_{1,5}$ is the migration image at row #1 and column # 5. The data form a matrix with M_{trc} traces, each trace in a column, and N_t samples per column. To do matrix multiplication in equation 3.40, \mathbf{d} must be vectorized to \mathbf{d}_v . To have the data as a vector, we put column $i + 1$ of \mathbf{d} below column i for all columns as we did in the previous section:

$$\mathbf{d}_v = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ \vdots \\ 2 \\ 0 \\ 0 \end{bmatrix}. \quad 3.41$$

Consequently, $\tilde{\mathbf{g}}_{1,5}^T$, must also be a row vector. If we put all rows of $\tilde{\mathbf{g}}_{1,5}^T$ next to each other, the desired row vector is obtained:

$$\tilde{\mathbf{g}}_{v1,5}^T = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ \dots \ 1 \ 0 \ 0]. \quad 3.42$$

Now, multiplication of $\tilde{\mathbf{g}}_{v1,5}^T$ by \mathbf{d}_v , is equal to the summation along the diagonal elements of $\tilde{\mathbf{g}}_{1,5}^T \mathbf{d}$ matrix in equation 3.38. Therefore, a calculation of the diagonal elements of the $\tilde{\mathbf{g}}_{1,5}^T \mathbf{d}$ matrix is replaced by the multiplication of the row vector $\tilde{\mathbf{g}}_{v1,5}^T$ with the column vector \mathbf{d}_v . This is equivalent to the summation of amplitudes along the flanks of the diffraction hyperbola:

putting all rows corresponding to the first column of the image matrix below each other and then repeating this procedure for the next column of the image matrix, $\hat{\mathbf{m}}_v$ is vectorized in the same manner that \mathbf{d}_v is vectorized.

In the calculation of \mathbf{g}^T , migration weights and other migration parameters must be included. Weight can be added to the \mathbf{g}^T matrix by replacing ones with the proper amplitude scales. As the migration weights are applied to the data in the most inner migration loop, \mathbf{g}^T can be calculated by going to the most inner loop and replacing the \mathbf{g}^T elements by the calculated migration weight. If instead of the DSR equation, a ray tracing program is used to calculate the migration time for each sample, this technique is extendable to prestack Kirchhoff depth modeling and migration. However, to add interpolation and antialiasing to the migration, the calculation of \mathbf{g} and \mathbf{g}^T are more complicated. For example, in the case of adding linear interpolation, the calculation of \mathbf{g} needs to be done with dividing the weight functions to two different adjacent samples in \mathbf{g} as shown in Table 3-5.

Table 3-5 Calculation of prestack modeling matrix with linear interpolation.

Subroutine: $\mathbf{g} =$ seismic modeling matrix

For $itrace = 1$ to M_{trace} % loop over traces

 For $iz = 1$ to N_z % loop over depth samples

 For $ix = 1$ to N_x % loop over CMPs

$time =$ from DSR equation % calculation of modeling time using DSR or ray

$w_{mig} =$ Migration Amplitude or Weight function

$it = floor(time/dt)$ % to calculate the corresponding time sample

$$d_t = 1 - \frac{time}{dt} + it$$

$$\mathbf{g}((itrace - 1) \times nt + it, (ix - 1) \times N_z + iz) = \mathbf{g}((itrace - 1) \times nt + it, (ix - 1) \times N_z + iz + dt * w_{mig})$$

$$\mathbf{g}((itrace - 1) \times nt + it + 1, (ix - 1) \times N_z + iz) = \mathbf{g}((itrace - 1) \times nt + it + 1, (ix - 1) \times N_z + iz - dt * w_{mig})$$

 End

 End

End

3.2.4 Example of modeling and migration matrices

As an example, suppose the assumed geometry includes only two sources, 10 receivers per source, 20 CMPs, 150 pseudo depth samples and our data has 250 time samples per trace. This geometry is shown in Figure 3.3, it appears to the velocity model shown in Figure 3.4. Modeling requires a 5000 by 3000 matrix, a relatively large matrix for such a small geometry and model. Non-zero elements of the \mathbf{g} matrix are shown in Figure 3.5a and Figure 3.5b. Figure 3.5c and Figure 3.5d show the corresponding \mathbf{G} matrix, modeling, in the presence of a Ricker wavelet.

Figure 3.6a and Figure 3.6b show the migration matrix, \mathbf{g}^T , for the same geometry and velocity model which is the transpose of Figure 3.5a and Figure 3.5b, respectively.

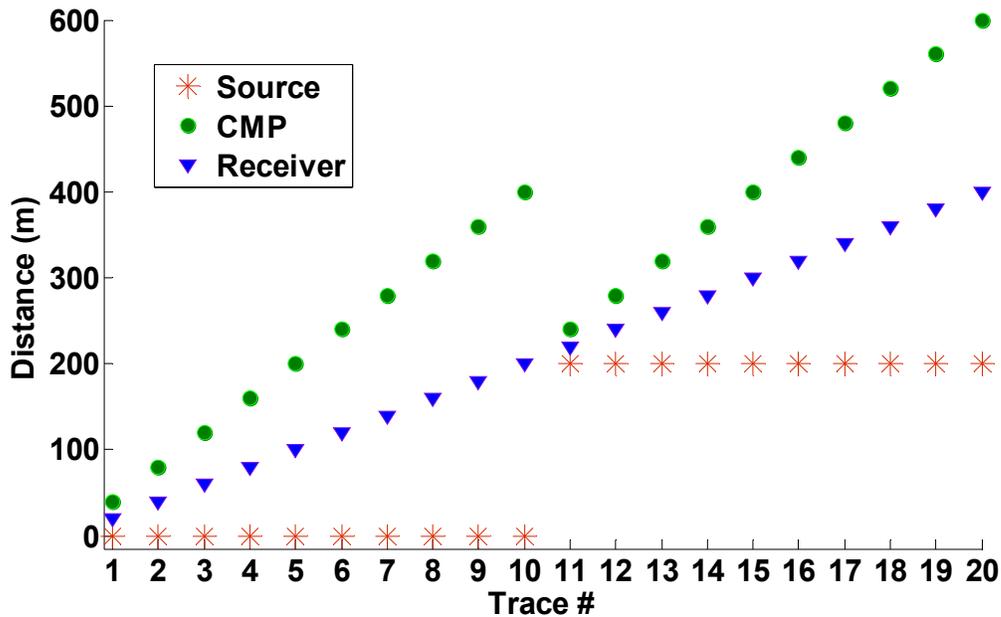


Figure 3.3 Acquisition geometry defined for construction of the synthetic modeling and migration matrices. Geometry includes two sources and 10 receivers per source. Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.

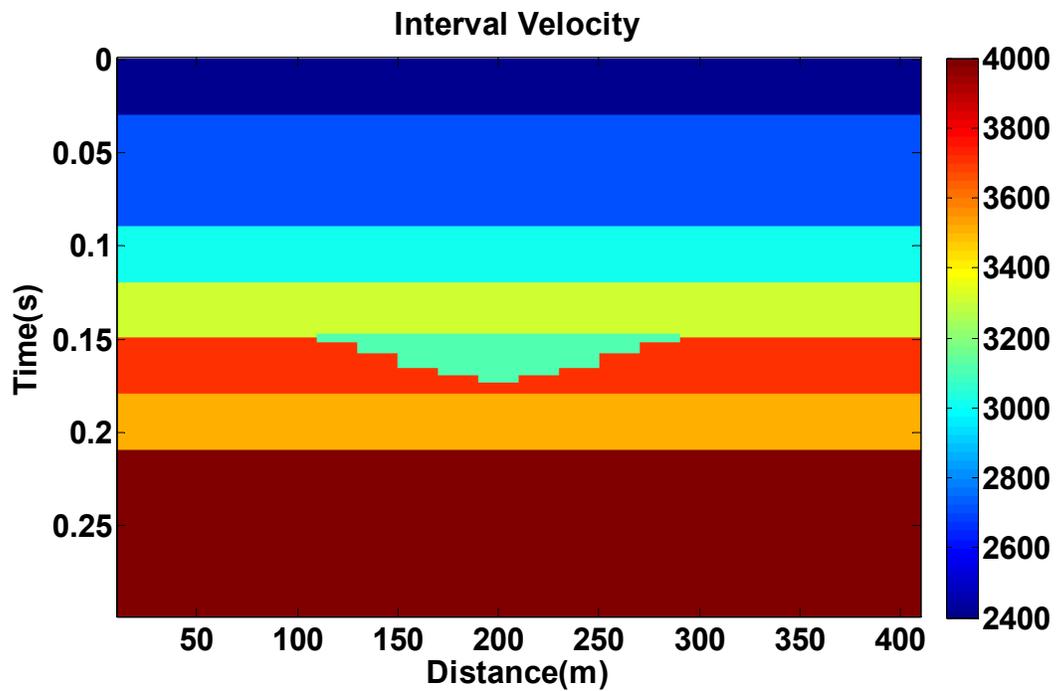
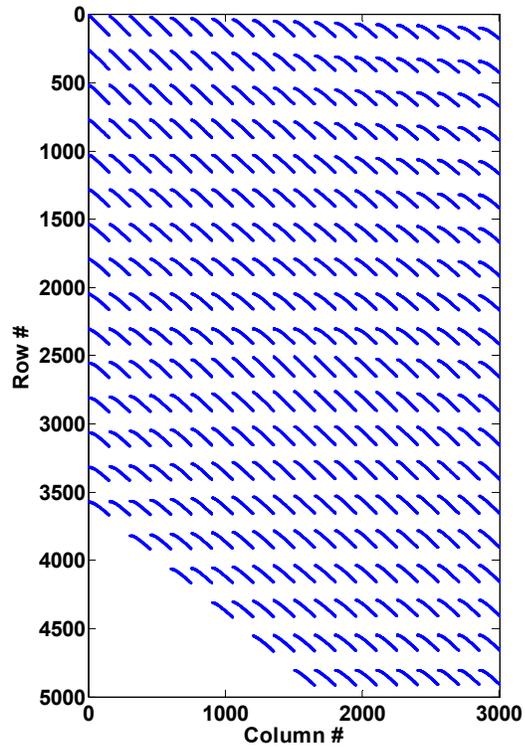
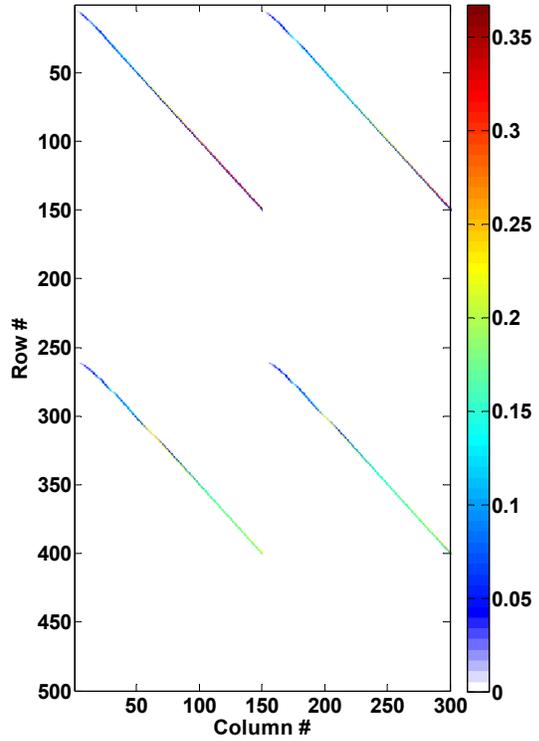


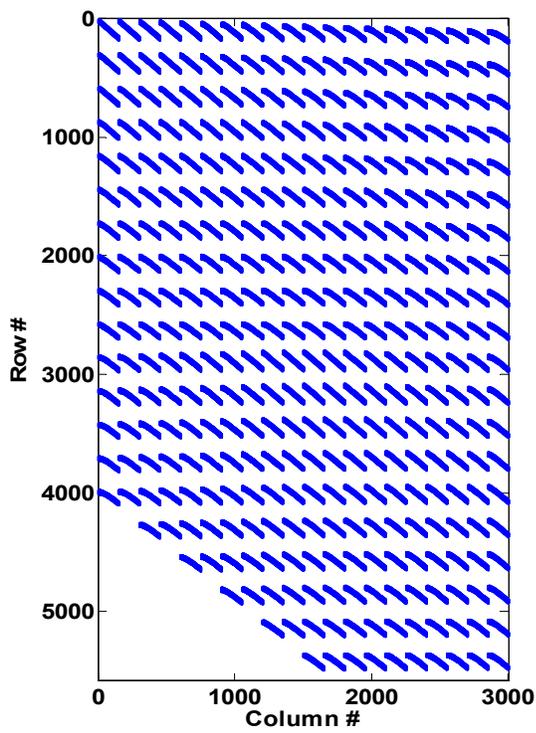
Figure 3.4 Interval velocity model used to construct the synthetic modeling and migration matrices.



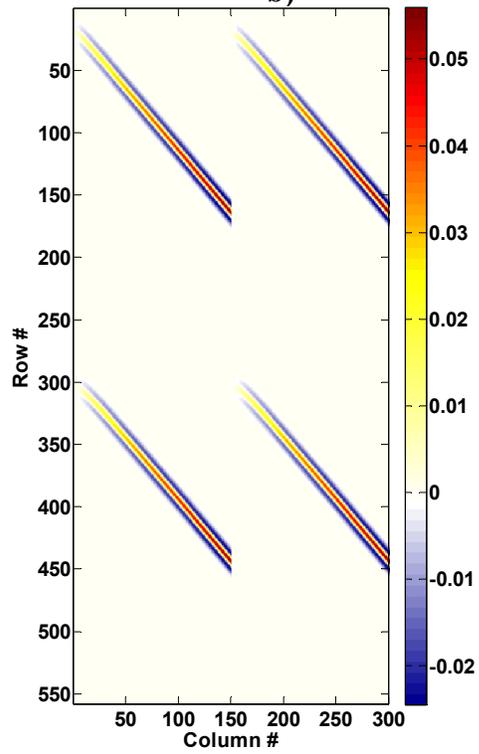
a)



b)



c)



d)

Figure 3.5 Non-zero elements of a) matrix g , and c) matrix G , close-up of the first 500 rows and 300 columns of b) g , and d) G matrices.

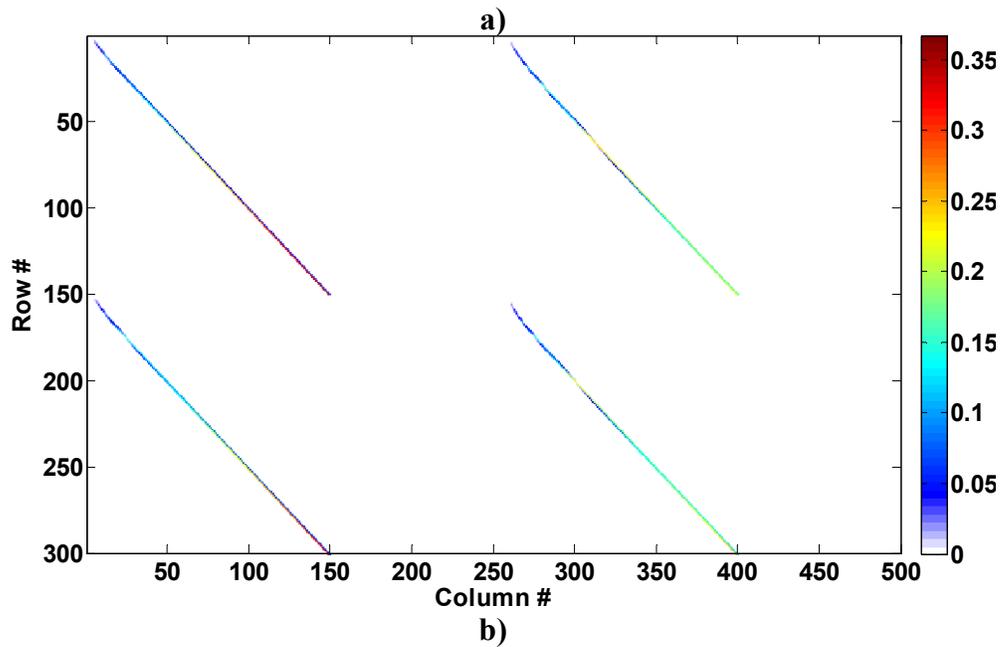
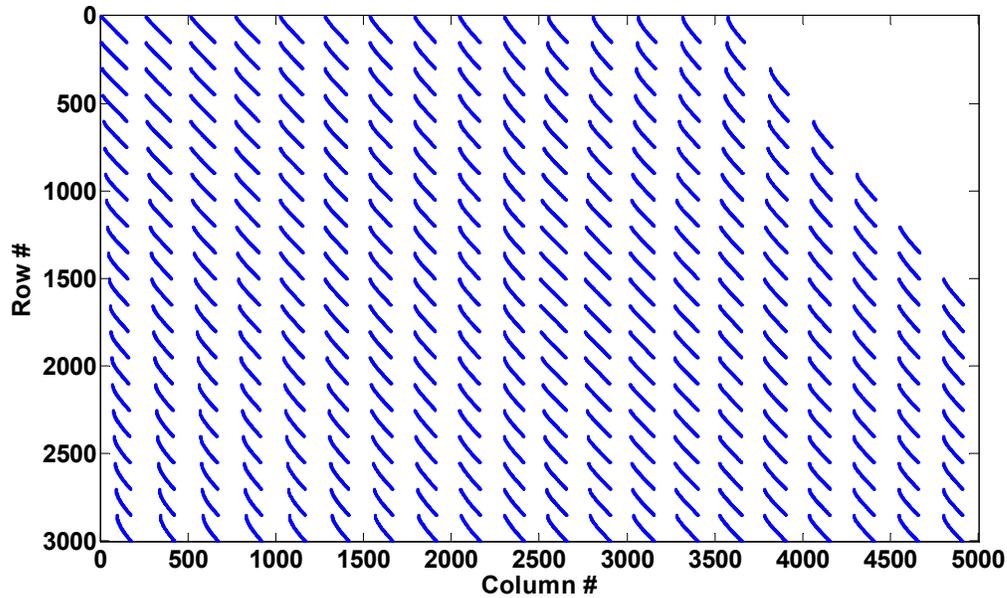


Figure 3.6 a) Non-zero elements of matrix g^T , b) Close-up of the first 500 rows and 300 columns.

Figure 3.7 and Figure 3.8 show the inversion matrices $g^T g$ and $G^T G$. Both matrices are large and dense with about 9 million elements for a small geometry and data set. These are the matrices that need to be inverted for finding the least squares prestack Kirchhoff migration.

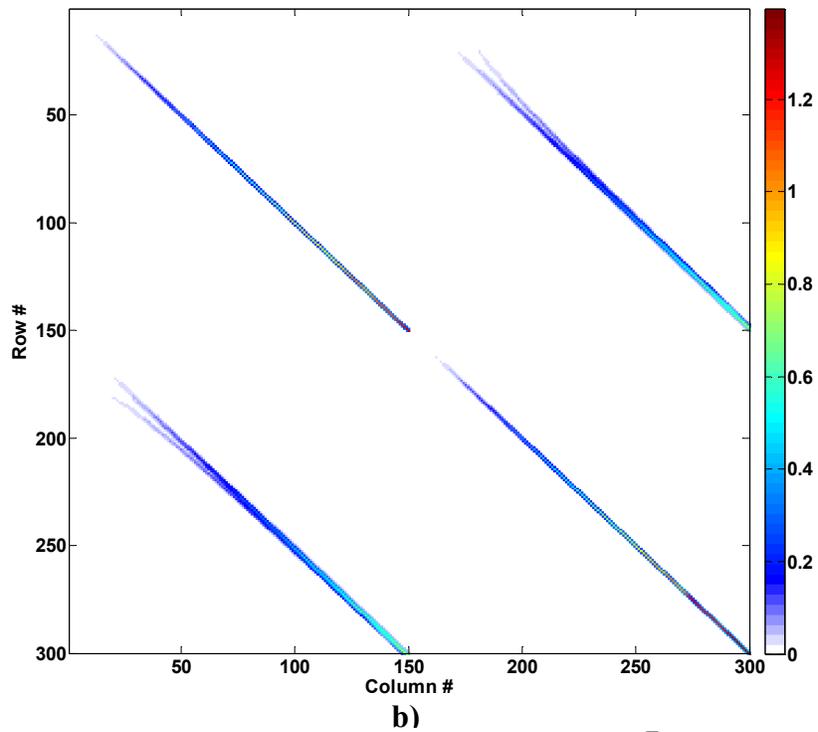
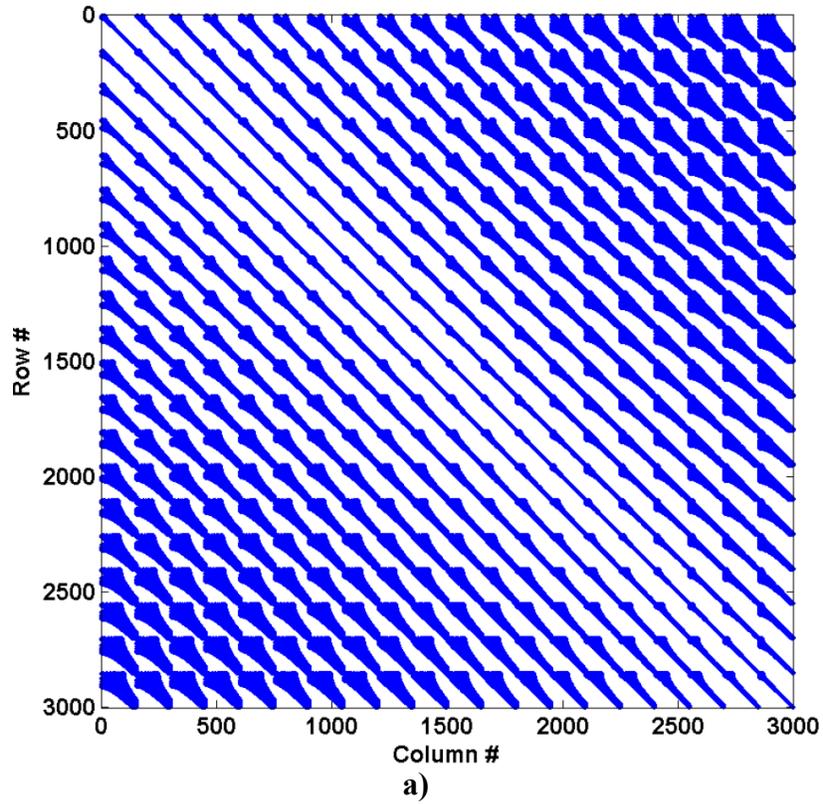


Figure 3.7 a) Non-zero elements of the Hessian matrix $g^T g$, b) Close-up of the first 300 rows and columns.

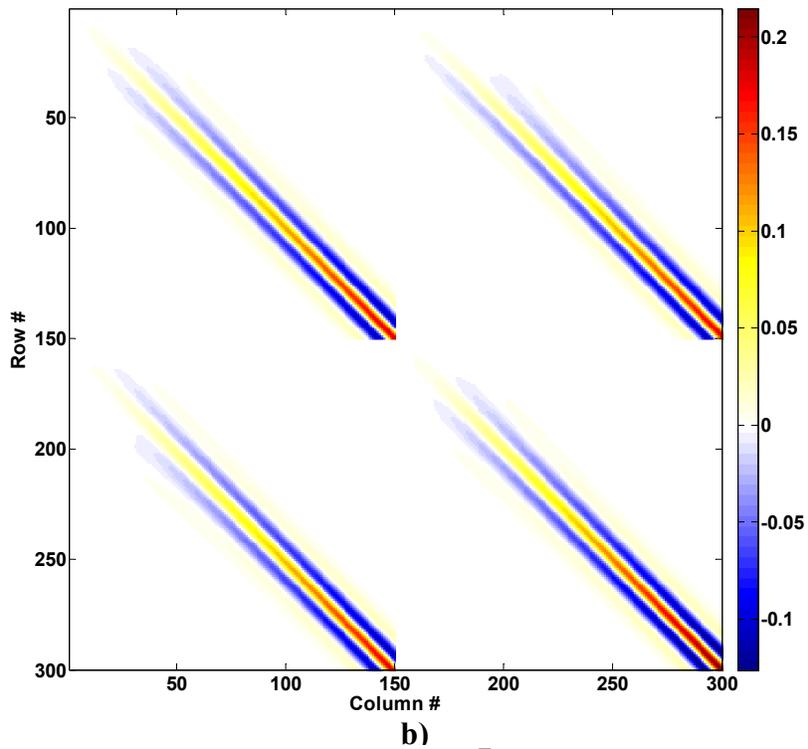
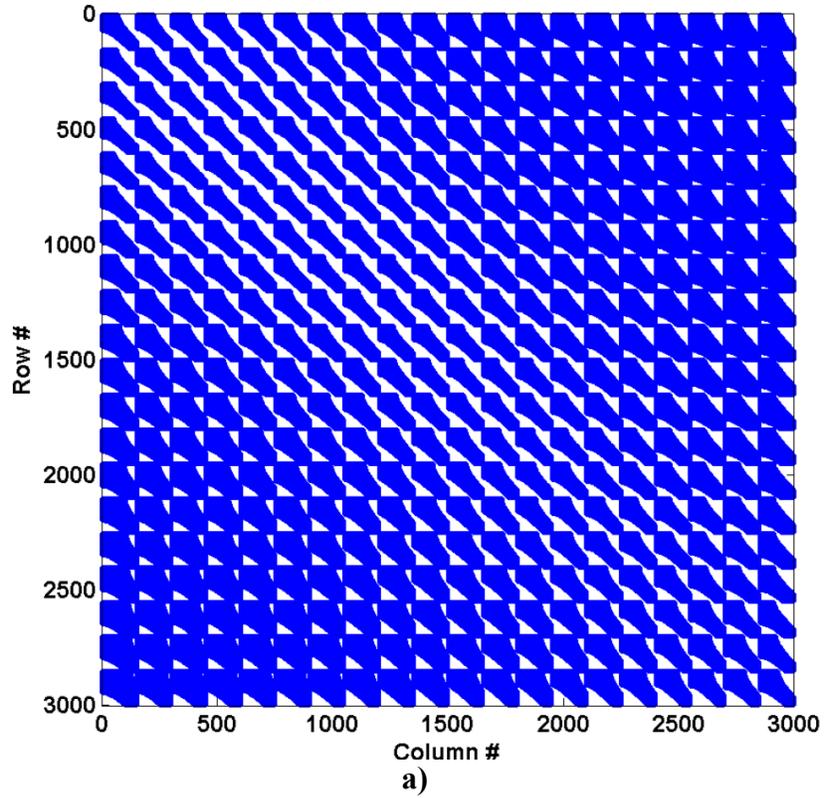


Figure 3.8 a) Non-zero elements of matrix $G^T G$, b) Close-up of the first 300 rows and columns. $G^T G$ is denser than the $g^T g$ matrix.

Figure 3.9 shows the $W^T W$ matrix. This is a semi-diagonal matrix. A 30 Hz Ricker wavelet with 30 samples is used here.

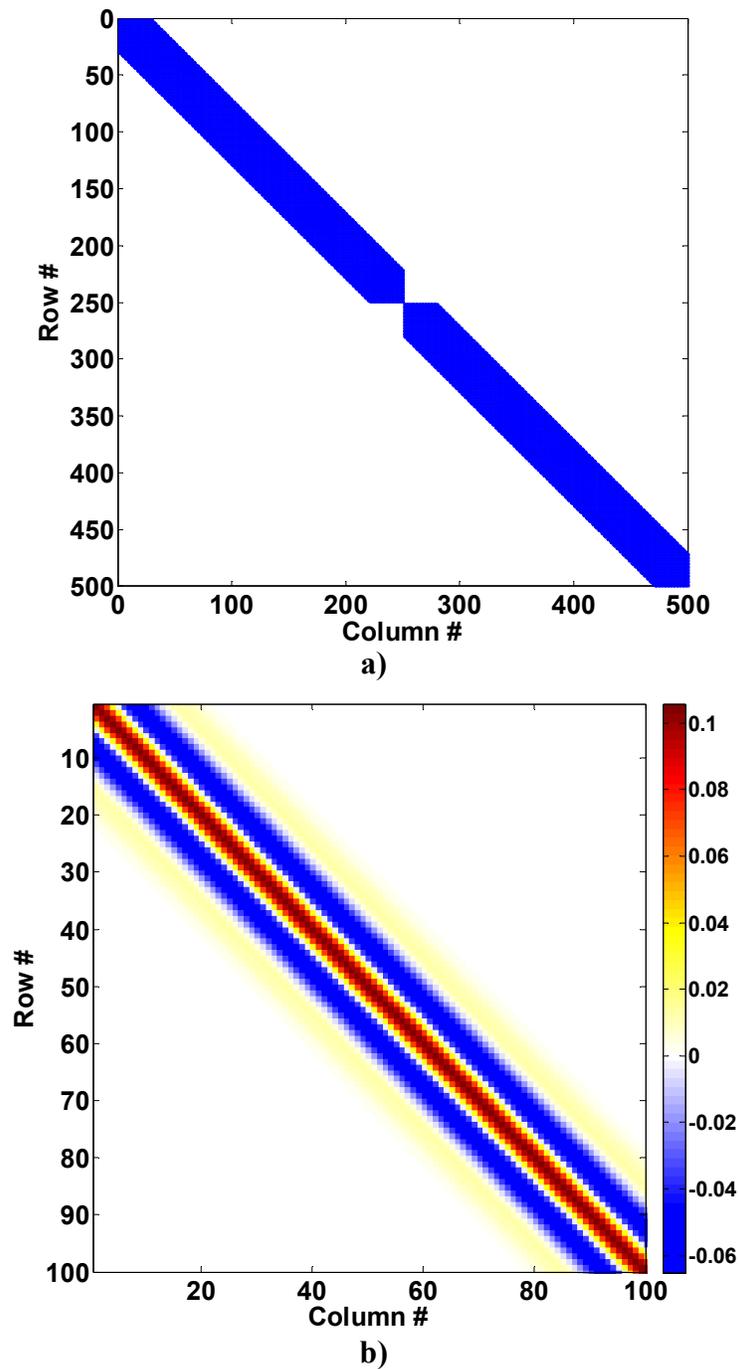


Figure 3.9 a) Non-zero elements of the first 500 rows and columns of matrix $W^T W$, b) Close-up of the first 100 rows and columns.

3.2.5 Feasibility of working with the matrix forms

When \mathbf{G} and \mathbf{G}^T are available in matrix form, they can be used for solving the least squares prestack Kirchhoff migration. It seems easier to use the matrix form when we need to perform Kirchhoff migration and modeling several times. It is also more feasible to add some regularization to such kinds of inversion. However, \mathbf{G} and \mathbf{G}^T are very large matrices. The size of \mathbf{G} (or \mathbf{G}^T) is equal to the number of model grids multiplied by the number of data samples. For a 10 km 2D seismic line with 75000 traces in 1000 CMPs, and each trace having 1000 samples, the size of \mathbf{G} will be $75000 \times 1000 \times 1000 \times 1000 = 7.5 \times 10^{13}$ samples. This matrix needs 600 terabytes of computer memory to be saved in double precision format which is far from the memory size of current computers. The difficulty is even more severe if 3D data is being used. In addition to the size, the matrix \mathbf{G} is not sparse enough to be saved and implemented using sparse techniques. Even considering it as a relatively sparse matrix, the Hessian, $\mathbf{G}^T \mathbf{G}$, is a dense matrix.

Matrix $\mathbf{G}^T \mathbf{G}$ is a symmetric matrix. Since entries of the \mathbf{G} matrix are the migration weights, all non-zero entries of $\mathbf{G}^T \mathbf{G}$ are positive. Consequently, the $\mathbf{x}^T \mathbf{G}^T \mathbf{G} \mathbf{x}$ matrix is positive for any \mathbf{x} vector with positive elements. Therefore, $\mathbf{G}^T \mathbf{G}$ is a positive definite matrix. This property is important because it makes the LSPSM solvable by the LSCG method (Section 3.5).

When all the off-diagonal elements of a square matrix are zero, it is called a diagonal matrix. A diagonal matrix is easy to invert since its inverse can be achieved by inverting the diagonal elements. A matrix is called strictly diagonally dominant if the magnitude of the diagonal element in each row is greater than the sum of the magnitudes of all other elements in that row:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad 3.46$$

where a_{ij} is the element at row i and column j . Diagonally dominant matrices may be effectively inverted by Jacobi or Gauss-Seidel methods. In a $\mathbf{G}^T \mathbf{G}$ matrix, diagonal elements have relatively large magnitudes. However, the high density of the matrix and the presence of many non-diagonal entries prevent this matrix from being diagonally dominant. Figure 3.10 shows the ratio of absolute values of the diagonal elements to the sum of the absolute values of nondiagonal entries for each row of the $\mathbf{G}^T \mathbf{G}$ matrix in the previous example. This and many other examples

show that the $\mathbf{G}^T \mathbf{G}$ matrix is a diagonally non-dominant matrix. It can be shown that adding some white noise or a reasonably large constant scalar to the diagonal elements of the $\mathbf{G}^T \mathbf{G}$ matrix does not change it into a diagonally dominant matrix. Size and structure of a $\mathbf{G}^T \mathbf{G}$ matrix cause solving the least squares Kirchhoff migration equation to be very expensive.

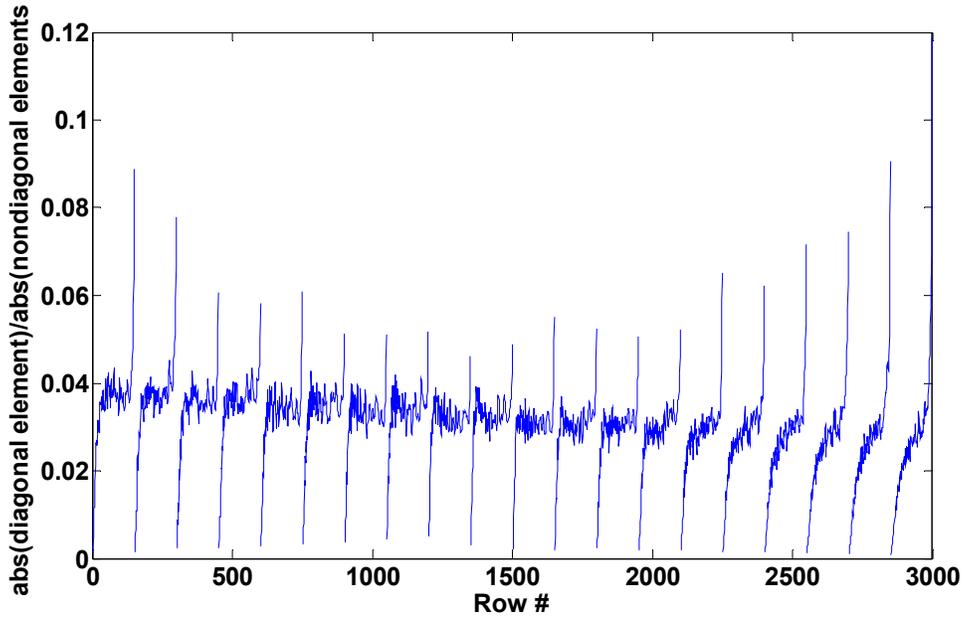


Figure 3.10 The ratio of absolute values of diagonal elements to the sum of absolute values of the non-diagonal elements for each row of $\mathbf{G}^T \mathbf{G}$. In a diagonally dominant matrix all values must be greater than one.

3.3 Solving the LSPSM equation

In equation 3.5, $\mathbf{G}^T \mathbf{G}$ is a Hessian matrix. For simplicity I may replace $\mathbf{G}^T \mathbf{G}$ with \mathbf{A} , $\mathbf{G}^T \mathbf{d}$ with \mathbf{b} , and \mathbf{m}_{LS} with \mathbf{m} for ease of notation. Using this notation, equation 3.1 becomes

$$\mathbf{A}\mathbf{m} = \mathbf{b}. \quad 3.47$$

Methods to solve equation 3.47 are placed in one of the two following main categories in linear algebra (Strang, 2006). Direct methods such as Gaussian elimination, which find the exact solution after a finite number of processes, and iterative methods, on the other hand, try to improve an initial estimation of \mathbf{m} to obtain the best approximation to the exact solution. In the stationary iterations methods, for example Jacobi, convergence steps are applied after each iteration, while in the gradient methods, for example the Conjugate Gradient (CG) and Krylove methods, changes are applied within the iteration to find the solution faster (Strang, 2006). In the

LSPSM equation, the Hessian matrix, $\mathbf{G}^T\mathbf{G}$, is a large matrix and cannot be solved using direct methods. The size of the \mathbf{G} matrix equals the number of grid points in the reflectivity model multiplied with the number of observations (data samples). The size of $\mathbf{G}^T\mathbf{G}$ is the square of the number of grid points in the model. Direct methods are not practical or efficient in such cases. Iterative methods are applicable to this problem, but they are relatively costly in comparison to the migration cost. The following section focuses on reducing the LSPSM cost or improving the image resolution by using standard multigrid, multilevel, and different versions of the CG methods.

The multigrid method is a fast and robust tool to solve an equation for finding both the low frequency and the high frequency content of the solution. In the multigrid method, an iterative solver such as Jacobi or Gauss-Seidel, which produces low frequency signal in residuals, is used to find a solution in a few iterations. By a restriction process, the main problem and the residuals in being transferred to a coarser grid, where the low frequency components act as a high frequency signal. The problem is solved in the coarse grid and then interpolated to a fine grid of the initial value for Jacobi iterations on the main problem. This procedure not only recovers the low frequency content of the solution more accurately, but also reduces the number of iterations on the main grid size and consequently the total computational cost.

Multigrid method is “often dramatically successful” (Strang, 2006). The multigrid method is evaluated for solving the Kirchhoff LSPSM equation. This research showed that the Hessian matrix of the Kirchhoff LSPSM was not diagonally dominant, and the standard method of the multigrid method, which uses Jacobi and Gauss-Seidel iterations, is not effective.

The multigrid method does not use CG methods as its iterative solver. I use a “multilevel” term when implementing iterative solvers rather than the Jacobi or Gauss-Seidel methods, for each grid size. The performance of LSCG (as an iterative solver for multilevel) is examined for some synthetic examples and the results are shown. Using the LSCG as an iterative solver for the multilevel slightly reduces the number of iterations for the same rate of convergence in the LSCG itself. Multilevel LSCG is effective when seismic data are split to chunks of different frequency bands where each chunk includes the frequency components of the lower frequency chunk. The advantage of using the Bi-Conjugate Gradient Stabilized (BiCGSTAB) method instead of LSCG method is also investigated.

An explanation of the multigrid method begins with the derivation of the Jacobi and Gauss-Seidel methods in the rest of the section. Elements of multigrid algorithms are defined, and the feasibility of using multigrid methods on solving LSPSM equation is investigated. CG and some CG-like methods are reviewed, and the feasibility, advantages, and disadvantages of using multilevel LSCG method for solving LSPSM equation are discussed.

3.4 Multigrid methods for solving the LSPSM equation

The feasibility of using the standard multigrid method to solve LSPSM is explained in this section. It will be shown why the standard multigrid methods cannot be used to solve the LSPSM equation.

If a problem is solvable by a multigrid method, it will be solved faster and with better recovery of the low frequency content than using the iterative solver only on the main grid. The ability of using the multigrid method to solve many types of partial differential equations more rapidly than the other iterative methods is demonstrated in Briggs et al. (2000).

Using multigrid methods for solving seismic problems is neither new nor a common idea. Saleck et al. (1993) and Bunks et al. (1995) implemented a multigrid method to perform seismic waveform velocity inversion on the Marmousi data set. Their idea was to eliminate the local minima of the objective function by solving the problem on a coarser grid in order to guarantee convergence to the global minimum and to avoid possible local minima. They showed the effectiveness of the method in diminishing the local minima in the coarse grid. They claimed that the method is able to reduce the total computational cost of the inversion.

The multigrid method is used to enhance the resolution of the seismic data by deconvolution (Millar and Bancroft, 2004), who also showed better recovery of the reflectivity and low frequencies using multigrid instead of the Gauss-Seidel method. However, they observed that the success of the method depends on the good estimation of the wavelet and also the frequency content of the data.

Millar and Bancroft (2005, 2006) expressed the surface consistent statics as a matrix operation and used a multigrid method for solving the corresponding least squares equation. Using multigrid for this problem did not reduce the computational cost significantly. However, it improved the recovery of the longer wavelength static corrections in the data.

Plessix (2007) studied the effects of using multigrid cycles for the 3D frequency domain wave equation migration. He considered the result of using multigrid on the undamped wave equation at seismic frequencies as the preconditioner for the BiCGSTAB method.

Yousefzadeh and Bancroft (2010a, 2010b) investigated the feasibility of using a standard multigrid method for solving the Kirchhoff LSPSM equation. They showed the reason which prevents the standard multigrid method from being an effective method to solve this inversion equation.

Multigrid methods use one special property of the Jacobi and Gauss-Seidel methods, the smoothing property, to converge to the solution quicker, and with better recovery of the low frequency content of the solution. To understand the multigrid method, it is necessary to understand smoothing as used in the Jacobi and Gauss-Seidel methods.

3.4.1 *Methods of Jacobi and Gauss-Seidel*

In equation 3.47, $\mathbf{A}\mathbf{m} = \mathbf{b}$, \mathbf{m} is the unknown desired exact solution. Direct methods such as Gaussian elimination give the exact solution after a finite number of operations. However, they are not efficient for systems with a large number of unknowns. Iterative methods start from an initial guess, \mathbf{m}_0 , as an approximation to the solution, and then try to improve the approximation by minimizing the residuals, the difference between $\mathbf{A}\mathbf{m}_i$ and \mathbf{b} vectors, where \mathbf{m}_i is the improved solution at the i^{th} iteration. Jacobi is the simplest iterative method. The derivation and algorithm of the Jacobi method is well documented. I used the method and notation of Briggs et al. (2000), Saad (2003), and Strang (2006). The last two references are freely available on the World Wide Web.

Considering \mathbf{v} as an approximation to the exact solution, \mathbf{m} , the algebraic error \mathbf{e} is the difference between two vectors:

$$\mathbf{e} = \mathbf{m} - \mathbf{v}. \quad 3.48$$

We desire to minimize \mathbf{e} . Since \mathbf{m} is unknown, \mathbf{e} is not directly measurable from equation 3.48. Residuals, an alternative measureable type of the error, may be used instead:

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{v}. \quad 3.49$$

Replacing \mathbf{b} from equation 3.47 into equation 3.49, residuals can be expressed by:

$$\mathbf{r} = \mathbf{A}\mathbf{m} - \mathbf{A}\mathbf{v} = \mathbf{A}(\mathbf{m} - \mathbf{v}) = \mathbf{A}\mathbf{e}. \quad 3.50$$

Equation 3.50, $\mathbf{r} = \mathbf{A}\mathbf{e}$, is called the “residual equation” (Briggs et al., 2000). With \mathbf{v} as an approximate solution, \mathbf{r} is computed from equations 3.49 and 3.50. Solving the residual equation for \mathbf{e} gives a new approximate solution using equation 3.48 in the form $\mathbf{m} = \mathbf{v} + \mathbf{e}$. Substitution of the residual equation into equation 3.48 gives:

$$\mathbf{m} = \mathbf{v} + \mathbf{P}^{-1}\mathbf{r}, \quad 3.51$$

where $\mathbf{P} \cong \mathbf{A}$, is a preconditioner matrix. This suggests iterations in the form of (Briggs et al., 2000):

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \mathbf{P}^{-1}\mathbf{r}, \quad 3.52$$

where k is the iteration number. In the Jacobi method, preconditioner \mathbf{P} is the diagonal matrix of \mathbf{A} , $\mathbf{P} = \mathbf{D}$. With this in mind, and splitting the matrix \mathbf{A} into a diagonal matrix \mathbf{D} , and negative of summation of the strictly lower and upper triangular matrices, \mathbf{L} and \mathbf{U} , $\mathbf{A} = \mathbf{D} + (-\mathbf{L} + -\mathbf{U})$, equation 3.47, with \mathbf{v} as an approximation to \mathbf{m} , can be expressed as:

$$\mathbf{D}\mathbf{v} = (\mathbf{L} + \mathbf{U})\mathbf{v} + \mathbf{b}. \quad 3.53$$

Multiplication of both sides with \mathbf{D}^{-1} leads to the following equation:

$$\mathbf{v} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{v} + \mathbf{D}^{-1}\mathbf{b}. \quad 3.54$$

which suggests Jacobi iterations in the form of:

$$\mathbf{v}^{k+1} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{v}^k + \mathbf{D}^{-1}\mathbf{b}. \quad 3.55$$

The advantage of choosing \mathbf{D} , the diagonal matrix of \mathbf{A} , as the preconditioner is that the diagonal matrices are easily invertible. By definition of the Jacobi iteration matrix, \mathbf{R}_J , as $\mathbf{R}_J = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$, the Jacobi method can also be expressed by (Briggs et al., 2000):

$$\mathbf{v}^{k+1} = \mathbf{R}_J\mathbf{v}^k + \mathbf{D}^{-1}\mathbf{b}. \quad 3.56$$

The weighted Jacobi matrix is a modification to the Jacobi method which gives a constant weight to the diagonal matrix in the form of (Briggs et al., 2000):

$$\mathbf{v}^{k+1} = ((1 - w)\mathbf{I} + w\mathbf{R}_J)\mathbf{v}^k + w\mathbf{D}^{-1}\mathbf{b}, \quad 0 < w < 2, \quad 3.57$$

where w is a real valued: weighting factor. By defining the weighted Jacobi iteration as $\mathbf{R}_w = (1 - w)\mathbf{I} + w\mathbf{R}_J$, the Jacobi method can be expressed by either,

$$\mathbf{v}^{k+1} = \mathbf{R}_w\mathbf{v}^k + w\mathbf{D}^{-1}\mathbf{b}, \quad 3.58$$

or,

$$\mathbf{v}^{k+1} = \mathbf{v}^k + w\mathbf{D}^{-1}\mathbf{r}^k .$$

3.59

As shown in the next section, the weighted Jacobi method converges to the solution faster than the Jacobi method itself. The Jacobi algorithm updates \mathbf{v}_i , $i = 1, 2, \dots, N$, at each iteration using the equation

$$v_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^N a_{ij} v_j^k \right), \quad 3.60$$

(Saad, 2003), where i is the component number of the \mathbf{b} and \mathbf{v} vectors. Starting from an initial value for \mathbf{v}^0 , all components of \mathbf{v}^{k+1} are calculated in each iteration, then \mathbf{v}^k is replaced by \mathbf{v}^{k+1} . This procedure repeats until the desired convergence is achieved.

Gauss-Seidel method is an improvement to the Jacobi method in which each component of \mathbf{v} is replaced as soon as it gets updated (Saad, 2003):

$$v_i^{k+1} = \frac{1}{a_{ii}} \left(- \sum_{j=1}^{i-1} a_{ij} v_j^{k+1} - \sum_{j=i+1}^N a_{ij} v_j^k - b_i \right), i = 1, 2, \dots, N . \quad 3.61$$

This procedure not only reduces the necessary memory requirement to keep all components of \mathbf{v}^{k+1} before updating, but also decreases the number of iterations for the same convergence rate. The speedup of Gauss-Seidel method might depend on the ordering of equations. By defining the Gauss-Seidel iteration by, $\mathbf{R}_G = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U}$, the Gauss-Seidel method is expressed in matrix form by:

$$\mathbf{V} \leftarrow \mathbf{R}_G \mathbf{V} + (\mathbf{D} - \mathbf{L})^{-1} \mathbf{b}, \quad 3.62$$

where “ \leftarrow ” shows the replacement of elements.

3.4.2 Jacobi convergence and the smoothing property

The convergence of the Jacobi (and the Gauss-Seidel) iterations are guaranteed if the magnitudes of all eigenvalues of \mathbf{R}_J are less than 1 (Strang, 2006):

$$|\lambda(\mathbf{R}_J)| < 1. \quad 3.63$$

The spectral radius of a matrix, ρ , is the maximum amount of its eigenvalues, $\rho(\mathbf{R}_J) = \max|\lambda(\mathbf{R}_J)|$. Therefore, the Jacobi and the Gauss-Seidel methods converge when $\rho(\mathbf{R}_J) < 1$. The speed of convergence depends on the size of $\rho(\mathbf{R}_J)$. Smaller $\rho(\mathbf{R}_J)$ results in faster convergence to the solution (Strang, 2006). Equivalently, Jacobi and Gauss-Seidel converge to the solution if matrix \mathbf{A} (in equation 3.47) is a diagonally dominant matrix. The fastest convergence (in one iteration) happens when \mathbf{A} is a diagonal matrix.

In order to explain the role of the Jacobi iterations in the multigrid method, consider as an example a system of equation 3.47 where \mathbf{A} is the second difference matrix:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & & & \\ -1 & 2 & -1 & \dots & & \\ 0 & -1 & 2 & & & \\ & \vdots & & & & 0 \\ & & & & 2 & -1 \\ & & & 0 & -1 & 2 \end{bmatrix}. \quad 3.64$$

The Jacobi iteration matrix is:

$$\mathbf{R}_J = \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 & & & \\ 1 & 0 & 1 & \dots & & \\ 0 & 1 & 0 & & & \\ & \vdots & & & & 0 \\ & & & & 0 & 1 \\ & & & & 0 & 1 & 0 \end{bmatrix}. \quad 3.65$$

The eigenvalues of \mathbf{A} are $\lambda_j(\mathbf{A}) = 2 - 2 \cos j\theta$, where $\theta = \frac{\pi}{N+1}$. Thus, $\lambda_j(\mathbf{R}_J) = \lambda(I - \frac{1}{2} \mathbf{A}) = \cos j\theta < 1$ and convergence is guaranteed but it is very slow. For example if $N = 4$, then \mathbf{R}_J has four eigenvalues as (Briggs et al., 2000):

$$\lambda(\mathbf{R}_J)_{j=1,2,3,4} = \cos \frac{\pi}{5}, \cos \frac{2\pi}{5}, \cos \frac{3\pi}{5}, \cos \frac{4\pi}{5} (= -\cos \frac{\pi}{5}). \quad 3.66$$

As observed, ignoring $\lambda_{j=4}$, λ_s are larger for smaller angles. It means that the convergence is slower for the lower frequencies of the solution. This is a general property of the Jacobi iterations. When solving a problem with the Jacobi method, answers with higher frequency components are retrieved first. If the program terminates after just a few iterations, the solution contains mostly high frequency components of the final solution, and the residuals contain mostly low frequency components.

Since some λ s are close to 1, convergence of Jacobi iterations is relatively slow for this example. As seen in equation 3.66, $|\lambda_1| = |\lambda_N|$, which means that the convergences for these two low and high frequencies are equal. This is not very appropriate for a multigrid solver. Eigenvalues for the weighted Jacobi methods are

$$\lambda_j(\mathbf{R}_w) = \lambda_j \left((1-w)\mathbf{I} + w \left(\mathbf{I} - \frac{1}{2}\mathbf{A} \right) \right) = \lambda_j \left(\mathbf{I} - \frac{w}{2}\mathbf{A} \right), \quad 3.67$$

which can be expressed as:

$$\lambda_j(\mathbf{R}_w) = 1 - w + w \cos j\theta. \quad 3.68$$

Optimal damping for the high frequency content is obtained when $w = 2/3$, (Strang, 2006). For instance with $N = 4$, \mathbf{R}_w has 4 eigenvalues as

$$\begin{aligned} \lambda(\mathbf{R}_w)_{j=1,2,3,4} &= \frac{1}{3} + \frac{2}{3} \left(\cos \frac{\pi}{5}, \cos \frac{2\pi}{5}, \cos \frac{3\pi}{5}, \cos \frac{4\pi}{5} \right) \\ &\cong 0.8727, 0.5393, 0.1273, -0.2060. \end{aligned} \quad 3.69$$

Higher values of the λ s for the low frequency components makes the weighted Jacobi method, with $w = 2/3$, a good smoother as the multigrid solver. Since weighted Jacobi eigenvalues are smaller than the Jacobi eigenvalues, the weighted Jacobi method converge faster than the Jacobi method.

This “smoothing” property is shown in the following example. Consider the system of equation

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad 3.70$$

where \mathbf{A} is the second difference matrix in equation 3.47, and $\mathbf{b} = [1 \ 0 \ 0 \ \dots \ 0 \ 0 \ 1]^T$, with $n = 64$. The trivial exact solution is $\mathbf{x} = [1 \ 1 \ \dots \ 1 \ 1]^T$. Let’s apply the weighted Jacobi method with $w = 2/3$ to solve this equation with $\mathbf{x}_0_j = 1 + \sin\left(\frac{jk\pi}{n}\right)$, $0 \leq j \leq n$, $1 \leq k \leq n-1$, the Fourier modes with the frequency k , plus one, as the initial values.

Figure 3.11a shows the convergence rate of the weighted Jacobi method for different initial values with $k = 1, 4, 16$, and 64 . By choosing an initial guess with higher frequency components (larger k s) the convergence to the solution is faster. The convergence rate is shown by error norm verses iteration number. The norm of the vector \mathbf{e} usually expresses the error. The Euclidean norm or L2-norm, $\|\mathbf{e}\|_2$, of a vector \mathbf{e} with n elements is expressed by:

$$\|\mathbf{e}\|_2 = \sqrt{\sum_{j=1}^n e_j^2}. \quad 3.71$$

The maximum norm, $\|\mathbf{e}\|_\infty$, is defined as the largest absolute value of the vector elements:

$$\|\mathbf{e}\|_\infty = \max |e_j|. \quad 3.72$$

The solutions corresponding to initial values with different frequency contents (k s) after 50 iterations are shown in Figure 3.11b. With higher frequency content in the initial value, the solution is closer to the exact solution, $\mathbf{x} = [1 \ 1 \ \dots \ 1 \ 1]^T$. With $k = 64$, the solution is good enough. It can be shown that the weighted Gauss-Seidel method behaves similarly.

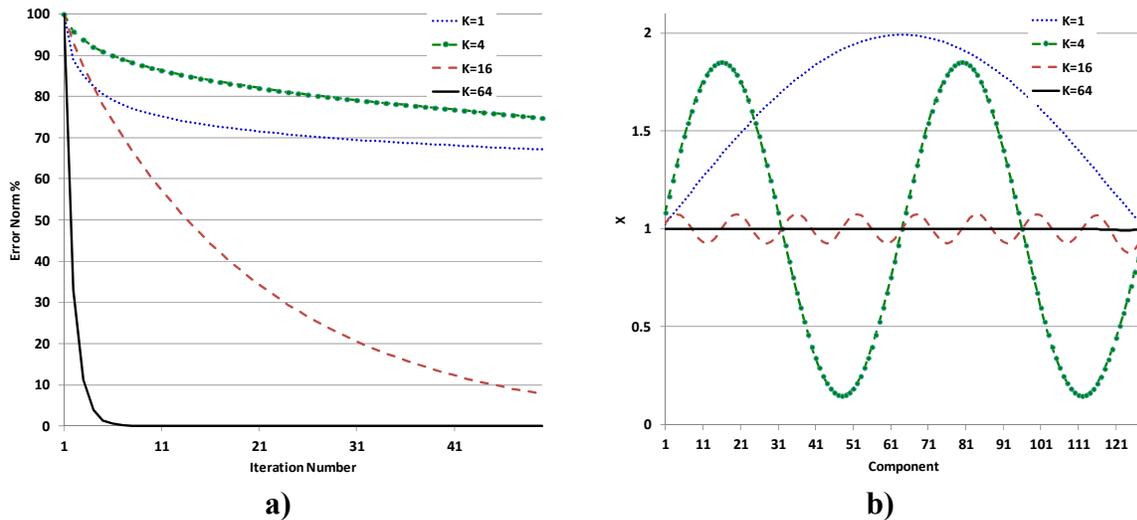


Figure 3.11 a) The Euclidean error norm versus the number of iterations for the weighted Jacobi iterations with different initial values to solve equation 3.70, b) Solutions after 50 iterations.

Suppose the initial guess is a superposition of all four Fourier modes,

$$\mathbf{x}_0 = 1 + \frac{1}{4} \left[\sin\left(\frac{j\pi}{n}\right) + \sin\left(\frac{4j\pi}{n}\right) + \sin\left(\frac{16j\pi}{n}\right) + \sin\left(\frac{64j\pi}{n}\right) \right]. \quad 3.73$$

The convergence rate for the weighted Jacobi iterations starting with this signal as the initial guess is shown in Figure 3.12a. The error norm rapidly decreases during the first four iterations. From the fifth iteration onwards, the convergence becomes slower. The fast decrease corresponds to the presence of the high frequency components in the initial value with the less rapid decrease of the error norm is due to the lower frequency components of the initial value

(Strang, 2006). Figure 3.12b shows the solution after 50 iterations. While with an initial guess with high frequency (Figure 3.11b), the solution comes very close to the exact solution, with an initial guess that is a combination of low and high frequencies, an acceptable solution is not achieved. Several more iterations are required to achieve a better solution.

When the weighted Jacobi method solves for the high frequency components of a solution in the first few iterations, the residuals consist of mostly low frequency components. It is a smoother version of the residuals. For this reason, weighted Jacobi and Gauss-Seidel methods are called smoothers.

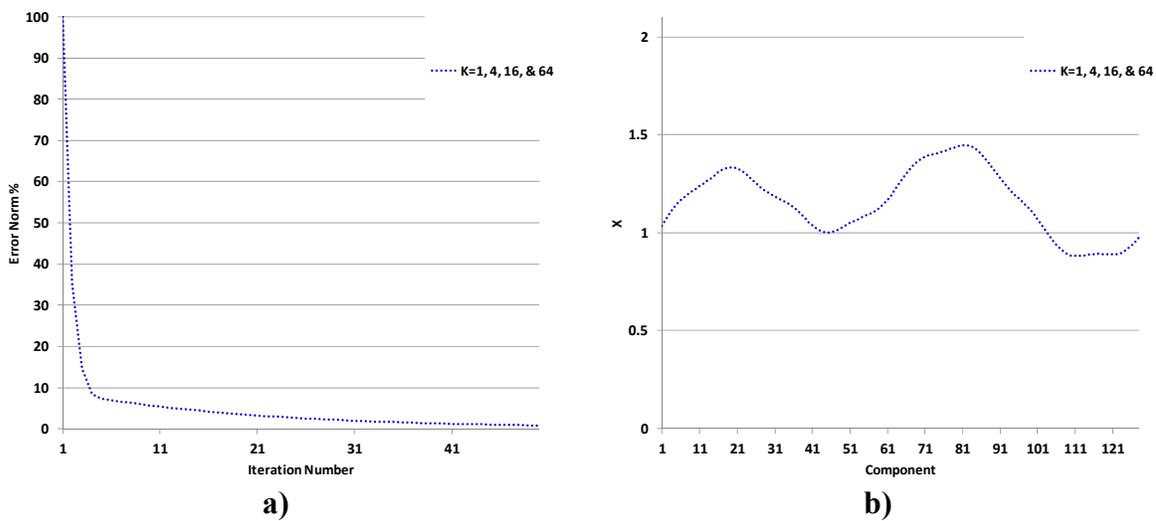


Figure 3.12 a) Convergence rate of the weighted Jacobi method when the initial solution is a superposition of both low and high frequencies to solve equation 3.70, b) Solution after 50 iterations.

3.4.3 Elements of the multigrid method

The smoothing property of the Jacobi and Gauss-Seidel methods leads to the multigrid idea. In the multigrid method, an iterative solver, weighted Jacobi or Gauss-Seidel in general, produces a low frequency signal in the residuals after a few iterations of equation 3.47. By a restriction process (down sampling), the main problem and residuals are transferred to a coarser grid (scale), $2h$, where the low frequency components act as high frequency components. When a low frequency signal acts as high frequency signal, a solution can be recovered faster by Jacobi iterations. Solving the original equation with an initial starting point from both fine and coarse

grids in the fine grid h , gives a solution which contains more low frequency components than when solving the equation with a vector of zeros as the initial guess (Strang, 2006).

There are two processes in the multigrid method in order to transfer the problem to the coarser ($2h$) or finer (h) grids (Strang, 2006): multiplying by a restriction matrix, \mathbf{R}_h^{2h} , which transfers the problem from fine grid, h , to the coarse grid, $2h$, and multiplying with an interpolation (or prolongation) matrix, \mathbf{I}_{2h}^h , which transfers a problem back to the finer grid. Many interpolation methods may be used. Linear is the simplest effective method of interpolation for the multigrid method. Using any interpolation method, a restriction matrix can be obtained by equation (Strang, 2006),

$$\mathbf{R}_h^{2h} = \frac{1}{2} (\mathbf{I}_{2h}^h)^T. \quad 3.74$$

As an example, a 1D linear interpolation matrix to transfer a vector $\mathbf{m}^{2h} = [m_1^{2h} \ m_2^{2h} \ m_3^{2h}]^T$ to a finer grid size has the following form:

$$\mathbf{I}_{2h}^h = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix}. \quad 3.75$$

Therefore, Interpolation to a finer grid will be performed as,

$$\mathbf{I}_{2h}^h \mathbf{m}^{2h} = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_1^{2h} \\ m_2^{2h} \\ m_3^{2h} \end{bmatrix} = \begin{bmatrix} m_1/2 \\ m_1 \\ (m_1 + m_2)/2 \\ m_2 \\ (m_2 + m_3)/2 \\ m_3 \\ m_3/2 \end{bmatrix} = \begin{bmatrix} m_1^h \\ m_2^h \\ m_3^h \\ m_4^h \\ m_5^h \\ m_6^h \\ m_7^h \end{bmatrix} = \mathbf{m}^h. \quad 3.76$$

The corresponding restriction matrix and its operation to transfer from a fine grid to coarse grid size are:

$$\mathbf{R}_h^{2h} \begin{bmatrix} m_1^h \\ m_2^h \\ m_3^h \\ m_4^h \\ m_5^h \\ m_6^h \\ m_7^h \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} m_1^h \\ m_2^h \\ m_3^h \\ m_4^h \\ m_5^h \\ m_6^h \\ m_7^h \end{bmatrix} = \begin{bmatrix} \hat{m}_1^{2h} \\ \hat{m}_2^{2h} \\ \hat{m}_3^{2h} \end{bmatrix} = \hat{\mathbf{m}}^{2h}. \quad 3.77$$

Interpolation and restriction processes on a 2D grid follow same roles as in 1D. The concept of interpolation and restriction are graphically shown in Figure 3.13.

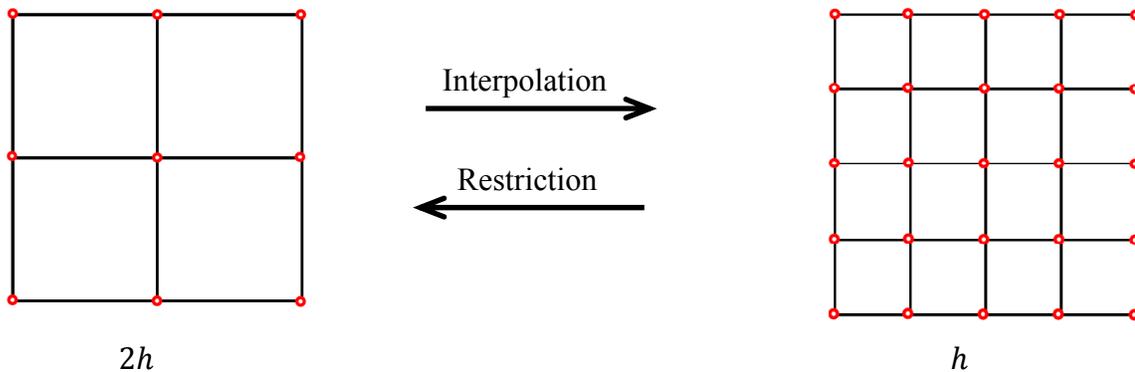


Figure 3.13 Interpolation and restriction processes illustrated schematically on a 2D grid.

Different types of multigrid methods are used in the mathematical literature. The v-cycle (with a lower case v) multigrid is the simplest. A v-cycle multigrid starts with a few (three for instance) iterations on the fine grid, then the error is transferred to a coarse grid by a restriction process, and iterations are performed on the coarse grid. Results being interpolated to the fine grid. The interpolated results are then used as the starting point in the weighted Jacobi method on the fine (main) grid.

A v-cycle multigrid includes only two grids: a fine grid (with the size of the main problem) and a coarse grid (Figure 3.14 and Figure 3.15). Iterations (weighted Jacobi or Gauss-Seidel) start with zero as the initial model to solve the equation $\mathbf{A}^h \mathbf{u}^h = \mathbf{b}^h$ on the fine grid where h corresponds to the size of main grid which is finest grid. After a few iterations, residuals to equation $\mathbf{A}^h \mathbf{u}^h = \mathbf{b}^h$, are calculated by $\mathbf{r}^h = \mathbf{b}^h - \mathbf{A}^h \mathbf{u}^h$. Then, multiplication with the restriction matrix converts \mathbf{r}^h to the coarser grid \mathbf{r}^{2h} , $\mathbf{r}^{2h} = \mathbf{R}_h^{2h} \mathbf{r}^h$, where $2h$ corresponds to the

first coarser grid with size equal to half of the fine grid size. Solving $A^{2h}e^{2h} = r^{2h}$ for e^{2h} on the coarse grid requires a few more iterations, and then the solution, e^{2h} , must be interpolated to the fine grid as e^h , $e^h = I_{2h}^h e^{2h}$. Finally, iterations to solve $A^h u^h = b^h$ starts with the improved initial value $u^h + e^h$. A few iterations on this grid size return a solution which includes more low frequency components than would solving the equation with a zero vector as the initial after the same number of iterations (Strang, 2006; Briggs et al., 2000).

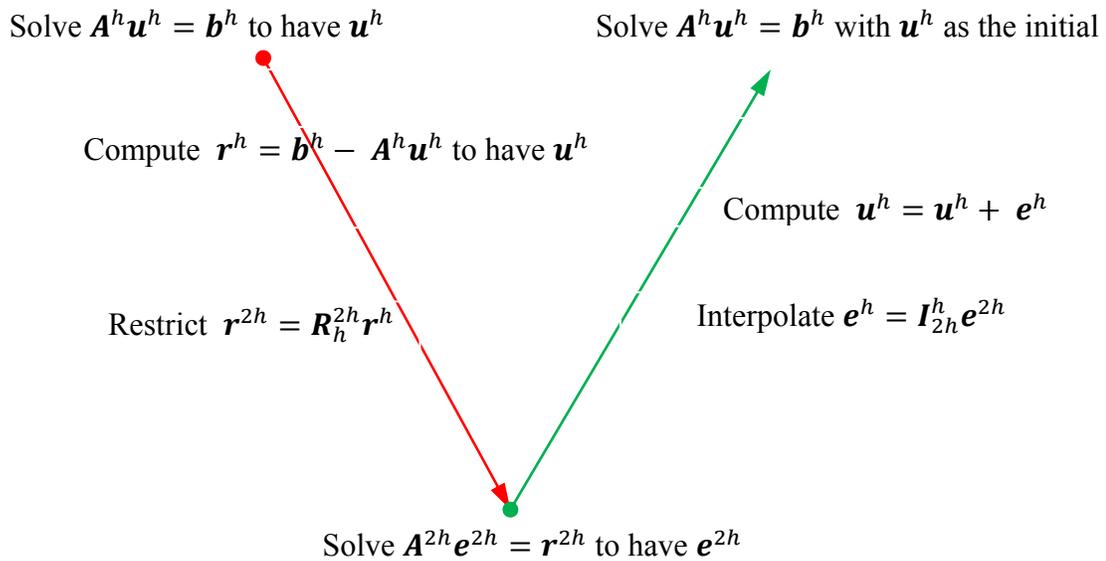


Figure 3.14 Schematic v-cycle multigrid. The main problem is solved on the fine grid, and low-frequency residuals are restricted to the coarse grid used to solve equation in the coarse grid. Solution in the coarse grid is interpolated to the fine grid and improved the initial value for iterations on the main grid.

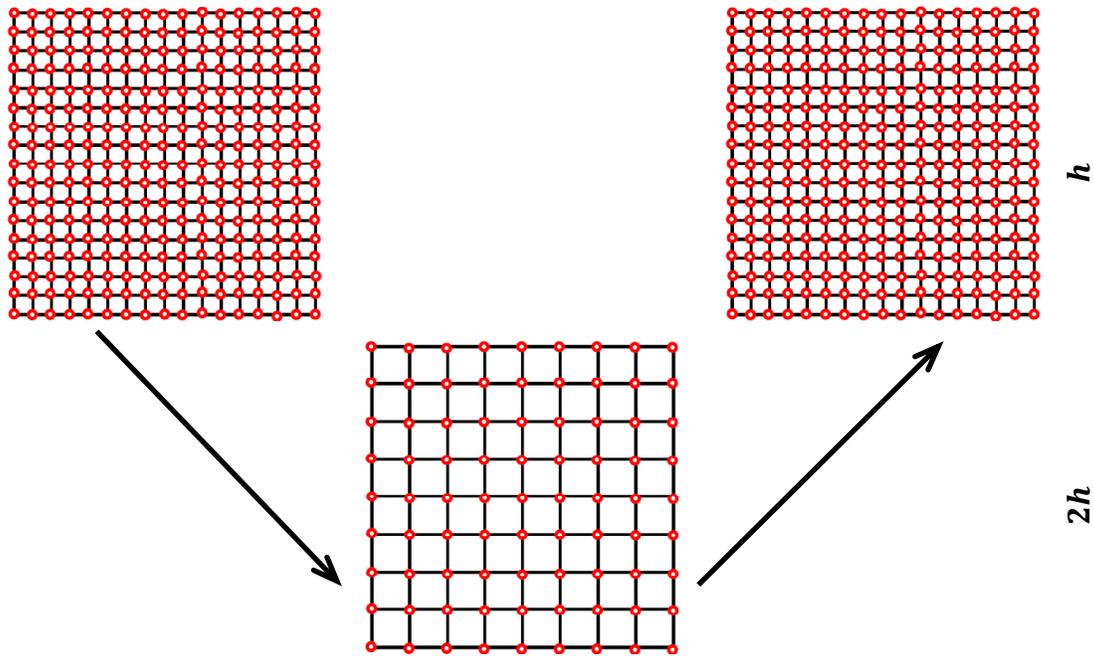


Figure 3.15 A v-cycle multigrid concept is shown on a 2D grid.

The v-cycle is the simplest algorithm in the multigrid methods. Very low frequencies are still low in the coarse grid, $2h$. It is possible to calculate the residuals in the coarse grid, $\mathbf{r}^{2h} = \mathbf{r}^{2h} - \mathbf{A}^{2h} \mathbf{e}^{2h}$, and restrict them into an even coarser grid $4h$ and repeat the procedure to a very coarse grid, nh . This algorithm is known as V-cycle (note the capital V) multigrid (Figure 3.16 and Figure 3.17). Consequently, the W-cycle algorithm performs more iterations on the coarser grids (Strang, 2006; Briggs et al., 2000) as seen in Figure 3.16. This method leads to a better recovery of the low frequencies than the V-cycle method. In the full multigrid (FMG), iteration starts on the coarsest grid, the solution is interpolated and used as the initial value for a grid that is one step finer. A v-cycle improves the result. Then, the result will be used for the one degree finer grid. A V-cycle improves this result and the process continues to arrive to the finest grid which is the size of the initial problem (Strang, 2006; Briggs et al., 2000).

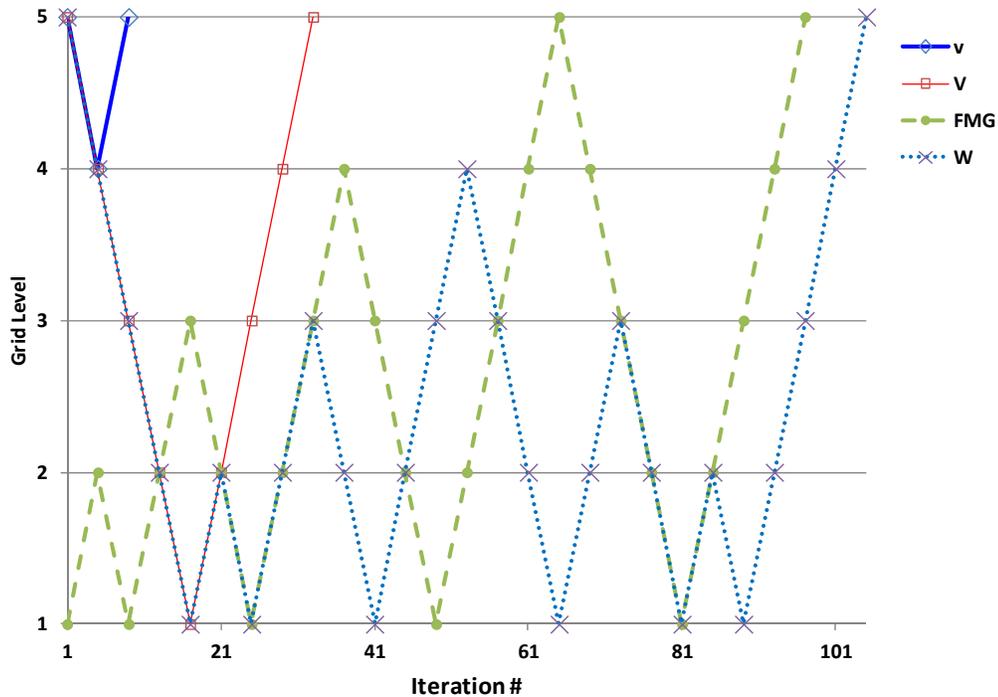


Figure 3.16 A comparison between v-cycle, V-cycle, W-cycle, and full multigrid methods.

To compare the advantages of using multigrid a method vs. the Jacobi method, consider the example in section 3.4.2. Equation 3.69 is solved by the full multigrid method with 25 iterations on each grid. Figure 3.18a shows the convergence of the full multigrid method after the last 25 iterations on the main grid. Compared with the convergence rate in Figure 3.12a, it shows the advantage of using multigrid to the Jacobi method. Figure 3.18b compares the result of the Jacobi method after 50 iterations with the result of the full multigrid method. After 25 iterations in each grid, multigrid resulting a good approximation to the exact solution, $\mathbf{x} = 1$, where the Jacobi method requires more iterations to return a good approximation to the real solution. The time spent in the coarse grids is very low in comparison to the time spent in the main problem (finest grid). Therefore, multigrid is cheaper than applying many Jacobi iterations to achieve the same solution in the main grid size.

A V-cycle multigrid or full multigrid with 25 iterations on each grid is a little more time consuming than performing 50 Jacobi iterations. However, the Jacobi method requires many more iterations to arrive at the same solution as the multigrid solution.

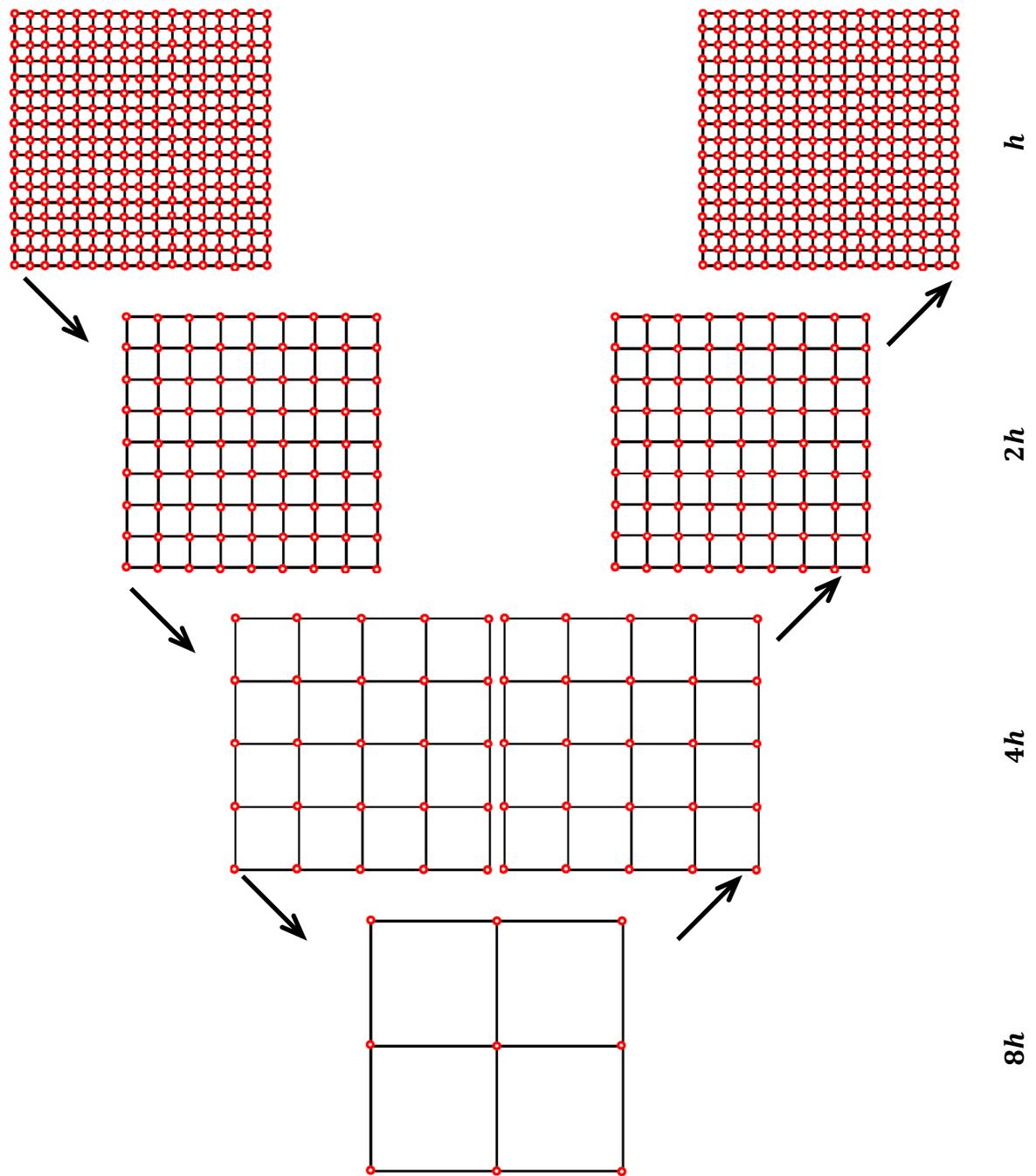


Figure 3.17 V-cycle multigrid concept shown on a 2D grid.

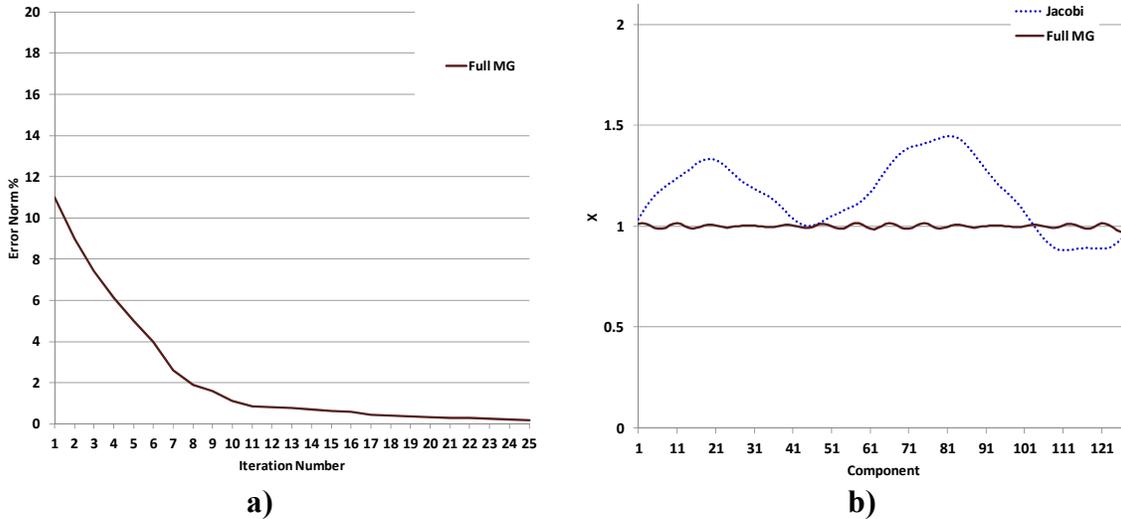


Figure 3.18 a) The residuals error norm versus the number of iterations is plotted for the last cycle of the full multigrid method. b) Comparison of the solution of the weighted Jacobi after 50 iteration with the solution of the full multigrid after 25 iterations on each grid.

3.4.4 Feasibility of solving the LSPSM equation with standard multigrid

The multigrid methods use weighted Jacobi or Gauss-Seidel as the smoother solver. In both solvers it is necessary to extract the diagonal elements of matrix \mathbf{A} (in equation 3.47) and invert it. Therefore, it is necessary to have \mathbf{A} (or $\mathbf{G}^T \mathbf{G}$) in the explicit or matrix form. From this point of view, multigrid may increase the speed of convergence, but it significantly increases the required memory to load matrix \mathbf{G} and to create matrix $\mathbf{G}^T \mathbf{G}$.

As shown in section 3.2.4, matrix $\mathbf{G}^T \mathbf{G}$ is a symmetric matrix with relatively higher values in the diagonal entries than the non-diagonal entries. However, the experience with the explicit form of $\mathbf{G}^T \mathbf{G}$ matrices for different patterns of seismic data acquisition shows the Hessians to be relatively dense matrices. The presence of relatively large entries in each row prevents diagonal entries from being dominant (larger than sum of absolute values of the non-zero elements). Since the matrix $\mathbf{G}^T \mathbf{G}$ is diagonally non-dominant, weighted Jacobi or Gauss-Seidel are not able to solve equation 3.1. Therefore, multigrid in its conventional formulation is not viable to solve the Kirchhoff LSPSM equation.

Many other approaches using multigrid methods in solving LSPSM equation may be considered. An alternative experiment applies the restriction operator several times on the matrix $\mathbf{G}^T \mathbf{G}$ and converts it to a very coarse matrix. Many synthetic examples show this experiment does

not change the diagonally non-dominancy of the matrix. In fact, the restricted matrix is denser than the original matrix. Therefore, the idea of transferring the problem from the main grid to a coarser grid, performing standard multigrid on the coarse grid, and using the interpolated result as the initial value for another iterative method such as CG, is not efficient in, or applicable to, these problems.

Another possibility would be reducing the size of \mathbf{G} (and $\mathbf{G}^T\mathbf{G}$) by solving for each column of the reflectivity model at each time ($ix = 1, \dots, Nx$). This procedure reduces the size of $\mathbf{G}^T\mathbf{G}$ from $(Nx \times Nz)^2$ to only $(1 \times Nz)^2$ where Nx and Nz are numbers of model grid points in the horizontal (usually equal to the number of CMPs) and vertical directions, respectively. By inverting for each column separately and putting the resulting inverted columns next to each other, the inversion image could be achieved. This separation may only be valid when there are only horizontal velocity variations in the subsurface. However, many experiments show that the matrix $\mathbf{G}_{ix}^T\mathbf{G}_{ix}$, is not a diagonally dominant matrix and consequently not solvable by Jacobi or Gauss-Seidel methods.

Therefore, the main precondition for the Kirchhoff LSPSM equation to be solvable by the standard solvers of the multigrid method is violated. As a result, at least in its standard definition, multigrid is not an effective method for solving the LSPSM problem.

3.5 Feasibility of solving LSPSM with CG and multilevel CG methods

Jacobi, Gauss-Seidel, and consequently standard multigrid methods are not applicable to solve the LSPSM equation. The CG method can handle the LSPSM equation. Since CG is not a multigrid solver, I used a multilevel solver for LSPSM equation in different data or model sizes with iterative methods other than the Jacobi or Gauss-Seidel solvers. CG methods are developed from the steepest descent method.

3.5.1 Steepest descent method

Steepest descent and CG are probably the most efficient methods to solve a large system of linear equations of the form

$$\mathbf{A}\mathbf{m} = \mathbf{b}, \quad 3.78$$

where \mathbf{A} is an n by n symmetric positive-definite sparse matrix, \mathbf{m} is the unknown vector to be determined, and \mathbf{b} is a known vector. The square matrix \mathbf{A} is called positive-definite if for any non-zero column vector \mathbf{m} with all entries real ($m \in \mathbb{R}$), the following inequality is valid:

$$\mathbf{m}^T \mathbf{A} \mathbf{m} > 0. \quad 3.79$$

Consider the quadratic function of a vector \mathbf{m} in the following form:

$$f(\mathbf{m}) = 1/2 \mathbf{m}^T \mathbf{A} \mathbf{m} - \mathbf{b}^T \mathbf{m} + c, \quad 3.80$$

where \mathbf{b} is a vector with the same length as \mathbf{m} and c is a constant scalar. Since \mathbf{A} is a positive-definite matrix, $f(\mathbf{m})$ has a paraboloid shape with a global minimum (Shewchuk, 1994). \mathbf{m} is a vector of n variables, and the gradient of $f(\mathbf{m})$ is defined by,

$$f'(\mathbf{m}) = \begin{bmatrix} \partial/\partial m_1 f(\mathbf{m}) \\ \partial/\partial m_2 f(\mathbf{m}) \\ \vdots \\ \partial/\partial m_n f(\mathbf{m}) \end{bmatrix} = 1/2 \mathbf{A}^T \mathbf{m} + 1/2 \mathbf{A} \mathbf{m} - \mathbf{b}. \quad 3.81$$

From linear algebra we know that the first derivative of a function gives the slope of the function at a certain point. Hence, $f(\mathbf{m})$ can be minimized by setting $f'(\mathbf{m}) = 0$:

$$1/2 (\mathbf{A}^T + \mathbf{A}) \mathbf{m} - \mathbf{b} = \mathbf{0}. \quad 3.82$$

Since \mathbf{A} is symmetric, equation 3.82 reduces to $\mathbf{A}\mathbf{m} = \mathbf{b}$. Therefore, the solution to $\mathbf{A}\mathbf{m} = \mathbf{b}$ is equivalent to finding the minimum of $f(\mathbf{m})$, where the gradients are zero. Steepest descent and CG iterations find the solution to equation 3.78 by finding the minimum of the corresponding quadratic form. Starting at any arbitrary initial point \mathbf{m}_i , the steepest descent method chooses the direction opposite to the maximum gradient, $-f'(\mathbf{m}_i) = \mathbf{b} - \mathbf{A}\mathbf{m}_i$, in which $f(\mathbf{m})$ decrease most quickly at each iteration (Shewchuk, 1994). Steepest descent algorithms follow this direction until the gradient changes. Then the algorithm changes the direction to a direction which is orthogonal to the previous direction to find a new minimum. This procedure

repeats until the global minimum of the paraboloid is found. Briefly, steepest descent iterations include these three main steps (Shewchuk, 1994):

Step one: A residual which measures how far the current solution is from the exact answer is computed. It also shows the direction of the steepest descent for the current iteration:

$$\mathbf{r}_i = \mathbf{b} - \mathbf{A}\mathbf{m}_i \quad 3.83$$

Step two: Calculate how far the line search has to go in the direction specified in the previous step:

$$\alpha_i = \frac{\mathbf{r}_i^T \mathbf{r}_i}{\mathbf{r}_i^T \mathbf{A} \mathbf{r}_i}. \quad 3.84$$

Step three: Update the solution by adding the appropriate distance calculated in the second step at the direction specified in step one:

$$\mathbf{m}_{i+1} = \mathbf{m}_i + \alpha_i \mathbf{r}_i. \quad 3.85$$

The first two steps include matrix-vector multiplications and are responsible for the high cost of the steepest descent algorithm.

3.5.2 CG and LSCG

CG and steepest descent methods use the same first iteration. In the CG method (Hestenes and Stiefel, 1952), which is an improvement over the steepest descent method, the new direction is orthogonal to all previous directions. Therefore, the CG method converges to the solution in less iterations than the steepest descent method. To avoid returning to one of the previously chosen directions, the third step of the steepest descent method is replaced by,

$$\mathbf{m}_{i+1} = \mathbf{m}_i + \alpha_i \mathbf{d}_i. \quad 3.86$$

where $\alpha_i = \mathbf{r}_i^T \mathbf{r}_i / \mathbf{d}_i^T \mathbf{A} \mathbf{d}_i$, is the step distance, and \mathbf{d}_i , $i = 1, \dots, n$, are a set of orthogonal search directions which are connected to the residuals by,

$$\mathbf{d}_{i+1} = \mathbf{r}_{i+1} + \beta_{i+1} \mathbf{d}_i, \quad 3.87$$

where $\beta_i = \mathbf{r}_{i+1}^T \mathbf{r}_{i+1} / \mathbf{r}_i^T \mathbf{r}_i$.

The ratios of the extremal eigenvalues of \mathbf{A} in equation 3.47 determine the convergence rate of the steepest descent method. The convergence rate of the CG method depends on the ill-conditioning of the whole matrix. Convergence is slower for matrices with larger condition numbers (more ill-conditioned problems). Ignoring the accumulation of the computer floating

points roundoff errors, convergence of the CG method to the exact solution in at most n iterations is guaranteed. From this point of view CG may be considered a direct method.

Similar to the steepest descent method, the CG method requires that \mathbf{A} be a symmetric and a positive definite matrix. However, it has most of the properties that an effective solver must have. Properties of the CG method include low memory space requirement, fast convergence to the solution, and the possibility of having access to the solution after each iteration (Hestenes and Stiefel, 1952). The original CG method advanced further to other improved types in order to speed up the convergence rate, eliminate the requirement of a positive definite matrix, generalize to complex systems, and solve nonlinear equations (for example see: Jacobs, 1986; Sonneveld, 1989; Van der Vorst, 1992; Hanger and Zhang, 2006; and Vujicic, 2006). The CG method is an effective solver for sparse matrices and has been widely used for solving geophysical inversion problems (Wang and Treitel, 1973; Koehler and Taner, 1985; Gersztenkorn et al., 1986; Scales, 1987).

Since the matrix \mathbf{G} is not square in general, the CG method is not directly applicable to solve the equation $\mathbf{G}\mathbf{m} = \mathbf{d}$. Scales (1987) adapted Hestenes and Stiefel's CG method to the least squares conjugate gradient (LSCG) method to be able to solve the normal equation 3.1. The LSCG method does not require the explicit form of \mathbf{A} (or equally $\mathbf{G}^T\mathbf{G}$ in equation 3.1) and directly works with the \mathbf{G} and \mathbf{G}^T matrices (Scales, 1987). Since there is no decomposition of \mathbf{G} or \mathbf{G}^T in the LSCG algorithm, it is possible to use operators instead of the multiplication of explicit forms of \mathbf{G} or \mathbf{G}^T matrices with the vectors. By replacing the method of CG with LSCG, the multiplication of the matrices \mathbf{G} or \mathbf{G}^T with vectors is replaced with applying forward (seismic modeling or de-migration) or adjoint (seismic migration) operators to the model or data, respectively. This procedure significantly reduces the required computer memory to load very large \mathbf{G} and \mathbf{G}^T matrices, and also avoids large matrix-vector multiplications. An LSCG algorithm to solve $\mathbf{G}^T\mathbf{G}\mathbf{m}_{LS} = \mathbf{G}^T\mathbf{d}$ is shown in Table 3-6.

Table 3-6 LSCG algorithm to solve the Kirchhoff LSPSM equation.

$\mathbf{m}_0 = \text{an initial guess or } \mathbf{m}_0 = \mathbf{0}$ $\mathbf{s}_0 = \mathbf{d} - \mathbf{G}\mathbf{m}_0$ $\mathbf{r}_0 = \mathbf{G}^T \mathbf{s}_0$ $\mathbf{p}_0 = \mathbf{r}_0$ $\mathbf{q}_0 = \mathbf{G}\mathbf{p}_0$ for $i = 0$: iterations limit $\alpha_{i+1} = \frac{\mathbf{r}_i \cdot \mathbf{r}_i}{\mathbf{q}_i \cdot \mathbf{q}_i}$ $\mathbf{m}_{i+1} = \mathbf{m}_i + \alpha_{i+1} \mathbf{p}_i$ $\mathbf{s}_{i+1} = \mathbf{s}_i - \alpha_{i+1} \mathbf{q}_i$ $\mathbf{r}_{i+1} = \mathbf{G}^T \mathbf{s}_{i+1} \quad \rightsquigarrow$ $\beta_{i+1} = \frac{\mathbf{r}_{i+1} \cdot \mathbf{r}_{i+1}}{\mathbf{r}_i \cdot \mathbf{r}_i}$ $\mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \beta_{i+1} \mathbf{p}_i$ $\mathbf{q}_{i+1} = \mathbf{G}\mathbf{p}_{i+1} \quad \rightsquigarrow$ endfor

There are two multiplications of \mathbf{G}^T and \mathbf{G} matrices with vectors (lines marked by \rightsquigarrow and \rightsquigarrow symbols, respectively) at each iteration. Each iteration using LSCG has at least twice the computational cost of one migration (or modeling). For example, the time required by 10 iterations in the LSCG is greater than the time required by 20 migrations. Nemeth et al., 1999, Duquet et al., 2000, and Yousefzadeh and Bancroft, 2012d used LSCG for solving the Kirchhoff LSPSM. Nemeth et al., 1999 inverted their Kirchhoff least squares migration normal equation using a preconditioned linear LSCG. I realized that after 10 to 15 iterations in the LSCG method, the resolution of the resulting LSPSM image is high enough for further seismic data analyses and data reconstruction. The reflectivity images in Chapter Two were obtained using the LSCG method.

Each Kirchhoff prestack migration includes calculations and the application of the DSR equation, proper weight function, rho filter, antialiasing filter, crosscorrelation, and interpolation. In order to have a modeling operator which is exactly the adjoint of the migration operator, the

adjoint of all these steps must also be considered in the modeling. This procedure causes the modeling operator to be even more expensive than the migration operator. For instance, adding a triangle antialiasing filter to the migration operator makes the modeling operator twice as expensive as the migration operator.

3.5.3 LSCG for solving a regularized LSPSM equation

Equation 3.1 is the simplest form of the LSPSM since there is no regularization term involved. The LSCG algorithm in Table 3-6 can be easily extended to solve a regularized LSPSM equation without a requirement of creating large matrices. A minimum or Euclidian norm is the simplest form of a regularization function, $\mathcal{R}(m) = \|m\|_2$, which make the inversion problem stable. It leads to the damped least squares solution, m_{DSL} , by solving the following equation,

$$(\mathbf{G}^T \mathbf{G} + \mu^2 \mathbf{I}) \mathbf{m}_{DSL} = \mathbf{G}^T \mathbf{d}. \quad 3.88$$

The LSCG method to solve equation 3.88 requires the system to be in the augmented matrix form:

$$\begin{bmatrix} \mathbf{G} \\ \mu \mathbf{I} \end{bmatrix} \mathbf{m}_{DSL} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix}. \quad 3.89$$

Considering $\begin{bmatrix} \mathbf{G} \\ \mu \mathbf{I} \end{bmatrix} = \check{\mathbf{G}}$ and $\begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} = \check{\mathbf{d}}$, equation 3.89 can be written as,

$$\check{\mathbf{G}} \mathbf{m}_{DSL} = \check{\mathbf{d}}. \quad 3.90$$

and solved using the LSCG algorithm as outlined in Appendix A.

With the smoothing in the offset direction as the regularization term, $\mathcal{R}(m) = \|\mathbf{D}_h m\|^2$, equation

$$(\mathbf{G}^T \mathbf{G} + \lambda^2 \mathbf{D}_h^T \mathbf{D}_h) \mathbf{m}_{SSL} = \mathbf{G}^T \mathbf{d}. \quad 3.91$$

must be augmented to

$$\begin{bmatrix} \mathbf{G} \\ \lambda \mathbf{D}_h \end{bmatrix} \mathbf{m}_{SSL} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix}. \quad 3.92$$

Appendix B shows the LSCG algorithm for solving this equation.

3.5.4 BiCG Stabilized method for solving the LSPSM equation

Since the condition number of the matrix $\mathbf{G}^T \mathbf{G}$ is the square of the condition number of the matrix \mathbf{G} , the convergence speed of the LSCG method is significantly less than the convergence speed of the CG method (Shewchuk, 1994). The BiConjugate Gradient (BiCG) method (Appendix C) is an alternative to the LSCG method which does not alter the condition number of the matrix and converges to the solution as fast as the CG method does. BiCG was originally developed for solving equation 3.47 where \mathbf{A} is not a symmetric matrix. However, BiCG is not applicable to LSPSM when we try to avoid using matrix forms of \mathbf{G} and \mathbf{G}^T . This is due to the requirement of a multiplication of \mathbf{A}^T with a model vector. Matrix \mathbf{A} in the BiCG algorithm is equivalent to matrix $\mathbf{G}^T \mathbf{G}$ in equation 3.1. Multiplication of $\mathbf{G}^T \mathbf{G}$ with the vector, \mathbf{m} , for instance, is possible by forward modeling of \mathbf{m} followed by the migration of the resulting data. However, since it is not possible to migrate the reflectivity model, \mathbf{m} , multiplication of $\mathbf{G} \mathbf{G}^T$ with \mathbf{m} is not feasible using operator forms of the modeling and migration.

The Conjugate Gradient Squared (CGS) method (Sonneveld, 1989) is an improved version of BiCG in which multiplication by \mathbf{A}^T is avoided. Each iteration of CGS includes three multiplications of $\mathbf{G}^T \mathbf{G}$ with vectors, which means each iteration requires three runs of Kirchhoff modeling and three runs of a Kirchhoff migration. I propose the use of another alternative, the BiConjugate Gradient Stabilized (BiCGSTAB) method (Van der Vorst, 1992). BiCGSTAB is a modified version of CGS for non symmetric matrices and has faster and smoother convergence. Each BiCGSTAB iteration includes two multiplications of \mathbf{A} to a vector, but there is no requirement of multiplying \mathbf{A}^T to any vector (Appendix D). This property makes the BiCGSTAB method applicable to solve the LSPSM equation.

LSPSM of the synthetic example solved by LSCG in Chapter Two is solved by BiCGSTAB for complete data. Figure 3.19 compares the convergence rates of the LSCG and BiCGSTAB methods for this problem. While the residuals decrease to 10% in 10 iterations in the LSCG method, they reach 10% in 4 iterations and as low as 2% in 10 iterations with the BiCGSTAB method. Four iterations in BiCGSTAB are roughly enough to achieve a high resolution image of the LSPSM. However, 60% fewer iterations with the BiCGSTAB method do not make this method 60% faster than the LSCG method. Each iteration in the LSCG method requires implementation of one modeling run and one migration run. BiCGSTAB, on other hand,

performs two modeling runs and two migration runs during each iteration which makes one BiCGSTAB iteration two times more expensive than one LSCG iteration.

Fortunately, in the BiCGSTAB's algorithm, each Kirchhoff modeling is instantly followed by a Kirchhoff migration on the modeled data. With some effort, it is possible to combine the Kirchhoff modeling and migration algorithms into a single combined algorithm which can act as the multiplication of $\mathbf{G}^T \mathbf{G}$ with the reflectivity model. This combined algorithm is significantly faster than applying two algorithms separately since it requires only one time calculation of some modeling/migration parameters such as migration times and weights. Considering the combined modeling-migration algorithm to be 50% faster than the separate algorithms, the BiCGSTAB method is roughly 40% faster than LSCG method to give the same resolution in the LSPSM image. Another important advantage of using the BiCGSTAB method is that it requires less memory than the LSCG method. The LSCG method loads two data size, $M_{trc} \times N_t$, variables and three model size, $N_x \times N_z$, variables simultaneously, into the computer memory. The BiCGSTAB method, on the other hand, holds one variable in data size and seven model sized variables in a computer memory. In a seismic experiment usually $N_x \ll M_{trc}$ and $N_z \approx N_t$, therefore, the LSCG method occupies almost two times more computer memory than the BiCGSTAB method. This is a significant improvement when working with a 3D or a large 2D data set.

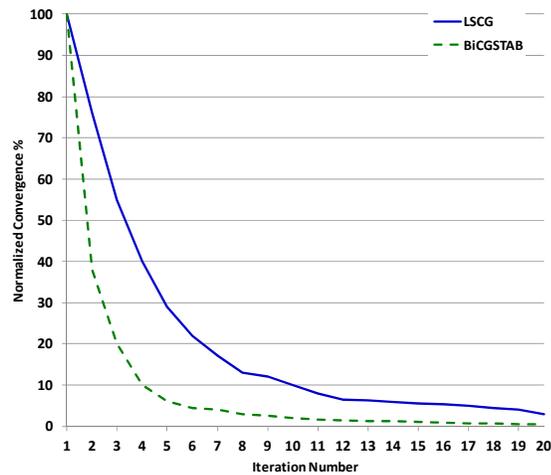


Figure 3.19 Convergence rate of LSPSM in 20 iterations in the LSCG (solid line) and BiCGSTAB (dashed line) methods.

3.5.5 Multilevel CG methods versus CG methods

LSCG and BiCGSTAB are very efficient methods to solve a linear LSPSM equation. After performing a few iterations with these methods, a high resolution image of the Earth's subsurface reflectivity is achieved. Multigrid is a general method that solves an equation in different grid sizes using an iterative solver. In order to be effective, multigrid requires the iterative method to be a smoother when used as the solver. The smoother must be able to find the high frequency components of the solution and leave the low frequency components in the residuals after a few iterations. When multigrid is not effective using its standard iterative smoothers for a LSPSM equation, other iterative methods such as CG-like methods may be considered.

However, the CG methods do not have a smoothing property which is the essential criterion for a multigrid method solver. In fact, the steepest descent and all CG-like methods are roughers and not smoothers (Shewchuk, 1994, Douglas, et al., 2003). Rougher solvers are able to recover the low frequency content of the solution slightly faster than the high frequency content. The residuals after the first few iterations consist of mostly high frequency components, a high oscillatory signal. The independence or slight roughness property of the convergence in CG methods to the frequency content of the solution is shown in the following examples.

Consider solving equation 3.69 where \mathbf{A} is the second difference matrix (equation 3.64) and \mathbf{b} is a vector of random numbers where $b_i \in [-1, 1]$, and $n = 64$. Assume $\mathbf{x}_0_j = \sin\left(\frac{jk\pi}{n}\right)$, $0 \leq j \leq n$, $1 \leq k \leq n - 1$, the Fourier modes with frequency k , as the initial value for the CG iterations. Since all its eigenvalues are positive (equation 3.66), matrix \mathbf{A} is positive definite (see Appendix E for another proof). Therefore, the CG methods are able to solve equation 3.69 in at most n iterations. Figure 3.20a shows the convergence in 10 iterations for initial values with different frequency content, $k = 1, 4, 8, 16, 32$. There is no relationship between the rate of convergence and the frequency content of the initial value. Starting with any initial value, CG converges to the exact solution after at maximum n iterations. Figure 3.20b compares the solutions for $k = 1$ and $k = 32$ after n iterations to the true \mathbf{x} . Both solutions exactly match with the true \mathbf{x} . It seems that there is no advantage in using any initial guess other than a vector with zero elements for this equation.

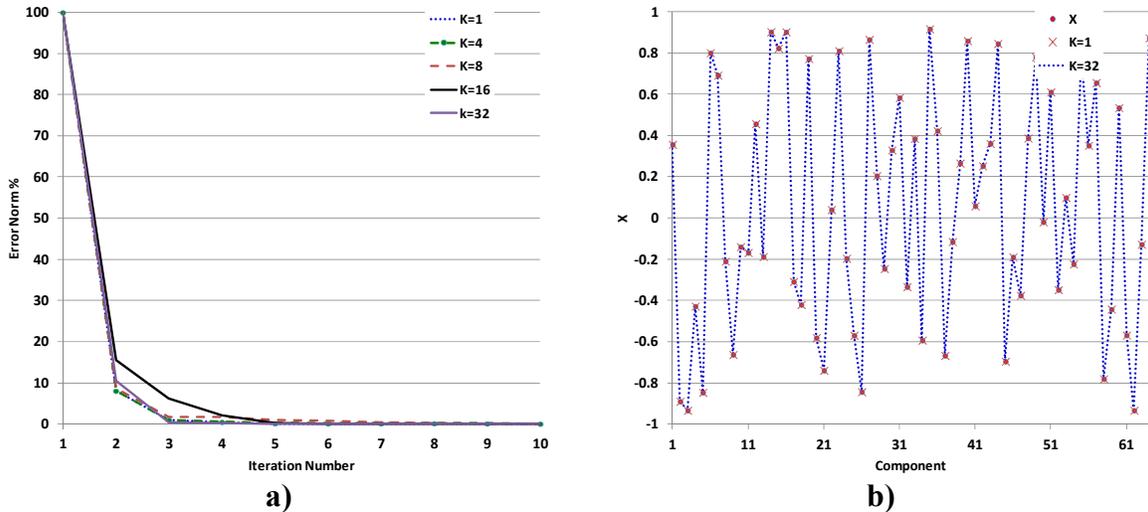


Figure 3.20 a) The Euclidean norm error versus the number of iterations is plotted for the CG iterations with different initial values to solve the linear equation 3.70 with b a random vector. b) Solutions after n iterations.

This property is investigated for solving a LSPSM equation. Figure 3.21 shows the convergence rate of LSCG for the synthetic model of Figure 3.4, with wavelets having different dominant frequencies. There is no improved convergence when the wavelet (and consequently the data) has higher frequency content. It seems that the convergence of the CG method for any linear problem such as LSPSM does not depend on the frequency content of the data. A similar conclusion can be reached by solving LSPSM with smoothing in the offset direction as the regularization term.

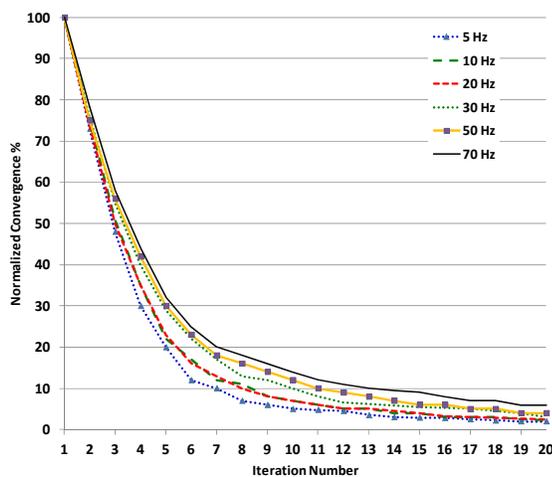


Figure 3.21 The convergence rate of the CG method for the LSPSM example in Chapter Two for data with different dominant frequency wavelets.

The roughness property of the CG is slightly visible when solving the LSPSM on the complex velocity model of the Marmousi data set. It is necessary to mention that the Kirchhoff time migration is not a proper method of migration for this geologically complex model. The Marmousi data set requires a depth migration method. However, subsurface imaging of the Marmousi data set is not the main goal of this study. Kirchhoff LSPSM on the Marmousi data set is used to investigate the roughness property of the CG method. Figure 3.22 shows the LSCG convergence rate of the Marmousi data set filtered by different band-pass filters. The convergence is not faster when using high frequency data. It seems that there is a slightly faster convergence for the very low frequency (less than 15 Hz) band of data which may be related to the roughness property of the CG method.

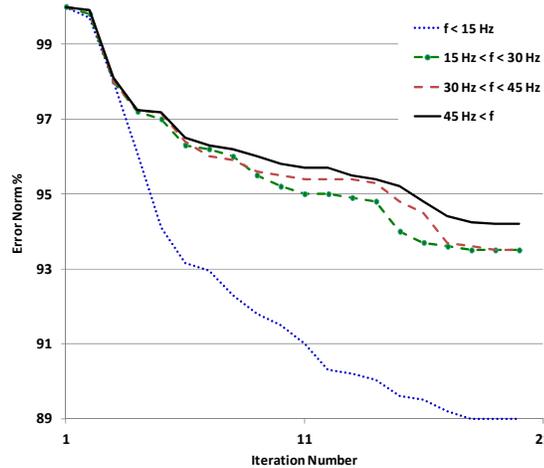


Figure 3.22 The convergence rate of the LSCG method used to solve LSPSM for the filtered versions of the Marmousi data set. Data are band-limited with a low pass ($< 15\text{Hz}$), two band pass ($15\text{Hz} < f < 30\text{Hz}$, and $30\text{Hz} < f < 45\text{Hz}$), and a high pass ($45\text{Hz} < f$) filters.

Since the CG method is a rougher, a multilevel CG method may not be effective in speeding up the LSPSM process. However, with the purpose of increasing image resolution, different approaches of applying multilevel CG to the LSPSM equation are investigated. There are at least three main approaches to applying a multilevel method to an LSPSM equation in the time and space domains. Restriction and interpolation of an LSPSM problem in the time and space domains can be applied in each, horizontal or distance, vertical or time, or both directions of the model in order to transfer a problem to a higher or lower grid size. As seen in the previous

section, since there is no change in the convergence rate for the lower frequency components of the data, applying multilevel LSCG in the vertical (time) direction does not improve the performance of the LSPSM.

The analysis of using multilevel LSCG with restriction and interpolation in the (horizontal) distance direction is investigated by comparison between multilevel LSCG and LSCG methods. Restriction to a coarser grid is performed by decimating half of traces (keeping one trace and deleting the next) from the migration image in order to move to a coarser grid. Our study showed that there will not be any faster convergence or increment in image resolution by using multilevel LSCG with interpolation and restriction in the horizontal direction. Figure 3.23 compares an image from multilevel LSCG with an image from LSCG. The improvement in the resolution of the multilevel LSCG image to the LSCG image is negligible.

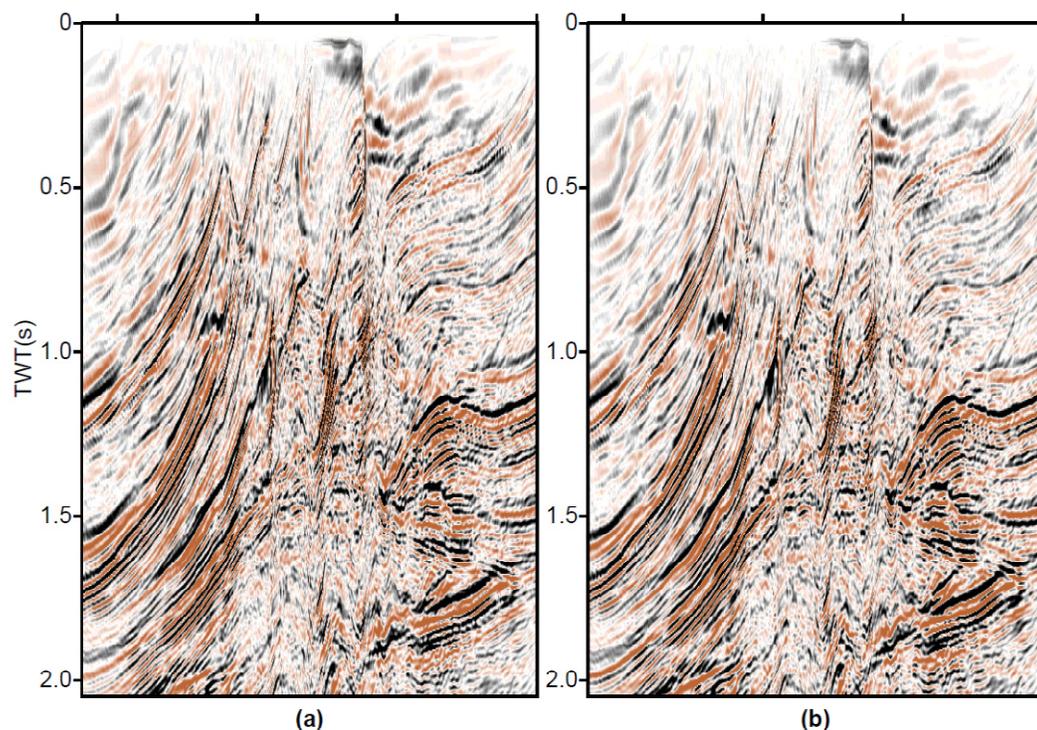


Figure 3.23 Comparison between a) multilevel LSCG and b) LSCG to solve regularized LSPSM of the Marmousi data set with 5 iterations on each grid.

Since CG is a rougher, a new method of multigrid, reverse- v (Λ) cycle, may be considered. In this method, a problem may be solved in the main grid size and then, results which include mostly high frequency content are interpolated to a finer grid where high frequency content now

act as low frequency content. In this new grid size, the problem is solved and a restricted solution is added to the solution in the main grid size to be used as the new initial value in the main problem.

However, this method is not effective in solving the LSPSM equation for two reasons: firstly, solving LSPSM in a grid finer than the main problem is very costly; secondly, as my example showed, the roughness property of CG is not as robust as the smoothness property of the Jacobi method. It seems that the convergence of the CG method is a little faster with an initial value with very low frequency content in the initial model. For moderate or higher frequencies, there is no clear relationship between the rate of convergence and the frequency content of the problem, at least for the aforementioned problems. I implement another alternative to take advantage of the rougher property of the CG methods by dividing the data to different levels of the temporal and spatial frequencies in a 2D frequency domain.

3.5.6 Multilevel LSCG with spatial down-sampling of data

The possibility of transforming seismic data from the time-space domain into the frequency-wavenumber domain and vice versa provides another method of splitting seismic data into the smaller grid sizes and applying multilevel LSCG for solving LSPSM equations. Instead of only splitting seismic data into the bands of temporal frequencies, Margrave et al. s' (2006) method spatially resamples seismic data into different frequency and wavenumber bands with larger trace spacing in the lower temporal frequency bands. They claimed that this method increases not only the stability of their forward operator and conjugate inverse (FOCI) method, but also increases the total computational speed by decreasing the number of traces in the low frequency chunks. The second advantage makes this method of down-sampling attractive in a multilevel LSCG method to solve the LSPSM equation.

Assume seismic data with Δx as the spatial sampling rate in the transverse coordinate to be band-limited, $\omega \in [\omega_{min}, \omega_{max}]$. Data can be divided into n , $n \geq 1$, chunks of separate temporal frequency bands as (Margrave et al., 2006):

$$[\omega_{min}, \omega_{max}] = [\omega_{min}, \omega_1) \cup [\omega_1, \omega_2) \cup \dots \cup [\omega_{n-1}, \omega_{max}]. \quad \mathbf{3.93}$$

Margrave et al. (2006) spatially resampled the trace spacing for the i^{th} chunk, $[\omega_{i-1}, \omega_i)$, from Δx to $\Delta x_i > \Delta x$ such that

$$\alpha k_{N_i} \leq k_{crit} \leq \beta k_{N_i}, \alpha < \beta \in [0,1], \quad 3.94$$

where $k_{N_i} = \pi/\Delta x_i$ is the Nyquist wavenumber and $k_{crit} = \omega/v_{crit}$, and v_{crit} is the velocity that determines the highest evanescent boundary. Margrave et al. (2006) proposed that the lowest velocity is a good choice for v_{crit} .

In order to avoid spatial aliasing due to incrementing the spatial sample size, an antialiasing filter should be applied to data to reject energy in the evanescent region while conserving energy in the wavelike regions. This can be implemented by defining a new Nyquist wavenumber for each chunk and zeroing data with greater wavenumbers. With n spatial location before resampling, and m_i wavenumbers after resampling, equation

$$\Delta x_i = \frac{n}{m_i} \Delta x, \quad 3.95$$

provides the spatial sample rate for j^{th} chunk.

By splitting seismic data into chunks of data with different temporal and spatial frequency bands, and using the roughness property of the CG methods, I found a relatively effective method for solving LSPSM. In this method, seismic data are separated into chunks of different frequency and wavenumber bands. LSPSM is performed on the chunk with the lowest frequency content (highest chunk number in this dissertation) and the result is being used as the initial value for the LSCG in the chunk with the next higher frequency content. The procedure continues to get to the first chunk. LSCG on the data starts with the result of the LSPSM from a high frequency chunk as the initial model. This method is not a type of multigrid because it does not benefit from the smoothing property of a solver. However, our experiments show better recovery of low frequency content of the solution using this method.

The method is shown with the synthetic example in Chapter Two. With this method, the complete synthetic seismic data is spilt into nine frequency bands. Table 3-7 shows the specification of each chunk for shot #5. As seen in this table, chunks with lower frequency content have a lesser number of traces. Therefore, LSCG iterations are faster for higher chunk numbers. Figure 3.24 shows the chunking on the FK spectrum on shotgather number 5.

I found the method to be effective when each chunk includes the frequency content of the next chunk (lower frequency chunk) and the energy of all traces in each chunk is normalized to the same level as the energy of the original seismic data before chunking. Therefore, I added the

frequency content of the higher chunk to the current chunk when transferring data from frequency-wavenumber domain to the time-space domain. Each resulting chunk of seismic data contains the frequency from 1.22 *Hz* to the upper range in that chunk in Table 3-7. The FK spectrum and seismic data in chunks 1, 5 and 9 of shot # 5 is shown in Figure 3.25.

Table 3-7 Frequency range, trace spacing, and the number of traces for each frequency band for the shot number 5.

Chunk number	Frequency range (Hz)		Trace spacing (m)	Number of traces in this shotgather
	Lower	Upper		
1	38.3	100	15.6	96
2	24.9	38.1	22.4	67
3	16.1	24.7	33.3	45
4	10.5	15.9	51.7	29
5	6.84	10.3	71.4	21
6	4.39	6.59	100	15
7	2.93	4.15	136	11
8	1.71	2.70	167	9
9	1.22	1.46	214	7

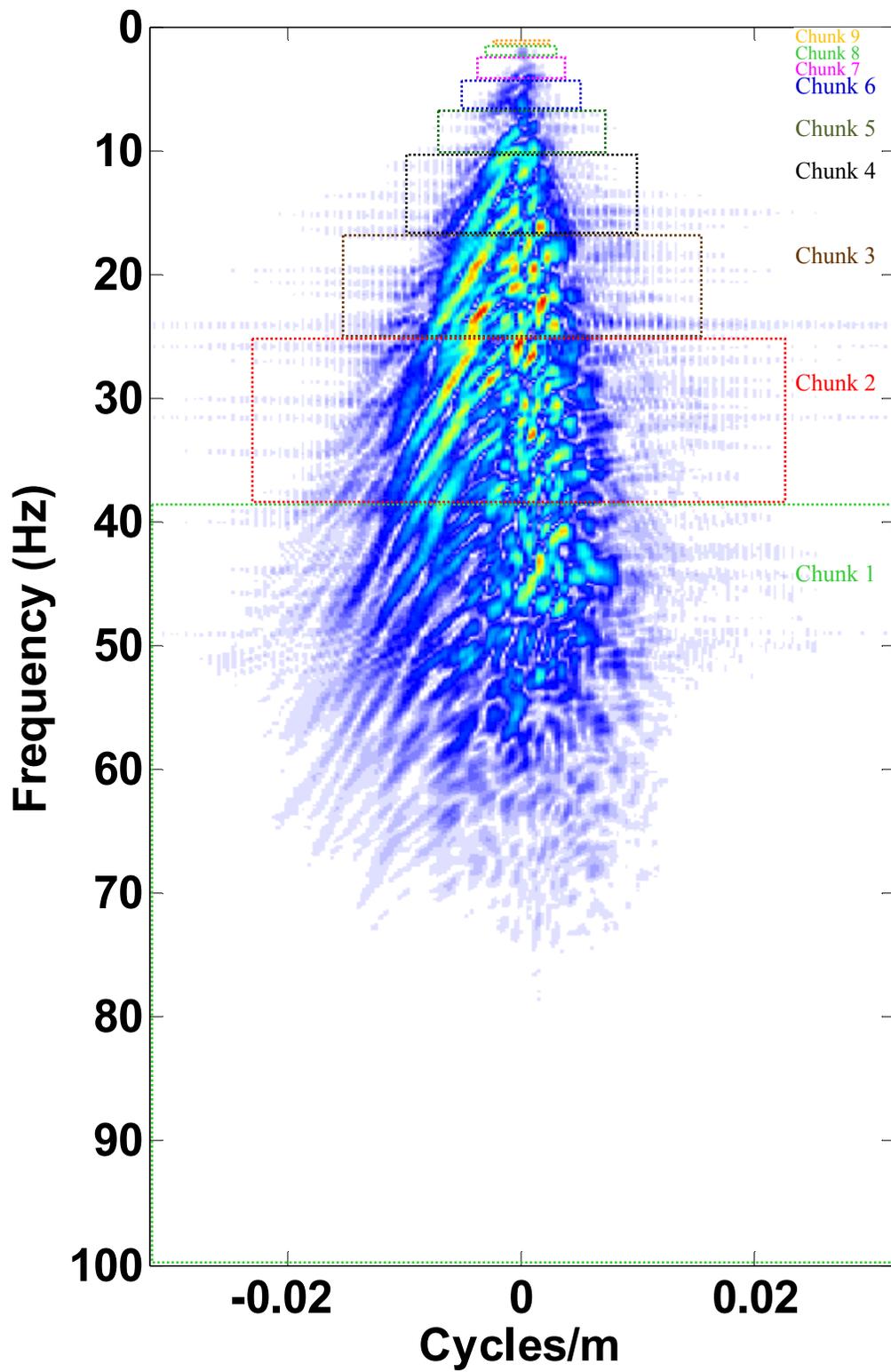
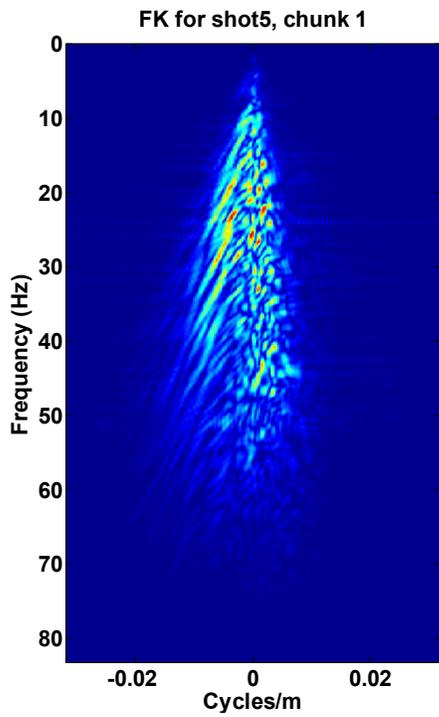
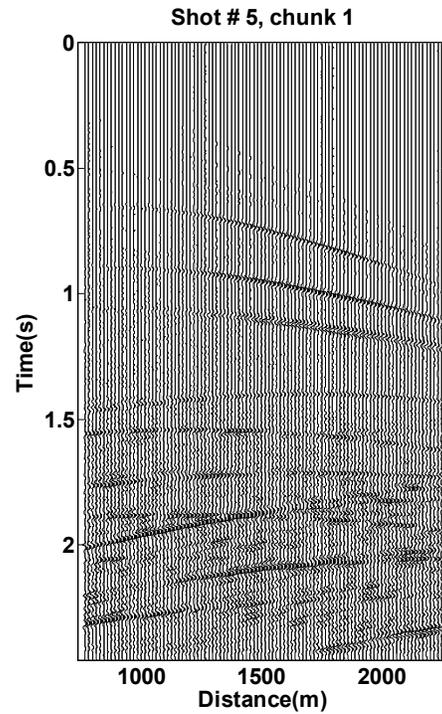


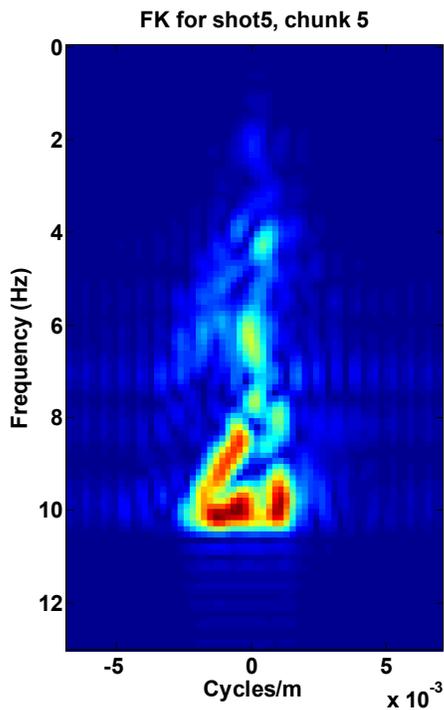
Figure 3.24 Frequency spectrum of shot number 5 with 9 chunking areas.



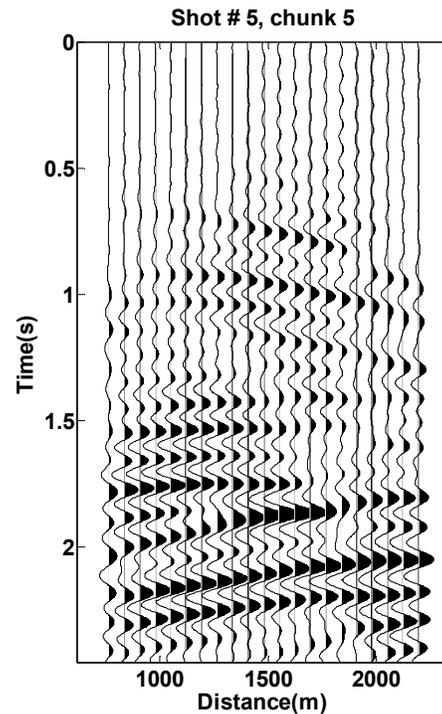
a)



b)



c)



d)

Figure 3.25 Frequency spectrum (a, c, and e) and corresponding data (b, d, f) for shot number 5 in chunks 1 (a,b), 5 (c,d), and 9 (e,f). Continued at the next page.

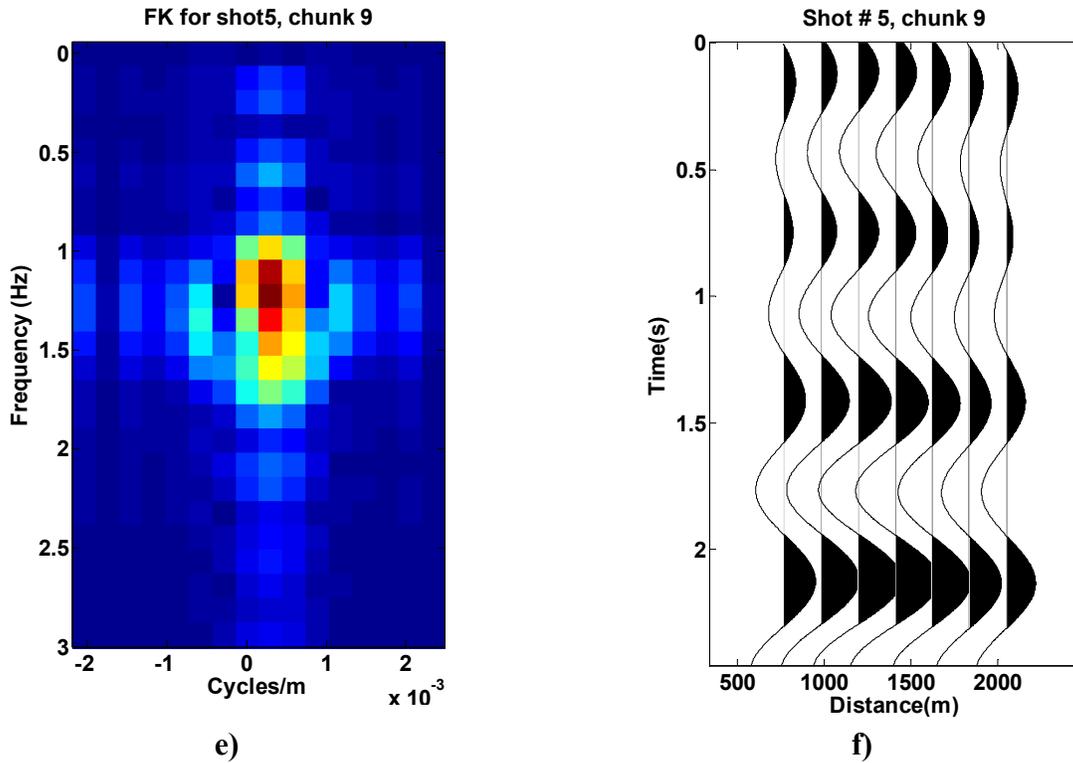


Figure 3.25. Continued.

LSCG is performed on chunk 9, the very low frequency-wavenumber data, with only three iterations. The resulting image is used as the initial model for three LSCG iterations on data in chunk 8 (which include lower frequencies from chunk 9). Then the current LSPSM image used for the next chunk and procedure continues until chunk 1 includes all lower frequency chunks. Finally, the method performs two LSCG iterations on the original seismic data with the image from LSPSM of chunk 1 as the starting point. This procedure provides a higher resolution of the final image in comparison to LSPSM of data with zero as the initial model with the same number of LSCG iterations.

As an example, Figure 3.26 compares the LSPSM of data in chunk 2 when, a) the initial guess for the LSCG iterations is a zero vector, with b) the result when the initial model resulting from the LSPSM of higher chunk numbers. This comparison shows better recovery of the reflectivity when LSCG uses the previous chunk as the starting model. Figure 3.27 compares the convergence rate in the LSCG iterations with zero models as the starting point and models from lower frequency chunks as the starting point. Convergence is faster when LSCG's starting point is not a zero vector.

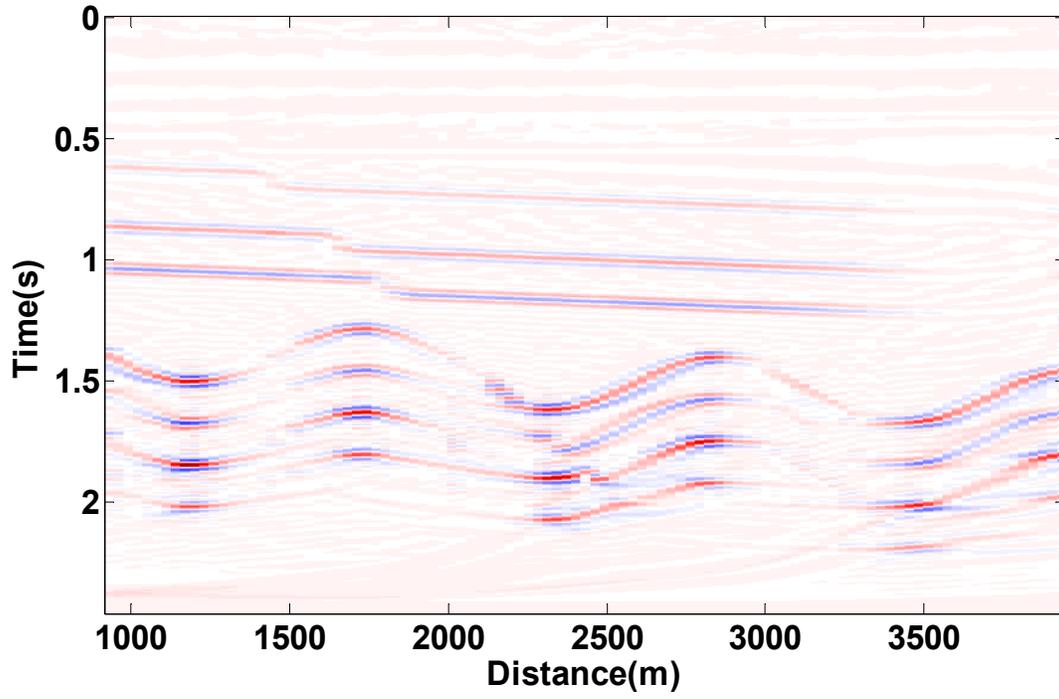
Figure 3.28a shows the resulting image on data in chunk 9 after three LSCG iterations with zero as the starting model. Figure 3.28b to Figure 3.28i show the LSPSM of each chunk after 3 iterations in the LSCG with the LSPSM from the previous chunk as the initial model. Finally, Figure 3.28j is the result of two LSCG iterations on data with the result from chunk 1 as the starting model. Figure 3.29 shows the LSPSM after eight LSCG iterations with zero as the initial model. Alternatively, Figure 3.28j is obtained achieved after only two LSCG iterations and shows a slightly higher resolution image than Figure 3.29 which is resulted from eight LSCG iterations.

It is important to mention that the total cost for acquiring Figure 3.28j includes the time spent for LSCG on the chunks. Low frequency chunks have fewer traces, and LSCG iterations are faster on small data. The average time spent on one LSCG iteration in all chunks is approximately twice the time spent on the original data. Therefore, the total cost for the multilevel LSCG is eight times the cost of one iteration on data of the same size. It means that the total cost of the multilevel LSCG is approximately equal to the cost of LSCG with zero as the starting model. Multilevel LSCG's cost can be lowered by reducing the number of chunks without altering the ability of the method in achieving a higher resolution image at the end. Images resulting from multilevel LSCG have a better resolution due to having more low frequency content from low frequency chunks. However, multilevel LSCG requires more computer memory for one to two additional data-size variables.

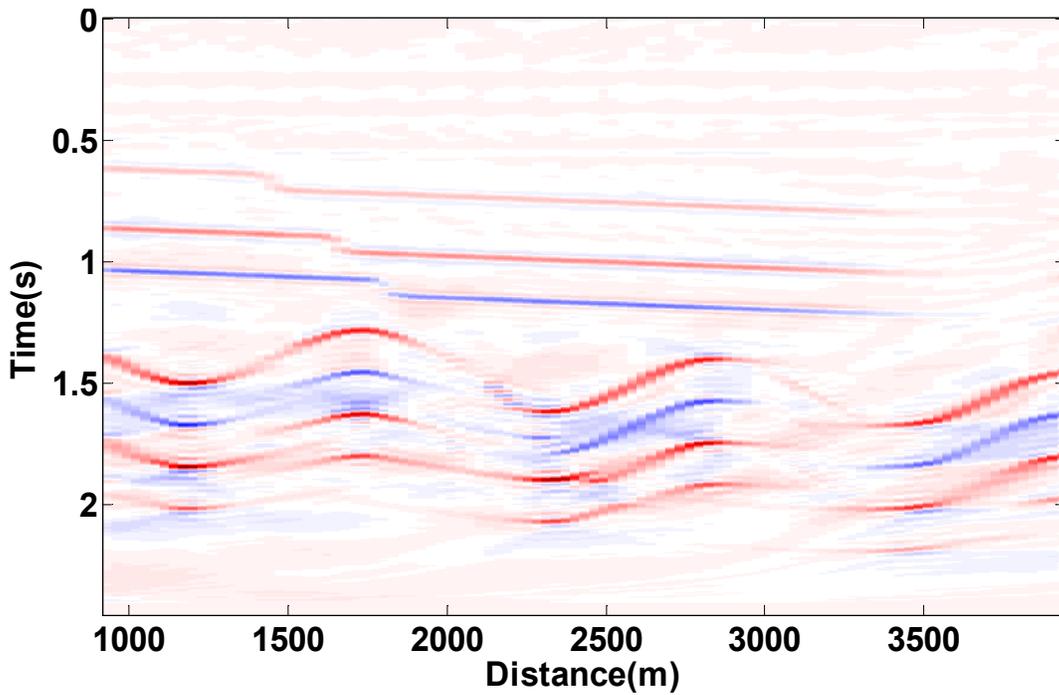
The ability to attenuate acquisition footprint resulting from incompleteness or irregularities in seismic data sampling is the main advantage of using LSPSM instead of migration. Since in the multilevel LSCG division of data to the desired chunks is performed in the frequency domain, the method works better for regularly and densely sampled traces. Data irregularities or coarse sampling introduce artifacts into chunked data and prevents the method from being effective.

To investigate the effectiveness of the method with coarsely sampled data, synthetic data used in this section is decimated by removing 5/6 of the data regularly and an LSPSM on the rest of data is run with multilevel LSCG. Figure 3.30 shows the resulting image on chunk1, and on data when lower frequency chunks are being used as the initial model in three LSCG iterations. The resulting images are noisier than the images resulting after the same number of iterations of LSCG with a zero model as the starting point as shown in Figure 3.31. Therefore,

multilevel LSCG may not be effective when data sampling is very coarse and irregular. In such cases LSCG or BiCGSTAB methods with a zero model as the starting point are more efficient.

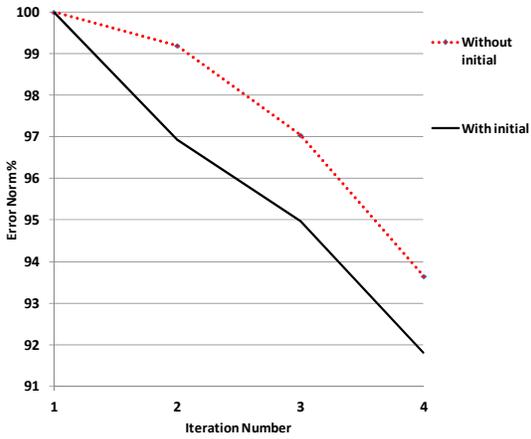


a)

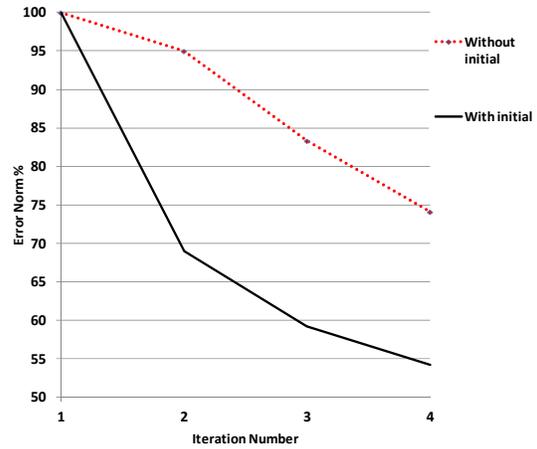


b)

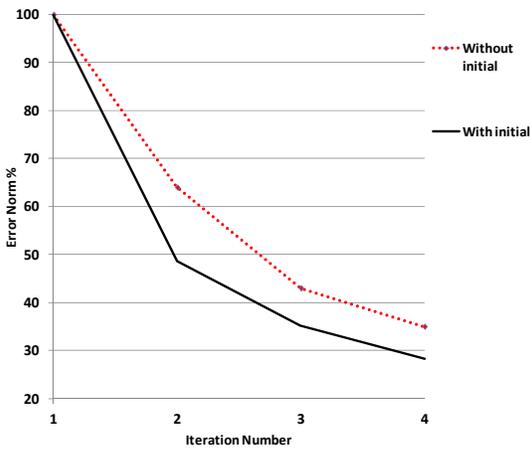
Figure 3.26 Comparison between an LSPSM image with a) a zero initial model, and b) an image resulting from a lower frequency chunk as the initial model, after same number of LSCG iterations.



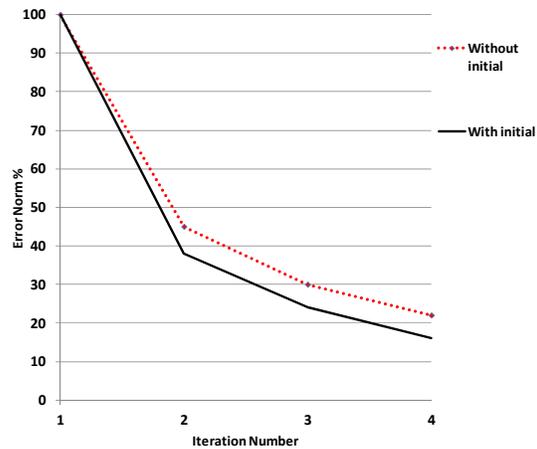
a)



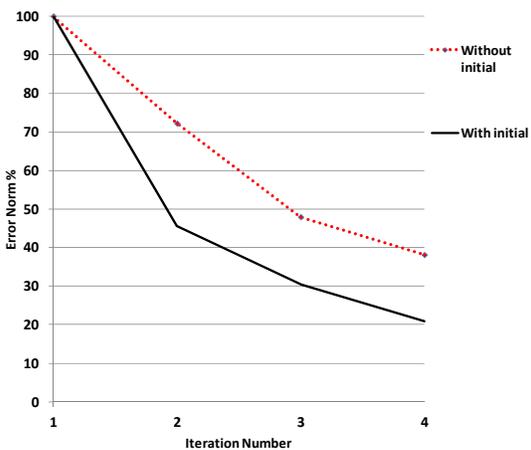
b)



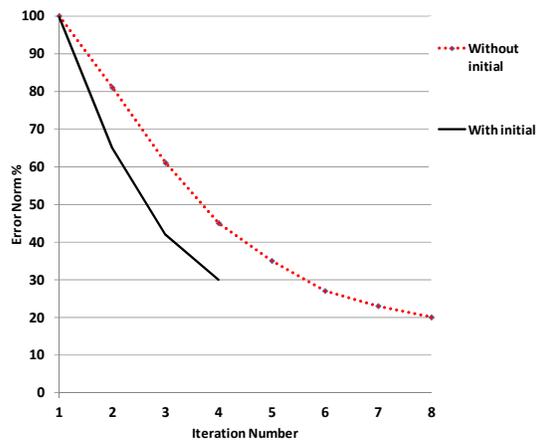
c)



d)

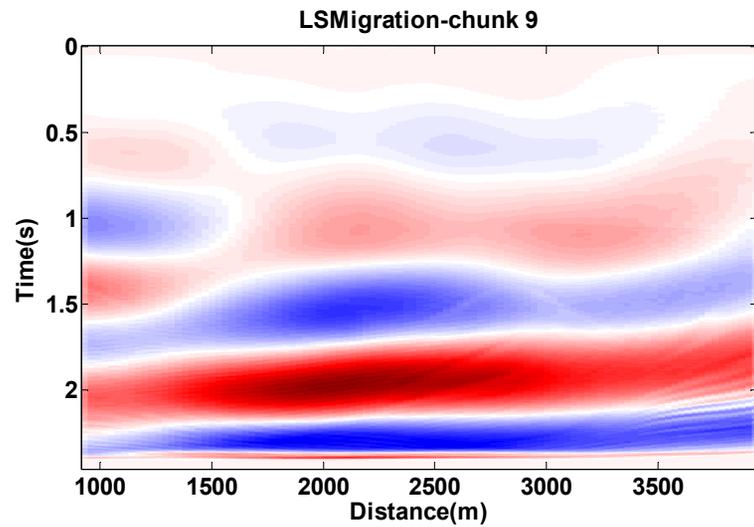


e)

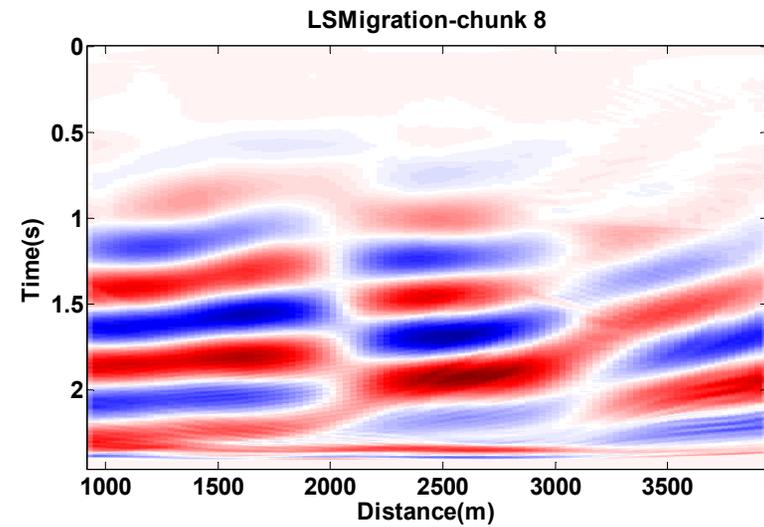


f)

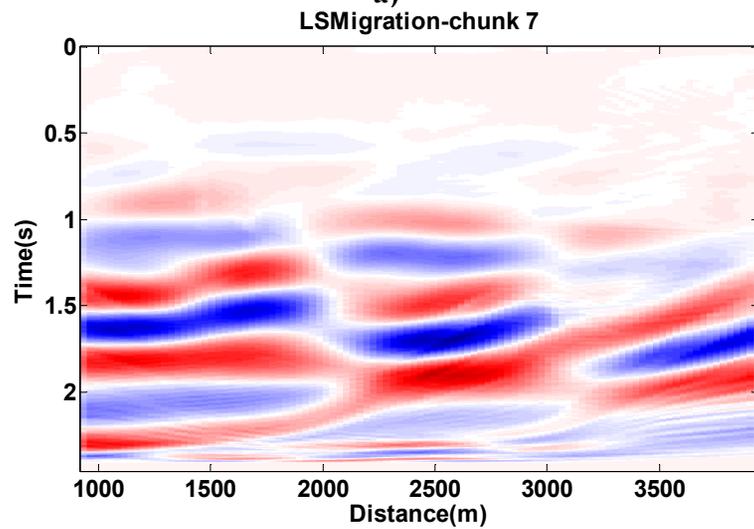
Figure 3.27 Comparison of convergence rates for the LSCG method when f) the starting model is zero, with the result from a higher chunk as the initial value for chunks a) 8, b) 6, c) 4, d) 2, and e) 1.



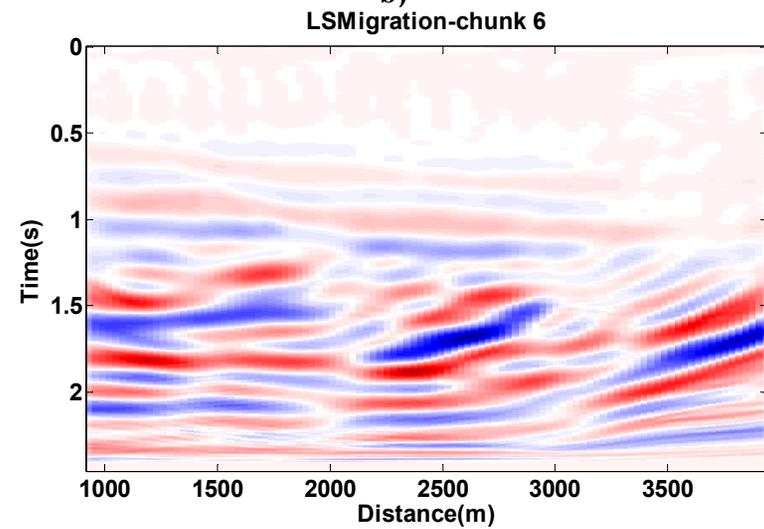
a)



b)

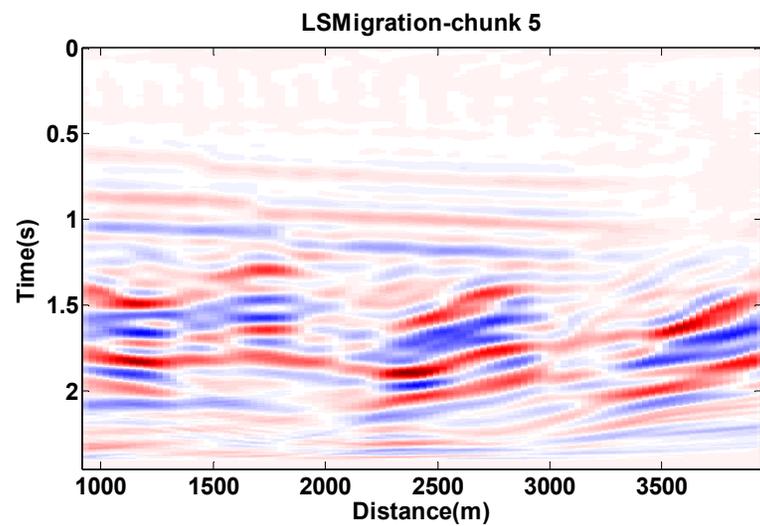


c)

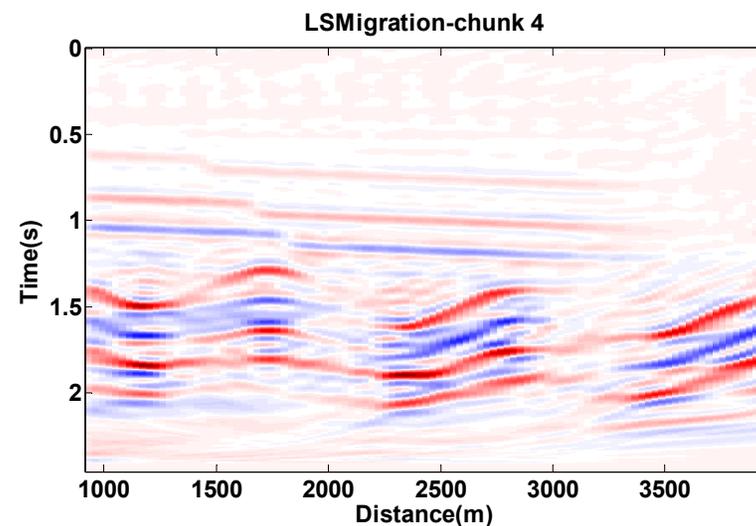


d)

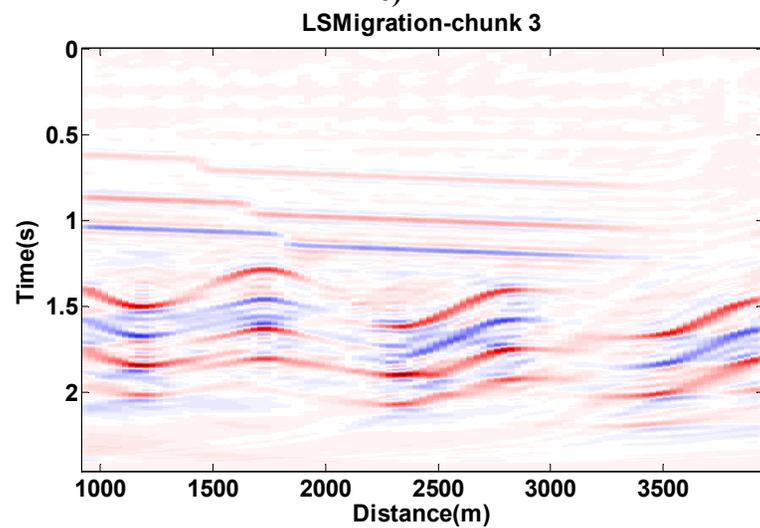
Figure 3.28 LSPSM images after three LSCG iterations, with the image resulting from a lower frequency chunk as the initial model. a), b), c), d), e), f), g), h), and i) correspond to chunk 9, 8, 7, 6, 5, 4, 3, 2, and 1, respectively.



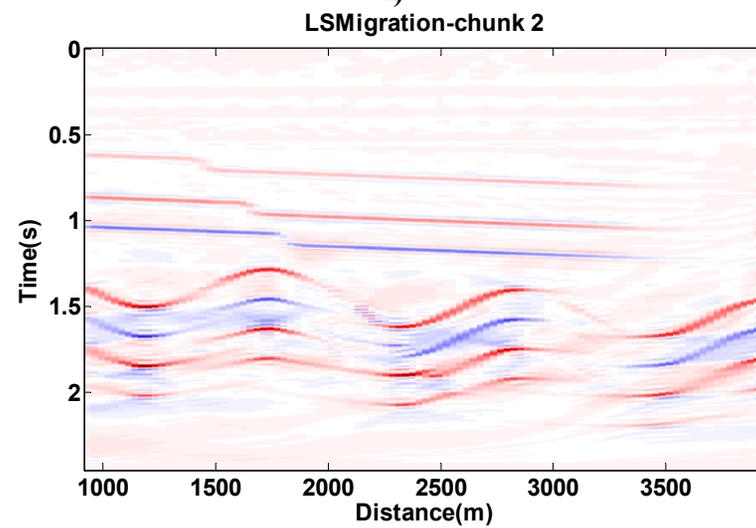
e)



f)

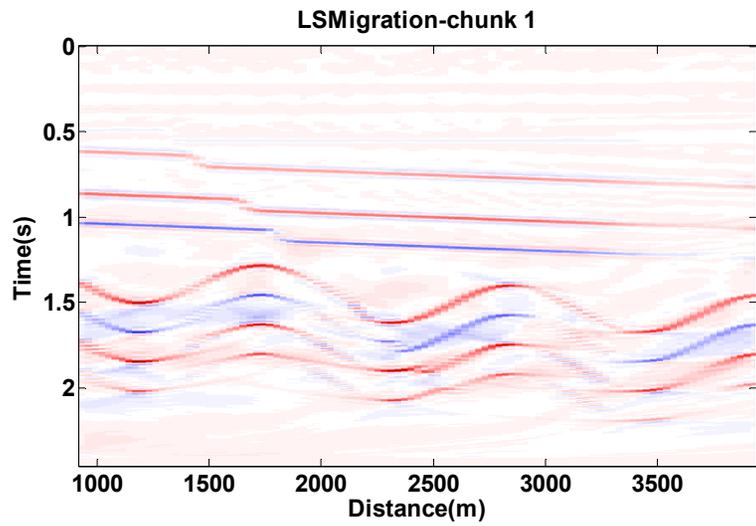


g)

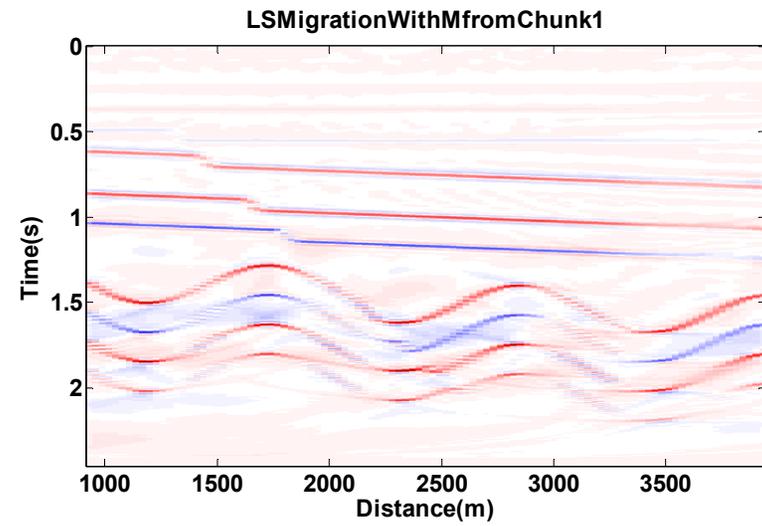


h)

Figure 3.28. Continued.



i)



j)

Figure 3.28. Continued.

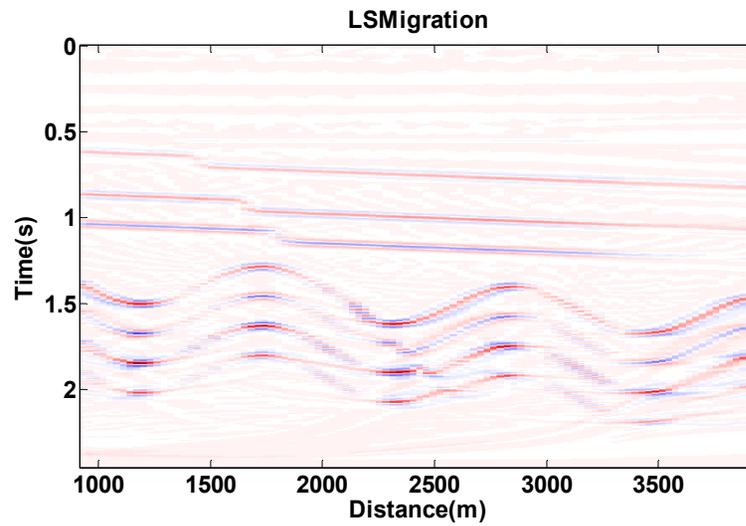


Figure 3.29 LSPSM images after eight LSCG iterations with zero as the initial model.

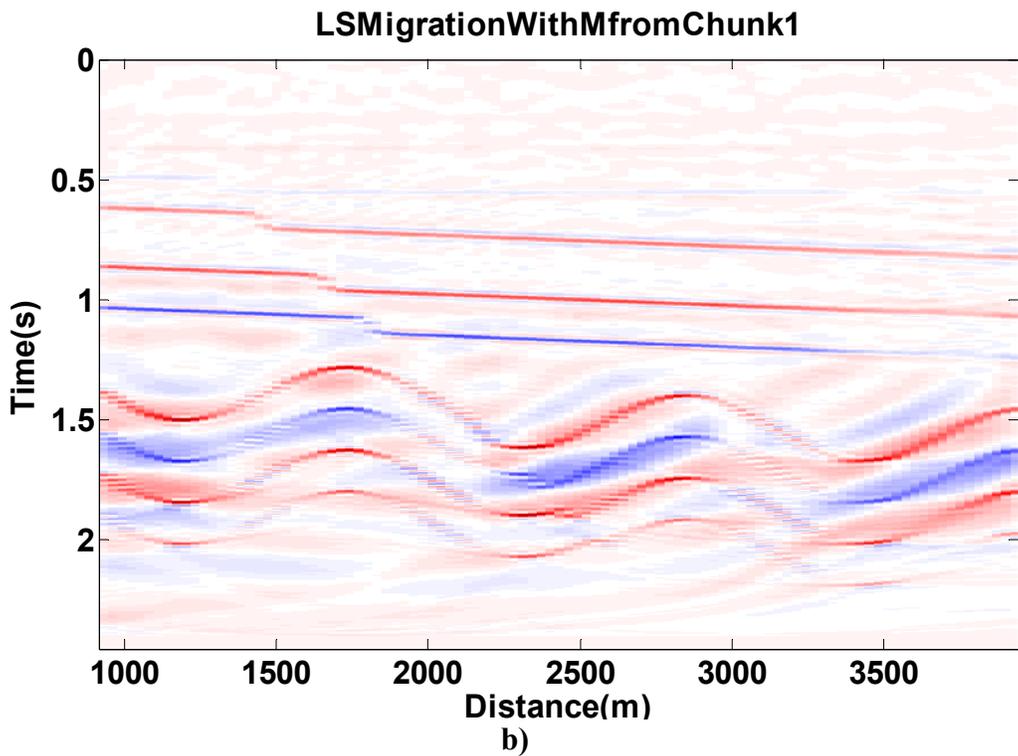
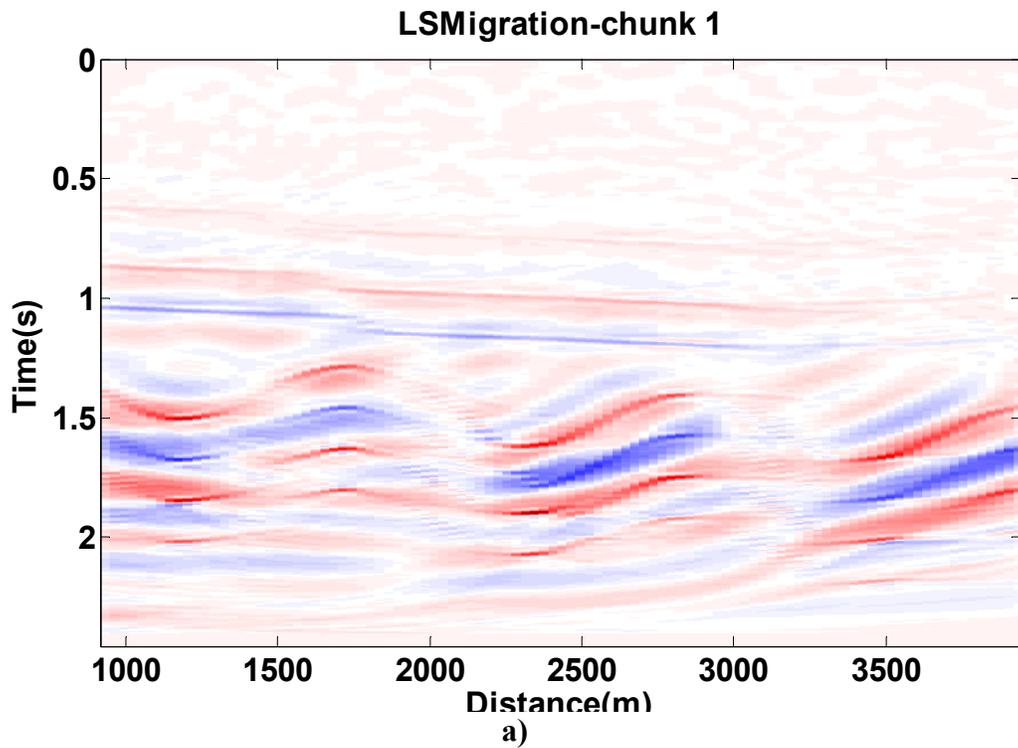


Figure 3.30 LSPSM images after three LSCG iterations on 84% regularly decimated data, a) in chunk 1 with the image resulting from a lower frequency chunk as the initial model, b) on data with the image resulting from chunk 1 as the initial model.

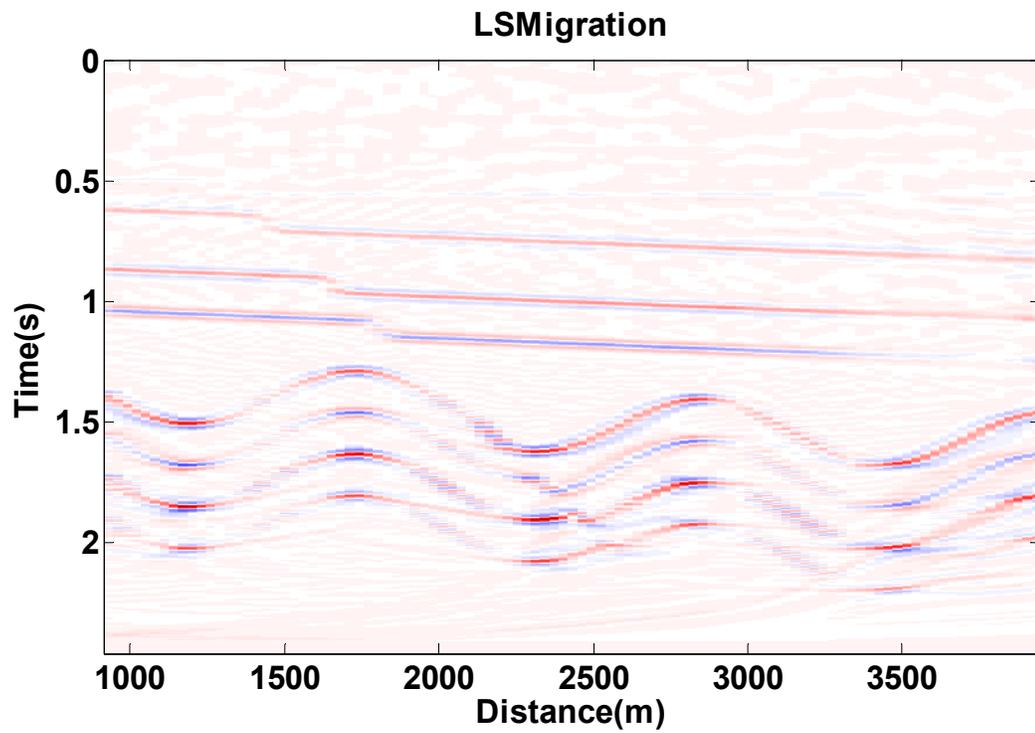


Figure 3.31 LSPSM images after three LSCG iterations on 84% decimated data with zero as the initial model.

3.6 Summary and Conclusions

Solving the LSPSM equation requires finding the solution to a large system of linear equations in the form of equation 3.1.

A technique for creating Kirchhoff modeling, migration, and inversion matrices using Kirchhoff migration algorithm is shown. Construction of the modeling and migration matrices goes to the inner most loop of the Kirchhoff modeling or migration algorithms. Once calculated, migration and modeling can be easily repeated by just multiplying the matrix to a data vector. However, the migration or modeling matrix is too large, which makes it impractical to be used on any real data sets. The method of construction of the matrix form from an algorithm can be used to create matrices whose multiplications can replace the execution of the corresponding algorithms.

Multigrid methods with their fast recovery of low frequency components of the solution are proven to be very successful in solving many PDEs. Numerical examples show that the LSPSM problem is not solvable by Jacobi or Gauss-Seidel iterations. Consequently, the standard multigrid method which uses Jacobi or Gauss-Seidel as iterative solvers is not applicable to the mentioned problem. The requirement of a large memory is another problem associated with this method.

CG-like methods are effective solvers for this equation. LSCG and BiCGSTAB methods have the advantage of using operators instead of matrices. BiCGSTAB provides a faster and smoother convergence rate than the LSCG.

The CG methods do not have a smoothing property. Therefore, using CG as the multigrid solver does not increase the convergence speed. Using multilevel LSCG combined with splitting seismic data in the frequency domain, slightly reduces the number of iterations for the same rate of convergence in comparison to the LSCG by introducing an initial low frequency value. It also provides a higher resolution LSPSM image for the same cost. For very irregularly and sparsely sampled data, LSCG and BiCGSTAB are appropriate choices.

4.1 Introduction

Industry moves to exploit smaller hydrocarbon resources in areas with complex geology to meet today's energy demand. Hence, more advanced imaging techniques are required to provide a higher resolution of subsurface images. LSPSM is one of these methods. Any advanced imaging method requires relatively accurate velocity information. Extracting proper velocity information is a main step for proper subsurface imaging. In addition to imaging purposes, velocity information may be used for time to depth conversion and noise filtering. It also provides a good knowledge of hydrocarbon reservoirs and helps geological interpretations since it strongly depends on lithology, porosity, fluid content, temperature, and the degree of compaction of the underground rocks.

LSPSM creates a high resolution image of the subsurface reflectivity. In order to recover the unknown reflectivity, \mathbf{m} , in LSPSM, Kirchhoff modeling and migration operators, \mathbf{G} and \mathbf{G}^T , must be defined as accurately as possible. These operators require a good estimation of the rms velocity to perform migration and modeling correctly.

In this chapter, the dependency of the LSPSM method on velocity is investigated. I used a synthetic data set to study the effect of using an inaccurate velocity model on an LSPSM image, and its usability for data reconstruction. I show that the convergence of the LSPSM method to the desired solution depends on the accuracy of the implemented velocity model. It is shown that LSPSM is more dependent on the accuracy of the velocity information than conventional Kirchhoff migration. This is a major problem when replacing the migration with the LSPSM. Without a reasonably accurate background velocity, LSPSM does not converge to the desired solution, and is not able to improve the final image resolution.

The high dependency of the LSPSM method on an accurate velocity model, and low sensitivity to incompletely or irregularly sampled data, makes this method effective for the velocity analysis of highly irregularly or incompletely sampled seismic data. In this chapter, I show how offset and shot domain common image gathers (CIG)s from LSPSM can improve the migration velocity analysis of very irregularly sampled data.

4.2 Migration velocity analyses

Proper subsurface velocity information leads to a well-focused migration image whereas an inaccurate velocity model distorts the migration image. However, image distortions have some useful velocity information that can be used for modifying or updating the velocity model.

The simplest method of velocity analysis is performed in the data domain. Data are sorted into common midpoint (CMP) gathers. Sorted CMP gathers are normal moveout (NMO) corrected several times using a plausible range of constant velocities. A correct velocity for an event results in flat events at that time. With a lower or higher implemented velocity, events will be over- or under-corrected, respectively. Flatness of the events can be used as a tool for finding the best velocity measurement at each time. The correct velocity at each time gives the maximum coherency at that time. Semblance is the most commonly used type of coherency measurement in an NMO corrected CMP gather, which can be calculated by,

$$S = \frac{1}{N} \frac{\sum_t (\sum_{i=1}^N a_{i,t(i)})^2}{\sum_t \sum_{i=1}^N a_{i,t(i)}^2}, \quad 4.1$$

where S is the semblance, $a_{i,t(i)}$ is the amplitude of the i^{th} trace in the gather at time $t(i)$, and N is the number of traces that contributes to the measurements. With this equation, semblance is the energy of the stack normalized by the energy of the components of the stack (Sheriff, 2006).

By calculating semblance at each time and for a range of constant velocities, a semblance panel is obtained. In this panel, the horizontal axis is the velocity, and the vertical axis represents the two way travelttime. By choosing the maximum semblance at each time, the best velocity for that time in the corresponding CMP position is achieved. Yilmaz (2008) mentions a complete list of other methods of coherency measurements which may be useful in velocity analysis.

NMO correction and semblance analysis assume having only horizontal reflectors. With dipping events, these methods give a higher velocity, referred to as the stacking velocities. Stacking velocity is related to the rms velocity by

$$V_s = \frac{V_{rms}}{\cos \beta}, \quad 4.2$$

where V_s and V_{rms} are stacking and rms velocities, respectively, and β is the real (geologic) dip of the subsurface reflector (Bancroft, 2007). With horizontal reflectors, rms velocity is equal to the stacking velocity. Consequently, with the rms velocity, NMO correction is not able to flatten

dipping events. Velocity analysis on the image domain instead of the data domain can be used with dipping layers to estimate the rms velocity.

Migration velocity analysis can be implemented by performing migration with a range of constant velocities to create many migration images. Then, searching for the best focusing of the image, a migration velocity function can be found. In a more efficient method, velocity analysis can be performed by measuring coherencies on each migration CIG after migrating data several times with a range of constant velocities.

Prestack Kirchhoff migration of 2D data produces a migrated section with two spatial coordinates, horizontal distance, x , and pseudo depth or time, τ . It is possible to divide the integration of Kirchhoff migration into a few subsets of different recording offsets (and azimuths in 3D). The result will be a cube (or hypercube, 4D) and each section is a “prestack partial image” as it is called after Biondi (2007). A CIG or Common Reflection Point (CRP) refers to a part of the image that corresponds to a specific subsurface location.

Offset domain CIGs are similar to CMP gathers after NMO correction. However, in CIGs, the focused energy is migrated and comes from the area below the CMP position. Hence, CIGs are more effective than the data domain CMP gathers for velocity estimation of the dipping layers. Velocity analysis based on the semblance method can be extended to the migration velocity analysis on the offset domain CIGs.

There are three common types of CIGs. The offset domain CIG is probably the easiest one to produce and to work with. In the offset domain CIG, the horizontal axis is the absolute source-receiver distance. In the angle domain CIG, the horizontal axis shows the incident angle parameter. In the shot domain CIG, the distance between image point and source is the horizontal axis. In all CIGs the vertical axis is the two way traveltime. Both, offset domain and shot domain CIGs can be easily created using prestack Kirchhoff migration. The focus in this thesis is on the offset domain CIGs and shot domain CIGs. I show the effect of inaccuracy in velocity information on the migration and LSPSM images and CIGs. Then, the relative advantages of using CIGs from LSPSM instead of CIGs from migration for velocity analysis are shown.

4.3 Effect of the velocity accuracy on the LSPSM

The high sensitivity of the LSPSM to the accuracy of the velocity model can be used as a tool for measuring the accuracy of the velocity (Yousefzadeh and Bancroft, 2012d). With a synthetic example I show how a minor inaccuracy in the implemented velocity model leads to a poor resolution of the LSPSM image and data reconstruction. Good data reconstruction requires an accurate velocity model. The convergence rate of the LSCG is another parameter to determine if the estimated velocity is accurate enough. I show that the best velocity model is the model that gives the least residuals in data reconstruction and optimal convergence in a few LSCG iterations.

I use a synthetic data set to evaluate the effect of the accuracy of the velocity model on the LSPSM method. I compare the resolution enhancement, the ability to reconstruct data, and the convergence rate, in the LSCG method for LSPSM by using an exact velocity model and a wrong velocity model. Consider the acquisition geometry shown in Figure 4.1, and the velocity model shown in Figure 4.2. The geometry assumes having 16 sources and 200 receivers per source. The source interval is 250 m and the receiver interval is 5 m. There are 201 image points with a 15 m interval on a 3 km 2D line. The velocity model includes both horizontal and dipping layers, and a fault. The interval velocity varies between 1600 *m/s* and 4160 *m/s*. After producing synthetic data for the model, one percent random noise is added to the data. The data are migrated and LSPSM-ed using the exact velocity model. I used the LSCG method with 10 iterations to solve the inversion equation. I then compare the result of using the exact velocity model with the result of using the wrong velocity model.

4.3.1 Effect on the resolution enhancement

The result of migration and LSPSM using the exact velocity model are shown in Figure 4.3 and Figure 4.4, respectively. All data are used in these processes and both, migration and LSPSM images, are a good depiction of the reflectivity model. LSPSM has higher resolution in the shallow part, and the fault is sharper. The result coincides with our expectation from implementing an LSPSM.

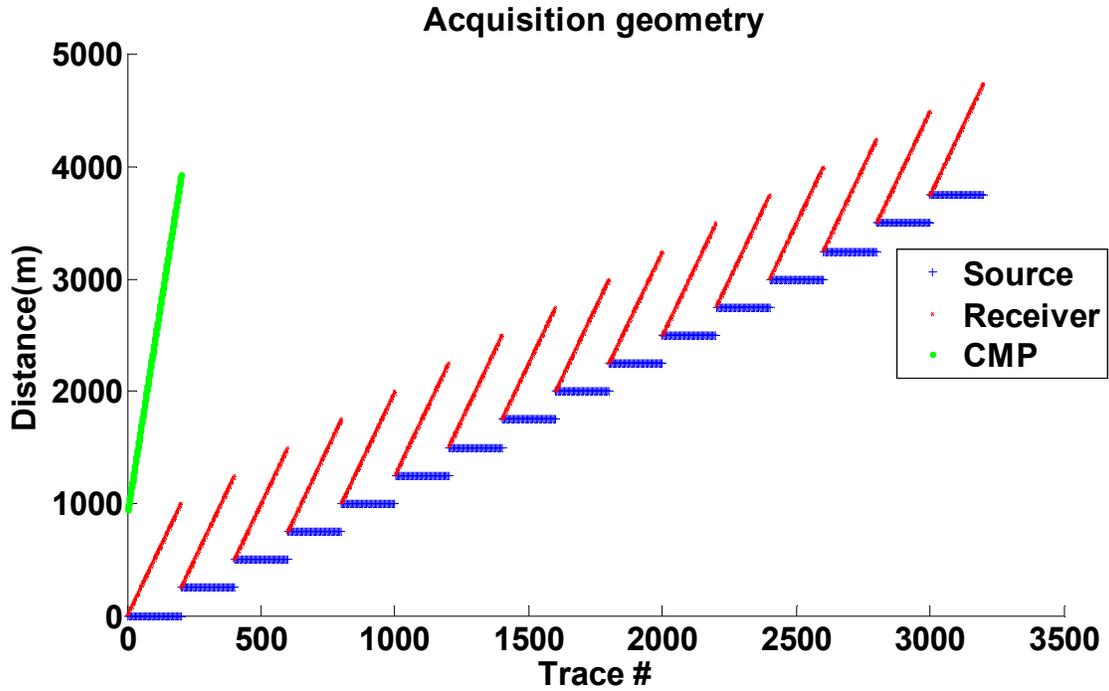


Figure 4.1 Geometry of seismic data modeling, includes 16 sources and 200 receivers per source. Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.

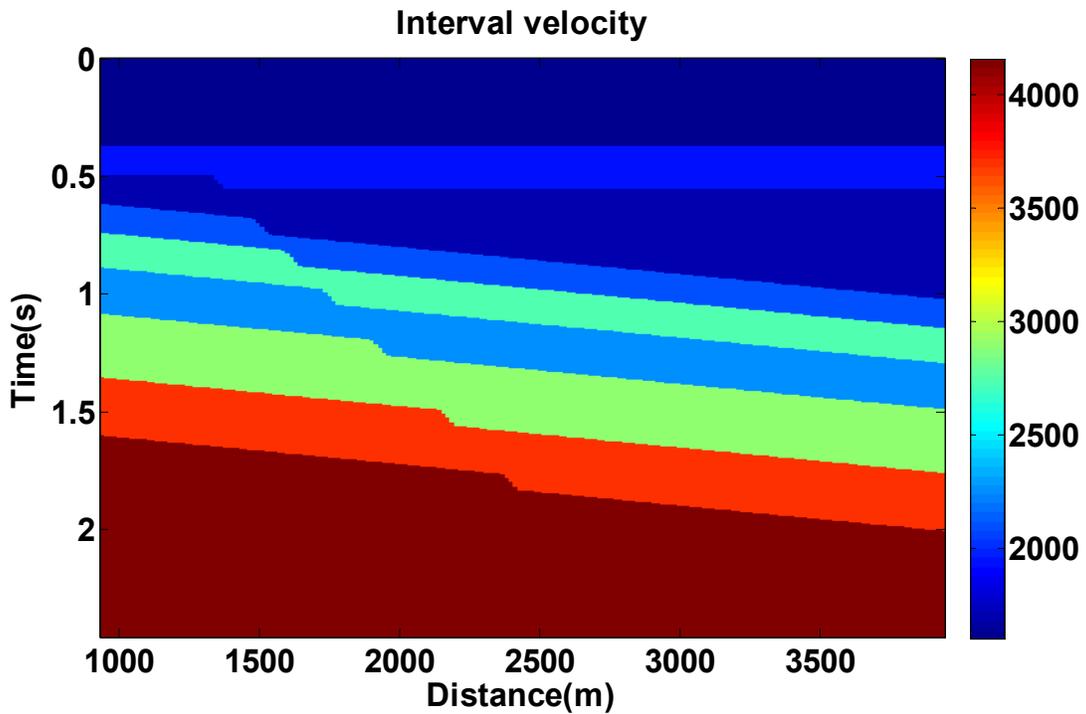


Figure 4.2 Interval velocity model used for studying the effect of velocity errors on LSPSM.

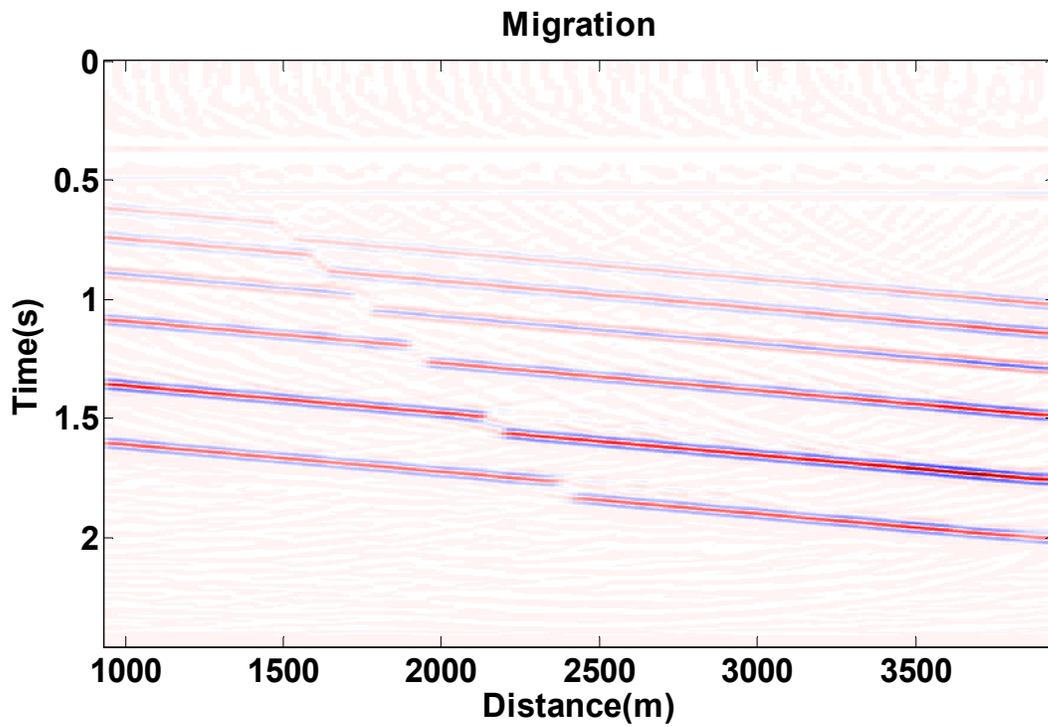


Figure 4.3 Migration of data with the exact velocity.

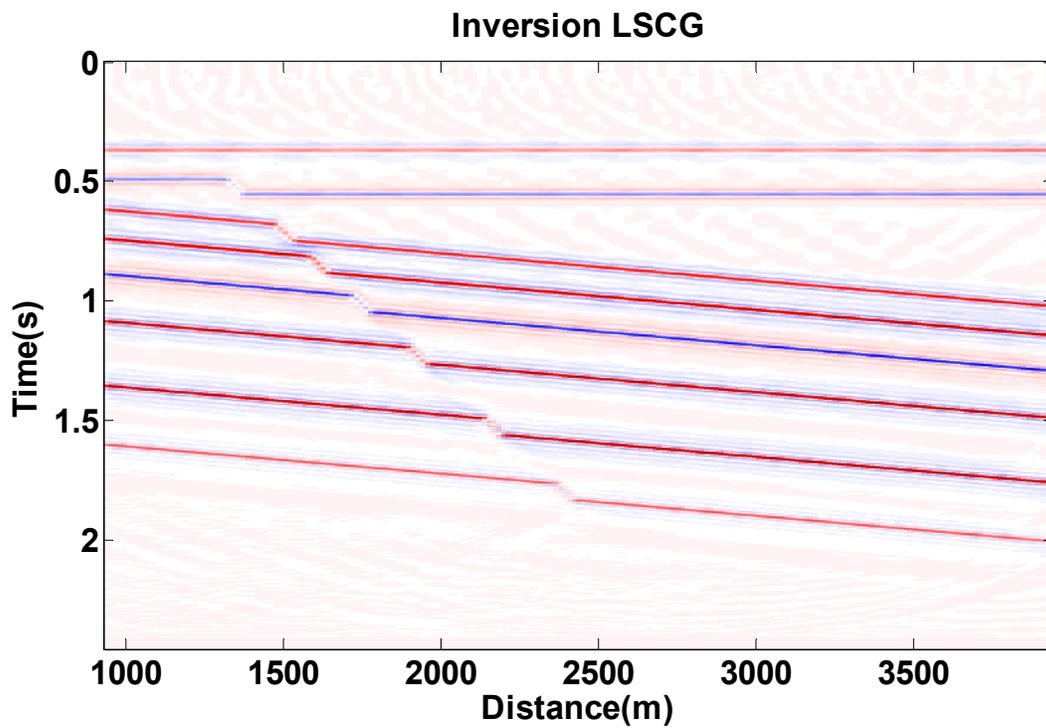


Figure 4.4 LSPSM image of data with the true velocity.

The data are then migrated with a velocity that is 5% higher than the true velocity, and the resulting image is shown in Figure 4.5. LSPSM using the same data and a 5% higher velocity is shown in Figure 4.6. Comparing these two figures shows that the migration images are less affected by the wrong velocity than the LSPSM. LSPSM with inaccurate velocity not only does not improve the resolution of the reflectivity image in comparison to migration, but it also seems to have introduced more artifacts and noise into the image.

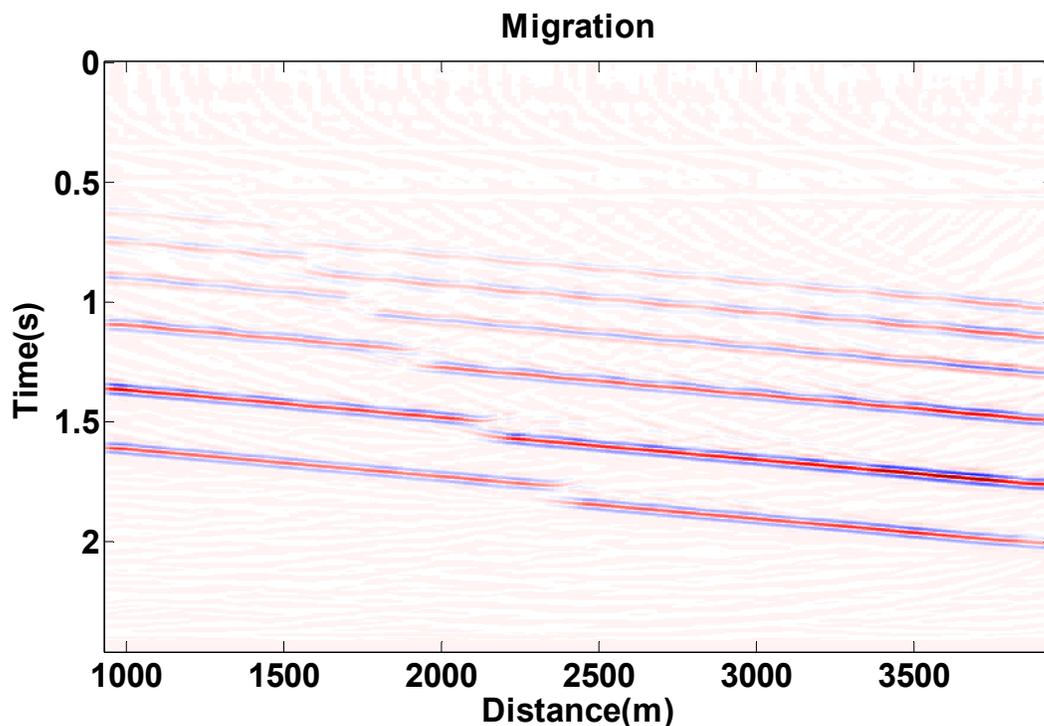


Figure 4.5 Migration of data with the 5% higher than the true velocity.

Consequently, resolution enhancement with the LSPSM method requires a good knowledge of the velocity model. I recommend the ability of the LSPSM for resolution enhancement as a factor to measure the accuracy of any estimated velocity.

4.3.2 Effect on data reconstruction

The sensitivity of the LSPSM to velocity accuracy can also be investigated with a view of the usability of the method for data reconstruction. Worse reconstruction of the data happens when more traces are missing. I used complete data in the LSPSM in the example in the previous

section. The resulting LSPSM images have been used for data reconstruction. Figure 4.7 is a test example that shows the first 50 traces from the fifth shot. Panels a, b, and c show the original data, reconstructed data, and residual, the difference between original and reconstructed data, respectively, when the exact velocity model is used. What remained in the residual panel is mostly the random noise that I added to the synthetic data, and a ghost of low amplitude reflection events.

Figure 4.8 show data reconstruction when I used a velocity 5% higher than the true velocity in LSPSM. The first 50 traces from fifth shot are shown in this figure. Panels a, b, and c show the original data, reconstructed data, and the difference between original and reconstructed data, respectively.

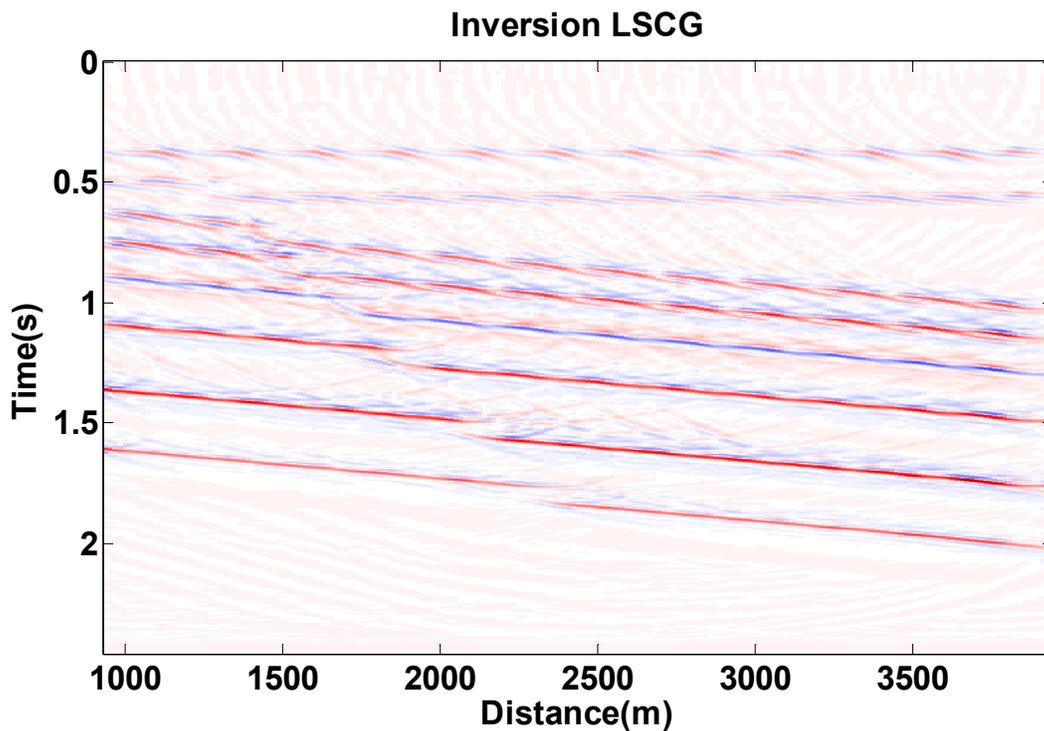


Figure 4.6 LSPSM of data with the 5% higher than the true velocity.

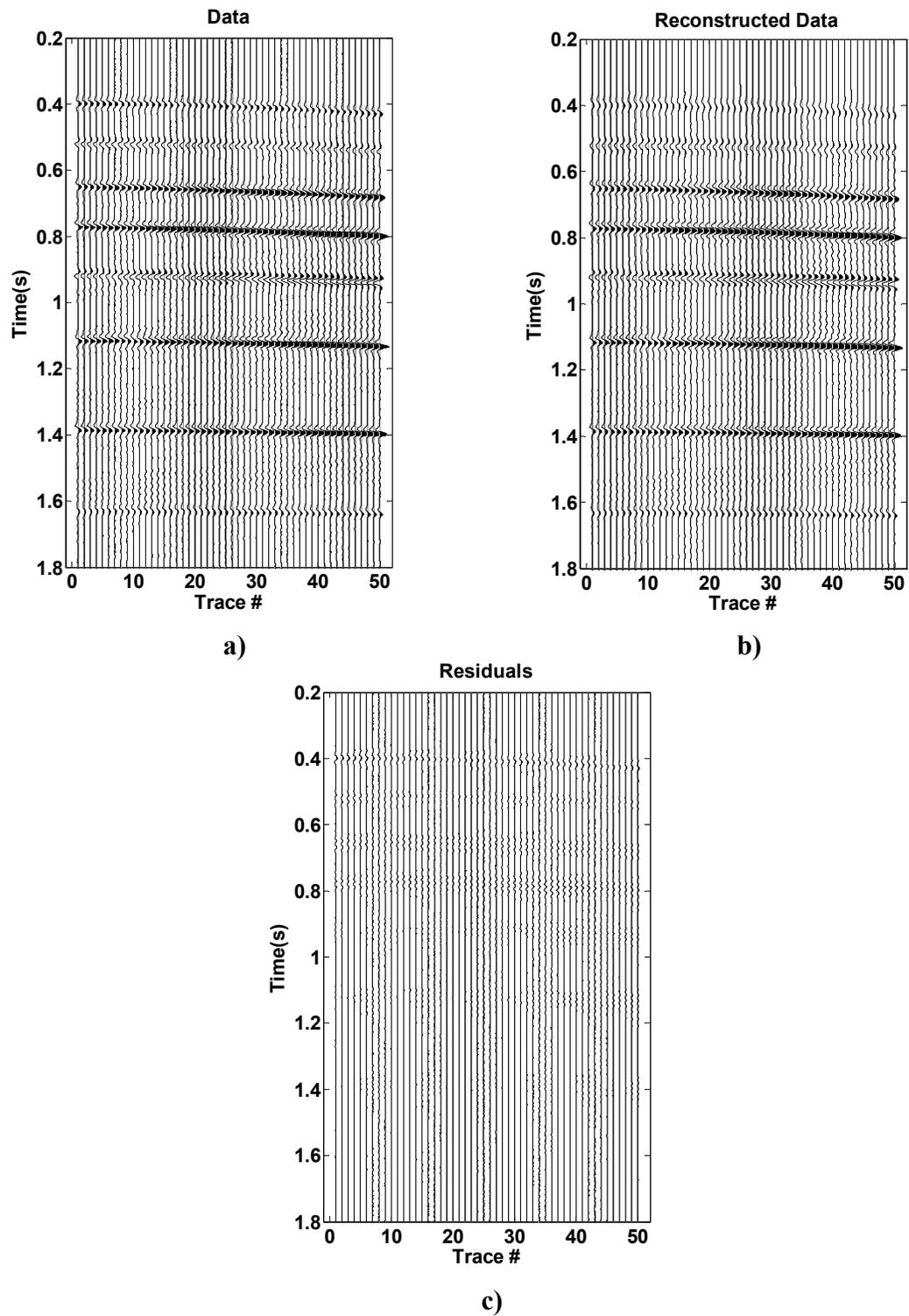


Figure 4.7 Reconstruction of data by LSPSM. Complete data are used and the velocity model is exact. a) original data, b) reconstructed data, and c) residuals.

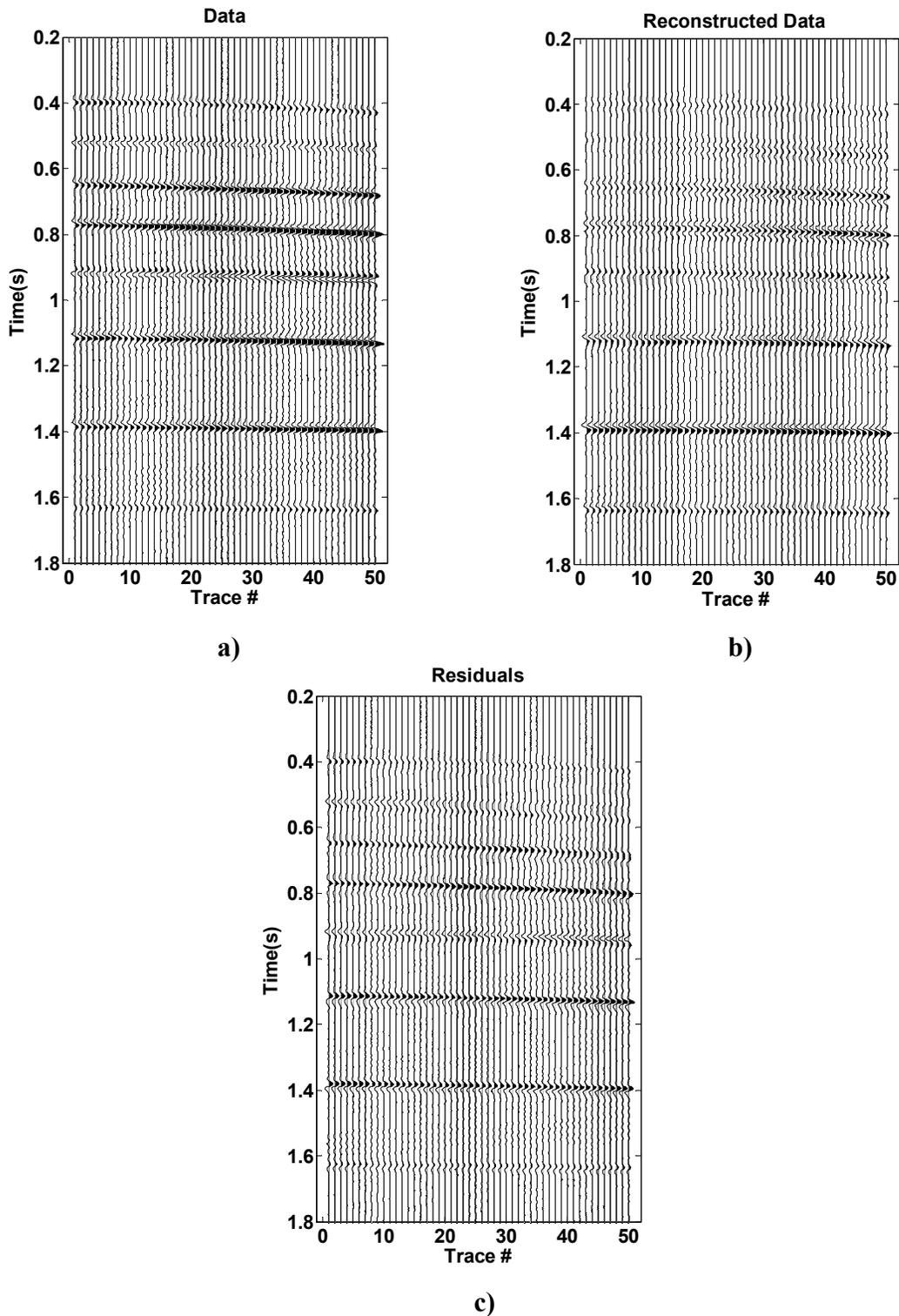


Figure 4.8 Reconstruction of data by LSPSM. Complete data are used and the velocity is 5% higher than the true velocity. a) original data, b) reconstructed data, and c) residuals.

Panel c in Figure 4.7 indicates a very good reconstruction of data. The error in Figure 4.8c shows that the data reconstruction was not successful because of using a velocity 5% higher than the true value. This is due to the high sensitivity of LSPSM to the accuracy of the velocity information. The same result is achieved when using a velocity which is lower than the true velocity. This provides another tool to measure the accuracy of the velocity model used in seismic imaging. An acceptable velocity model should be able to reconstruct data with high accuracy.

4.3.3 The convergence rate of LSCG

LSPSM's resolution enhancement and data reconstruction fails with an inexact velocity model as shown in the previous sections. I realized that the wrong velocity also affects the convergence rate of the LSCG method. As mentioned in Chapter Two, the convergence rate is the relative Euclidean norm of the difference between the observed data and the synthetic data obtained after each iteration of an iterative method.

Figure 4.9 shows the convergence rate of the LSCG method for the synthetic examples in the previous sections. Using the exact velocity, the residual converges to 10% in four iterations. However, the convergence will not go down to less than 50% with a 5% higher velocity, and to 65% with a 10% higher velocity. As the velocity moves farther from the exact velocity, the convergence gets slower, and the convergence value gets larger. I define an acceptable velocity model when the convergence rate is about 60% in a few iterations with real data.

Therefore, the improvement in image resolution, the ability of data reconstruction, and the quality of the LSCG's convergence rate are three important factors that can be used to check the accuracy of a velocity model. In the remainder of this chapter, I extend the velocity analysis on the migration CIGs to the LSPSM CIGs where I expect higher resolution in the LSPSM CIGs due to the higher sensitivity of the LSPSM to the velocity. I will show this method to be very effective when data are highly decimated.

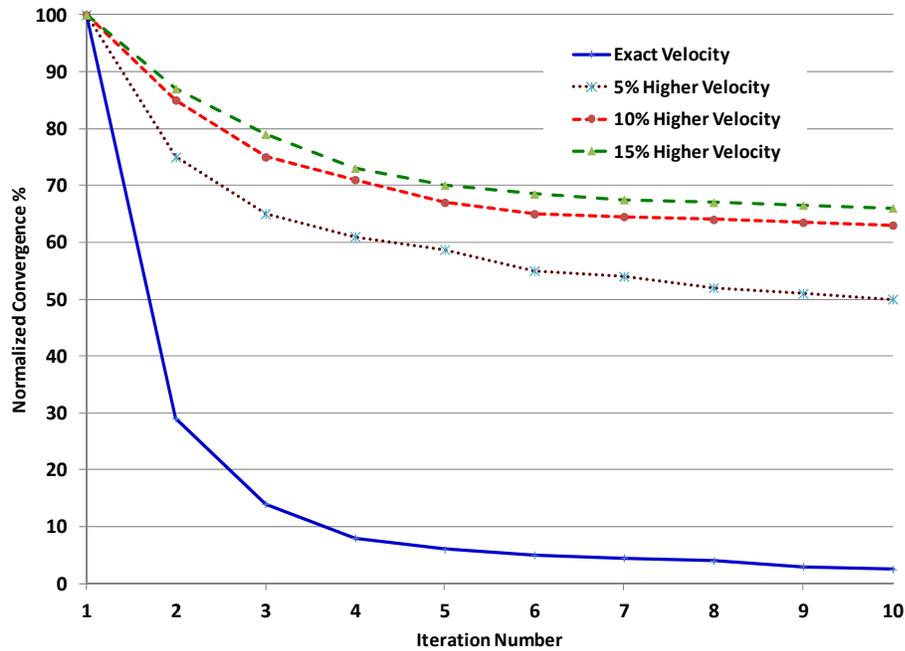


Figure 4.9 Normalized convergence rate of the LSCG method in 10 iterations, with true, and a 5% higher velocity model.

4.4 Velocity analysis on LSPSM offset domain CIGs

It is very convenient with Kirchhoff migration to construct offset domain CIGs by splitting traces into 10 offset bins, for example, and performing migration and LSPSM on each offset bin data separately. Then the final image can be reconstructed by stacking images from all common offset bins. However, the un-stacked images are useful for some analyses that may require some information about amplitude variation versus offset, for example. They also provide a good environment for velocity analysis. Velocity analysis on the offset domain migration CIGs is useful in the area with dipping layers.

I chose 10 as the number of offset bins and performed migration and LSPSM on the synthetic data in the previous section. Figure 4.10 shows the offset domain CIG at $x = 2450 m$ after migration with the exact velocity (a), and with 5% higher velocity (b). Panels (a) and (b) in Figure 4.11 show LSPSM CIGs at the same location with the true velocity (a) and when the implemented velocity is 5% higher (b).

The offset domain CIGs from LSPSM have a slightly higher resolution especially in the shallower parts of the CIGs with a true velocity, as expected. The events are not flattened when a higher velocity model is used (this may be viewed by looking sideways along each event, eg. at

1.58 s). LSPSM CIG is noisier with the implementation of the wrong velocity. The higher resolution of the offset domain CIGs from LSPSM, compared to migration, makes them attractive for correcting the velocity.

I extended the idea of using migration CIGs for velocity analysis to the LSPSM CIGs of incomplete data. Therefore, instead of performing constant velocity migration, I do constant velocity LSPSM on the offset binned data and produce a semblance spectrum for each LSPSM CIGs (Yousefzadeh et al., 2011, 2012).

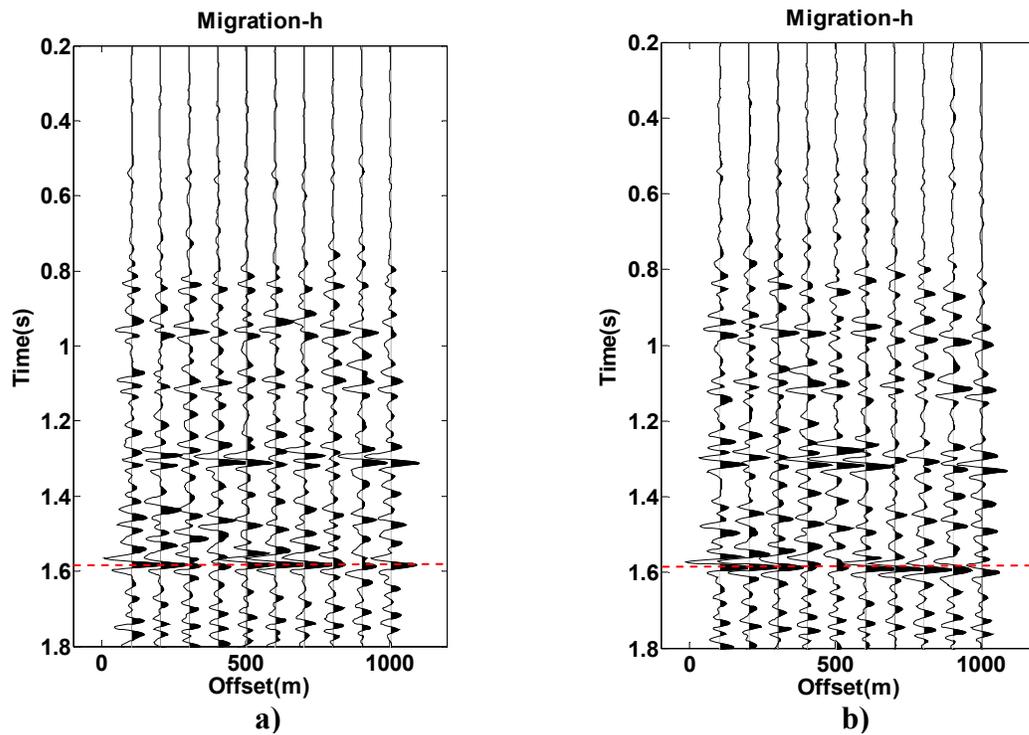


Figure 4.10 Migration offset domain CIG at $x = 2422\text{ m}$ with the a) true velocity and b) with a 5% higher velocity.

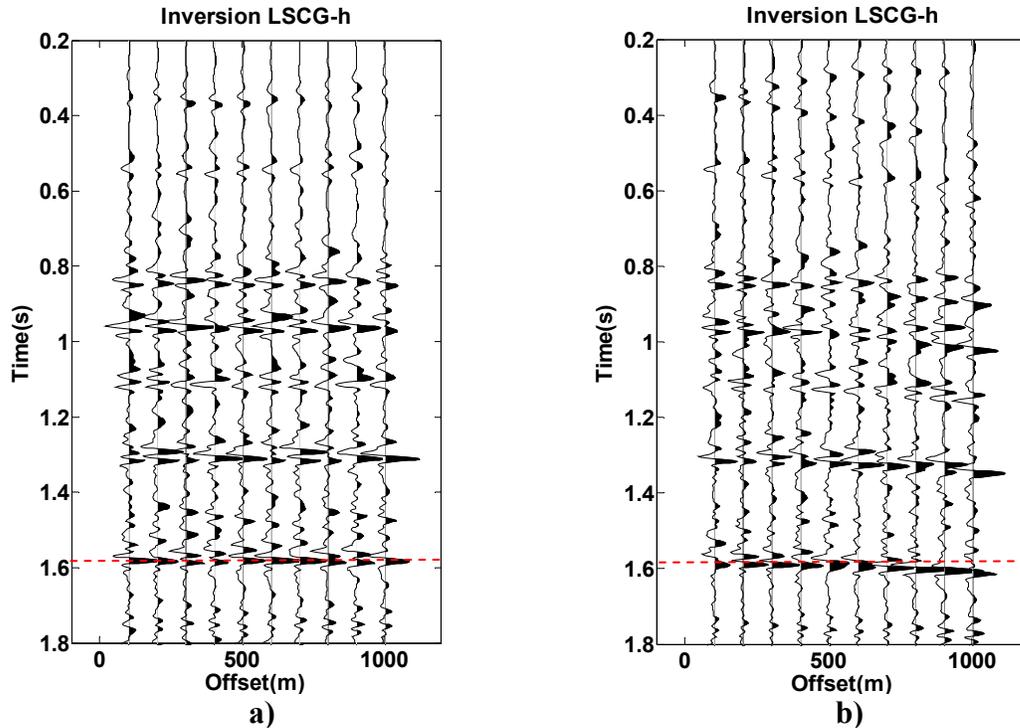


Figure 4.11 LSPSM offset domain LSPSM CIG at $x = 2422\text{ m}$, a) with the true velocity and b) with a 5% higher velocity.

In this section, I demonstrate the use of LSPSM offset domain CIGs for velocity analysis and compare it with velocity analysis based on migration offset domain CIGs using the model in Figure 4.2. For this example I performed velocity analysis on 90% regularly decimated traces using the semblance method on CMP gathers, offset domain CIGs from migration, and offset domain CIGs from LSPSM. The velocity range starts from 1550m/s and with the increment of 25m/s , and increases to 3500m/s .

Figure 4.12, Figure 4.13, and Figure 4.14 show the semblance spectrums (a) at $x = 2422\text{m}$ in the middle of the section, and (b) at $x = 3922\text{m}$ at the edge of the model, with only 10% of the data. Figure 4.12 shows the semblance spectrum from CMP gathers, Figure 4.13, and Figure 4.14 show the semblance spectrum for the offset domain CIG from migration and LSPSM, respectively.

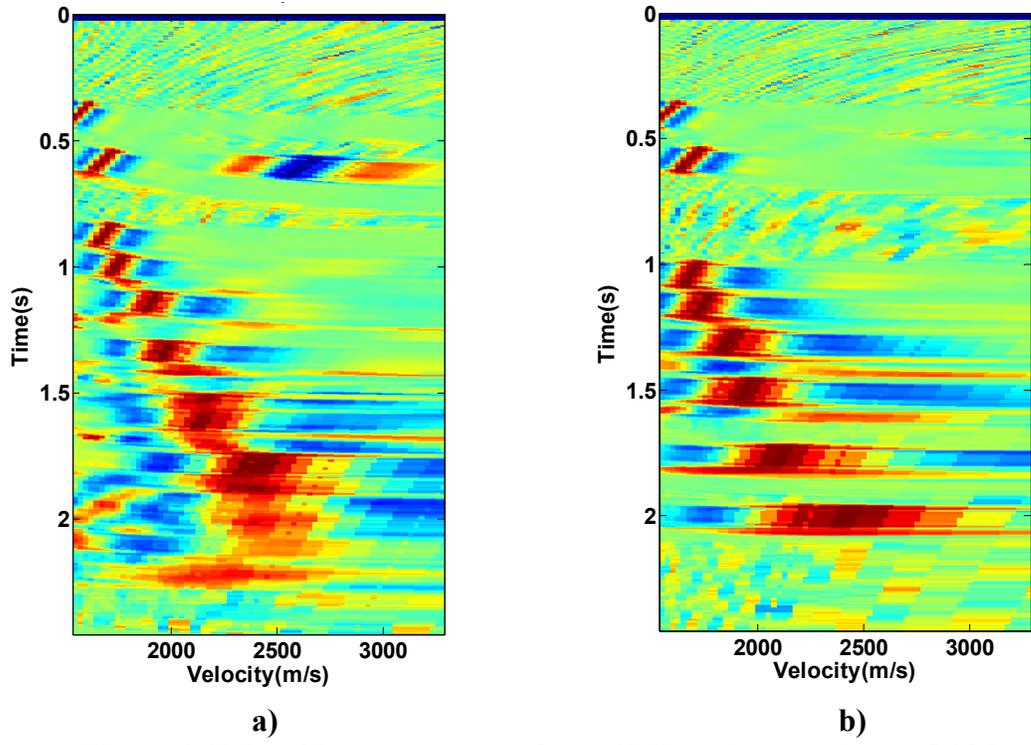


Figure 4.12 Semblance spectrum from CMP gathers at a) $x = 2422\text{ m}$ and b) $x = 3922\text{ m}$ using 10% of regularly selected data.

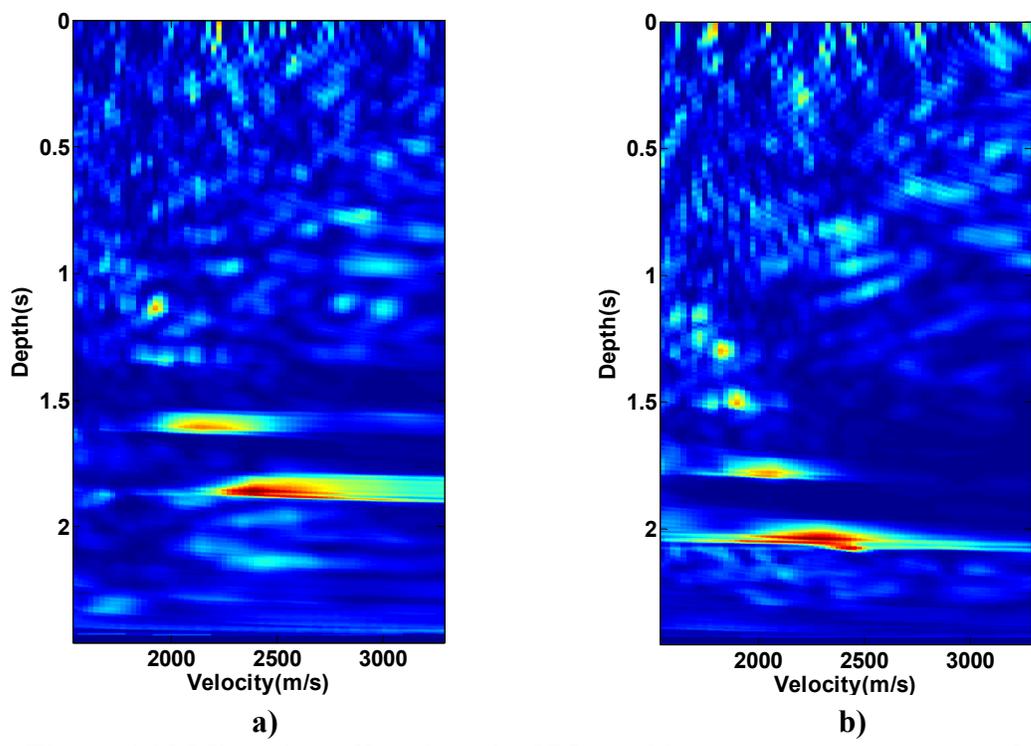


Figure 4.13 Migration offset domain CIG semblance spectrum at a) $x = 2422\text{ m}$, and b) $x = 3922\text{ m}$ with 10% of regularly selected data.

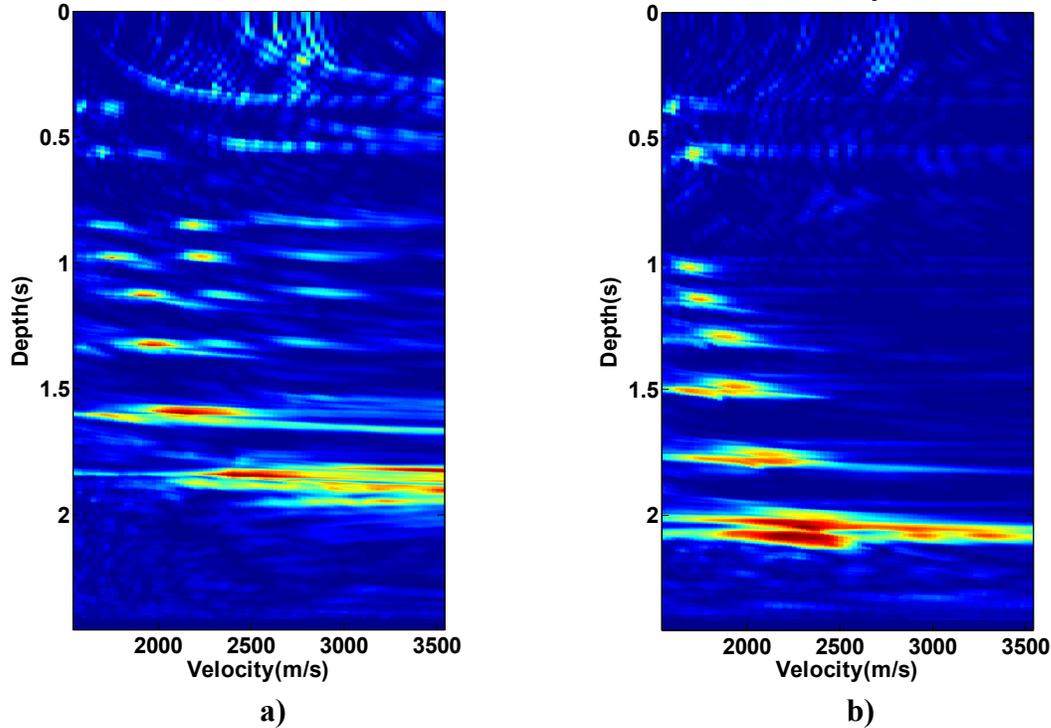


Figure 4.14 LSPSM offset domain CIG semblance spectrum at a) $x = 2422\text{ m}$, and b) $x = 3922\text{ m}$ with 10% of regularly selected data.

The semblance spectrum is noisy due to the highly decimated data. Because of the presence of dipping layers, the velocity at each depth is higher than the true velocity, as shown in the semblance of the CIGs. The comparison between the semblance of migration offset domain CIGs, with the semblance of the LSPSM offset domain CIGs, shows that LSPSM gives a slightly better image for velocity picking, especially in shallower parts. This is due to the low sensitivity of the LSPSM method to data incompleteness, and its high sensitivity to velocity. Hence, velocity analysis on offset domain migration CIGs is recommended in the case of having highly decimated data. In the next section velocity analysis on the shot domain CIGs is shown.

4.5 Velocity analysis on LSPSM shot domain CIGs

The idea of velocity analysis on LSPSM offset domain CIGs is extended to the LSPSM shot domain CIGs. I use LSPSM shot domain CIGs for the velocity analysis and compare them to a velocity analysis based on migration shot domain CIGs.

In the shot domain CIGs, the horizontal axis shows the distance between a source (shot) and the corresponding image positions. Therefore, the number of traces in each CIG is equal to

the number of shot points. The number of sources for a 2D seismic line can be a relatively large number, therefore, I limited the number of traces in the shot domain CIGs by binning the maximum source to image point distance to a reasonably small number. The number of bins depends on the maximum shot to image point distance, source interval, and the complexity of the area. The best choice for the number of bins is when there are between two to five shot points stacked into one CIG bin.

For all analyses in this section, the synthetic data is created with the acquisition geometry and the velocity model in section 4.3. The geometry includes 16 sources, then 8 is a reasonable number for binning the source image point distance. Binning to the higher number introduces some artifacts to the migration of each bin. With this binning, and using the complete data set, migration and LSPSM are performed.

Figure 4.15 shows two migration shot domain CIGs in the surface positions $x = 2422m$ with the exact velocity implemented (a) and with 10% higher than true velocity (b). Figure 4.16 shows two LSPSM shot domain CIGs from the same position and when the exact velocity is used (a) and with a 10% higher than true velocity (b).

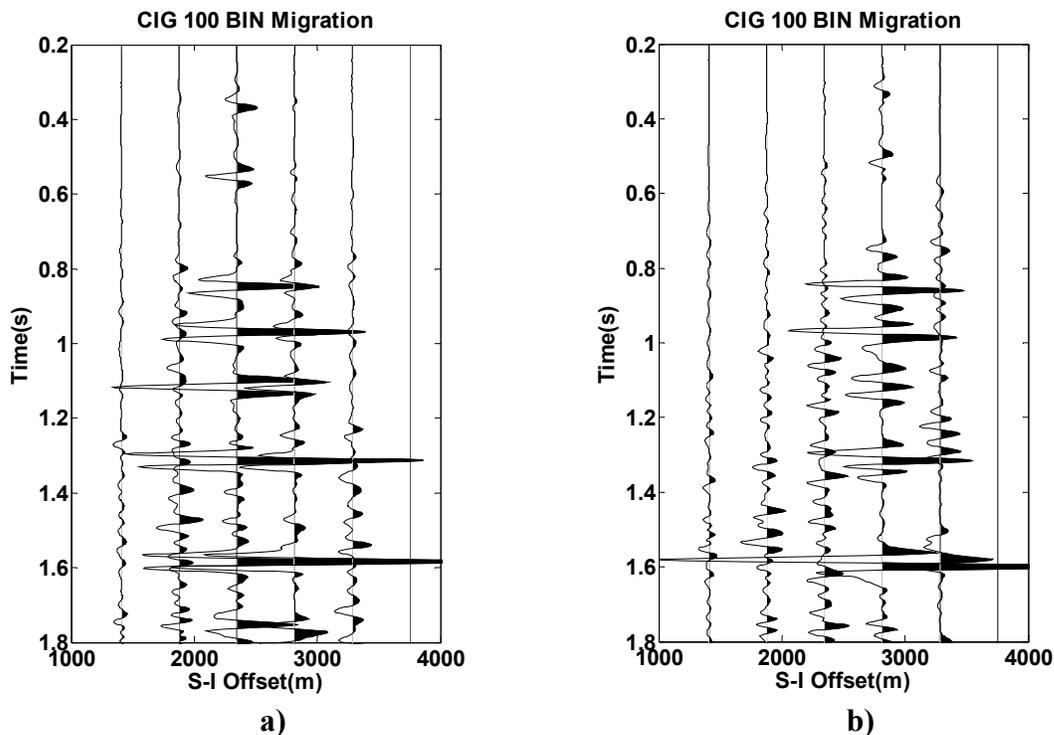


Figure 4.15 Migration shot domain CIGs at position $x = 2422 m$, a) using the exact velocity and b) using a velocity 10% higher than the exact value.

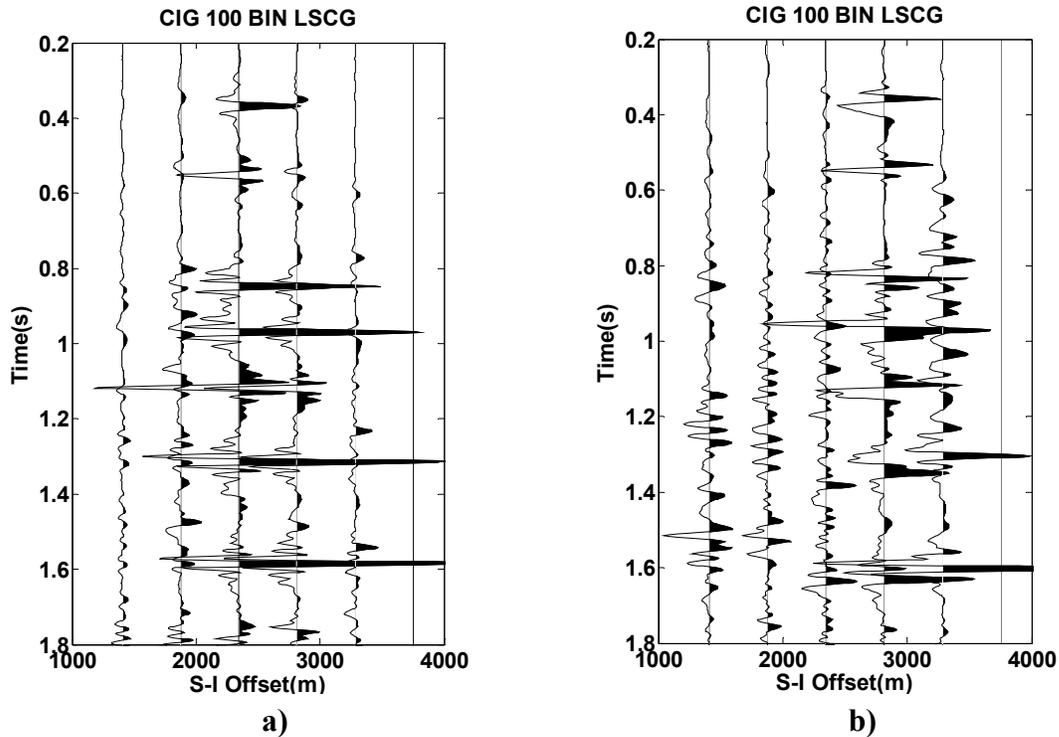
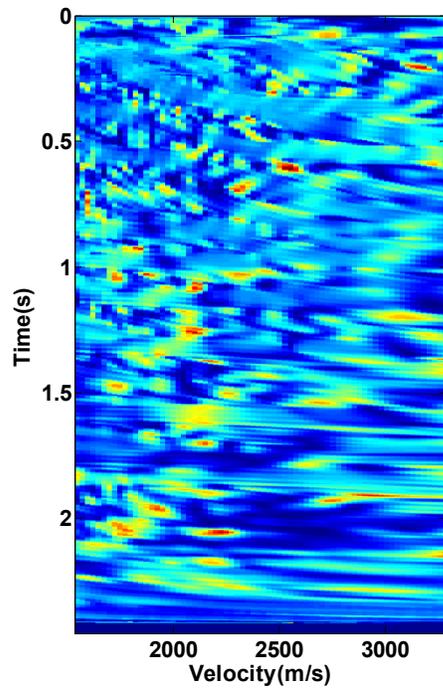


Figure 4.16 LSPSM shot domain CIGs at position $x = 2422\text{ m}$, a) using the exact velocity, and b) using a velocity 10% higher than the exact value.

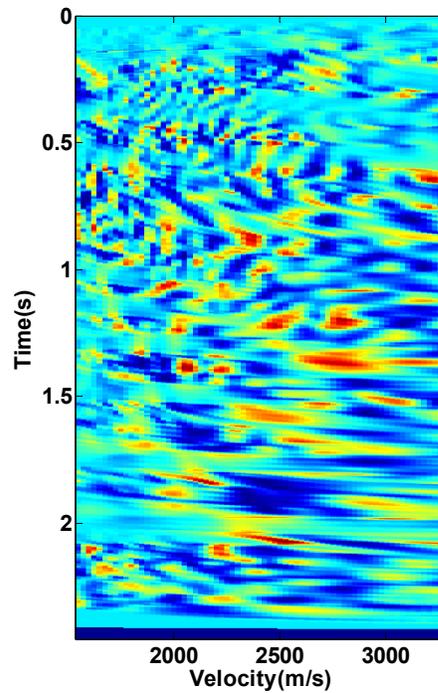
Comparing the CIGs in Figure 4.15 and Figure 4.16 shows a noticeable difference between migration shot domain CIGs and LSPSM shot domain CIGs with incorrect velocities. LSPSM CIGs are noisier, and the coherency between events in the LSPSM CIGs is lower. The incoherency increases when there is more decimation of data.

The high sensitivity of the shot domain LSPSM CIGs to the velocity accuracy, makes shot domain LSPSM CIGs attractive for velocity analysis. I performed velocity analysis on the shot domain CIGs using the same method that I used for semblance analysis on the offset domain CIGs. Using constant velocities within a range that starts at 1550 m/s and increments 25 m/s to 3500 m/s , I performed constant velocity migration and LSPSM on only 10% of regularly sampled data and I created a semblance spectrum for each CIG.

Figure 4.17 shows the semblance spectrum of the migration shot domain CIGs at the positions $x = 2422\text{ m}$, at the middle, and $x = 3922\text{ m}$, at the right hand side of the model, with 90% of data regularly decimated. Figure 4.18 shows the semblance spectrum on the LSPSM shot domain CIGs.

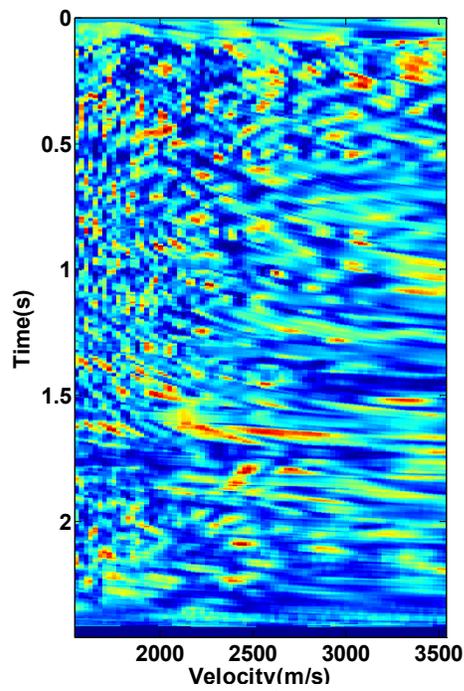


a)

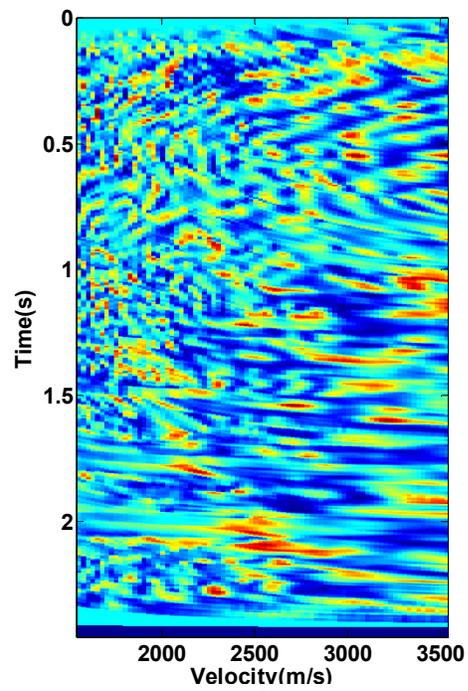


b)

Figure 4.17 Migration shot domain CIG semblance spectrum at a) $x = 2422 m$, and b) $x = 3922m$, with 10% of the data.



a)



b)

Figure 4.18 LSPSM shot domain CIG semblance spectrum at a) $x = 2422 m$, and b) $x = 3922m$, with 10% of the data.

The achieved semblance panels are noisy, and there is no clear contrast to choose the best velocity at each time. This is due to the strong incoherency of shot domain CIGs when the incorrect velocity is used. Fortunately, there are other methods of coherency measurements for velocity analysis. I found that the unnormalized crosscorrelation sum, (XC), is an effective quantity for measuring the coherency in the shot domain migration and LSPSM CIGs. XC is defined by,

$$XC = \frac{1}{2} \sum_t \left\{ \left[\sum_{i=1}^N a_{i,t(i)} \right]^2 - \sum_{i=1}^N a_{i,t(i)}^2 \right\}, \quad 4.3$$

where $a_{i,t(i)}$ is the amplitude of the i^{th} trace at time $t(i)$ and N is the number of traces that contributes to the measurements in a defined window (Yilmaz, 2008). Yilmaz (2008) interprets XC as “half the difference between the output energy of the stack and the input energy”.

Figure 4.18 shows the XC spectrum for the migration shot domain CIG at the surface positions $x = 24220m$ and $x = 3922m$, when 90% of data are regularly decimated. A comparison of Figure 4.16 and Figure 4.18 shows the relative effectiveness of using XC instead of semblance for velocity analysis on the shot domain CIGs.

Figure 4.20 shows the XC spectrums for the LSPSM shot domain CIGs at the same positions when only 10% of the data are used. The improvement in the XC resolution for velocity analysis by using LSPSM shot domain CIGs instead of migration shot domain CIGs is noticeable.

Using the LSPSM shot domain CIGs XC spectra I extracted three velocity functions at the left, middle, and right edge of the model as shown in Figure 4.21. These figures show a good agreement between the true rms velocity (solid line) and the analyzed rms velocities. Figure 4.22 compares the true rms velocity with the extracted velocity. The extracted velocity is very close to the true velocity. However, it is not possible to reveal the effect of the Fault in the model.

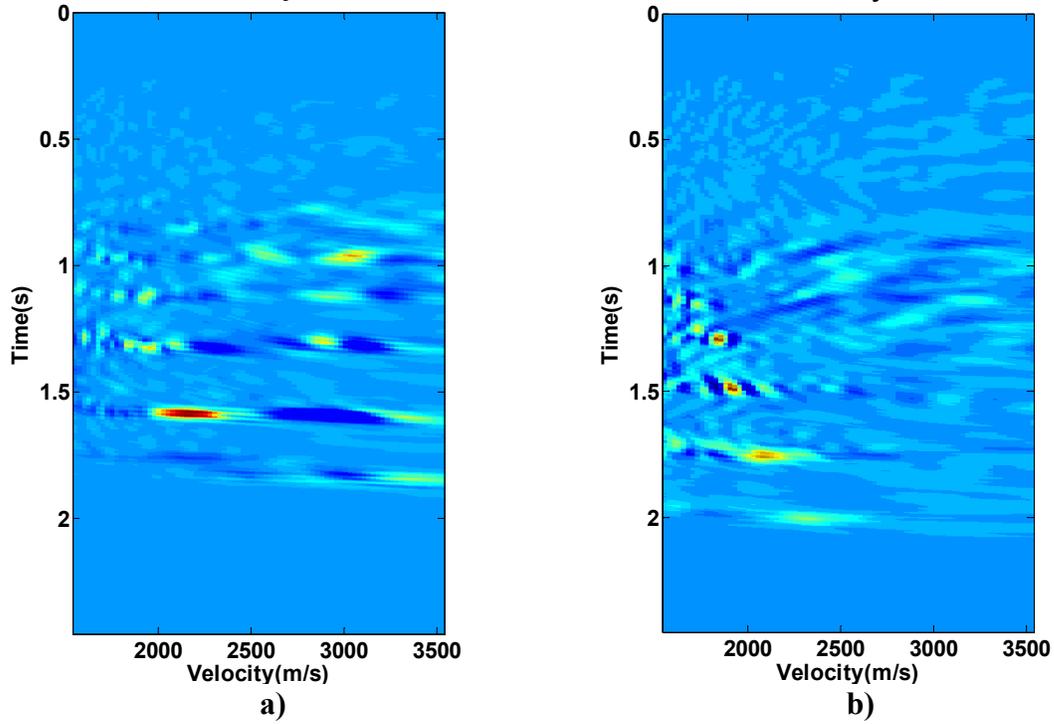


Figure 4.19 Migration shot domain CIG XC spectrum at a) $x = 2422 m$, and b) $x = 3922m$, with 10% of the data.

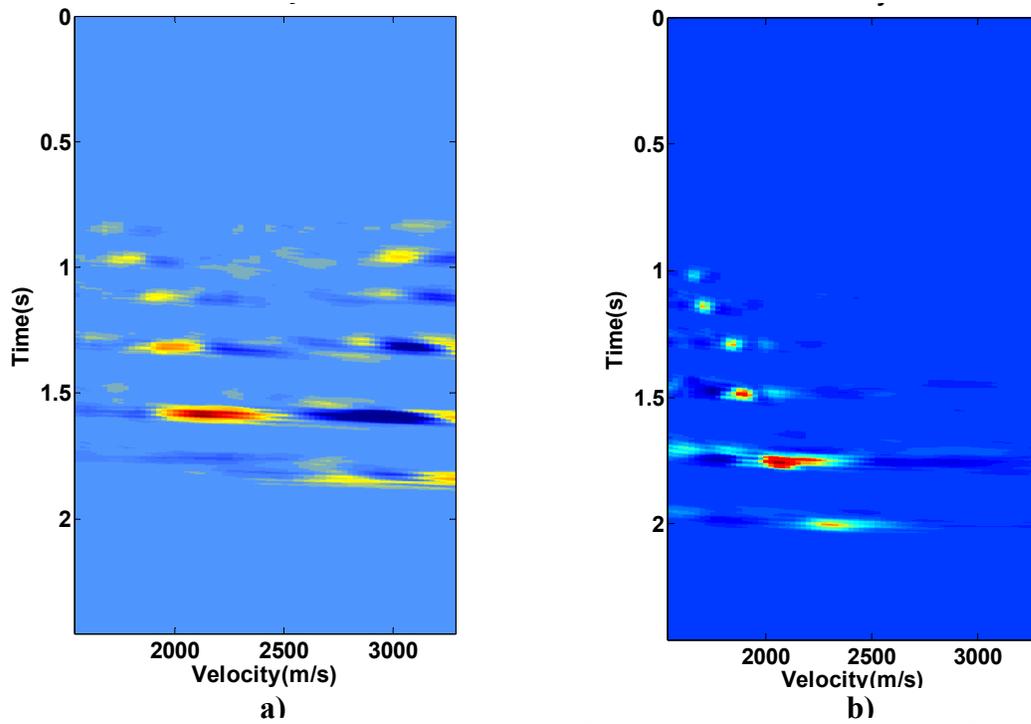
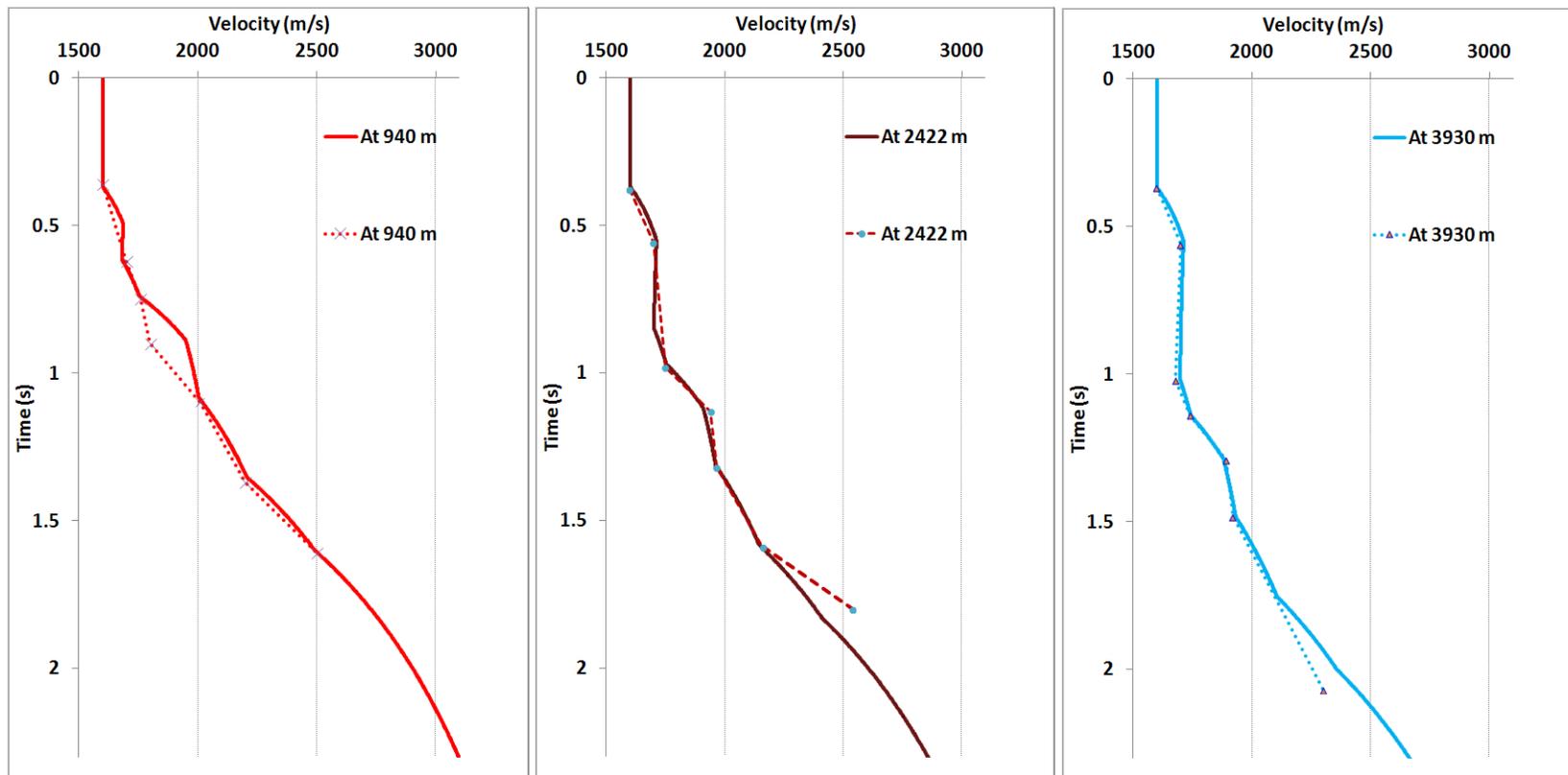
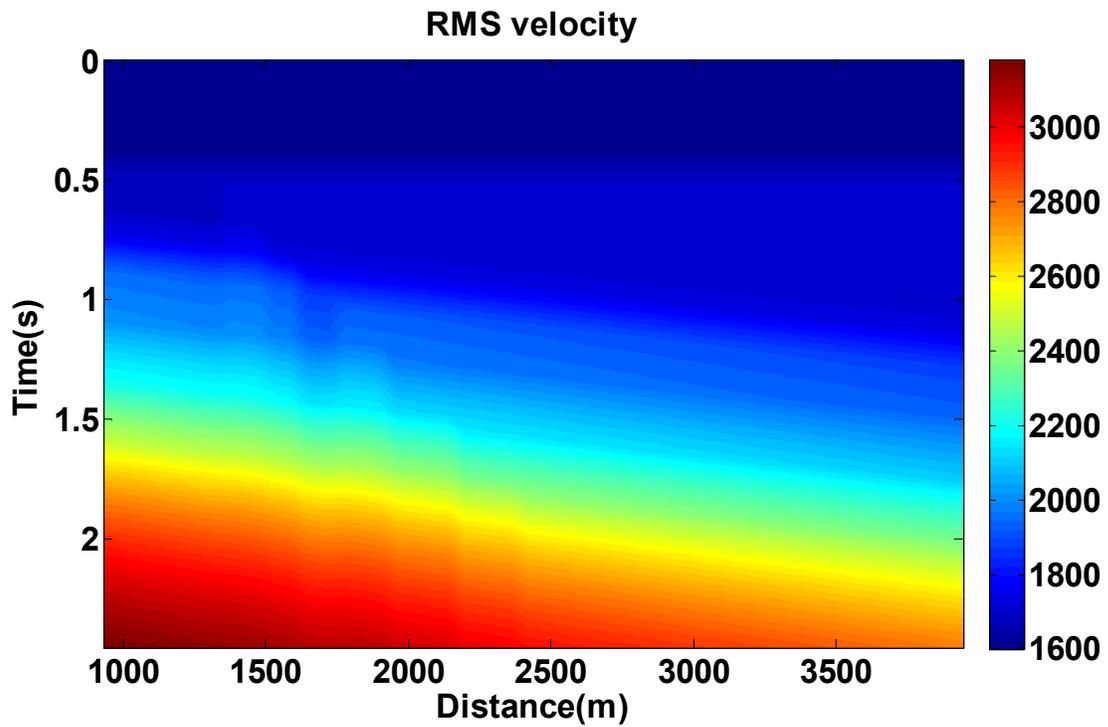


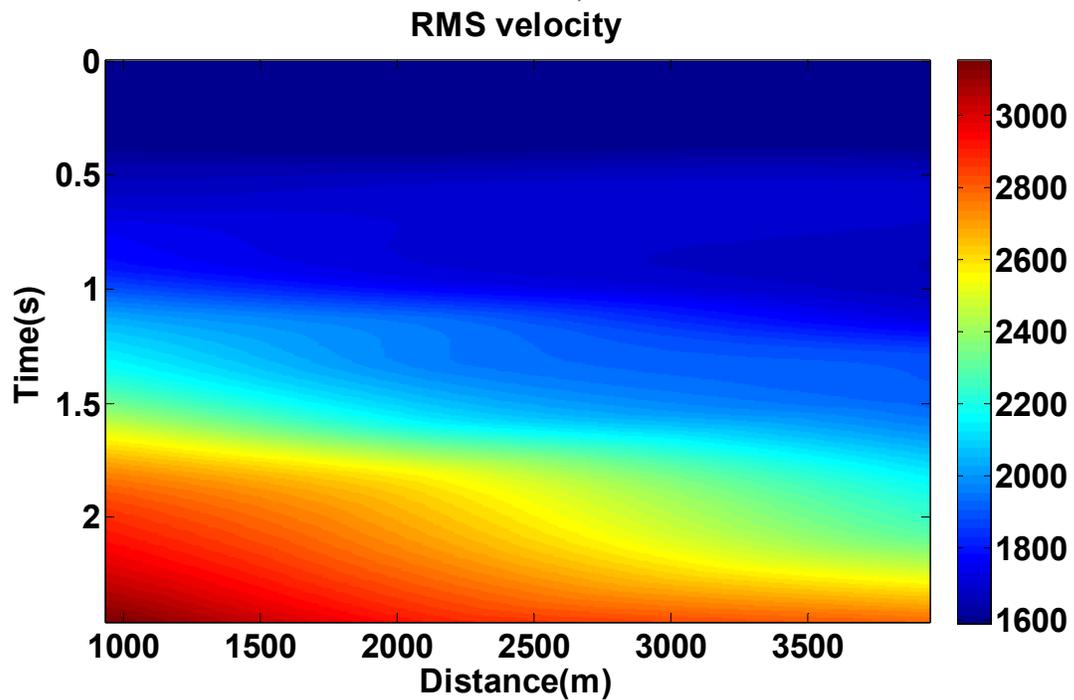
Figure 4.20 LSPSM shot domain CIG XC spectrum at a) $x = 2422 m$, and b) $x = 3922m$ with 10% of the data.



a) b) c)
Figure 4.21 Comparison between the true and the analyzed velocity function at 3 positions.



a)



b)

Figure 4.22 a) RMS velocity model used to generate synthetic data, and b) the extracted velocity using the XC method on the LSPSM shot domain CIGs.

The decimated data are migrated with the velocity information extracted by this method on the 10% of regularly decimated data. Figure 4.23 shows the migration and Figure 4.24 shows the LSPSM results. The migration image is blurred, but LSPSM is able to give a high resolution image with less blurriness which is comparable with the result in Figure 4.4 which uses all data and the exact velocity information.

The image in Figure 4.24 is used for the reconstruction of the 90% missing data. Figure 4.25 shows the data reconstruction results based on the first 50 traces from the fifth shot gather. The difference between original and reconstructed data is shown in panel d. Comparison between residuals in this panel with the residuals in panel d in Figure 4.7 shows that the data reconstruction was relatively successful.

To see the degree of accuracy of the extracted velocity information, it is also useful to compare the convergence rate of the LSPSM with the true velocity and when the obtained velocity is used. Figure 4.26 shows the convergence rate when true, higher, and extracted velocities are used in LSPSM. In one iteration, LSPSM with the extracted velocity converges to 35% of the original difference where this amount with the true velocity is 29%. Finally, in 10 iterations, the extracted velocity causes the convergence rate to go down to 5%, only 3% more than when the true velocity has been used in the LSPSM method.

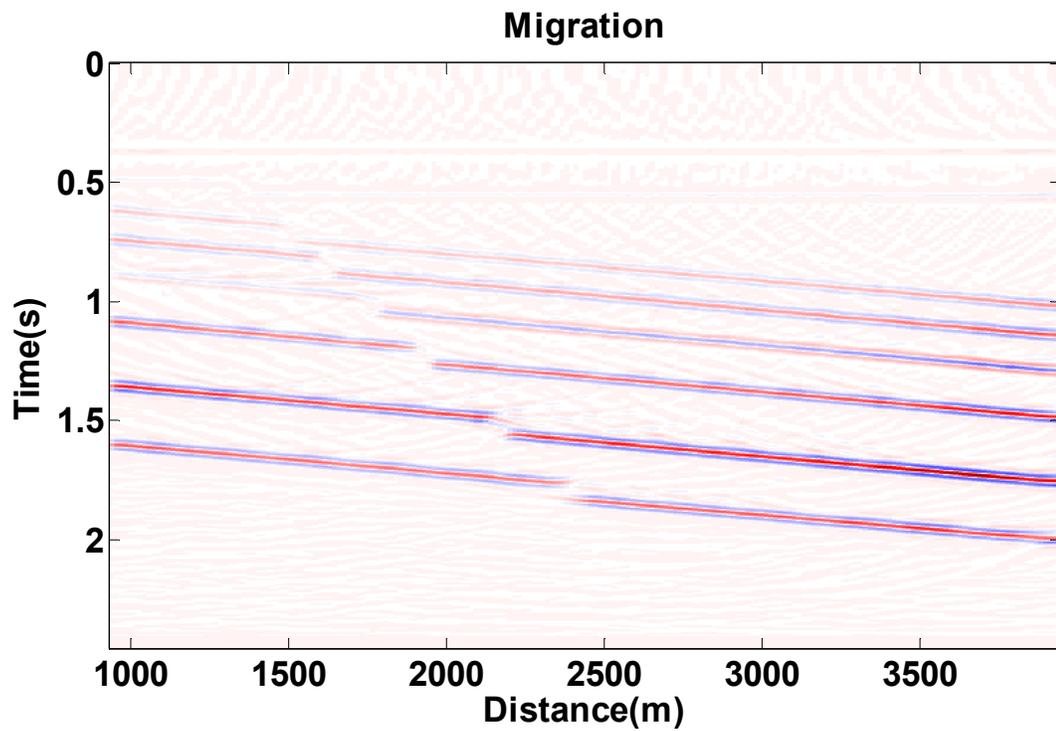


Figure 4.23 Migration of the decimated data with the extracted velocity model.

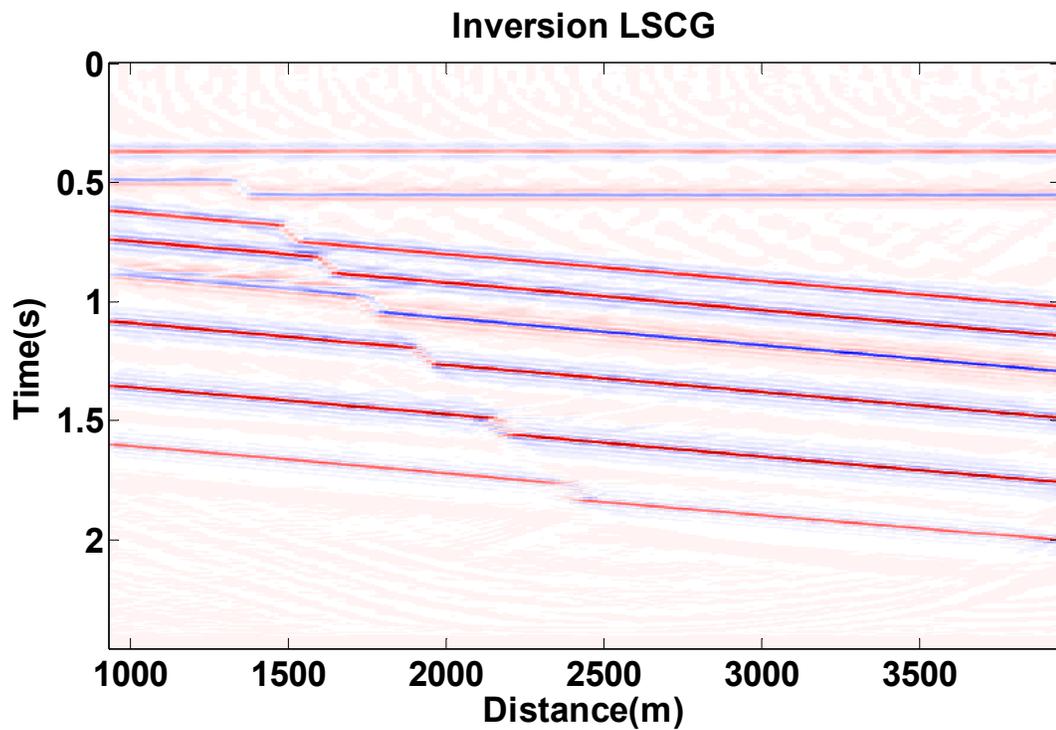


Figure 4.24 LSPSM of the decimated data with the extracted velocity model.

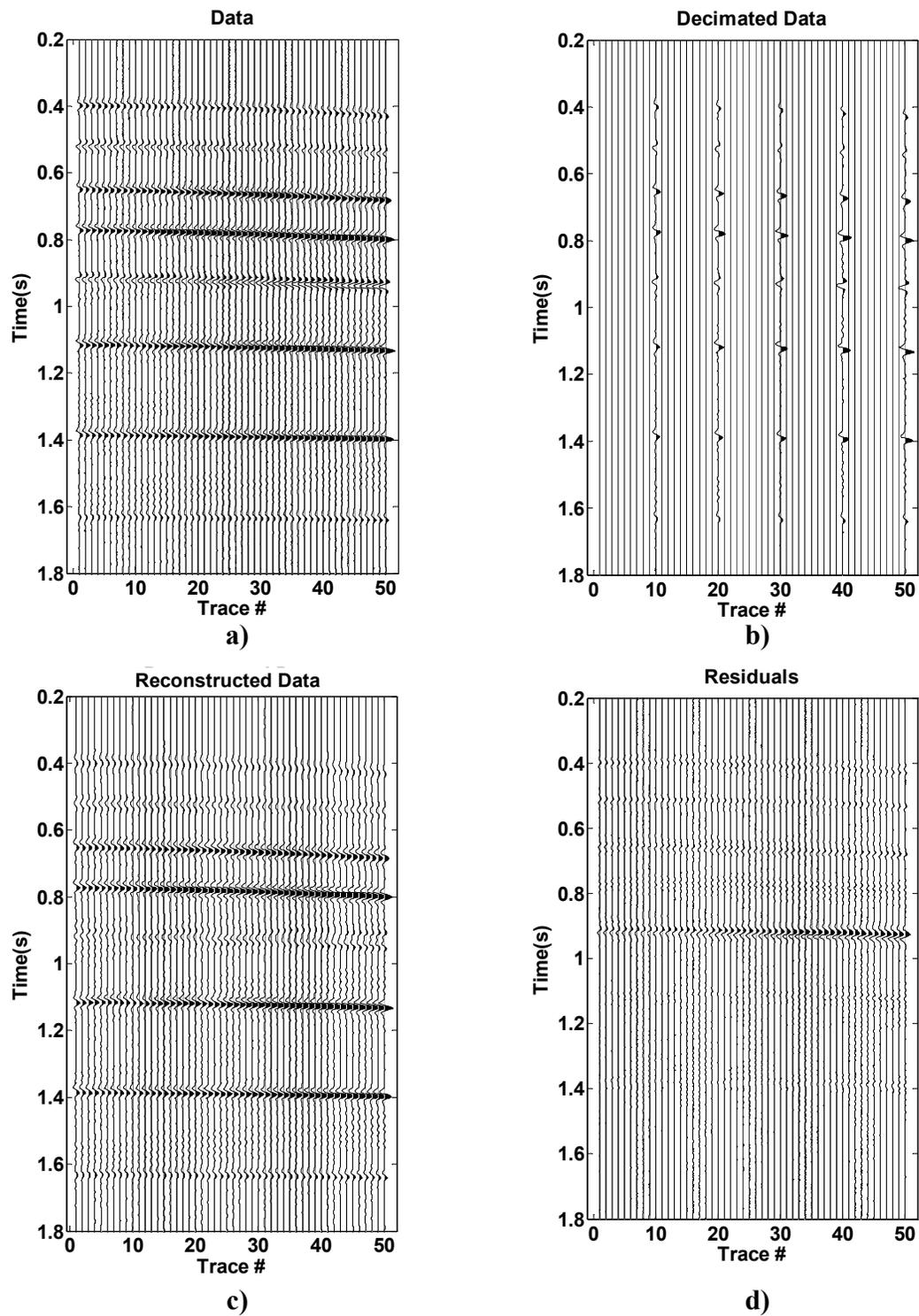


Figure 4.25 Reconstruction of data with the extracted velocity; a) original data, b) decimated data, c) reconstructed data, and d) residual.

Since each iteration in the LSPSM takes twice the time of a migration, this method of velocity analysis on LSPSM CIGs seems to be a very costly procedure. However, doing 10 iterations on LSPSM with 10% of data is only twice as expensive as a velocity analysis on migration CIGs using all data.

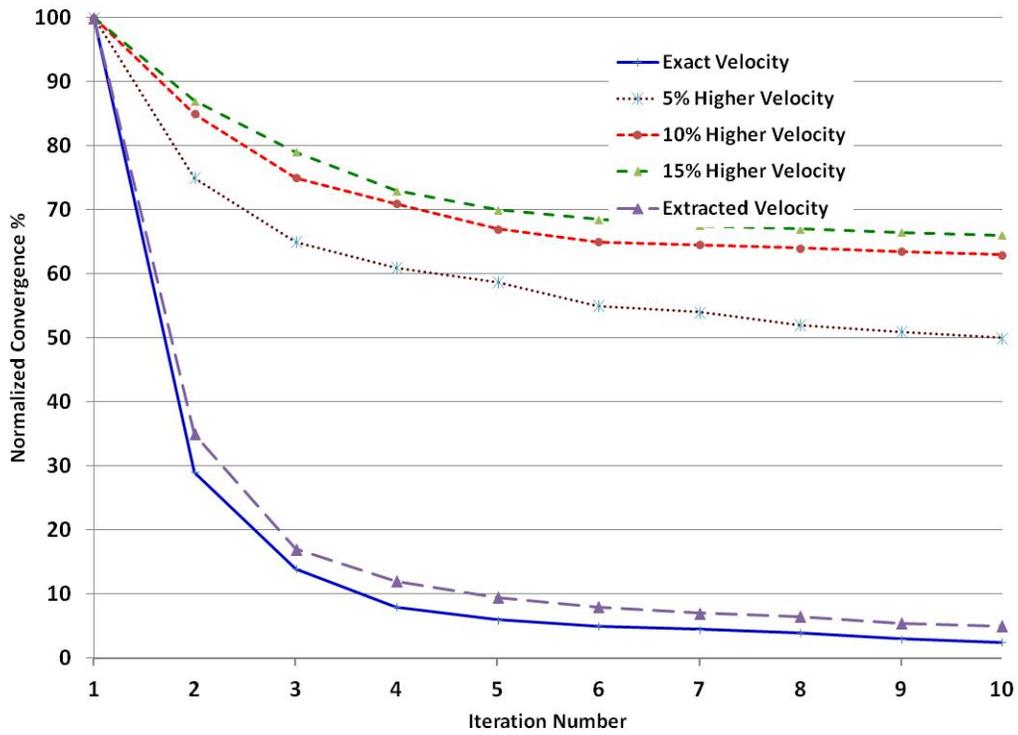


Figure 4.26 Normalized convergence rate of the LSCG method in 10 iteration with true, higher, and extracted velocities.

4.6 Summary and Conclusion

Using Kirchhoff LSPSM with proper velocity information gives a higher resolution image than the conventional Kirchhoff migration and produces less acquisition footprint.

The sensitivity of the LSPSM method to the velocity accuracy can be used to evaluate the velocity model. I showed the effect of using an inaccurate velocity model on resolution enhancement of LSPSM and data reconstruction as well as the convergence rate of the LSCG method. The best velocity model is the one that improves the LSPSM image relative to the conventional migration image by attenuating acquisition footprint, provides good data reconstruction, and gives a good convergence rate in the LSCG.

With dipping reflectors, semblance velocity analysis in the data domain fails to give the correct rms velocity. The semblance method on the offset domain CIGs, or unnormalized crosscorrelation on the shot domain CIGs from migration, can be used instead.

I showed that velocity analysis with the semblance method on the migration offset domain CIGs, which is a robust tool for migration velocity analysis, can be extended to the LSPSM offset domain CIGs. This gives better results when the data are highly irregularly or incompletely sampled. I showed that the velocity extracted by this method is accurate enough to give a high resolution image in LSPSM and works well for data reconstruction. The same result is achieved by using an unnormalized crosscorrelation on the shot domain CIGs.

Replacement of migration CIGs with LSPSM CIGs for velocity analysis is a costly procedure. However, it gives better results with incomplete or irregular data.

Chapter Five: LSPSM/INVERSION FOR PRE- AND POSTSTACK TIME LAPSE STUDIES

5.1 Introduction

Time lapse or 4D, (x, y, z, t) , where time is the fourth dimension, is the study of seismic data recorded at different times. This study includes recording and analysing a secondary seismic survey after a period of time in order to detect subtle differences or changes in the physical properties of the hydrocarbon reservoirs or injection sites. These changes can be due to either production or injection of a fluid (oil, gas, water, steam, etc.) from or into the reservoirs, or sequestration of CO_2 . Usually, the first seismic survey is called the baseline survey and subsequent ones are called monitor surveys. Assuming all acquisition parameters, instrumentation, environmental noise, near surface effects, and processing procedures are exactly equal, the comparison between two final migrated images may show the effect of fluid movement in the reservoir. The effect may be a small difference in the travelttime of an event or a change in seismic attributes, such as reflectivity. Time lapse processing is a general name for this process.

One of the first successful time lapse surveys was performed in 1987 in the Holt Fireflood reservoir to show the movement of the gas/oil contact (Greaves and Fulp, 1987). More than 100 time lapse seismic surveys were performed around the world by 2001 (Lumley, 2001). Time lapse studies are not restricted to the comparison of poststack images; for example, Vedanti and Sen (2009) performed prestack time lapse study by inversion of prestack data for elastic parameters to track the thermal front of an *in situ* combustion project.

Time lapse methods become more demanding when hydrocarbon reservoirs become depleted and go into the phase of tertiary recovery. A tertiary recovery phase may entail gas or solvent injection into reservoirs. Solvent injection may greatly change the physical properties of the rock. Physical changes must be significant enough to be tractable from their seismic responses. In addition to proper data acquisition and processing, a time lapse study is feasible in shallow reservoirs with unconsolidated rocks with high porosity and permeability and low net pressure. Otherwise, it may be difficult to detect fluid exchange in the seismic data.

Re-gridding, phase and time shift correction, and match filtering (to match the static time shift, phase, and frequency content of two surveys) are the usual processing steps toward a

reliable 4D seismic study. We are looking for two seismic sections from the same area where any difference between time delays, amplitudes, impedance or any other attribute reflects only the changes in the physical properties in the reservoirs.

In this chapter, I use synthetic examples and real data to show how different acquisition geometries between baseline and monitor surveys lead to the different migration artifacts for the same model. Dead receivers in permanently planted geophone surveys will leave some artifacts in the migration of monitor surveys that are different from baseline survey artifacts. I show how LSPSM can attenuate these effects and provide comparable images.

5.2 Separate LSPSM inversion of time lapse data

Throughout this section, LSPSM is performed separately on both the baseline and monitor surveys, where no caveat is introduced. I show how LSPSM of time lapse data sets can help to remove the effect of different acquisition geometries or lost receivers, and provide reliable high resolution images as shown in the next two subsections. Joint inversion of time lapse seismic data is discussed in section 5.3.

5.2.1 Reducing the effect of different acquisition geometries

Ignoring the effect of different environmental noise, near surface effects, and processing procedures, a key point in comparing two seismic surveys is that both, baseline and monitor surveys, have identical or similar acquisition geometries. However, this is not always feasible. The baseline survey may be an old survey with a limited number of sources and receivers. Monitor surveys may use more modern equipment which allow better and denser data acquisition planning and gathering. There may be some new surface obstacles that prevent new data acquisition from matching the baseline geometry. Changes in acquisition geometries may be reduced by placing permanent receivers under the surface. However, some receivers may not function properly after a period of time. Lost geophones will leave new artifacts in the migration image of the monitor data.

In marine data acquisition, there is poor control on the positioning of the hydrophones due to streamer feathering. The effect of streamer feathering is larger for far offset receivers. The

ocean bottom cable (OBC) method of data acquisition can be used instead of a streamer; however, it is more expensive and has the same problem as permanently planted geophones.

Migration is always accompanied by acquisition footprint. The pattern of acquisition footprint depends on the acquisition geometry. Different acquisition geometries between old and new surveys leave different artifacts in the migrated images of the baseline and monitor surveys. Therefore, time lapse artifacts may dominate the changes in the model parameters.

Consider the baseline survey experiment as

$$\mathbf{d}_0 = \mathbf{G}_0 \mathbf{m}_0, \quad 5.1$$

where \mathbf{d}_0 , \mathbf{G}_0 , and \mathbf{m}_0 are the recorded data, forward modeling operator, and reflectivity for the baseline survey, respectively.

Assume the Earth's reflectivity changes from \mathbf{m}_0 to \mathbf{m}_1 after a period of time as:

$$\mathbf{m}_1 = \mathbf{m}_0 + \Delta\mathbf{m}, \quad 5.2$$

where \mathbf{m}_1 is the reflectivity at the time of the monitor surveying, and $\Delta\mathbf{m}$ is the difference in reflectivity between two data acquisitions. The monitor survey records data, \mathbf{d}_1 , which mathematically is expressed by

$$\mathbf{d}_1 = \mathbf{G}_1 \mathbf{m}_1, \quad 5.3$$

where \mathbf{G}_1 is the forward modeling operator of the monitor survey.

Migration of the two surveys gives:

$$\widehat{\mathbf{m}}_0 = \mathbf{G}_0^T \mathbf{d}_0, \quad 5.4$$

for the baseline survey and

$$\widehat{\mathbf{m}}_1 = \mathbf{G}_1^T \mathbf{d}_1, \quad 5.5$$

for the monitor survey, where $\widehat{\mathbf{m}}_0$ and $\widehat{\mathbf{m}}_1$ are migration of baseline and monitor surveys, respectively.

Even when model is not changing, ie. $\mathbf{m}_1 = \mathbf{m}_0$, $\widehat{\mathbf{m}}_0$ and $\widehat{\mathbf{m}}_1$ will be different due to different acquisition parameters between \mathbf{G}_0 and \mathbf{G}_1 .

When two acquisition geometries are similar, and change in the velocity of the modeling operators is negligible, $\mathbf{G}_0 \sim \mathbf{G}_1$, and we may write:

$$\widehat{\mathbf{m}}_1 = \mathbf{G}_0^T \mathbf{d}_1 = \mathbf{G}_0^T \mathbf{G}_0 \mathbf{m}_1 = \mathbf{G}_0^T \mathbf{G}_0 (\mathbf{m}_0 + \Delta\mathbf{m}) = \widehat{\mathbf{m}}_0 + \mathbf{G}_0^T \mathbf{G}_0 \Delta\mathbf{m}, \quad 5.6$$

or,

$$\widehat{\mathbf{m}}_1 - \widehat{\mathbf{m}}_0 = \mathbf{G}_0^T \mathbf{G}_0 \Delta \mathbf{m}, \quad 5.7$$

which states that the difference between the baseline and monitor migration images is proportional to the changes in the model parameters between the two surveys. However, since $\mathbf{G}_0^T \mathbf{G}_0$ is not a unitary operator, the difference in migration images is not exactly equal to the real changes in the reflectivity.

The migration images $\widehat{\mathbf{m}}_0$ and $\widehat{\mathbf{m}}_1$ must be cross-equalized to remove the effect of non-repeatability of data acquisition and migration artifacts before generating the time lapse image. A few cross-equalization methods have been proposed. For example, Rickett and Lumley (2001) suggested a cross-equalization flow, including two runs of a match filter application after regridding the data for amplitude, phase and bandwidth balancing.

LSPSM is an effective method to reduce the acquisition artifacts to make the final images comparable. Separate damped LSPSM of the baseline and monitor surveys,

$$\mathbf{m}_{DSL0} = (\mathbf{G}_0^T \mathbf{G}_0 + \mu_0^2 \mathbf{I}_0)^{-1} \mathbf{G}_0^T \mathbf{d}_0, \quad 5.8$$

and

$$\mathbf{m}_{DSL1} = (\mathbf{G}_1^T \mathbf{G}_1 + \mu_1^2 \mathbf{I}_1)^{-1} \mathbf{G}_1^T \mathbf{d}_1, \quad 5.9$$

provides images that are less affected by the corresponding acquisition geometries. Therefore, they represent the changes in the reflectivity model better than the migration images. This implies that, when $\mathbf{m}_1 = \mathbf{m}_0$, then

$$\mathbf{m}_{DSL0} \sim \mathbf{m}_{DSL1}, \quad 5.10$$

even when $\mathbf{G}_0 \neq \mathbf{G}_1$.

In addition to this, the ability of data reconstruction by LSPSM provides another reliable domain for the comparison between two surveys, the data domain. Data sets reconstructed from two surveys into a new geometry make the prestack time lapse studies more feasible and reliable.

In this subsection, I compare the differences between migration and LSPSM images from the same model but with different acquisition geometries for the baseline and the monitor surveys. I follow the methodology in equations 5.8 and 5.9 to compute the LSPSM images of the baseline and monitor surveys. Also I show the reconstructed data from the two surveys and compare them with the original data. For these purposes, consider the velocity model in Chapter Two in Figure 5.1. I now compare the data generated using this model with two different

acquisition geometries as the baseline and monitor surveys, without any change in the model parameters.

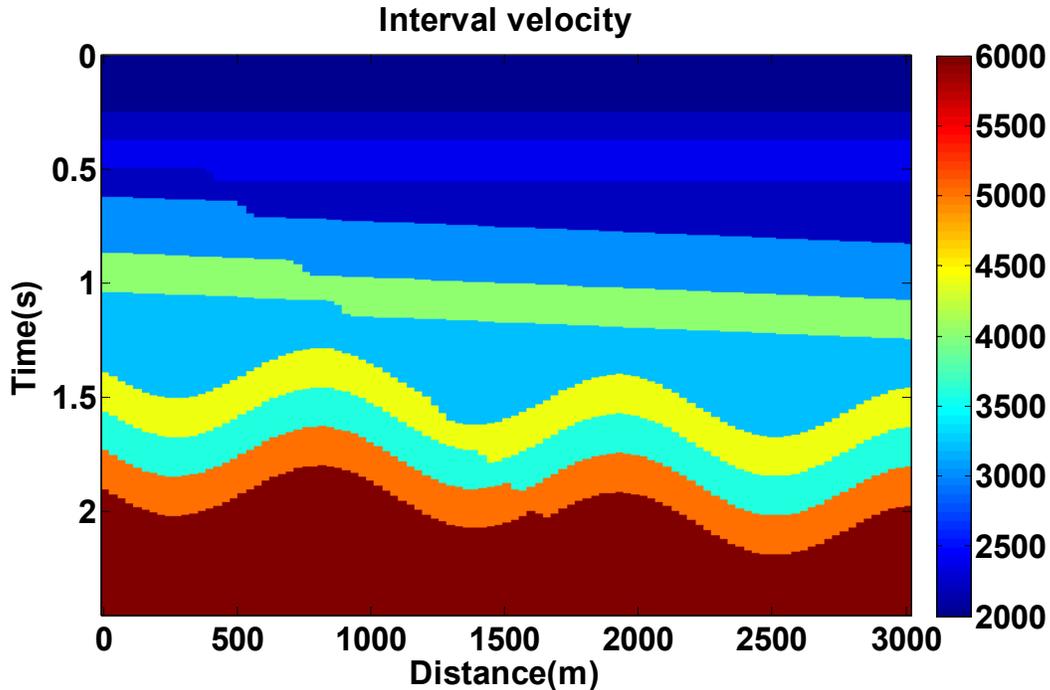


Figure 5.1 Velocity model used for forward modeling and data generation.

5.2.1.1 Case I: both surveys have dense data sampling

Consider the baseline survey has the acquisition geometry shown in Figure 5.2a. There are 32 sources and 100 receivers per source (3200 traces). Source spacing is 93.75 m and receiver spacing is chosen to be 18.75 m in order to have a fold of 10. Acquisition geometry of the monitor survey is shown in Figure 5.2b, and includes 20 sources and 200 receivers per source. The source spacing is 150 m and receiver spacing is 15 m to keep the fold at 10. Baseline and monitor synthetic data are generated with these geometries and 1% random noise is added to both data sets. The source wavelet and the other parameters are equivalent for both experiments. These geometries are significantly different; however, due to dense sampling, they should produce accurate migration images. Consequently the difference between the two migration images should be negligible. In the following subsections, I show that this is not necessarily true. Even dense data sampling produces a type of acquisition footprint which may be different from

the acquisition footprint of a subsequent dense survey of the same area. It is then shown how LSPSM can improve the time lapse imaging.

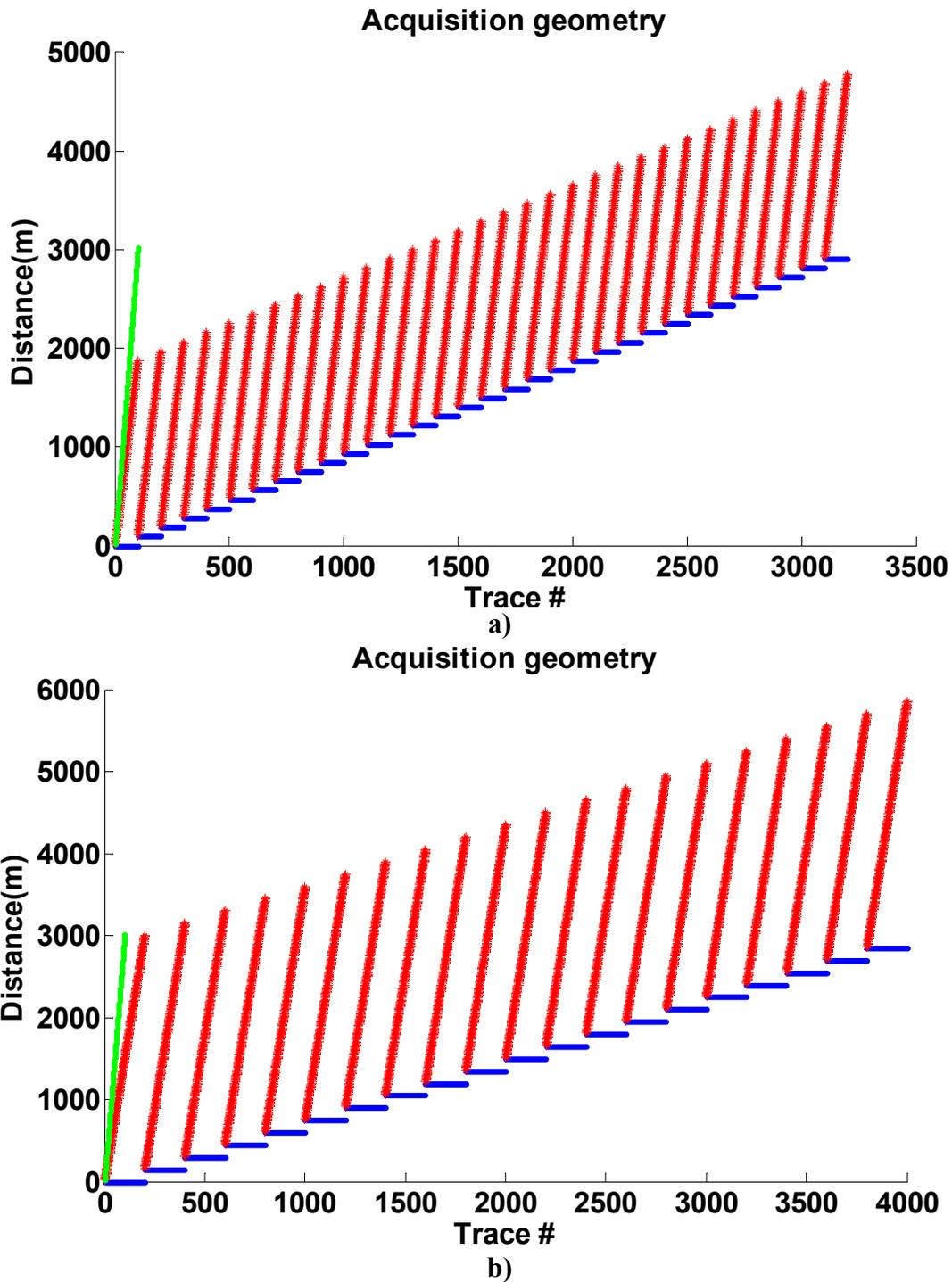


Figure 5.2 Acquisition geometry for a) baseline, and b) monitor surveys Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.

5.2.1.1.1 Comparing migration and LSPSM images for poststack time lapse studies

Migration images from the baseline and monitor surveys are shown in Figure 5.3. All migration parameters are identical for both surveys. The exact velocity model is used in the migrations and both images have good quality. To perform a comparison between these two migrations, the monitor image is subtracted from the baseline image. The difference image is shown in Figure 5.4 with the same amplitude scale. The difference should be zero or the difference between random noise of the two surveys. However, due to the presence of different acquisition footprint, the difference image shows some changes in the model which are not due to the physical properties of the model.

Figure 5.5 shows the LSPSM images for both, baseline and monitor surveys. Both data sets produced high resolution images. Figure 5.6 shows the difference between two LSPSM images. Comparing Figure 5.4 with Figure 5.6 shows that the LSPSM method has significantly reduced the acquisition footprint and returned images that are more reliable than the migration images for a time lapse study. It is very important in any time lapse study that both surveys be minimally affected by surveying effects.

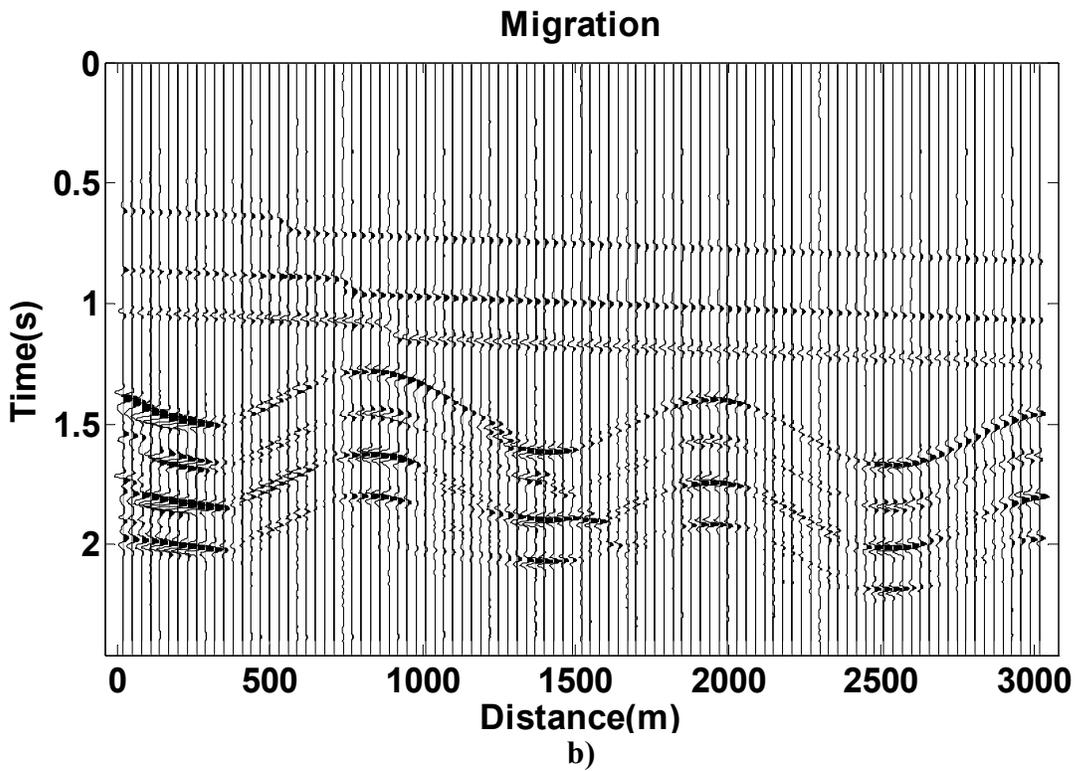
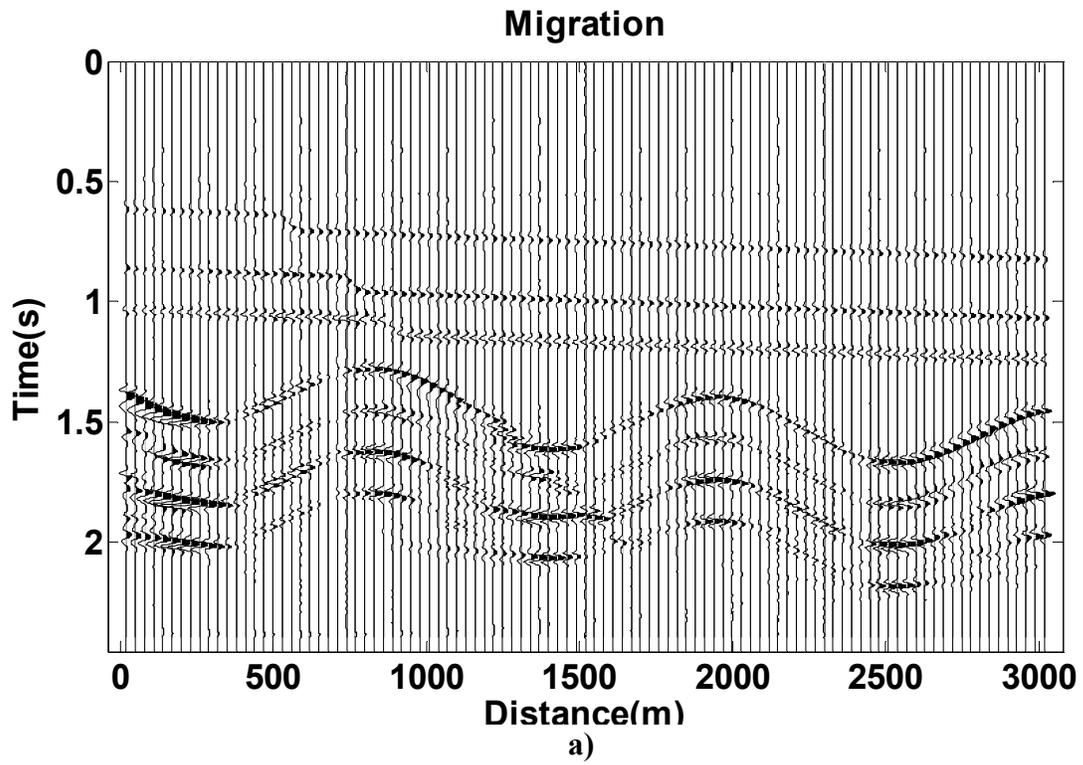


Figure 5.3 Migration of data set from a) baseline, and b) monitor survey.

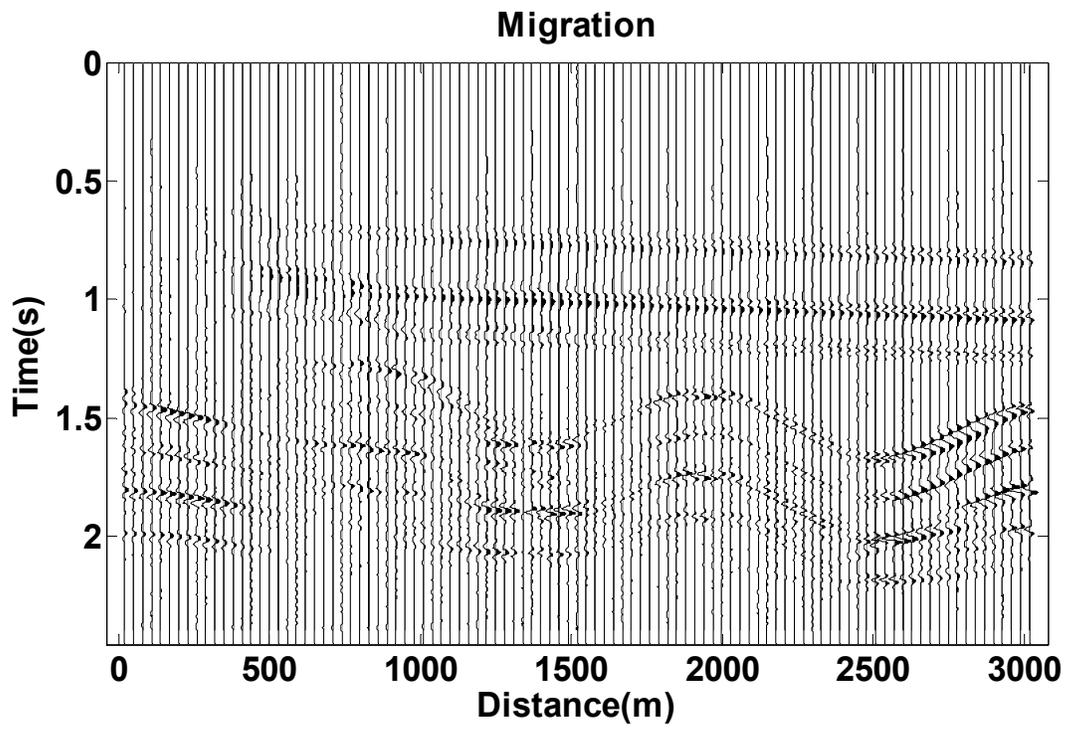


Figure 5.4 The difference between two migration images in Figure 5.3

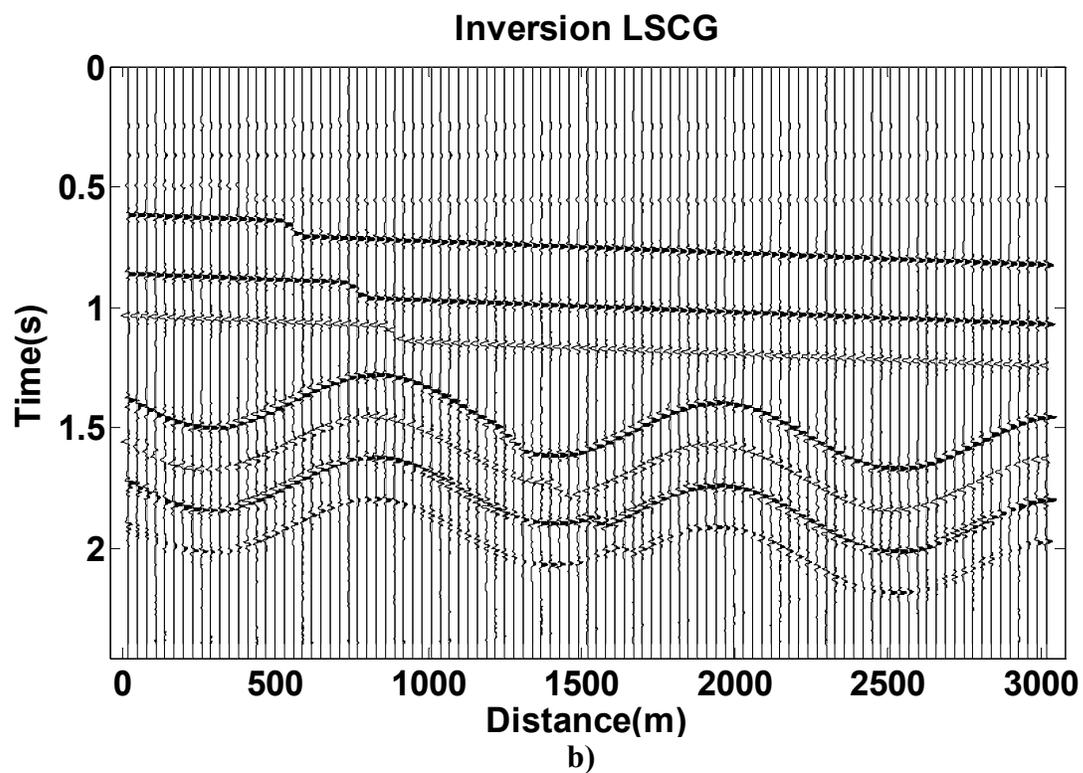
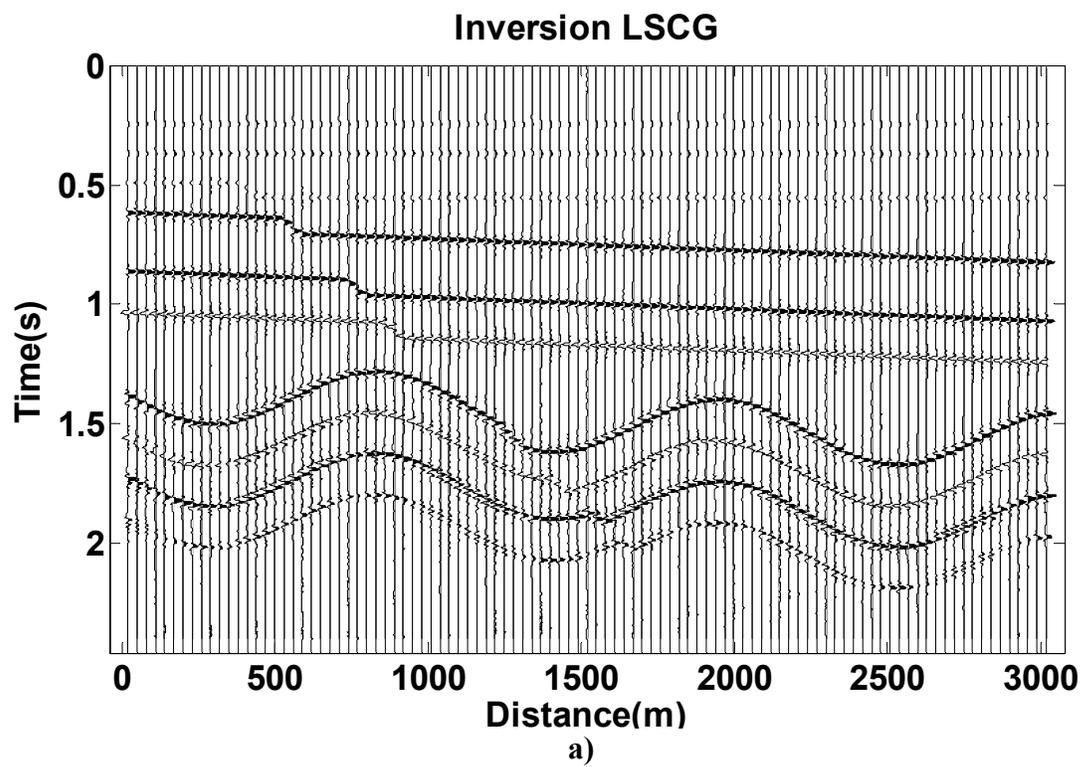


Figure 5.5 LSPSM images of the data set from a) baseline, and b) monitor surveys.

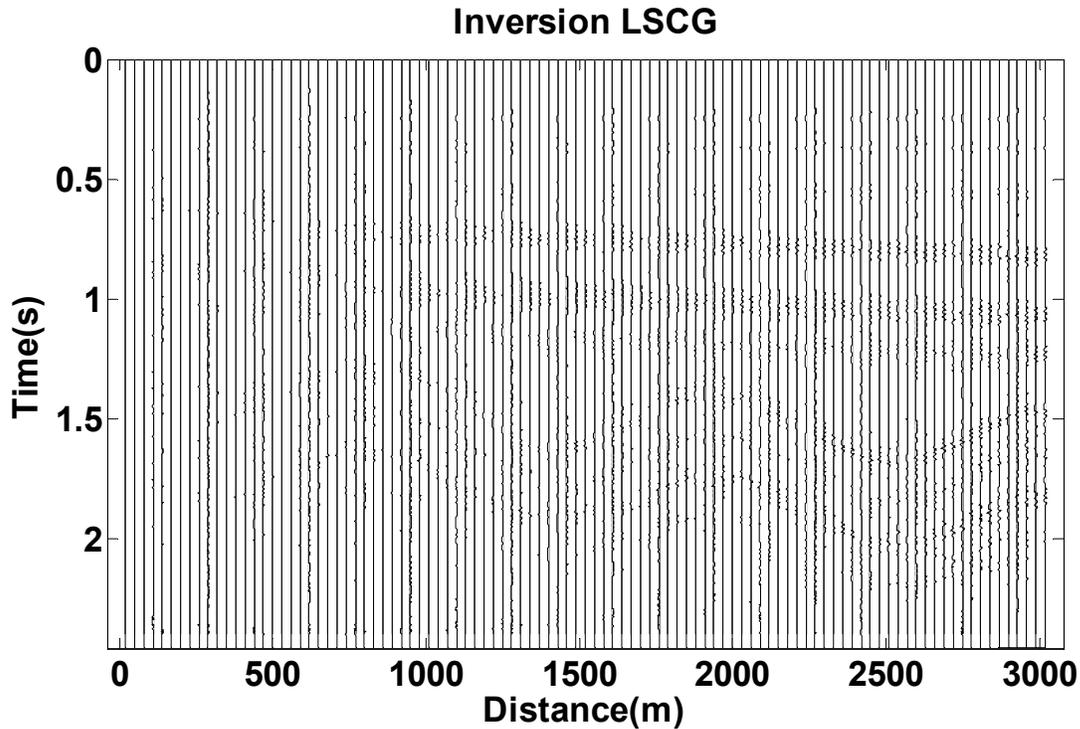


Figure 5.6 The difference between the two LSPSM images in Figure 5.5.

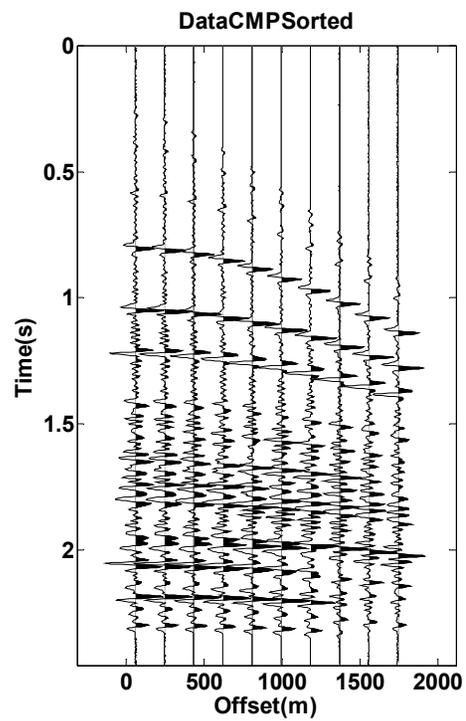
5.2.1.1.2 Comparing data and reconstructed data for prestack time lapse studies

Prestack time lapse studies provide very useful information about fluid flow using AVO methods. Noise and other undesired events such as multiples may get attenuated by stacking or migration, but they remain in the prestack data. In addition, prestack data are more affected by the different acquisition geometries of the baseline and monitor surveys. Therefore, prestack time lapse studies require more attention to detail when considering time lapse differences. Two surveys may have different offset spacings for the same CMP gather. Data reconstruction by LSPSM can provide two new baseline and monitor data sets for a given geometry in which CMPs have equivalent numbers of traces and offsets. A given geometry can be either a baseline or a monitor acquisition geometry or a new designed survey. Consequently, data reconstructed from two different surveys are made comparable.

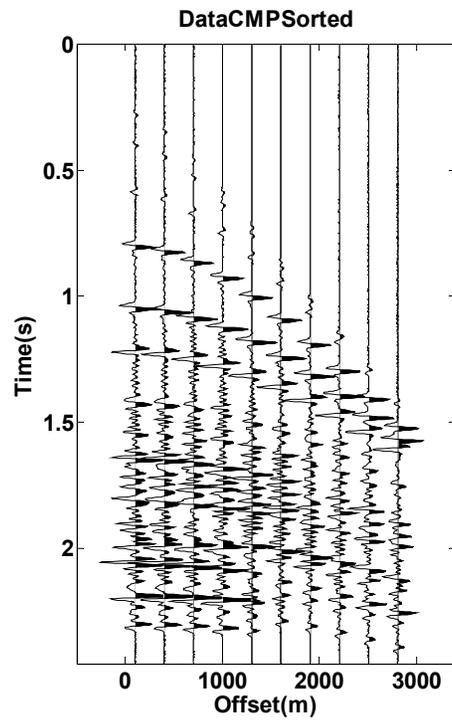
For instance, consider a CMP point at 2000 m in the previous example. Figure 5.7 show the CMP gather from baseline (a) and monitor (b) surveys. Both CMP gathers have 10 traces. However, a comparison is not feasible because of the differences in the traces' offsets.

Subtracting the two CMP gathers results in the CMP in panel (c) in Figure 5.7, which is not a desirable difference.

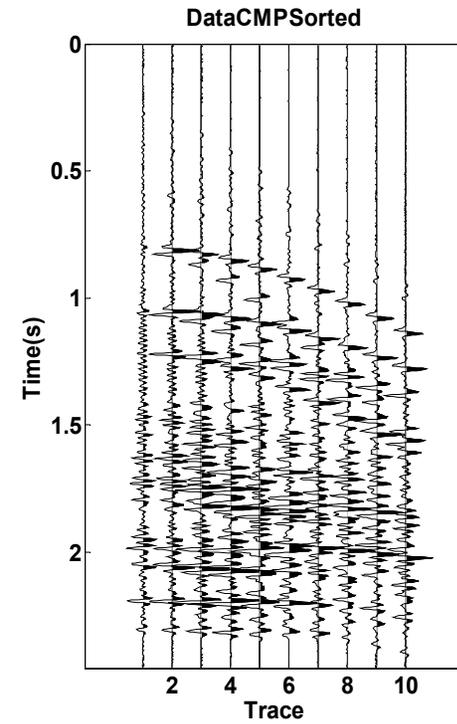
Using the LSPSM method with the geometry of the baseline survey, data from both surveys are reconstructed. Figure 5.8 shows the reconstructed data from baseline survey, monitor survey, and the difference for the CMP at 2000 m. Since there is no change in the model parameters, two reconstructed CMP gathers are similar, and the difference panel shows very low energy residuals. Therefore, data reconstruction of both, baseline and monitor surveys, makes the prestack time lapse study feasible. This is very useful in AVO inversion studies, before and after fluid or steam injection, to track the movements of a fluid/gas boundary or changes in the reservoir temperature due to steam injection.



a)



b)



c)

Figure 5.7 CMP gathers from a) baseline and b) monitor surveys and c) their difference.

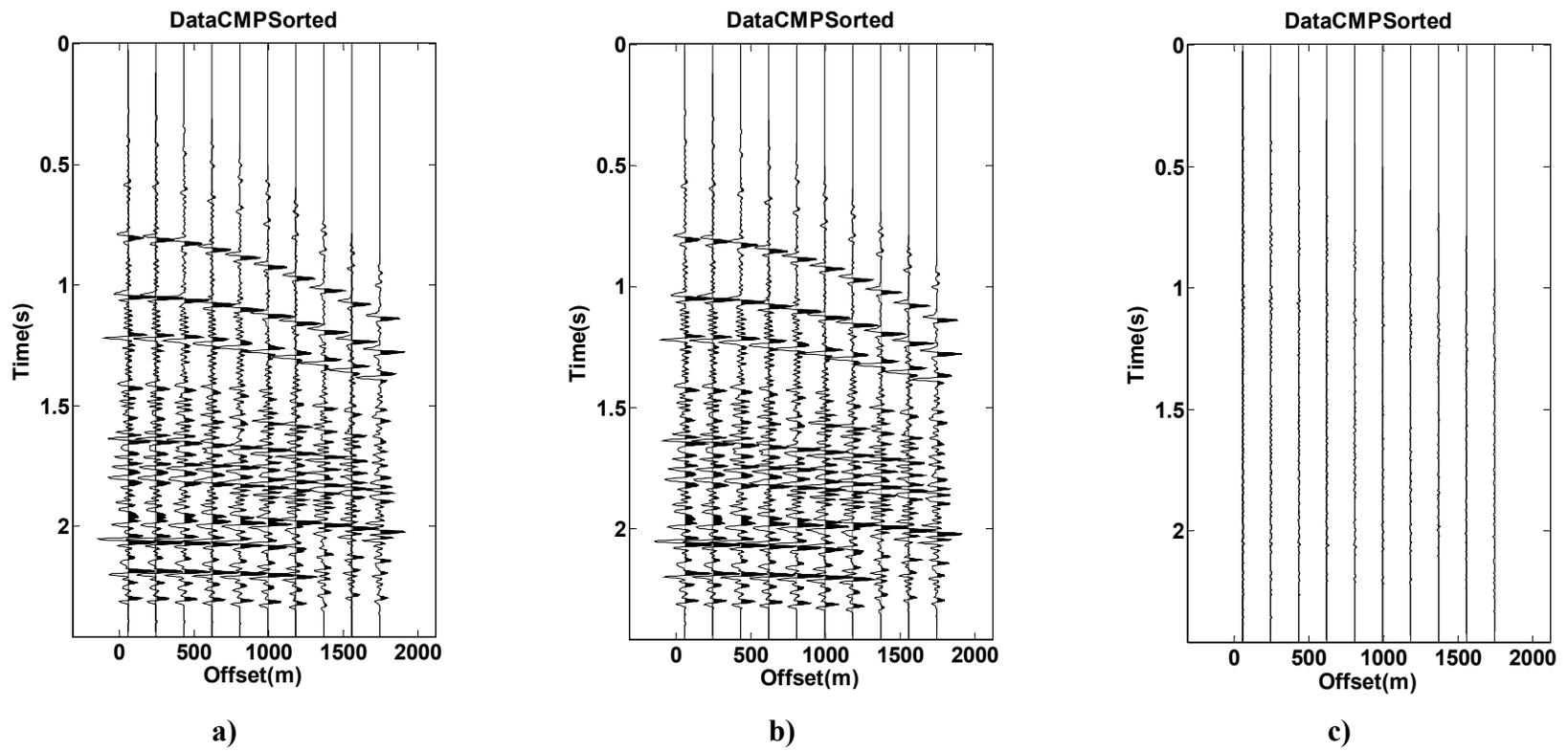


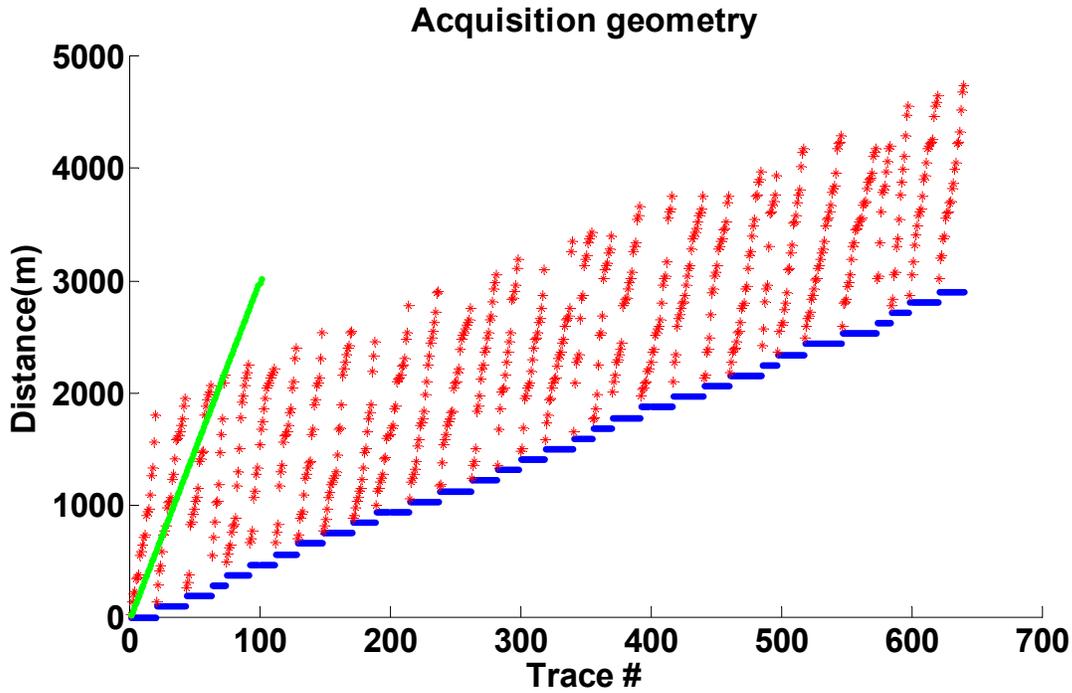
Figure 5.8 Reconstructed CMP gathers from a) baseline and b) monitor surveys and c) their difference.

5.2.1.2 Case II: both surveys are 80% decimated

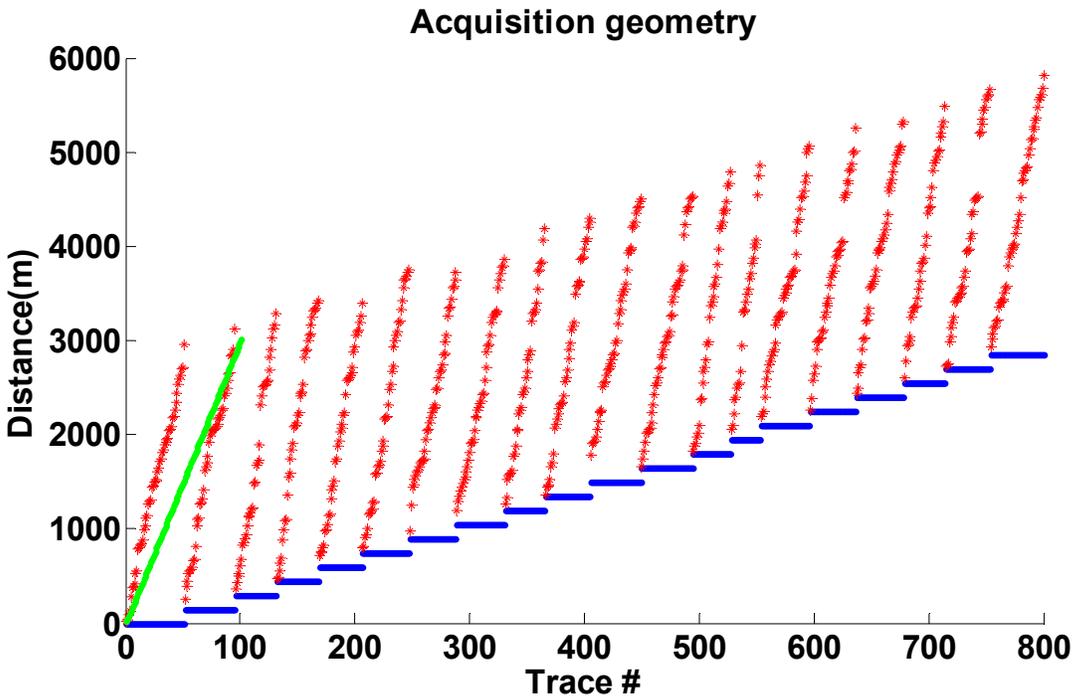
In the example of the previous section, both surveys had very dense and regular data sampling. With coarse and irregular data sampling, migration images will have more acquisition artifacts, and time lapse studies become unreliable. Prestack data of the two surveys will also have very irregular offset spacing in the CMP gathers. In such cases, implementation of an imaging method which is less affected by irregular sampling is very important. I show the ability of LSPSM to make the time lapse study of very irregularly sampled data possible.

Consider the same baseline and monitor surveys as in the previous section. I randomly removed 80% of traces from both surveys. Figure 5.9 show the resulting acquisition geometries. The decimated data from the two surveys are migrated. Figure 5.10a shows the difference between the two migration images. Acquisition footprint left many artifacts in this image which makes a poststack time lapse study unreliable. However, most of these artifacts are removed by the LSPSM method as seen in Figure 5.10b which shows the difference between LSPSM images from baseline and monitor survey data sets.

A prestack time lapse study for this experiment is almost impossible due to the low and random numbers of traces in each CMP gather. One solution is to merge adjacent CMP gathers to a supergather. Figure 5.11 show two CMP supergathers for the baseline and monitor surveys. Five adjacent CMPs are combined to create these gathers. Due to irregularities in the offset position of traces, the comparison between the two supergathers is still not practical. Figure 5.12 shows the reconstructed data which are CMP sorted. LSPSM is used for data reconstruction of decimated data from both surveys into the undecimated baseline survey. The residual in panel (c), the difference between two reconstructed CMPs, has very low energy signals. Therefore, any subtle changes in the physical properties of the reservoir rocks which is detectable by the seismic method, would be detectable in the LSPSM reconstructed data.

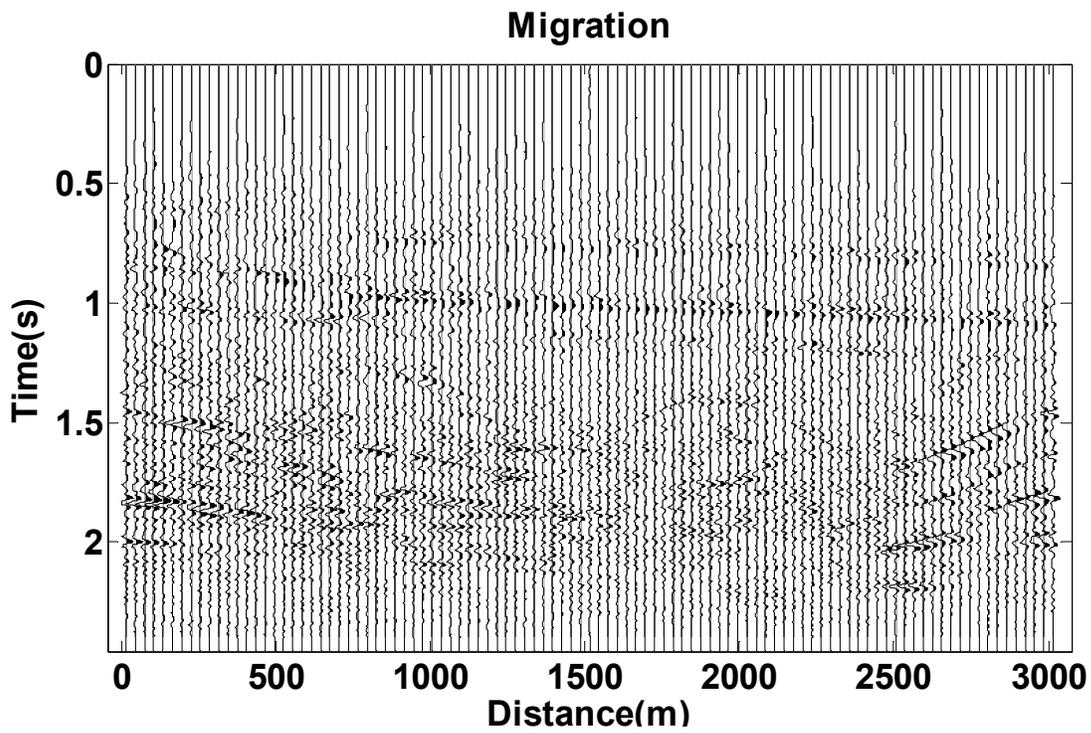


a)

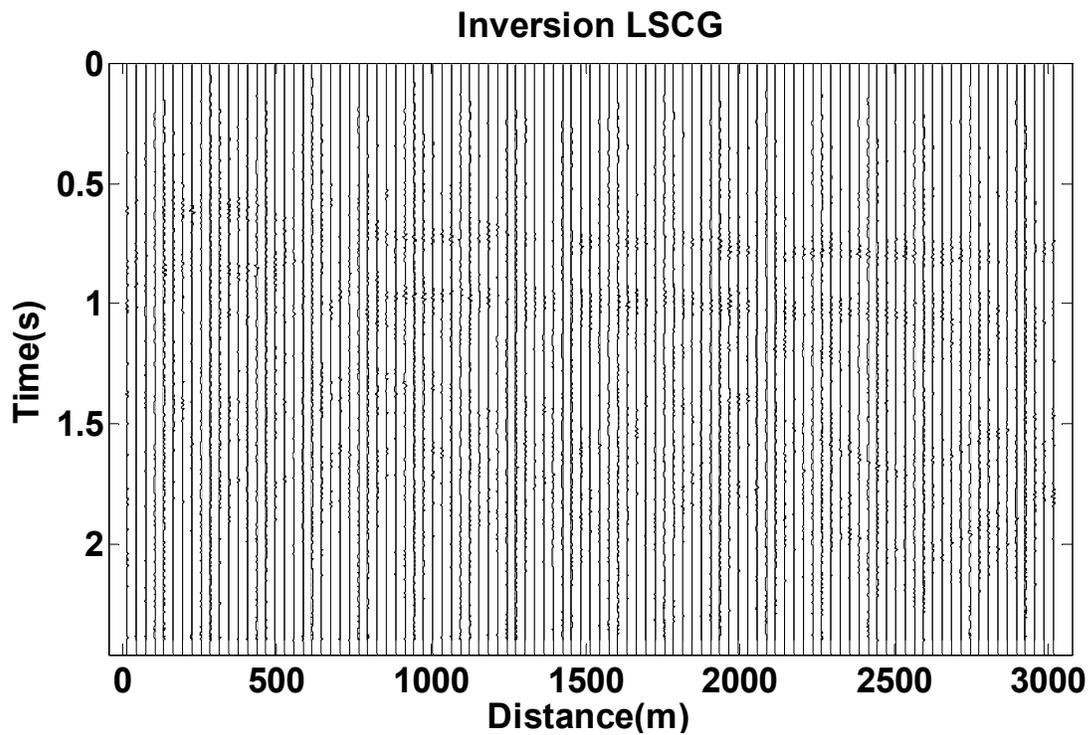


b)

Figure 5.9 Acquisition geometry for a) decimated baseline, and b) monitor surveys. Blue and red: sources and receivers positions for each seismic trace, positions of the image points are shown in green.



a)



b)

Figure 5.10 The difference between two a) migration, and b) LSPSM images.

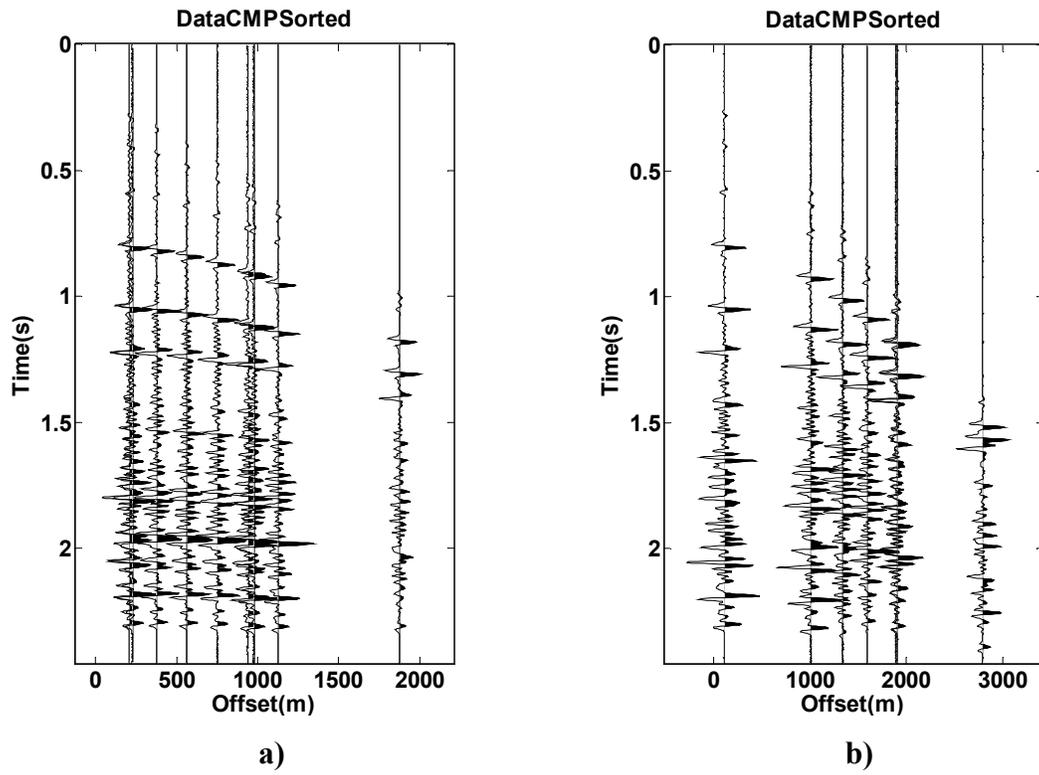
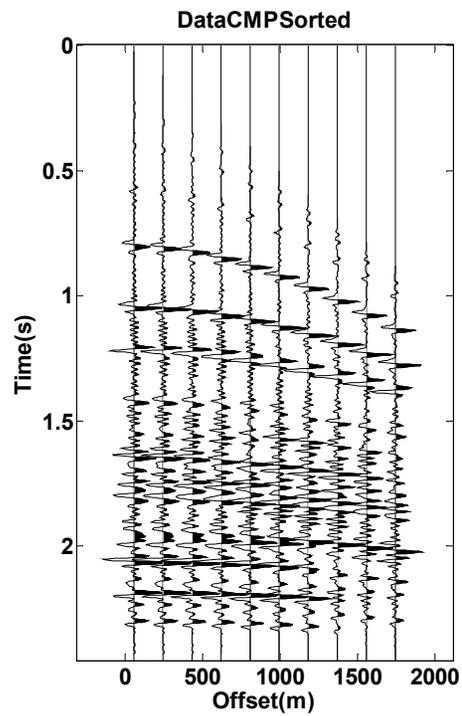
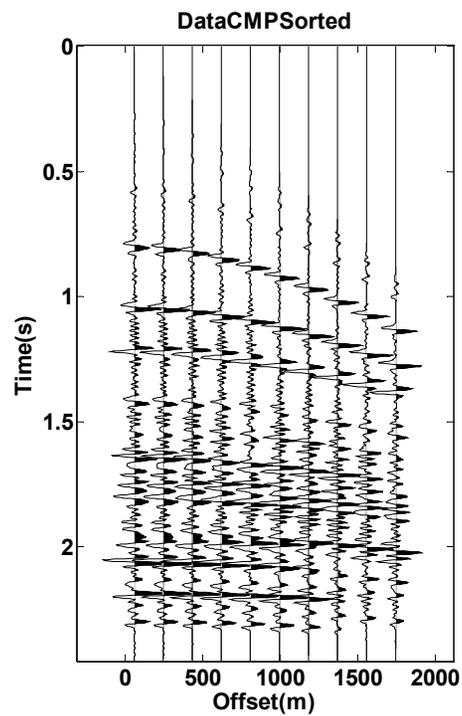


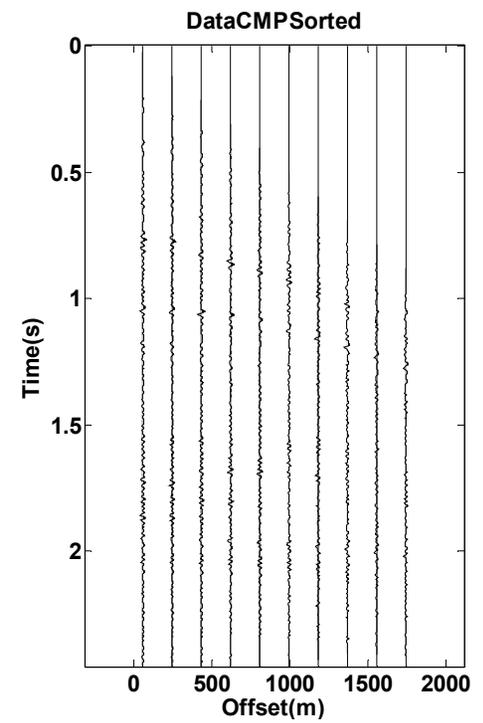
Figure 5.11 CMP-bin supergathers from a) baseline, and b) monitor surveys.



a)



b)



c)

Figure 5.12 Reconstructed CMP gathers from a) baseline, and b) monitor surveys, and c) their difference.

5.2.2 Reducing the effect of losing receivers

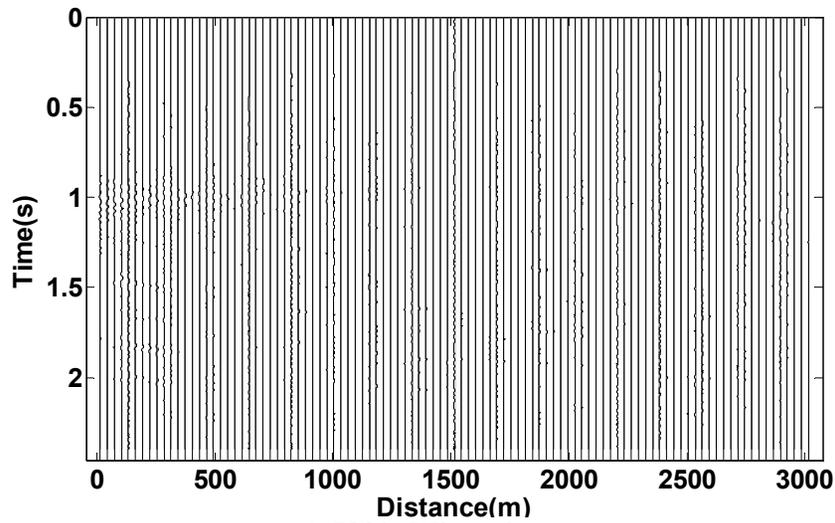
The ability of the LSPSM for reducing the effects of different acquisition geometries in the baseline and monitor surveys was shown in the previous sections. Attempts to mimic baseline acquisition geometry in the monitor surveys may be avoided by permanently planting receivers in the ground. In marine surveys, streamer acquisition may be replaced by OBC acquisition with the cables left at the bottom of the sea for future monitor data acquisitions. Replacing streamers with the OBC method is more beneficial since it makes the recording of shear waves possible. In both methods, planted geophones or OBC acquisition, the receiver geometry remains unchanged. Consequently, similar acquisition artifact patterns will be introduced into the migration images, and time lapse studies are feasible. Source locations must also be repeatable.

After a period of time some receivers may not function properly. Therefore, monitor surveys become affected by having a fewer number of receivers as the deployed receiver system becomes older. Losing receivers over time produces more artifacts in the monitor survey migration images. In order to have the same artifacts in the baseline survey, traces that correspond to the dead geophones must be removed from the baseline data. Alternatively, I show how this effect can be minimized by replacing migration with the LSPSM method, as separate LSPSM of the baseline and decimated data provides images that are less affected by the loss of receivers than a migration image.

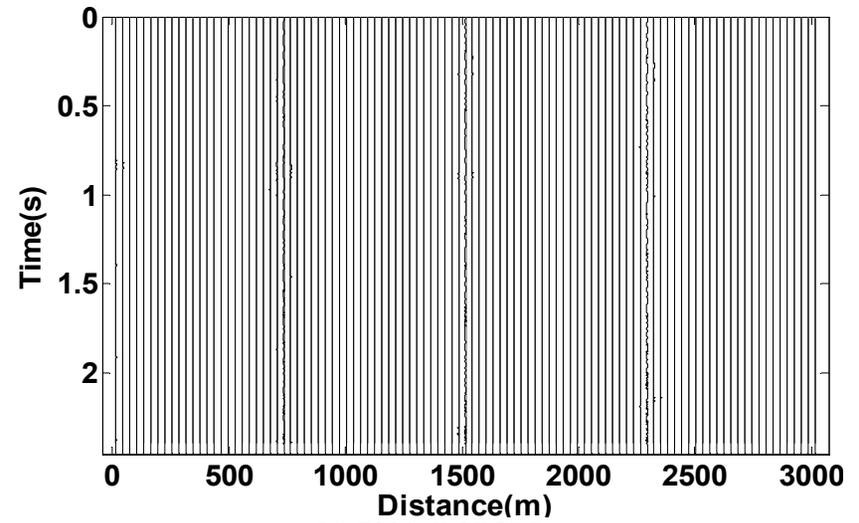
5.2.2.1 Comparing migration and LSPSM images

Consider the baseline survey in the previous section. I assume that six monitor seismic surveys are performed in this area with permanently planted receivers. Source positions are not changing from baseline to monitor survey. However, I assume that a large percentage of receivers stop functioning with each new monitor survey. The baseline survey has 3200 traces, however, only 50%, 30%, 20%, 10%, 5%, and 2% of traces are functioning in the future monitor surveys. The last three cases are very extreme and not realistic. Data from the monitor survey are migrated and LSPSM inverted. The left hand side panels of Figure 5.13 shows the difference between migration of baseline data and six monitor surveys while the right hand side panels show the difference between LSPSM image of baseline data and six monitor surveys. As shown, due to increasing acquisition footprint in the monitor surveys as a result of losing more receivers, the difference between migration images significantly increases. LSPSM images are less affected

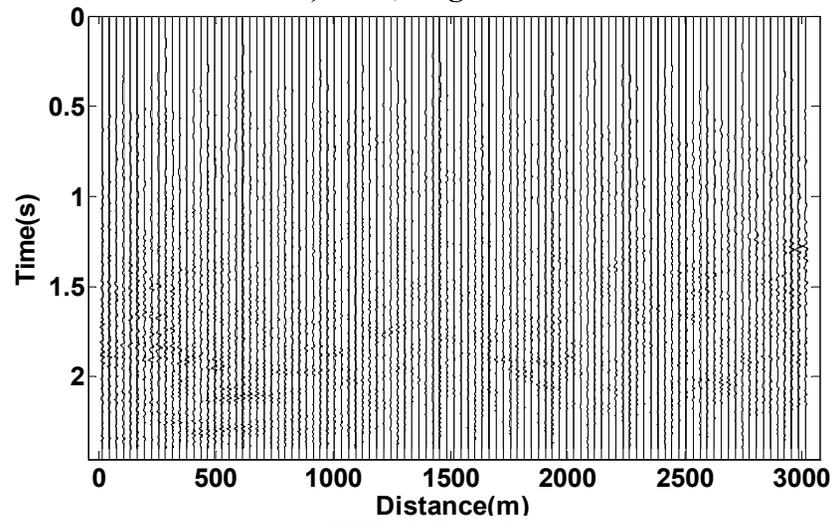
by the loss of receivers, and there is no noticeable difference between the baseline LSPSM image and the monitor LSPSM images. Since it can produce a high resolution time lapse image, using LSPSM to compensate for the difference in acquisition artifacts is a better choice than to ignore some receivers from baseline survey to make it similar to the monitor survey.



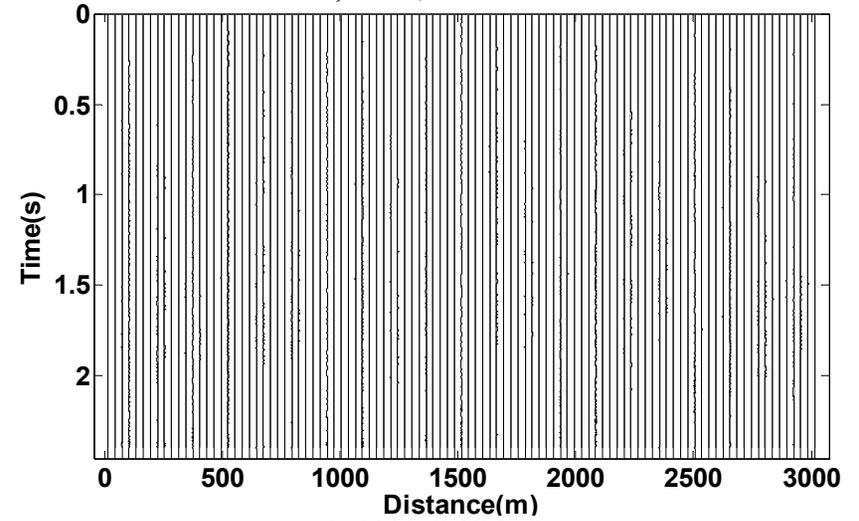
a) 50%, Migration



b) 50%, LSPSM

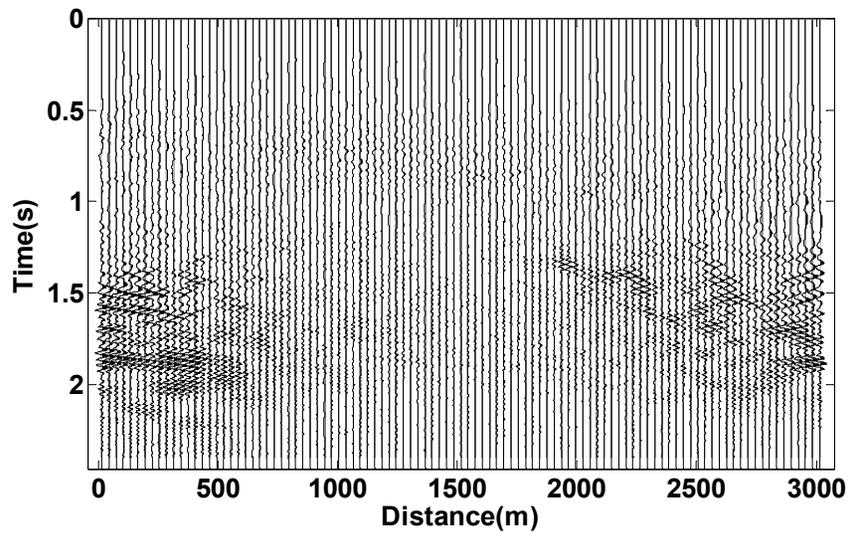


c) 30%, Migration

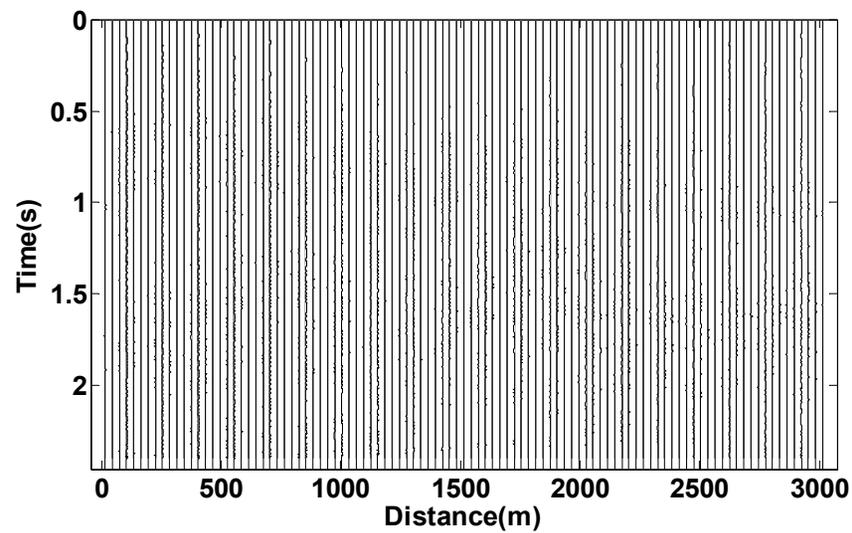


d) 30%, LSPSM

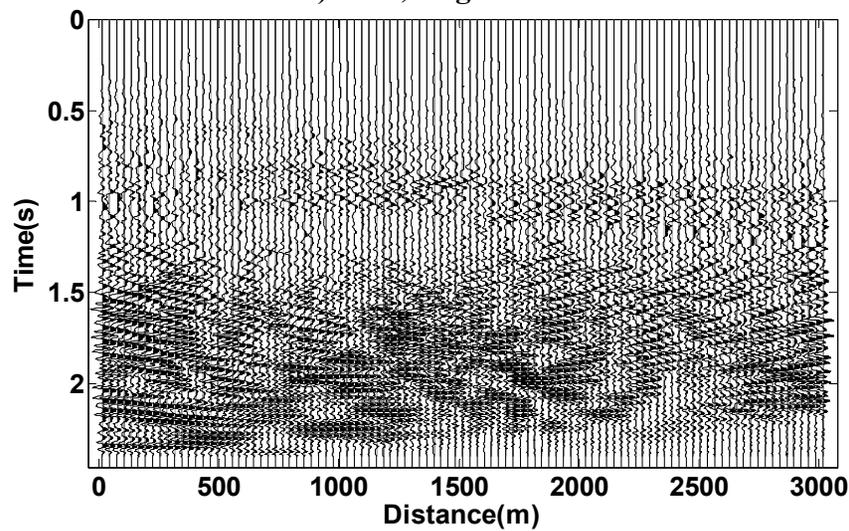
Figure 5.13 Migration difference (a and c) and LSPSM difference (b and d) with using 50% (a and b) and 30% (c and d) of data.



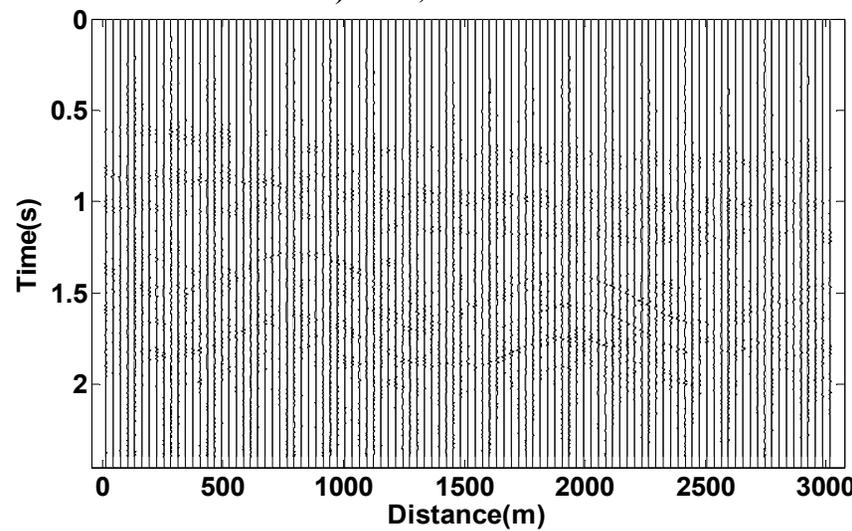
e) 20%, Migration



f) 20%, LSPSM

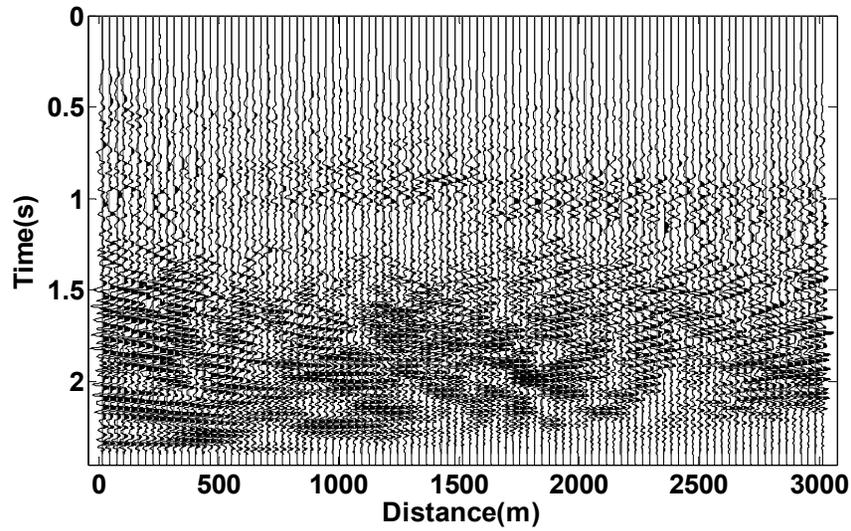


g) 10%, Migration

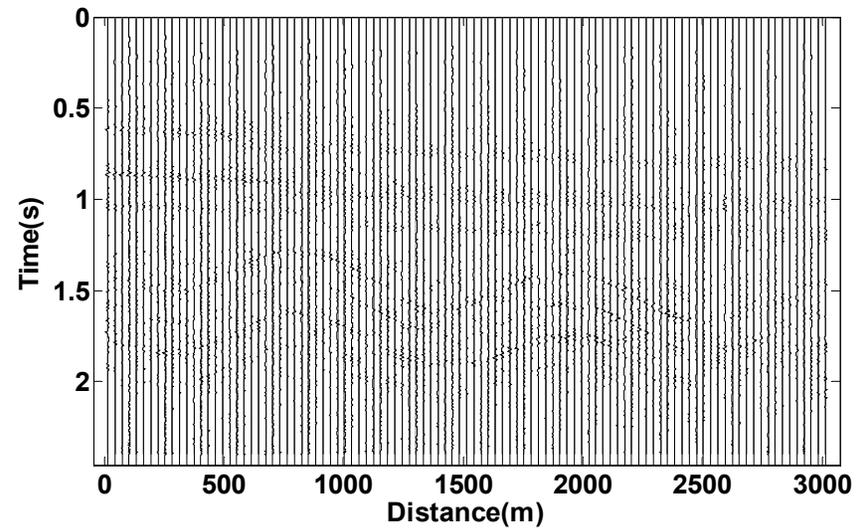


h) 10%, LSPSM

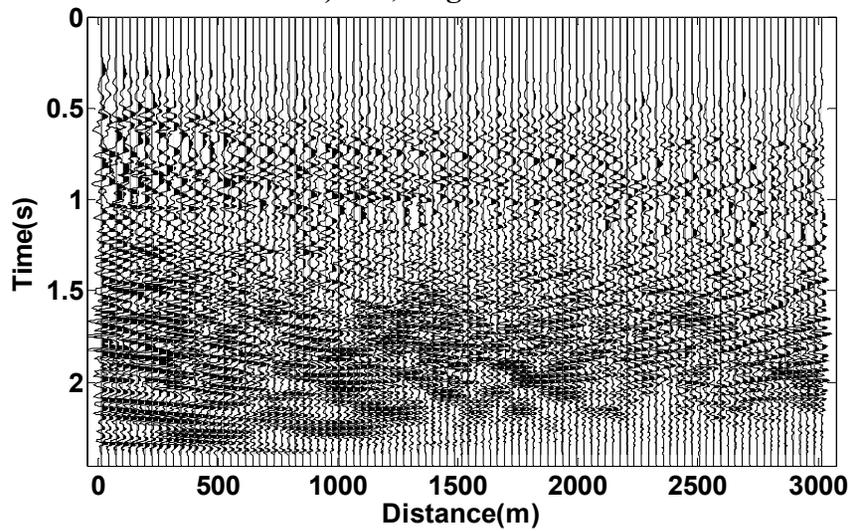
Figure 5.13 Continued: using 20% (e and f) and 10% (g and h) of data.



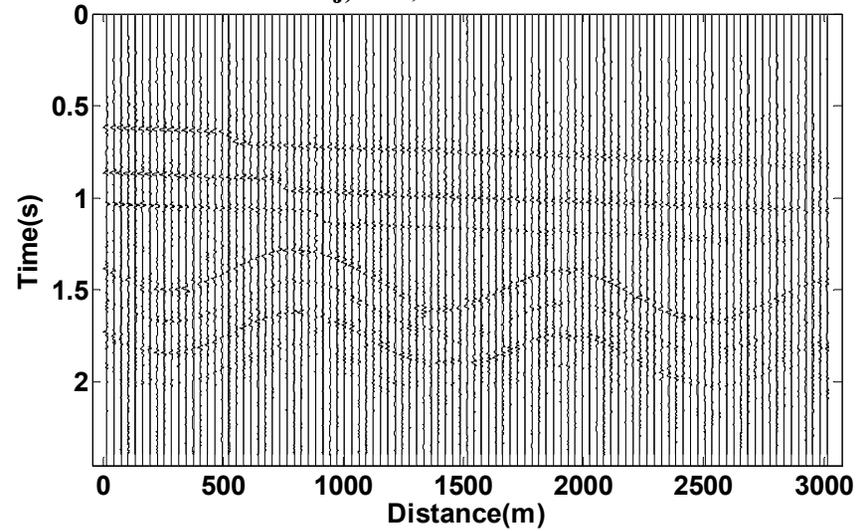
i) 5%, Migration



j) 5%, LSPSM



k) 2%, Migration



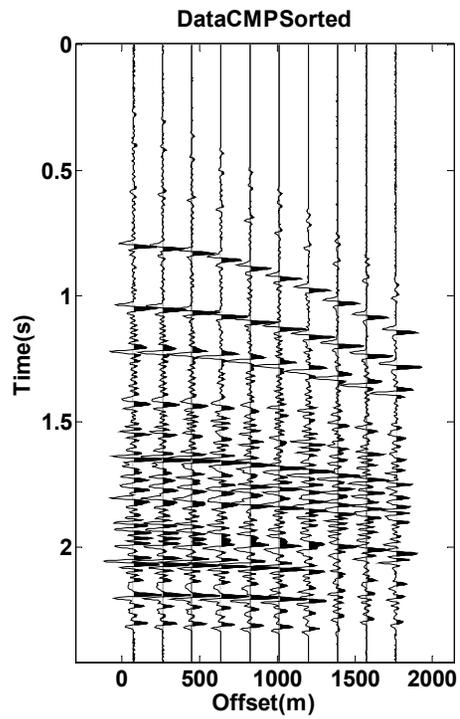
l) 2%, LSPSM

Figure 5.13 Continued: using 5% (i and j) and 2% (k and l) of data.

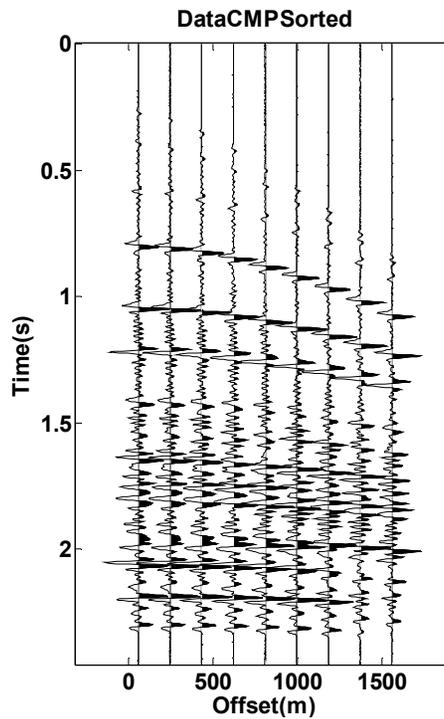
5.2.2.2 Comparing data and reconstructed data from two surveys

Figure 5.14 shows the effect of losing receivers in the monitor surveys on a CMP gather at the position of 2000 m. Losing more receivers in the newer surveys makes the comparison between prestack data sets more difficult or even impossible. Figure 5.15 shows the reconstructed CMP gather by LSPSM on the left and the difference between original and reconstructed data on the right, for different levels of decimation. The reconstruction after decimation is surprisingly successful, even for the extreme cases.

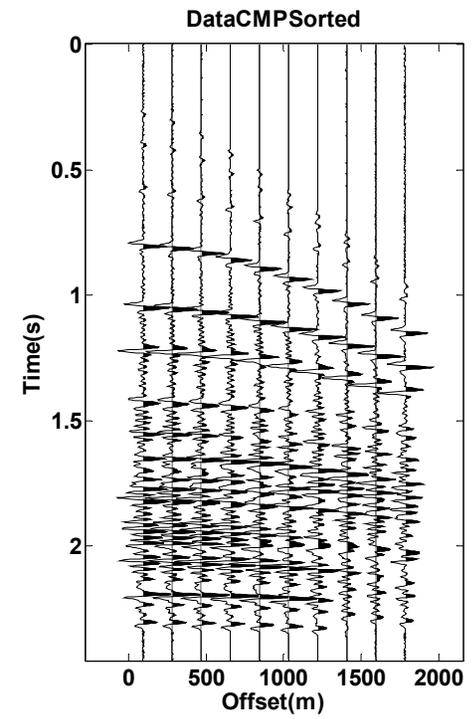
Therefore, the effect of losing receivers can be compensated by data reconstruction with LSPSM. This method helps keeping the fold at the same level as the baseline survey for comparing AVO effects of the baseline and monitor surveys.



a)

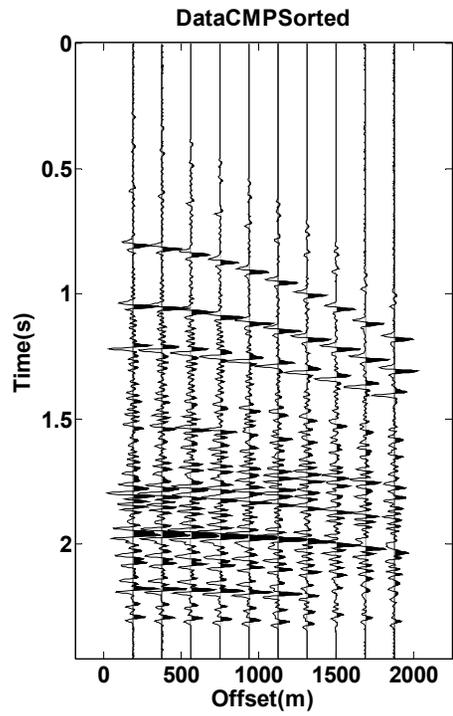


b)

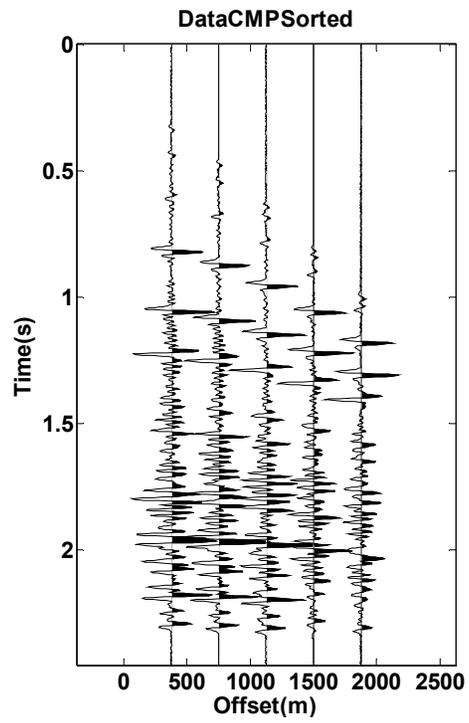


c)

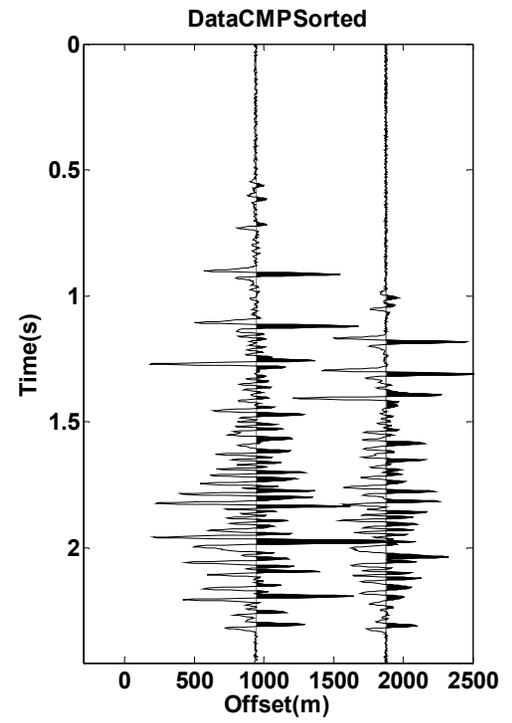
Figure 5.14 CMP gathers from monitor surveys after 50%, 30%, 20%, 10%, 5%, and 2% in panels a), b), c), d), e), and f) respectively.



d)

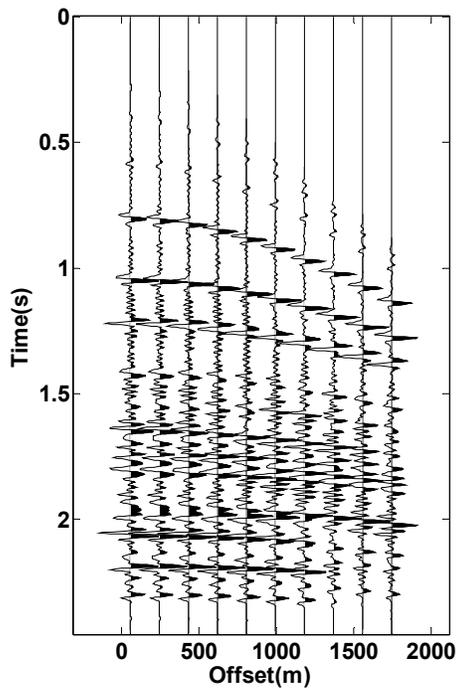


e)

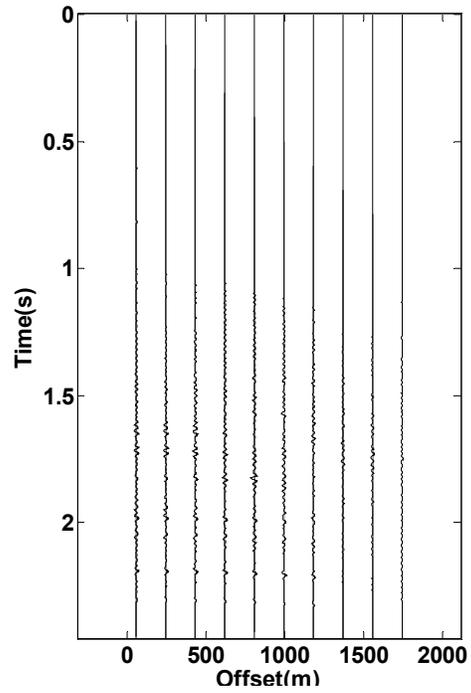


f)

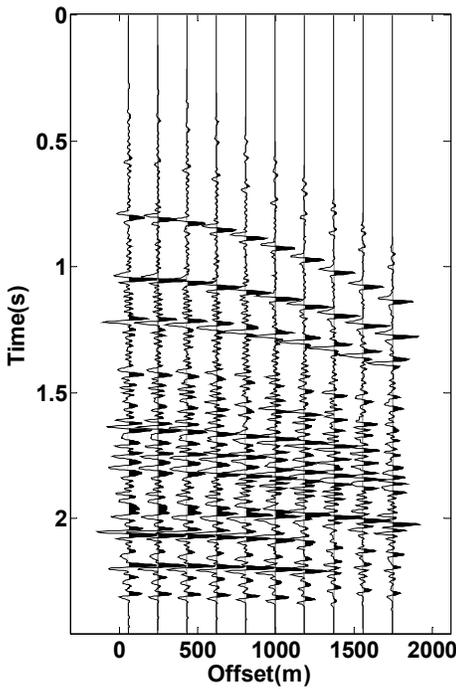
Figure 5.14 Continued.



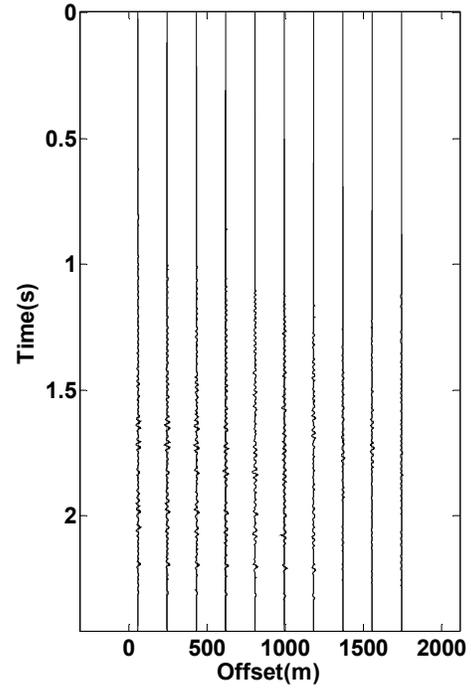
a) 50%



b) 50%

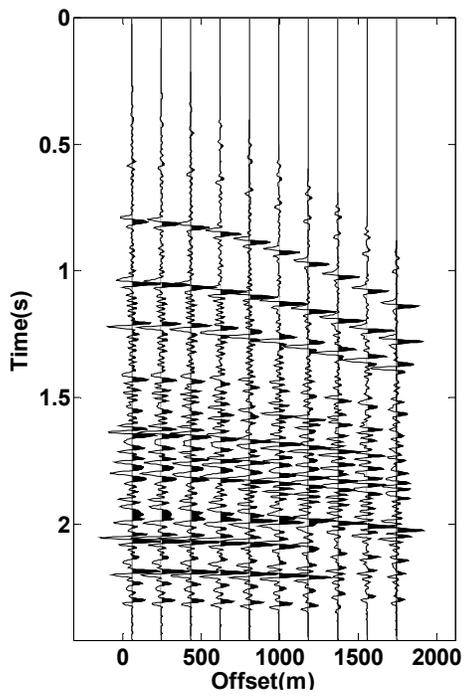


c) 30%

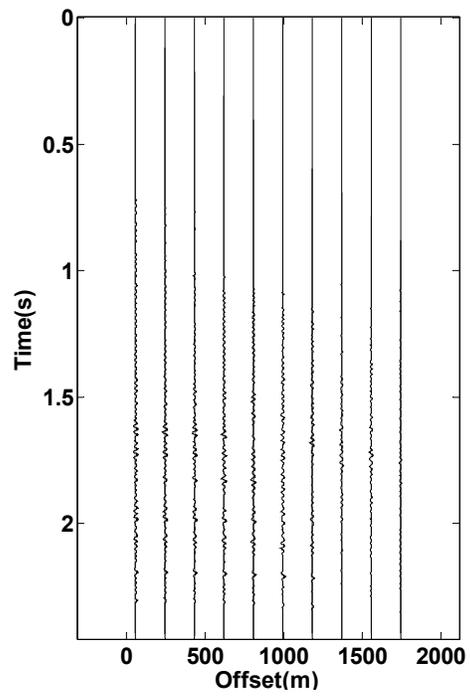


d) 30%

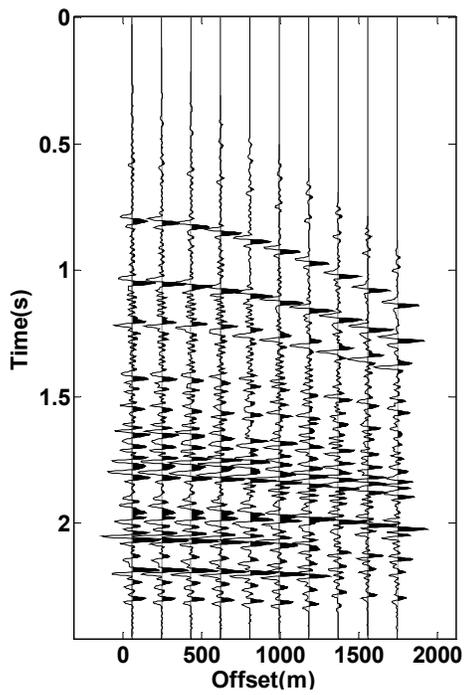
Figure 5.15 Reconstructed CMP gathers using LSPSM on the left and the difference from baseline data on the right, with (a,b) 50%, (c,d) 30%, (e,f) 20%, (g,h) 10%, (i,j) 5%, and (k,l) 2% of data.



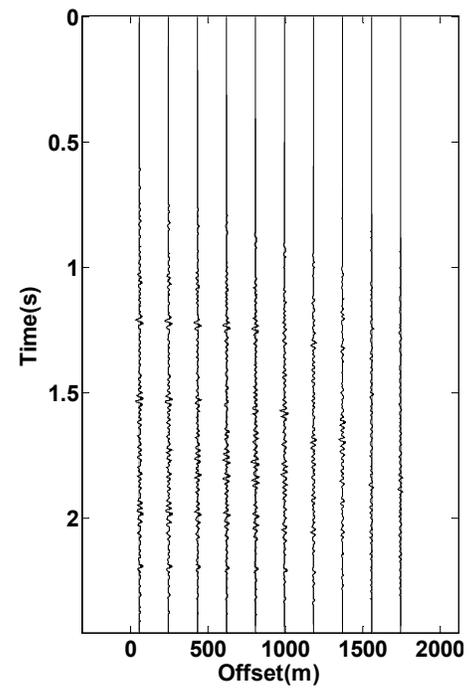
e) 20%



f) 20%

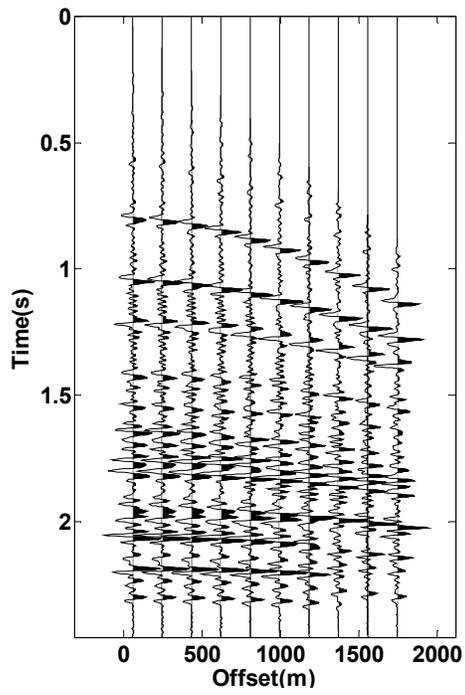


g) 10%

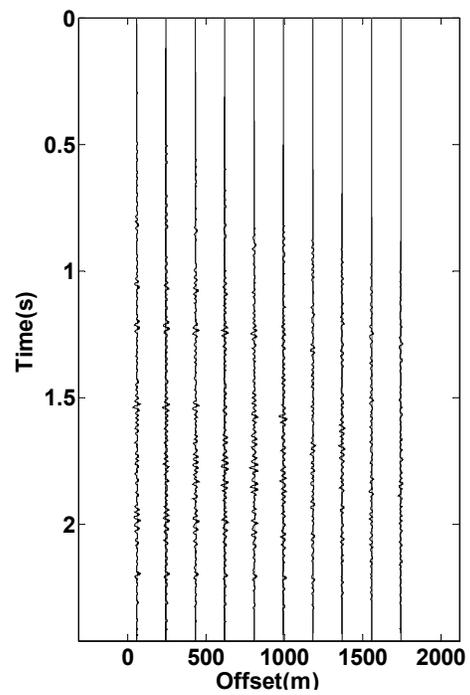


h) 10%

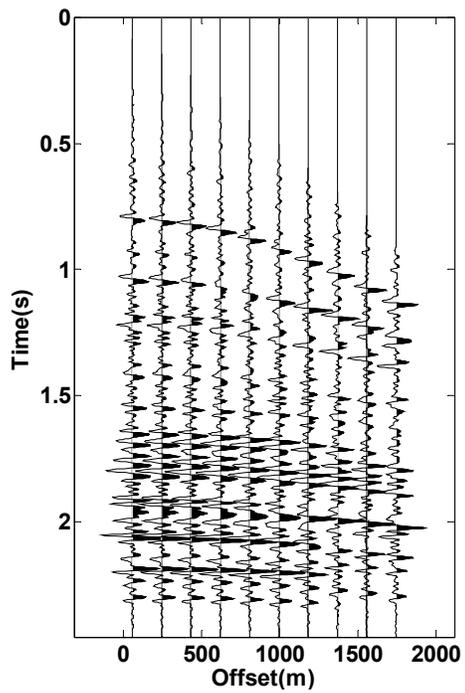
Figure 5.15 Continued.



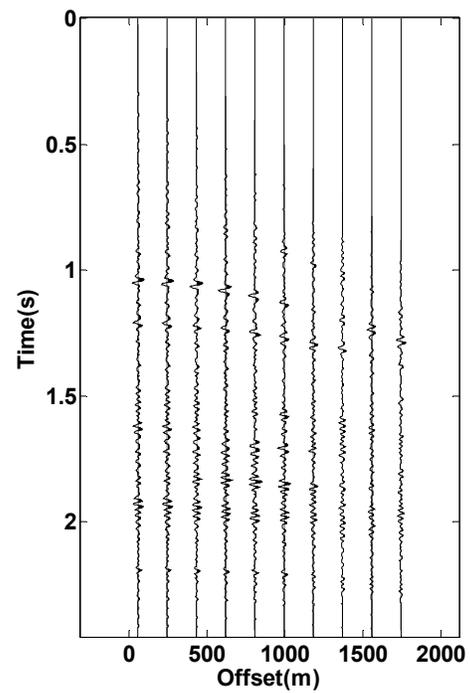
i) 5%



j) 5%



k) 2%



l) 2%

Figure 5.15 Continued.

5.2.3 High resolution detection of reflectivity changes

In this section I assume that the baseline and monitor surveys have similar geometries but the velocity, and consequently the reflectivity model, changes from baseline to monitor survey. I compare changes in the baseline and monitor migration and LSPSM images.

Consider that the baseline geometry at the beginning of this chapter applies to the monitor surveys, too. Assume that the velocity of the dipping layer at 1 s drops by 15% from 4000 m/s to 3400 m/s before the monitor survey. Figure 5.16 shows the new velocity model used for the generation of the synthetic monitor survey data.

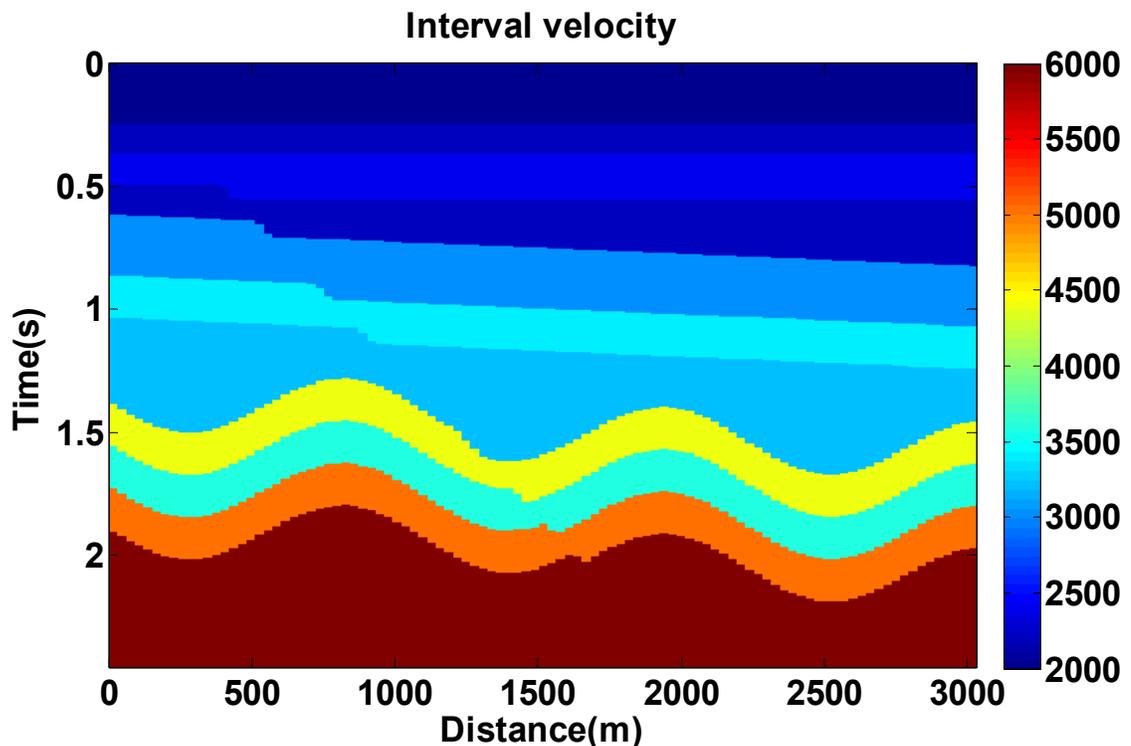


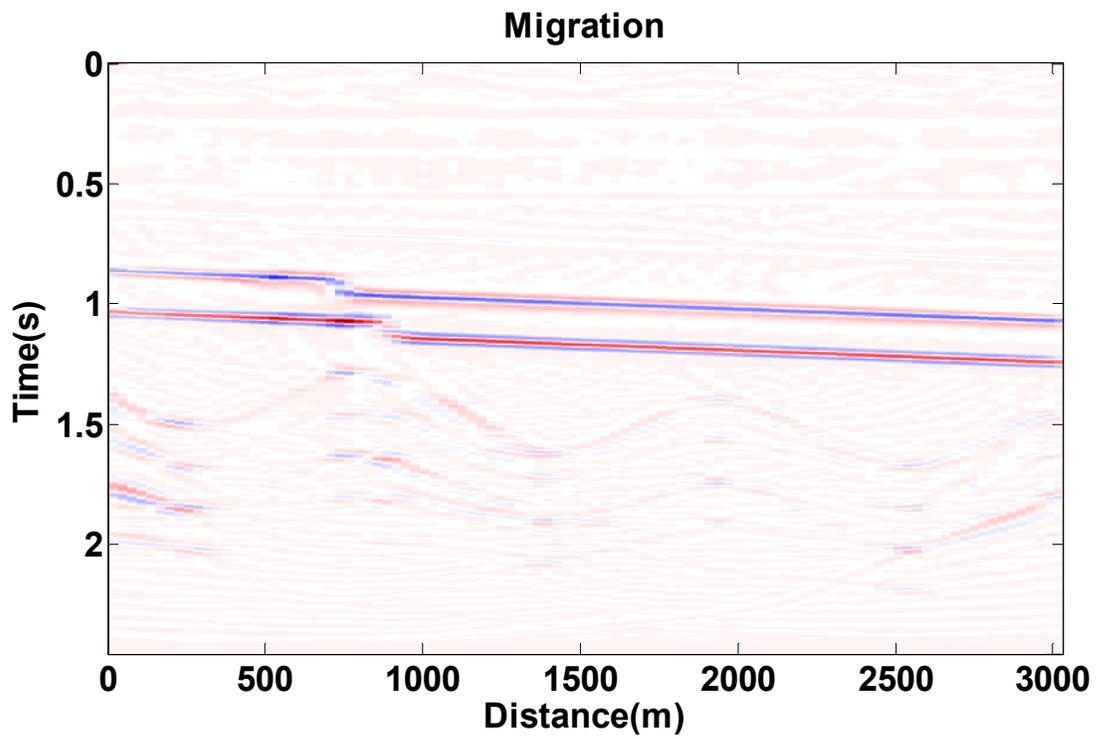
Figure 5.16 Velocity model used in the monitor survey.

Data are produced for baseline and monitor surveys with the corresponding velocity models. After adding 1% random noise, the data are migrated. Figure 5.17a shows the difference between migration images of the baseline and monitor survey. The resulting time lapse image shows the difference in reflectivity due to velocity change. Since the only change in the velocity model is the dipping layer at 1 s, all other events below 1.3 s are migration artifacts and not

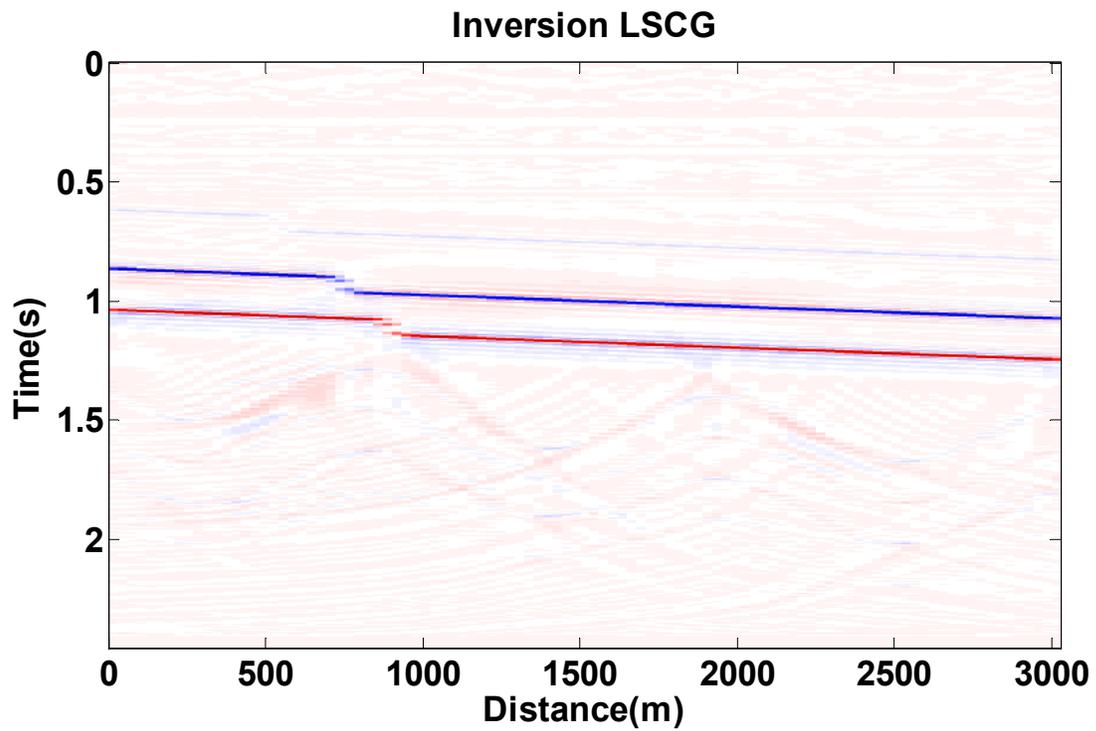
changes in the model parameters. Figure 5.17b shows the difference between LSPSM images. Comparison between the two images shows that the LSPSM time lapse image has a higher resolution than the migration time lapse image, and artifacts in the migration time lapse image are attenuated in the LSPSM image.

Figure 5.18 shows the baseline and monitor survey data for a CMP at 2000 m. Figure 5.18c shows the difference between baseline and monitor surveys. Events above a time of 1 s are cancelled. Due to the time shift below 1 s, the model parameters of events below the time of change are not comparable.

Figure 5.19 shows the reconstructed baseline data, monitor data, and their difference. LSPSM with 20 LSCG iterations is used for data reconstruction. However, the velocity model of the baseline survey is used in forward modeling and for the reconstruction of monitor survey data. The difference panel in Figure 5.19c can be considered as the difference between baseline and monitor data (Figure 5.18c) when the effect of the time shift is removed from data below the region of the velocity change.

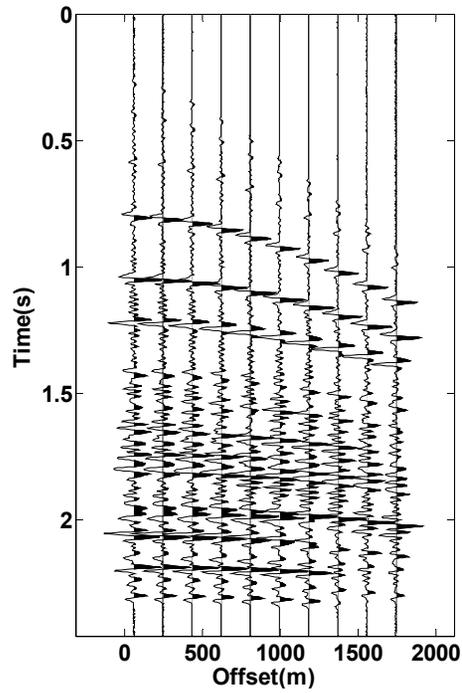


a)

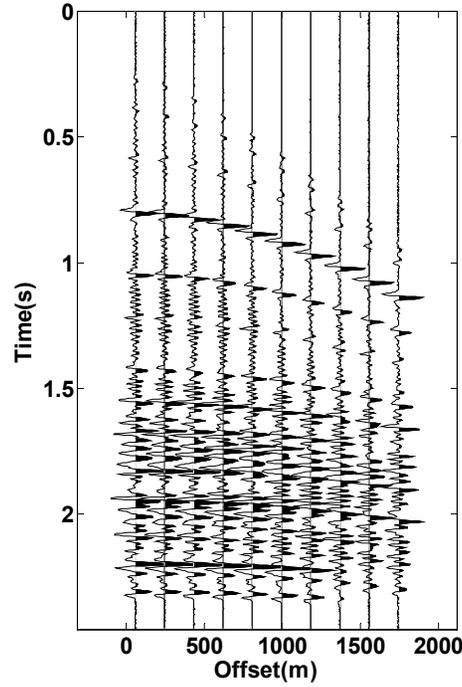


b)

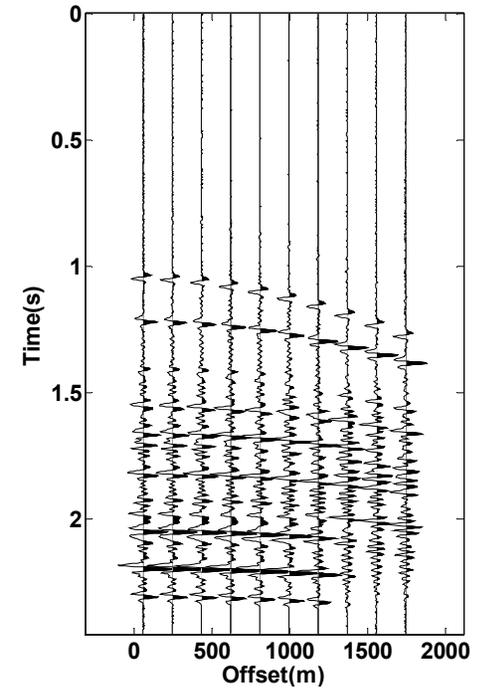
Figure 5.17 a) Migration and b) LSPSM time lapse images resulting from subtraction of baseline and monitor images.



a)

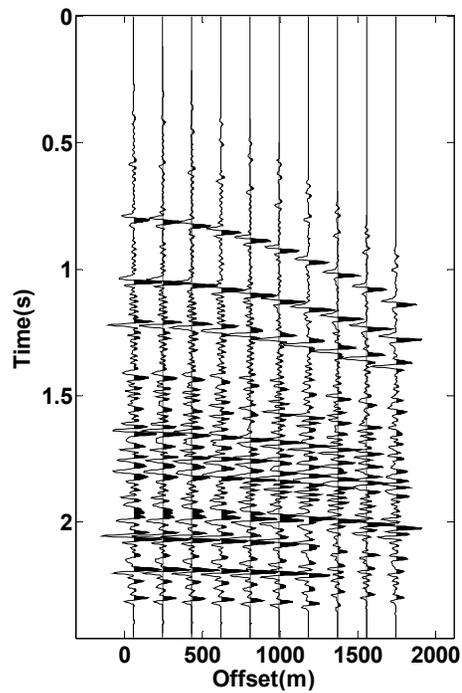


b)

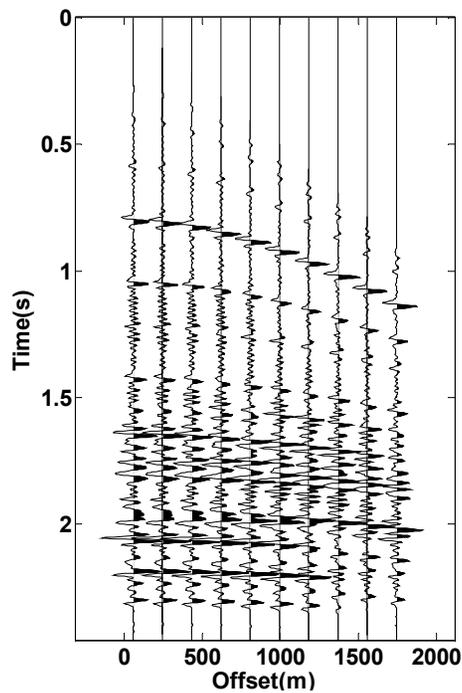


c)

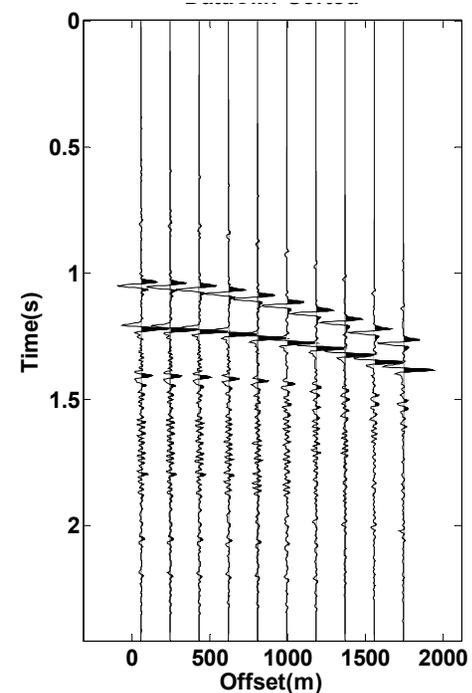
Figure 5.18 CMP gathers from a) baseline and b) monitor surveys, and c) their difference.



a)



b)



c)

Figure 5.19 Reconstructed CMP gathers from a) baseline and b) monitor surveys, and c) their difference.

5.3 Joint inversion of time lapse data by LSPSM

The advantages of using LSPSM and data reconstruction methods for time lapse studies of seismic data was discussed in section 5.2. I performed a separate LSPSM for each baseline and monitor data sets, ignoring the presence of the other data set. In this section, I show the simultaneous inversion of the baseline and monitor survey data using a joint inversion.

Ayeni and Biondi (2010) performed least squares joint inversion of time lapse data using two related formulations. Their first formulation dealt with the simultaneous inversion of multiple images, referred to as the “Regularized Joint inversion of Multiple Images” (RJMI), the second formulation dealt with the inversion of the baseline and the difference (time lapse) images, referred to as the “Regularized Joint inversion for Image Differences” (RJID). The RJMI method returns baseline and monitor images, and the output of the RJID method is the baseline image and differences between the baseline and monitor image. Ayeni and Biondi (2010) used approximations to the wave equation least squares Hessian matrices to perform their inversion in the image domain. Implementing least squares in the image domain enabled them to target-orient their equation and reduce the high cost of the wave equation least squares migration.

In this study, I use Kirchhoff LSPSM for the joint inversion of time lapse data which is cheaper than the wave equation least squares inversion. To do so with the same methodology that Ayeni and Biondi (2010) implemented, I combine the two cost functions for the separate damped LSPSM/inversion of the baseline survey (Yousefzadeh and Bancroft, 2012c)

$$J_0(\mathbf{m}_0) = \|\mathbf{G}_0\mathbf{m}_0 - \mathbf{d}_0\|^2 + \mu_0^2\|\mathbf{m}_0\|^2, \quad 5.11$$

and the monitor survey

$$J_1(\mathbf{m}_1) = \|\mathbf{G}_1\mathbf{m}_1 - \mathbf{d}_1\|^2 + \mu_1^2\|\mathbf{m}_1\|^2, \quad 5.12$$

into the Multiple Image Joint Inversion (MIJI) cost function as

$$J_{MIJI}(\mathbf{m}_0, \mathbf{m}_1) = \left\| \begin{bmatrix} \mathbf{G}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_1 \end{bmatrix} \begin{bmatrix} \mathbf{m}_0 \\ \mathbf{m}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mu_0 & 0 \\ 0 & \mu_1 \end{bmatrix} \begin{bmatrix} \mathbf{m}_0 \\ \mathbf{m}_1 \end{bmatrix} \right\|^2. \quad 5.13$$

This is similar to RJMI. The time lapse image can be computed by

$$\Delta\mathbf{m}_{miji} = \mathbf{m}_{miji1} - \mathbf{m}_{miji0}, \quad 5.14$$

where \mathbf{m}_{miji0} and \mathbf{m}_{miji1} are the baseline and monitor survey reflectivity images resulting from MIJI, respectively.

Alternatively, joint inversion can be performed to obtain a time lapse image, $\Delta\mathbf{m}$, directly by minimizing the Image Difference Joint Inversion (IDJI) cost function as

$$J_{IDJI}(\mathbf{m}_0, \Delta\mathbf{m}) = \left\| \begin{bmatrix} \mathbf{G}_0 & \mathbf{0} \\ \mathbf{G}_1 & \mathbf{G}_1 \end{bmatrix} \begin{bmatrix} \mathbf{m}_0 \\ \Delta\mathbf{m} \end{bmatrix} - \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mu_0 & 0 \\ 0 & \mu_1 \end{bmatrix} \begin{bmatrix} \mathbf{m}_0 \\ \Delta\mathbf{m} \end{bmatrix} \right\|^2. \quad 5.15$$

This is similar to the RJID formulation.

I used LSCG to minimize these cost functions. As an example, ignoring regularization terms, $\mu_0 = \mu_1 = 0$, a simplified LSCG algorithm to minimize J_{IDJI} is shown in Table 3-6. As seen in this table, each iteration in this LSCG algorithm includes performing two migrations and two modelings. Therefore, each iteration is at least four times as expensive as one time migration of the baseline survey data. Tests showed that the convergence of the above mentioned joint inversion algorithms were slower than the convergence of separate LSPSMs in the previous section.

I performed MIJI and IDJI on the synthetic example in section 5.2.3. Results are shown after 50 iterations using LSCG in Figure 5.20 and Figure 5.21. The resulting time lapse image from MIJI is shown in Figure 5.22. Figure 5.23 and Figure 5.24 show the baseline and time lapse images resulting from IDJI, respectively.

Comparison between Figure 5.22 and Figure 5.24 shows that a higher resolution time lapse is achieved by MIJI. This seems to be due to the difference between the structure of the Hessian matrices in the MIJI and IDJI. The IDJI forward modeling matrix is 50% denser than the MIJI matrix. The MIJI matrix is symmetric where the IDJI matrix is not. Therefore, MIJI is better solved by the LSCG method.

The comparison between a separate LSPSM time lapse image (Figure 5.17a) and MIJI (Figure 5.22) shows that the MIJI has slightly less artifacts in the resulting time lapse image than the time image from a separate LSPSM. However, it is necessary to mention that a time lapse image of MIJI is obtained after 50 LSCG iterations which is equal to the cost of 200 migrations while the separate LSPSM image is achieved after 20 iterations on each data sets which is equal to the cost of 80 migrations.

Table 5.1 LSCG algorithm for solving image difference joint inversion equation.

$\mathbf{m}_0 =$ an initial guess or $\mathbf{m}_0 = \mathbf{0}$	
$\Delta \mathbf{m} = \mathbf{0}$	
$\mathbf{s}_0 = \mathbf{d}_0 - \mathbf{G}_0 \mathbf{m}_0$	
$\mathbf{s}_1 = \mathbf{d}_1 - \mathbf{G}_1 (\mathbf{m}_0 + \Delta \mathbf{m})$	
$\mathbf{r}_1 = \mathbf{G}_1^T \mathbf{s}_1$	
$\mathbf{r}_0 = \mathbf{G}_0^T \mathbf{s}_0 + \mathbf{r}_1$	
$\mathbf{p}_0 = \mathbf{r}_0$	
$\mathbf{p}_1 = \mathbf{r}_1$	
$\mathbf{q}_0 = \mathbf{G}_0 \mathbf{p}_0$	
$\mathbf{q}_1 = \mathbf{G}_1 (\mathbf{p}_0 + \mathbf{p}_1)$	
for $i = 0$: iterations limit	
$\alpha_{i+1} = \frac{\mathbf{r}_{0i} \cdot \mathbf{r}_{0i} + \mathbf{r}_{1i} \cdot \mathbf{r}_{1i}}{\mathbf{q}_{0i} \cdot \mathbf{q}_{0i} + \mathbf{q}_{1i} \cdot \mathbf{q}_{1i}}$	
$\mathbf{m}_{0i+1} = \mathbf{m}_{0i} + \alpha_{i+1} \mathbf{p}_{0i}$	
$\Delta \mathbf{m}_{i+1} = \Delta \mathbf{m}_i + \alpha_{i+1} \mathbf{p}_{1i}$	
$\mathbf{s}_{0i+1} = \mathbf{s}_{0i} - \alpha_{i+1} \mathbf{q}_{0i}$	
$\mathbf{s}_{1i+1} = \mathbf{s}_{1i} - \alpha_{i+1} \mathbf{q}_{1i}$	
$\mathbf{r}_{1i+1} = \mathbf{G}_1^T \mathbf{s}_{1i+1}$	≈≈
$\mathbf{r}_{0i+1} = \mathbf{G}_0^T \mathbf{s}_{i+1} + \mathbf{r}_{1i+1}$	≈≈
$\beta_{i+1} = \frac{\mathbf{r}_{0i+1} \cdot \mathbf{r}_{0i+1} + \mathbf{r}_{1i+1} \cdot \mathbf{r}_{1i+1}}{\mathbf{r}_{1i} \cdot \mathbf{r}_{1i}}$	
$\mathbf{p}_{0i+1} = \mathbf{r}_{0i+1} + \beta_{i+1} \mathbf{p}_{0i}$	
$\mathbf{p}_{1i+1} = \mathbf{r}_{1i+1} + \beta_{i+1} \mathbf{p}_{1i}$	
$\mathbf{q}_{0i+1} = \mathbf{G}_0 \mathbf{p}_{0i+1}$	↔
$\mathbf{q}_{1i+1} = \mathbf{G}_1 (\mathbf{p}_{0i+1} + \mathbf{p}_{1i+1})$	↔
endfor	

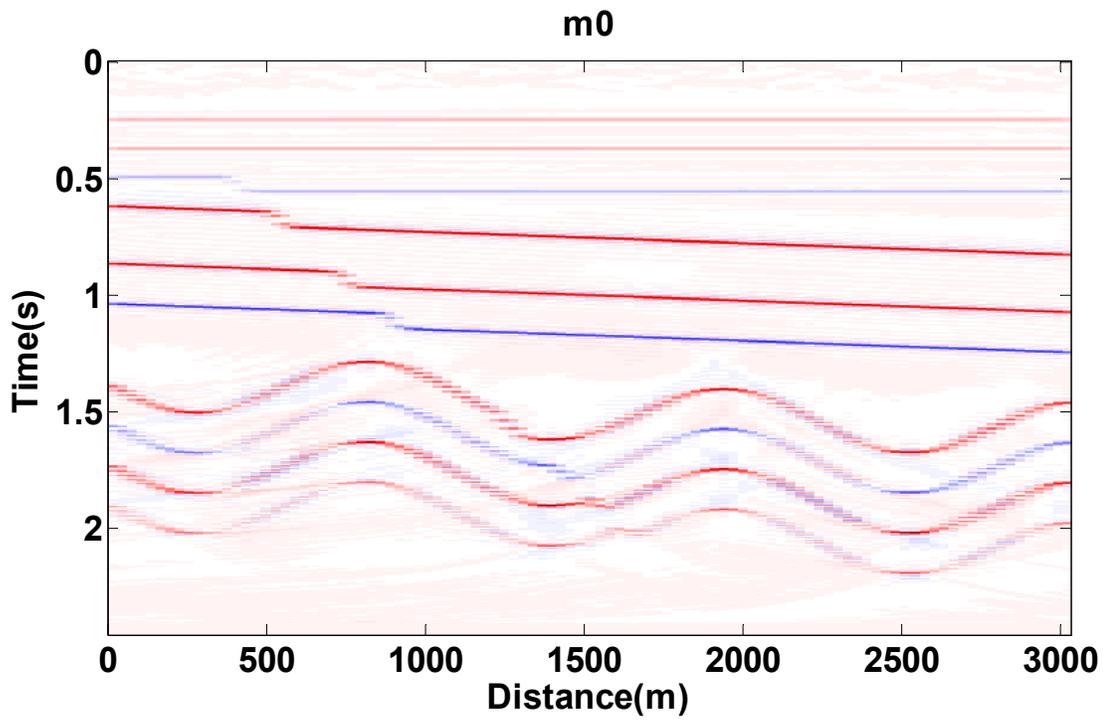


Figure 5.20 Reflectivity of baseline survey achieved by MIJI.

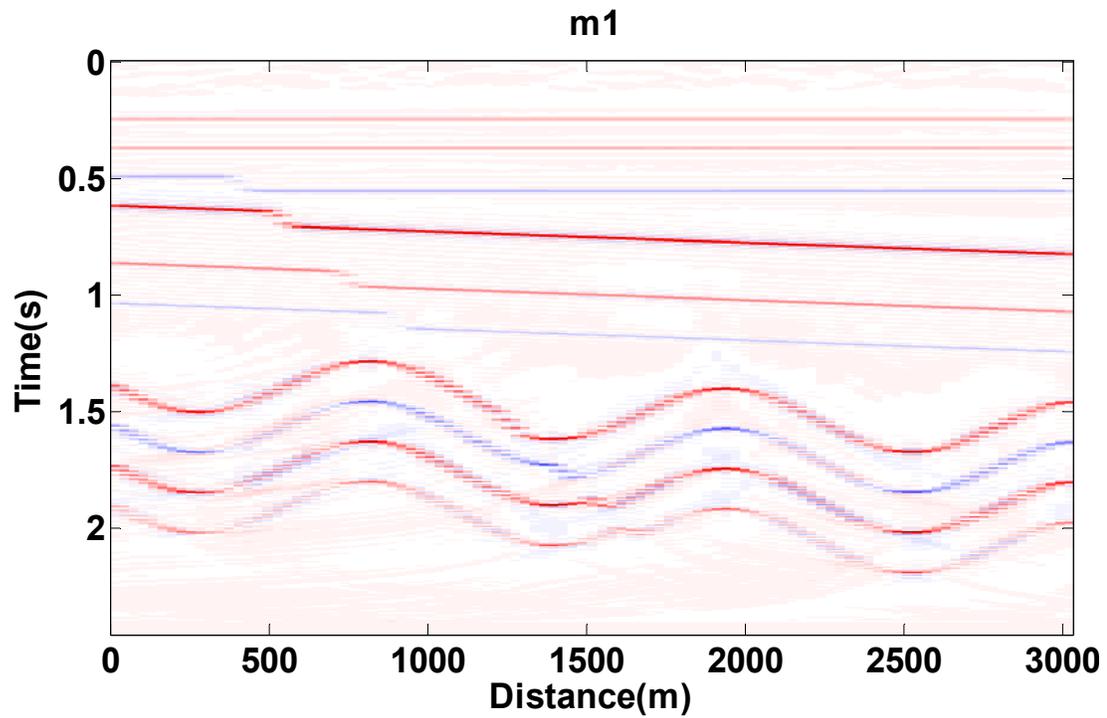


Figure 5.21 Reflectivity of monitor survey achieved by MIJI.

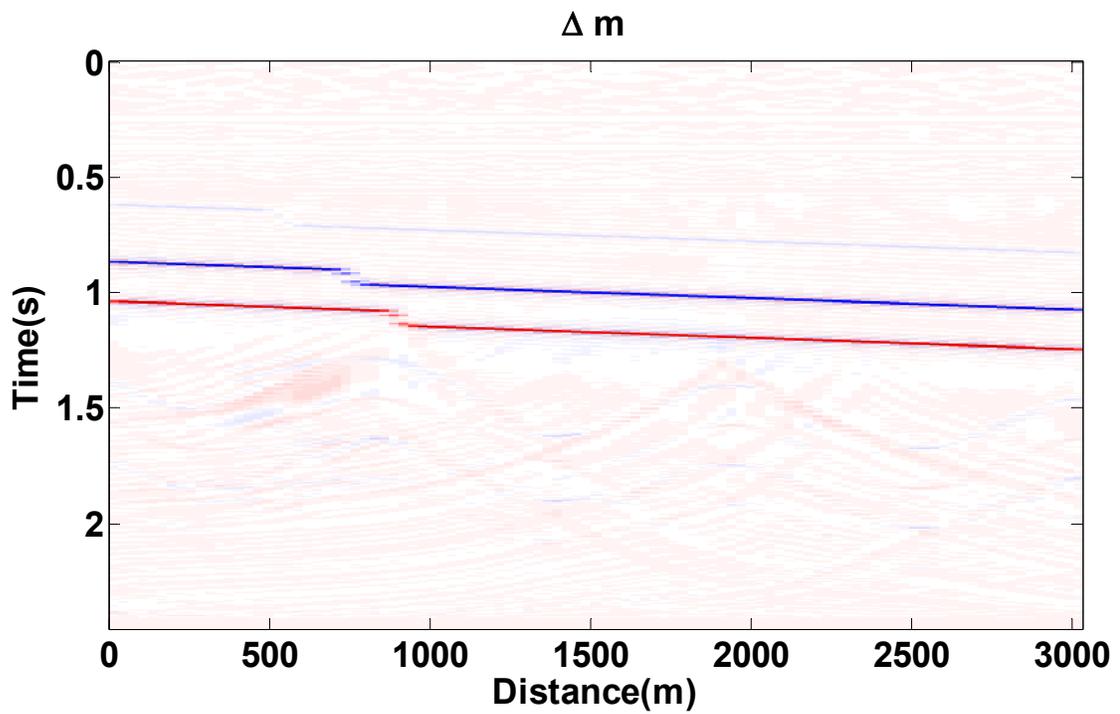


Figure 5.22 Difference between reflectivity of baseline and monitor surveys by MIJI.

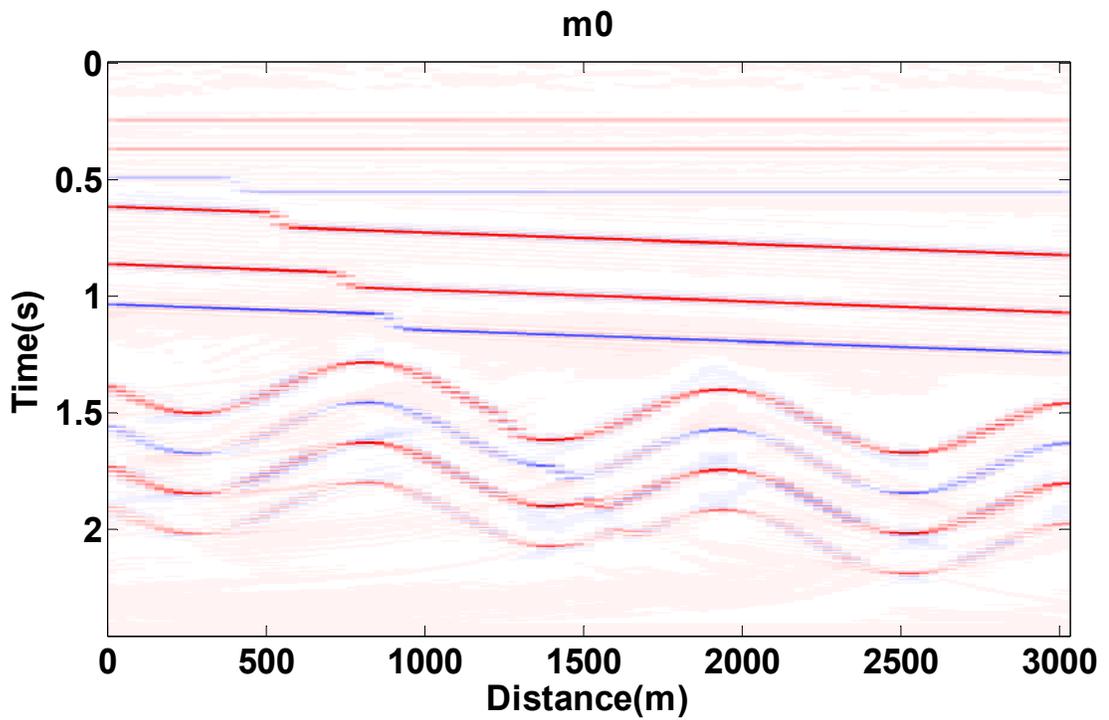


Figure 5.23 Reflectivity of baseline survey achieved by IDJI.

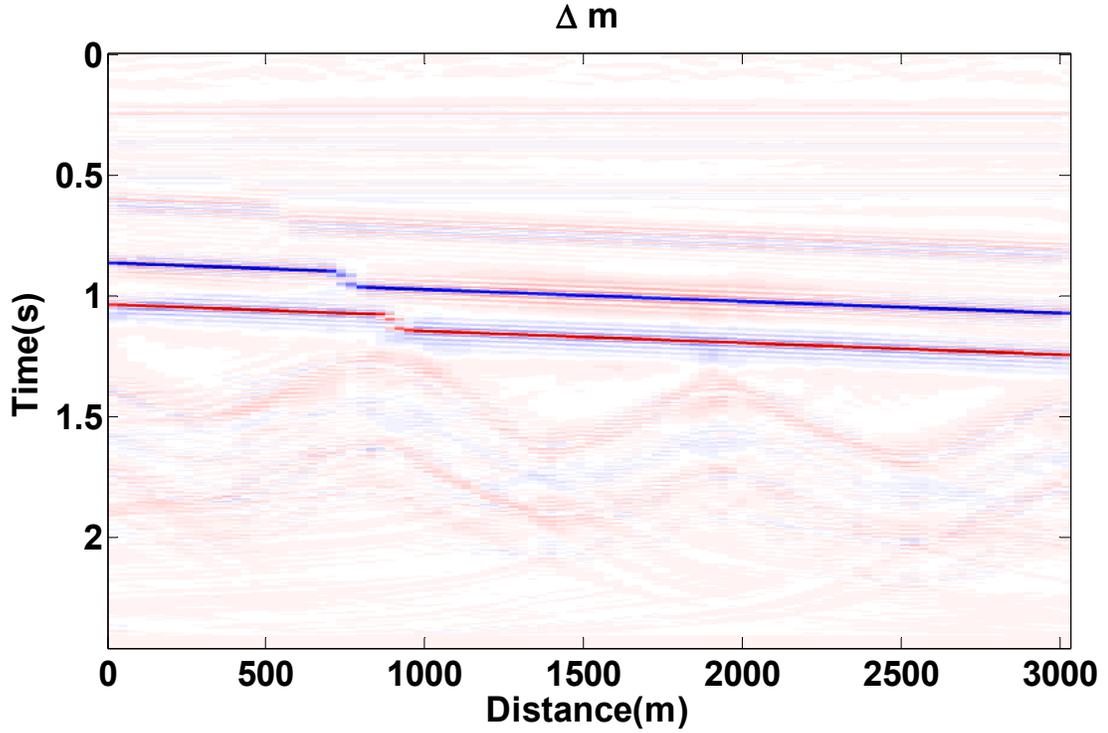


Figure 5.24 Time lapse image from reflectivity by IDJI.

The formulation for the joint inversion of the baseline and monitor survey data can be extended to the joint inversion of a baseline and several monitor surveys. An outline of the method follows, however, no data was processed using this method.

For example, if the subsurface reflectivity model changes from \mathbf{m}_1 to $\mathbf{m}_2 = \mathbf{m}_1 + \Delta\mathbf{m}_1$ between the two monitor surveys, the second monitor survey data, \mathbf{d}_2 , can be expressed by

$$\mathbf{d}_2 = \mathbf{G}_2 \mathbf{m}_2 = \mathbf{G}_2(\mathbf{m}_1 + \Delta\mathbf{m}_1) = \mathbf{G}_2(\mathbf{m}_0 + \Delta\mathbf{m} + \Delta\mathbf{m}_1), \quad 5.16$$

where \mathbf{G}_2 is the forward modeling operator of the second monitor survey. Then, the corresponding joint inversion methods can retrieve the \mathbf{m}_0 , \mathbf{m}_1 , and \mathbf{m}_2 via MIJI or \mathbf{m}_0 , $\Delta\mathbf{m}$, and $\Delta\mathbf{m}_1$ via IDJI by minimizing the following cost functions,

$$J_{MIJI}(\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2) = \left\| \begin{bmatrix} \mathbf{G}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_2 \end{bmatrix} \begin{bmatrix} \mathbf{m}_0 \\ \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mu_0 & 0 & 0 \\ 0 & \mu_1 & 0 \\ 0 & 0 & \mu_2 \end{bmatrix} \begin{bmatrix} \mathbf{m}_0 \\ \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix} \right\|^2, \quad 5.17$$

and

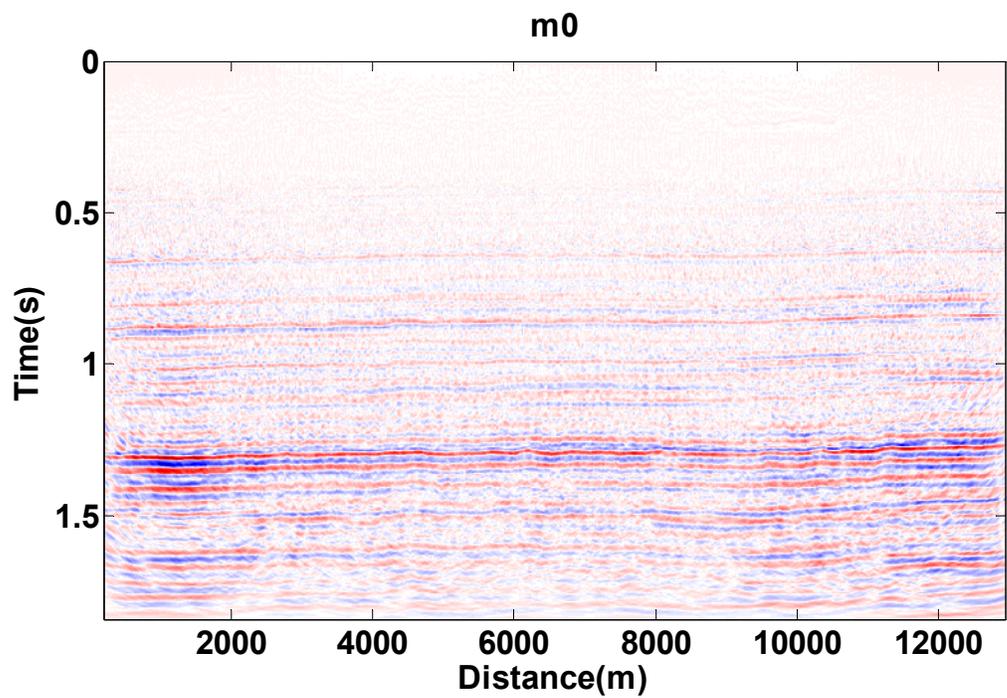
$$J_{IDJI}(\mathbf{m}_0, \Delta\mathbf{m}, \Delta\mathbf{m}_1) = \left\| \begin{bmatrix} \mathbf{G}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_1 & \mathbf{G}_1 & \mathbf{0} \\ \mathbf{G}_2 & \mathbf{G}_2 & \mathbf{G}_2 \end{bmatrix} \begin{bmatrix} \mathbf{m}_0 \\ \Delta\mathbf{m} \\ \Delta\mathbf{m}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \mu_0 & 0 & 0 \\ 0 & \mu_1 & 0 \\ 0 & 0 & \mu_2 \end{bmatrix} \begin{bmatrix} \mathbf{m}_0 \\ \Delta\mathbf{m} \\ \Delta\mathbf{m}_1 \end{bmatrix} \right\|^2. \quad 5.18$$

Each iteration in the LSCG method to solve equations 5.17 and 5.18 is at least six times more expensive than the migration of the baseline data. This means that 20 iterations of LSCG to invert a time lapse study with two monitor surveys is 40 times more costly than the migration of each data set.

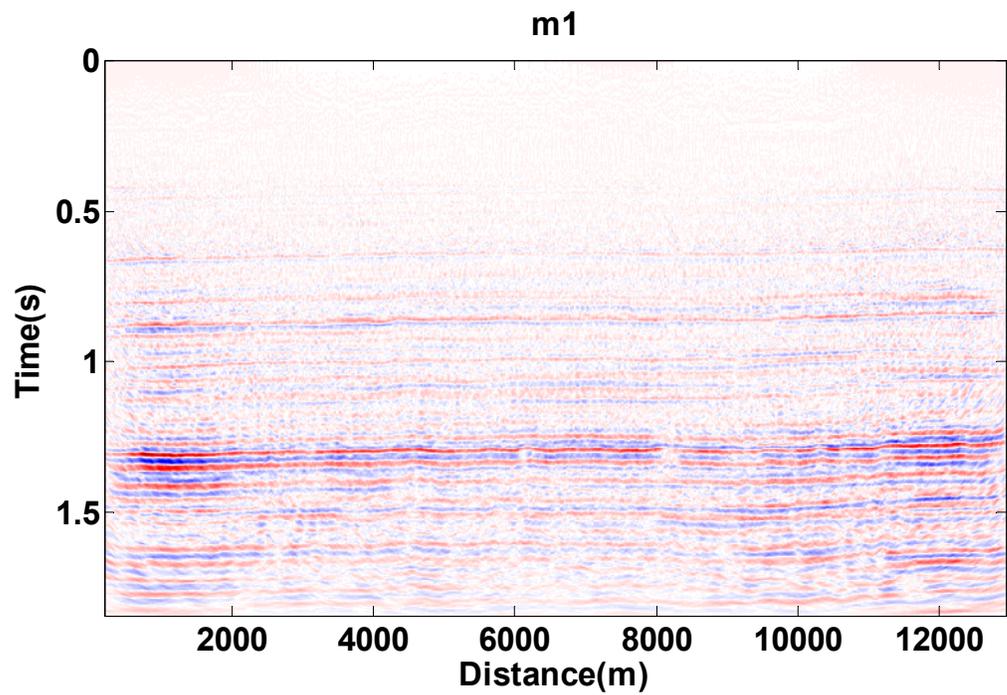
5.4 Real data examples

The ability of LSPSM to enhance the resolution and to reconstruct the missing data from the NEBC real data set is shown in Chapter Two. The NEBC data set is not a time lapse data set. However, I consider the two sets of 10% randomly selected data from the NEBC survey as the baseline and the monitor data. I found this method to be a useful way to compare the two surveys which differ only in their geometry. Since there is no change in the model parameters, the resulting time lapse images should have no major reflections.

The baseline and monitor survey data are migrated. Figure 5.25 shows the migration image of the baseline and the monitor data sets. For a better comparison between the time lapse and the inversion images, all images in this section are shown without applying any kind of AGC or amplitude correction. Figure 5.26 is the migration time lapse image, which shows the difference between the baseline and the monitor migration images. Acquisition footprint resulting from high decimation of the original data remains in the time lapse image. The baseline and monitor surveys are jointly inverted for the baseline and monitor images. Results of the MIJI are shown in Figure 5.27 and Figure 5.28. Figure 5.29 is the corresponding time lapse image. Results of IDJI for baseline and monitor surveys are shown in Figure 5.30. Results of both MIJI and IDJI are obtained after 10 LSCG iterations. Comparison between three time lapse images, migration time lapse image in Figure 5.26, MIJI time lapse image in Figure 5.29, and IDJI in Figure 5.30b show the ability of the joint inversion methods at improving attenuation of migration time lapse artifacts. The best time lapse image is achieved with the MIJI method. The IDJI method requires many more LSCG iterations to obtain a time lapse image similar to MIJI time lapse image.



a)



b)

Figure 5.25 Migration of data set from a) baseline, and b) monitor survey selected from NEBC data set.

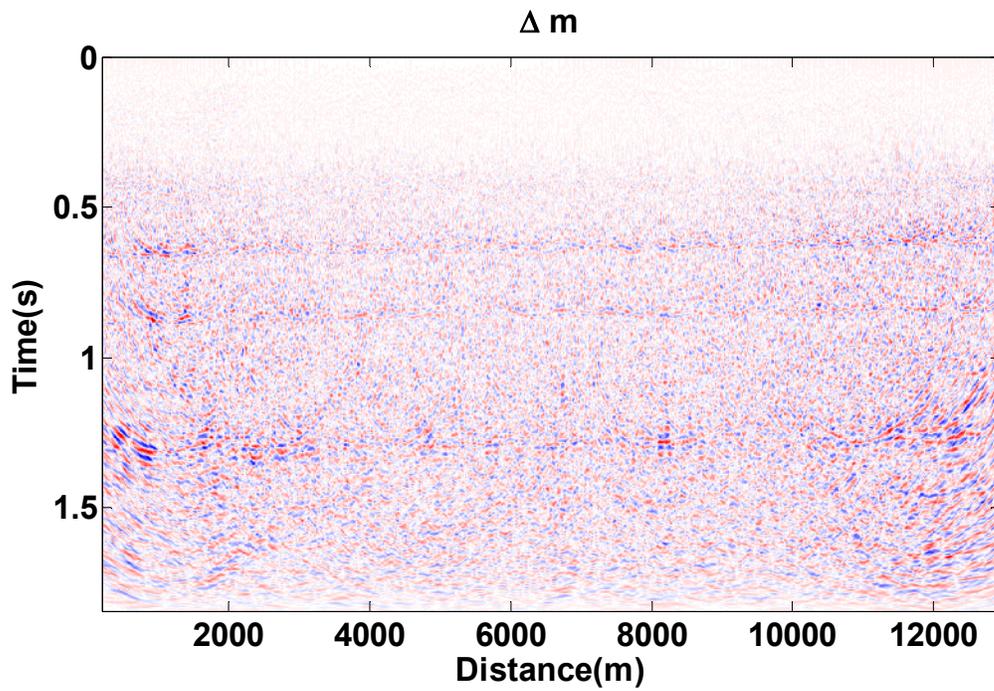


Figure 5.26 The difference between two migration images in Figure 5.25.

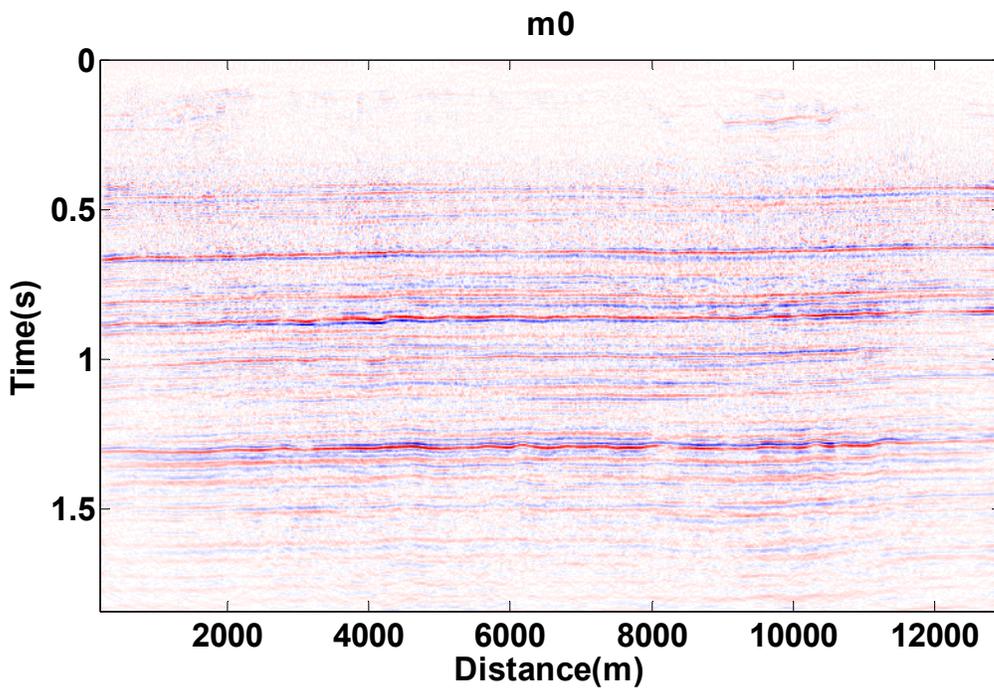


Figure 5.27 The baseline image obtained from the time lapse joint inversion (MIJI).

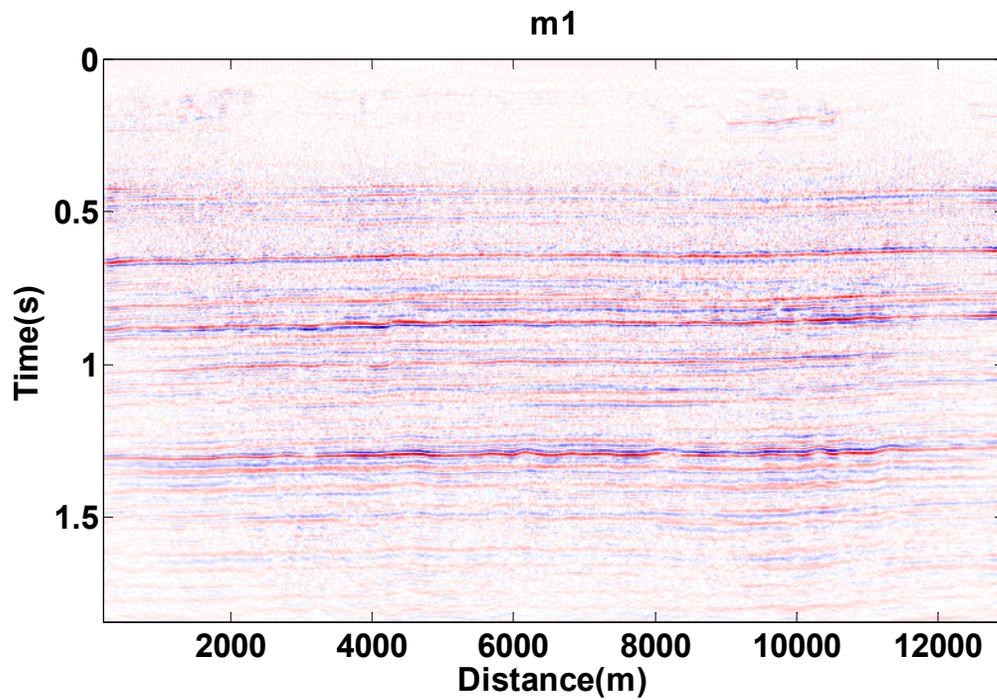


Figure 5.28 The monitor image obtained from the time lapse joint inversion (MIJI).

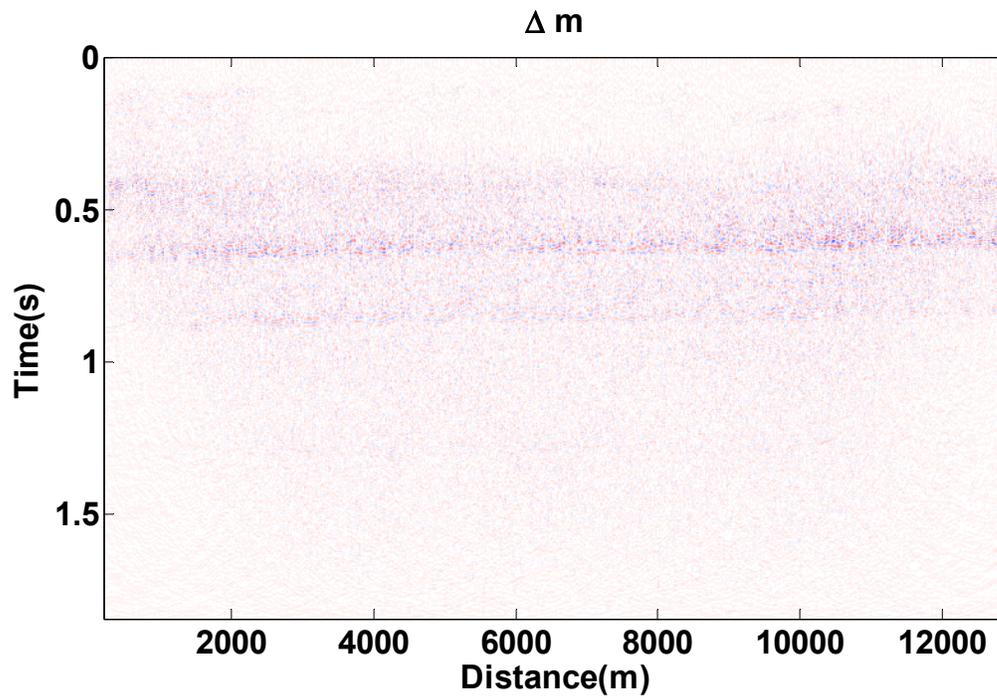


Figure 5.29 The difference between reflectivity of baseline and monitor surveys by MIJI.

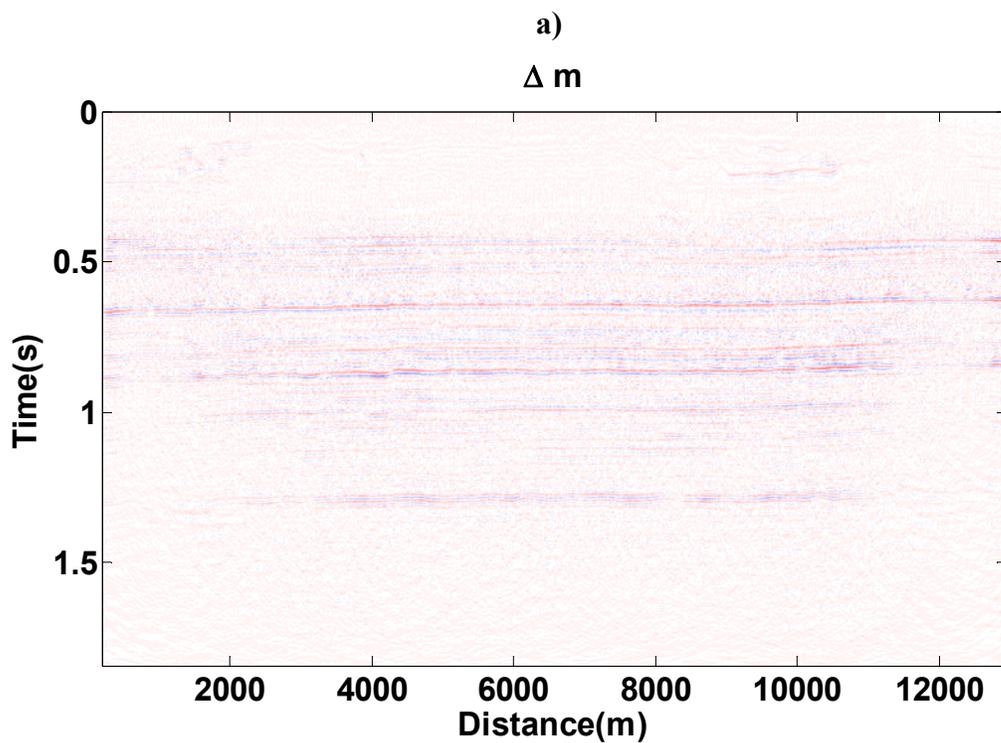
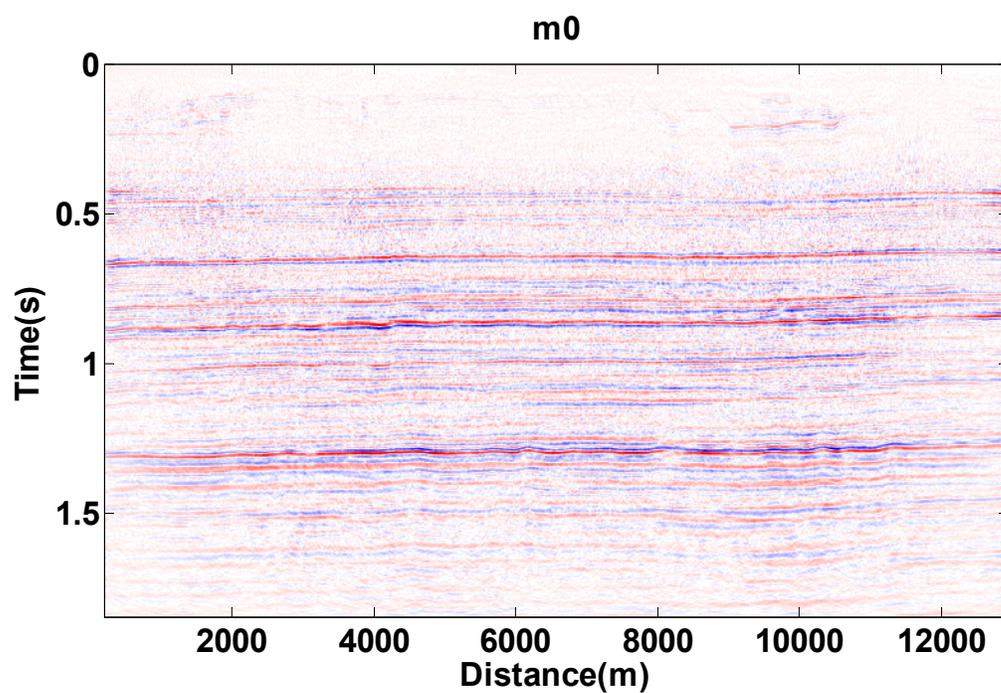


Figure 5.30 Image of a) the baseline survey and b) time lapse obtained by IDJI.

5.5 Summary and Conclusion

The usefulness of separate and joint LSPSM/inversion for time lapse data has been investigated in this chapter.

This analysis assumed that there is no difference in the acquisition instruments, environmental noise, near surface effects, processing flows, and parameters, for both baseline and monitor surveys. It is the difference in acquisition geometries that leaves different artifacts at the migration images. These artifacts may mask the reflectivity in the time lapse image. It was shown how LSPSM of both baseline and monitor data sets can attenuate acquisition footprint and create reliable time lapse images. The reconstructed data from two surveys make the prestack time lapse studies more easily feasible. Formulations of the joint inversion of time lapse data to invert for the baseline image and time lapse image by LSCG method are derived.

It is important to mention that this is an expensive procedure. Each iteration of the joint inversion costs more than four times that of performing a single conventional migration. Other corrections such as removing the near surface effects and attenuating multiples must be performed before inverting the data.

Chapter Six: CONCLUSION

Recorded seismic data are a discrete representation of the acoustic/elastic wavefield measured on the ground's surface. Any imaging algorithm is affected by the incomplete sampling of data. Kirchhoff migration is an effective, low cost method of migration which easily handles irregularities of seismic data sampling. However, irregular or sparsely sampled data produce artifacts in the migration image. Migration artifacts are due to either insufficient sampling of data, acquisition footprint, or the inaccuracy of the imaging algorithm, or any combination of the above.

Artifacts due to sparse or irregular sampling of data may be reduced using LSPSM by minimizing a cost function which includes the difference between observed data and modeled data, called the misfit function.

LSPSM significantly reduces the acquisition footprint in the migration image and produces a high resolution image. Then, the high resolution image of LSPSM can be used for data interpolation and reconstruction. As shown in this study, replacing migration with LSPSM is a costly procedure which also requires a good estimation of the subsurface velocity model.

The feasibility of using standard multigrid methods for reducing the cost of the LSPSM method is investigated. I showed how migration, and consequently inversion, can be expressed by matrix multiplications. I showed that the standard multigrid method is not a suitable solver for the LSPSM equation for at least two reasons:

- 1) The LSPSM Hessian is not a diagonally dominant matrix and consequently is not solvable by Jacobi or Gauss-Seidel methods. Therefore, a conventional multigrid method is not able to solve the LSPSM equation.
- 2) The Hessian matrix is too large. The size of the matrix is equal to the square of the number of grid points in the model space. Therefore, it is not possible to work directly with the Hessian matrix with presently available computers.

I showed how multilevel methods can be used to solve the LSPSM equation. The ability of the BiCG Stabilized method and multilevel LSCG with spatial downsampling of data is shown in this dissertation.

LSPSM is very sensitive to the accuracy of the background velocity model. The improvement of an LSPSM image in a few iterations, and the ability of data reconstruction, are factors which can be used to evaluate the accuracy of an imaging velocity.

The high sensitivity of LSPSM to imaging velocity, and its low sensitivity to the irregularity of data sampling, make LSPSM a practical choice for velocity analysis of very incompletely sampled data. Velocity analysis on the migration CIGs can be extended to the LSPSM CIGs. The unnormalized crosscorrelation spectrum of shot domain LSPSM CIGs gives a high resolution panel for velocity estimation of very irregularly and sparsely sampled data. This can provide an imaging velocity model that is accurate enough to improve the resolution of a migration image using LSPSM in a few LSCG iterations, and give an acceptable data reconstruction.

Migration time lapse images always have artifacts due to different acquisition geometries of baseline and monitor surveys. The separate and joint LSPSM/inversion of time lapse seismic data provides high resolution time lapse images which are more reliable than the migration time lapse images. This is very important when it is not possible to have a monitor survey that exactly follows the baseline survey. A streamer marine data acquisition survey is an example of an unrepeatable acquisition geometry. Data reconstruction of time lapse projects with LSPSM is very important for performing a prestack time lapse study.

In closing, it is necessary to mention that in comparison to a conventional migration, the LSPSM method is a very expensive process. The requirement of a good estimate of the velocity model is also very important. Since Kirchhoff time migration does not account for multipathing of seismic signals, data must be multiple attenuated before LSPSM.

The LSPSM method is less affected by using an inaccurate wavelet, approximate migration weight, or aperture, as long as the modeling operator is the exact transpose of the migration operator. Due to the high flexibility of the Kirchhoff method, LSPSM can be extended to converted wave or surface microseismic imaging.

References

- Aoki N., and G. T. Schuster, 2009, Fast least squares migration with a deblurring filter: *Geophysics*, **74**, WCA83-WCA93.
- Ayeni, G., and B. Biondi, 2010, Target-oriented joint least-squares migration/inversion of time-lapse seismic data sets: *Geophysics*, **75**, R62-R73.
- Bancroft, J. C., 2007, A practical understanding of pre- and poststack migrations: SEG publications.
- Baysal, E., D. D. Kosloff, and J. W. C. Sherwood, 1983, Reverse time migration: *Geophysics*, **48**, 1514-1524.
- Biondi, B. L., 2007, Concepts and applications in 3D seismic imaging: SEG publications.
- Blais, J. A. R., 2010, Least squares for practitioners: Mathematical problems in engineering, vol. **2010**, Article ID: 508092.
- Briggs, W. L., V. E. Henson, and S. F. McCormick, 2000, A multigrid tutorial: second edition, SIAM.
- Bunks, C., F. M. Saleck, S. Zaleski, and G. Chavents, 1995, Multiscale seismic waveform inversion: *Geophysics*, **60**, 1457-1473.
- Claerbout, J. F., and S. M. Doherty, 1972, Downward continuation of moveout corrected seismograms: *Geophysics*, **37**, 741-768.
- Cole, S., and M. Karrenbach, 1997, Least squares Kirchhoff migration: Stanford Exploration Project, **75**, 107-120.
- Douglas, C. C., G. Hasse, and U. Langer, 2003, A tutorial on elliptic PDE solvers and their parallelization: SIAM.
- Duquet, B., K. J. Marfurt, and J. A. Dellinger, 2000, Kirchhoff modeling, inversion for reflectivity, and subsurface illumination: *Geophysics*, **65**, 1195-1209.
- Gazdag, J., 1978, Wave equation migration with the phase-shift method: *Geophysics*, **43**, 1342-1351.

- Gersztenkorn, A., J. Bednar, and L. Lines, 1986, Robust iterative inversion for the one-dimensional acoustic wave equation: *Geophysics*, **51**, 357-368.
- Greaves, R. J., and T. J., Fulp, 1987, Three-dimensional seismic monitoring of an enhanced oil recovery process: *Geophysics*, **52**, 1175-1187.
- Gulunay, N., 2003, Seismic trace interpolation in the Fourier transform domain: *Geophysics*, **68**, 355-369.
- Hagedoorn, J. G., 1954, A process of seismic reflection interpretation: *Geophysical Prospecting*, **2**, 85-127.
- Hampson, D., 1986, Inverse velocity stacking for multiple elimination: *J. Can. Soc. Expl. Geophys.*, **22**, 44-55.
- Hanger, W. W., and H. Zhang, 2006, A survey of the nonlinear conjugate gradient methods: *Pacific Journal of Optimization*: **2**, 35-58.
- Hestenes, M., and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems: *Nat. Bur. Standards J. Res.*, **49**, 403-436.
- Hill, N. R., 1990, Gaussian beam migration: *Geophysics*, **55**, 1416-1428.
- Hu J., G. T. Schuster, and P. A. Valasek, 2001, Poststack migration deconvolution: *Geophysics*, **66**, 939-952.
- Jacobs, D. A. H., 1986, A generalization of the conjugate gradient method to solve complex systems: *IMA Journal of Numerical Analysis*, **6**, 447-452.
- Ji, J. 1997, Least squares imaging, datuming, and interpolation using the wave equation: *Stanford Exploration Project*, **75**, 121-135.
- Jiang, Z., and G. T. Schuster, 2003, Target oriented least squares migration: *SEG Expanded Abstract*, **22**, 1043-1046.
- Koehler, F., and M. Taner, 1985, The use of conjugate-gradient algorithms in the computation of predictive deconvolution operators: *Geophysics*, **50**, 2752-2758.

- Kostov, C., 1990, Multichannel seismic experiment with a drill bit source: Stanford Exploration Project, **63**.
- Kuehl, H., 2002, Least-squares wave-equation migration/inversion: PhD thesis dissertation, Univ. of Alberta.
- Kuehl, H., and M. D. Sacchi, 2001, Split-step WKB least-squares migration/inversion of incomplete data: 5th Soc. Expl. Geophys. of Japan Internat. Symp. Imaging Technology.
- Lambare, G., J. Virieux, R. Madariaga, and S. Jin, 1992, Iterative asymptotic inversion in the acoustic approximation: *Geophysics*, **57**, 1138-1154.
- Levenberg, K., 1944, A method for the solution of certain nonlinear problems in least squares: *Quarterly of Applied Mathematics*, **2**, 164-168.
- Lines, L. R., and S. Treitel, 1984, Tutorial, A review of least squares inversion and its application to geophysical problems: *Geophysical Prospecting*, **32**, 159-186.
- Lumley, D. E., 2001, The next wave in reservoir monitoring: The instrumented oil field: *The Leading Edge*, **20**, 640-648.
- Margrave, G. F., 2001, Numerical methods of exploration seismology: Department of Geology and Geophysics, The University of Calgary.
- Margrave, G. F., H. D. Geiger, S. M. Al-Saleh, and M. P. Lamoureux, 2006, Improving explicit seismic depth migration with stabilizing Wiener filter and spatial resampling: *Geophysics*, **71**, S111-S120.
- Marquardt, D. W., 1963, An algorithm for least squares estimation of non-linear parameters: *Journal of the Society of Industrial and Applied Mathematics*, **11**, 431-441.
- MathWorks: MathWorks Corporation, Last accessed 28 Dec. 2012. <<http://www.mathworks.com/help/signal/ref/convmtx.html>>
- McMechan G. A., 1983, Migration by extrapolation of time-dependent boundary values: *Geophysical Prospecting*, **31**, 413-420.
- Menke, W., 1984, *Geophysical data analysis: discrete inverse theory*: Academic Press Inc.

- Millar, J., and J. C. Bancroft, 2004, Multigrid deconvolution of seismic data: 74th Annual International Meeting, SEG Expanded Abstract, **23**, 1945-1948.
- Millar, J., and J. C. Bancroft, 2005, Using multigrid for surface consistent statics: 75th Annual International Meeting, SEG Expanded Abstract, **24**, 2189-2192.
- Millar, J., and J. C. Bancroft, 2006, Long wavelength solutions to the surface consistent equations: 76th Annual International Meeting, SEG Expanded Abstract, **25**, 3091-3094.
- Morse, P. M., and H. Feshbach, 1953, Methods of theoretical physics, part I: McGraw-Hill.
- Nemeth, T., C. Wu, and G. T. Schuster, 1999, Least-squares migration of incomplete reflection data: Geophysics, **64**, 208-221.
- Plessix, R. E., 2007, A Helmholtz iterative solver for 3D seismic-imaging problems: Geophysics, **72**, P.SM185-SM194.
- Porsani, M., 1999, Seismic trace interpolation using half-step prediction filters. : Geophysics, **64**, 1461-1467.
- Rickett, J. E., and D. E. Lumley, 2001, Cross-equalization data processing for time-lapse seismic reservoir monitoring: A case study from the Gulf of Mexico: Geophysics, **66**, 1015-1025.
- Saad, Y., 2003, Iterative methods for sparse linear systems, second edition: Society for Industrial and Applied Mathematics, PA, USA, //www.umn.edu (Accessed 07 Jun, 2012).
- Sacchi, M. D., J. Wang, and H. Kuehl, 2004, High resolution imaging: 66th EAGE's Conference & Technical Exhibition, Expanded Abstracts, D032.
- Saleck, F. M., C. Bunks, S. Zaleski', and G. Chavent, 1993, Combining the multigrid and gradient methods to solve the seismic inversion problem: 63th Annual International Meeting, SEG Expanded Abstract, 688-691.
- Scales, J., 1987, Tomographic inversion via the conjugate gradient methods: Geophysics, **52**, 179-185.
- Scales, J. A., 1997, Theory of seismic imaging: Samizdat Press, Release 2.2.

- Scales, J. A., and L. Tenorio, 2001, Tutorial: Prior information and uncertainty in inverse problems: *Geophysics*, **66**, 389-397.
- Scales, J. A., and R. Snieder, 2000, The anatomy of inverse problems: *Geophysics*, **65**, 1708-1710.
- Schneider, W. A., 1978, Integral formulation for migration in two-dimensions and three-dimensions: *Geophysics*, **43**, 49-76.
- Sheriff, R. E., 2006, *Encyclopedic dictionary of applied geophysics: fourth edition*, SEG publications.
- Shewchuk, J. R., 1994, *An introduction to the conjugate gradient method without the agonizing pain*: School of Computer Science, Carnegie Mellon University.
- Sonneveld, P., 1989, CGS, a fast Lancosz-type solver for non-symmetric linear systems: *SIAM J. Sci. Stat. Comput.*, **10**, 36-52.
- Spitz, S., 1991, Seismic trace interpolation in the $f - x$ domain: *Geophysics*, **56**, 785-796.
- Stolt, R. H., 1978, Migration by Fourier transform: *Geophysics*, **43**, 23-48.
- Strang, G., 18.086 *Mathematical Methods for Engineers II*: Spring 2006. (Massachusetts Institute of Technology: MIT OpenCourseWare), <http://ocw.mit.edu> (Accessed 07 Jun, 2012). License: Creative Commons BY-NC-SA.
- Tang, Y. 2007, Least squares migration of incomplete data sets with regularization in the subsurface-offset domain: *Stanford Exploration Project*, **125**, 105-121.
- Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: *Geophysics*, **49**, 1259-1266.
- Ulrych, T. J., M. D. Sacchi, and A. Woodbury, 2001, Tutorial: A Bayes tour of inversion: *Geophysics*, **66**, 55-69.
- Van der Vorst, H., 1992, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems: *SIAM J. Sci. Stat. Comput.*, **13**, 631-634.

- Vedanti, N., and M. K. Sen, 2009, Seismic inversion tracks in situ combustion: A case study from Balol oil field. *India: Geophysics*, **74**, 103-112.
- Vujicic, M. R., 2006, Finite element solution of transient heat conduction using iterative solvers: *Engineering Computations*, **23**, 408 – 431.
- Wang, J., 2005, 3-D Least-squares wave equation AVP/AVA migration of common azimuth data: PhD thesis dissertation, Univ. of Alberta.
- Wang, R., and S. Treitel, 1973, The determination of digital Wiener filters by means of gradient methods: *Geophysics*, **38**, 310-326.
- Whitmore, N. D., 1983, Iterative depth migration by backward time propagation: 53rd Annual International Meeting, SEG Expanded Abstract, 382-385.
- “Circulant matrix.” Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 30 Nov. 2012. Web. 28 Dec. 2012. <http://en.wikipedia.org/wiki/Circulant_matrix>
- Yilmaz, Oz, 2008, *Seismic data analysis: processing, inversion, and interpretation of seismic data*: SEG publications.
- Yousefzadeh, A., and J. C. Bancroft, 2010a, Feasibility of using multigrid methods for solving least squares prestack Kirchhoff migration equation: CSEG Expanded Abstracts, 2010.
- Yousefzadeh, A., and J. C. Bancroft, 2010b, Solving least squares Kirchhoff migration using multigrid methods: 80th Annual International Meeting, SEG Expanded Abstract, **29**, 3135-3139.
- Yousefzadeh, A., and J. C. Bancroft, 2011, Writing Kirchhoff migration/modelling in a matrix form, CREWES Annual Research Report, Vol. **23**.
- Yousefzadeh, A., and J. C. Bancroft, 2012a, Kirchhoff imaging in a matrix form. Part I: modelling, CSEG Expanded Abstracts, 2012.
- Yousefzadeh, A., and J. C. Bancroft, 2012b, Kirchhoff imaging in a matrix form. Part II: least squares Migration, CSEG Expanded Abstracts, 2012.
- Yousefzadeh, A., and J. C. Bancroft, 2012c, LSPSM/inversion for pre- and poststack time lapse studies: CREWES Annual Research Report, Vol. **24**.

- Yousefzadeh, A., and J. C. Bancroft, 2012d, Velocity evaluation using least squares prestack migration: 82nd Annual International Meeting, SEG Expanded Abstract.
- Yousefzadeh, A., J. C. Bancroft, and G. F. Margrave, 2011, Migration velocity analysis of incomplete/irregular data using least squares Kirchhoff migration, CREWES Annual Research Report, Vol. **23**.
- Yousefzadeh, A., J. C. Bancroft, and G. F. Margrave, 2012, Migration velocity analysis of incomplete data using LSPSM CIGs, CSEG Expanded Abstracts, 2012.
- Yu J., and G. T. Schuster, 2003, Migration deconvolution vs. least squares migration: SEG Expanded Abstract, **22**, 1047-1050.
- Yu J., J. Hu, G. T. Schuster, and R. Estill, 2006, Prestack migration deconvolution: Geophysics, **71**, S53-S62.

Appendix A: LSCG ALGORITHM FOR SOLVING DAMPED LSPSM EQUATION

The LSCG algorithm shown in table 3.6 can be extended to solve the damped LSPSM equation $(\mathbf{G}^T \mathbf{G} + \mu \mathbf{I}) \mathbf{m}_{DSL} = \mathbf{G}^T \mathbf{d}$, where \mathbf{I} stands for the identity matrix and \mathbf{m}_{DSL} stands for the damped least square solution. Like the LSCG algorithm in table 3.6, this algorithm performs one seismic modeling run and one migration run at each iteration.

$\mathbf{m}_0 =$ an initial guess or $\mathbf{m}_0 = \mathbf{0}$

$$\mathbf{s}_0 = \mathbf{d} - \mathbf{G}\mathbf{m}_0$$

$$\tilde{\mathbf{s}}_0 = \mu \mathbf{I}\mathbf{m}_0$$

$$\mathbf{r}_0 = \mathbf{G}^T \mathbf{s}_0 + \mu \mathbf{I}\tilde{\mathbf{s}}_0$$

$$\mathbf{p}_0 = \mathbf{r}_0$$

$$\mathbf{q}_0 = \mathbf{G}\mathbf{p}_0$$

$$\tilde{\mathbf{q}}_0 = \mu \mathbf{I}\mathbf{p}_0$$

For $i = 1$: iterations limit

$$\omega_0 = \mathbf{r}_{i-1} \cdot \mathbf{r}_{i-1}$$

$$\alpha = \frac{\omega_0}{\mathbf{q}_{i-1} \cdot \mathbf{q}_{i-1} + \tilde{\mathbf{q}}_{i-1} \cdot \tilde{\mathbf{q}}_{i-1}}$$

$$\mathbf{m}_i = \mathbf{m}_{i-1} + \alpha \mathbf{p}_{i-1}$$

$$\mathbf{s}_i = \mathbf{s}_{i-1} - \alpha \mathbf{q}_{i-1}$$

$$\tilde{\mathbf{s}}_i = \tilde{\mathbf{s}}_{i-1} - \alpha \tilde{\mathbf{q}}_{i-1}$$

$$\mathbf{r}_i = \mathbf{G}^T \mathbf{s}_i + \mu \mathbf{I}\tilde{\mathbf{s}}_i$$

$$\omega = \mathbf{r}_i \cdot \mathbf{r}_i$$

$$\beta = \frac{\omega}{\omega_0}$$

$$\omega_0 = \omega$$

$$\mathbf{p}_i = \mathbf{r}_i + \beta \mathbf{p}_{i-1}$$

$$\mathbf{q}_i = \mathbf{G}\mathbf{p}_i$$

$$\tilde{\mathbf{q}}_i = \mu \mathbf{I}\mathbf{p}_i$$

Endfor

Appendix B: LSCG ALGORITHM FOR SOLVING SMOOTH LSPSM EQUATION

The LSCG algorithm to solve a regularized LSPSM with smoothing in the offset direction as the regularization term, $\mathcal{R}(m) = \|\mathbf{D}_h \mathbf{m}\|_2$, requires inverting the $\mathbf{G}^T \mathbf{G} + \lambda^2 \mathbf{D}_h^T \mathbf{D}_h$ term in the equation $(\mathbf{G}^T \mathbf{G} + \lambda^2 \mathbf{D}_h^T \mathbf{D}_h) \mathbf{m}_{SSL} = \mathbf{G}^T \mathbf{d}$, where \mathbf{m}_{SSL} is the smooth least squares solution. This algorithm is shown below. Iteration continues until the desired model is obtained, or a predefined number of iterations have been executed. Usually more than 10 and less than 20 iterations are required to get an acceptable solution. A higher amount of the regularization weight changes the matrix condition number and makes the convergence slower.

$\mathbf{m}_0 =$ an initial guess or $\mathbf{m}_0 = \mathbf{0}$

$$\mathbf{s}_0 = \mathbf{d} - \mathbf{G}\mathbf{m}_0$$

$$\tilde{\mathbf{s}}_0 = \mathbf{D}_h \mathbf{m}_0$$

$$\mathbf{r}_0 = \mathbf{G}^T \mathbf{s}_0 + \lambda \mathbf{D}_h^T \tilde{\mathbf{s}}_0$$

$$\mathbf{p}_0 = \mathbf{r}_0$$

$$\mathbf{q}_0 = \mathbf{G}\mathbf{p}_0$$

$$\tilde{\mathbf{q}}_0 = \lambda \mathbf{D}_h \mathbf{p}_0$$

For $i = 1$: iterations limit

$$\omega_0 = \mathbf{r}_{i-1} \cdot \mathbf{r}_{i-1}$$

$$\alpha = \frac{\omega_0}{\mathbf{q}_{i-1} \cdot \mathbf{q}_{i-1} + \tilde{\mathbf{q}}_{i-1} \cdot \tilde{\mathbf{q}}_{i-1}}$$

$$\mathbf{m}_i = \mathbf{m}_{i-1} + \alpha \mathbf{p}_{i-1}$$

$$\mathbf{s}_i = \mathbf{s}_{i-1} - \alpha \mathbf{q}_{i-1}$$

$$\tilde{\mathbf{s}}_i = \tilde{\mathbf{s}}_{i-1} - \alpha \tilde{\mathbf{q}}_{i-1}$$

$$\mathbf{r}_i = \mathbf{G}^T \mathbf{s}_i + \lambda \mathbf{D}_h^T \tilde{\mathbf{s}}_i$$

$$\omega = \mathbf{r}_i \cdot \mathbf{r}_i$$

$$\beta = \frac{\omega}{\omega_0}$$

$$\omega_0 = \omega$$

$$\mathbf{p}_i = \mathbf{r}_i + \beta \mathbf{p}_{i-1}$$

$$\mathbf{q}_i = \mathbf{G}\mathbf{p}_i$$

$$\tilde{\mathbf{q}}_i = \lambda \mathbf{D}_h \mathbf{p}_i$$

Endfor

Appendix C: BICG ALGORITHM

The BiConjugate Gradient (BiCG) method was developed for solving non symmetric matrices. However, it is numerically unstable. The algorithm is shown here. Each iteration requires one matrix vector multiplication of the form $(GG^T) \mathbf{p}$, where \mathbf{p} is a model vector (in the line marked by " \Leftarrow "). A difficulty arises in the $G^T \mathbf{p}$ multiplication: migration of a model vector is neither mathematically nor geophysically possible. Therefore, this method is not applicable to solve an LSPSM when using only operator forms of the migration and modeling. BiCGSTAB does not require this multiplication and is more stable.

$\mathbf{m}_0 =$ an initial guess or $\mathbf{m}_0 = \mathbf{0}$

$$\mathbf{r}_0 = G^T \mathbf{d}$$

$$\mathbf{d} = G \mathbf{m}_0$$

$$\mathbf{r}_0 = \mathbf{r}_0 - G^T \mathbf{d}$$

$$\rho_0 = 1$$

$$\tilde{\mathbf{r}}_0 = \mathbf{r}_0$$

For $i = 1$: iterations limit

$$\rho = \tilde{\mathbf{r}}_{i-1} \cdot \mathbf{r}_{i-1}$$

$$\beta = \frac{\rho}{\rho_0}$$

$$\rho_0 = \rho$$

$$\mathbf{p}_i = \mathbf{r}_i + \beta \mathbf{p}_{i-1}$$

$$\tilde{\mathbf{p}}_i = \tilde{\mathbf{r}}_i + \beta \tilde{\mathbf{p}}_{i-1}$$

$$\sigma = \tilde{\mathbf{p}}_i \cdot G^T G \mathbf{p}_i$$

$$\alpha = \frac{\rho}{\sigma}$$

$$\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha G^T G \mathbf{p}_i$$

$$\tilde{\mathbf{r}}_i = \tilde{\mathbf{r}}_{i-1} - \alpha G G^T \tilde{\mathbf{p}}_i \quad \Leftarrow$$

$$\mathbf{m}_i = \mathbf{m}_{i-1} + \alpha \mathbf{p}_i$$

Endfor

Appendix D: BICGSTAB ALGORITHM FOR SOLVING LSPSM

The BiConjugate Gradient Stabilized (BiCGSTAB) method (Van der Vorst, 1992) is a developed method that has the advantages of both, the BiCG method, since it works with non-symmetric matrices, and the CGS method, since the algorithm does not require the multiplication of the transpose matrix ($\mathbf{G}\mathbf{G}^T$) with a vector.

\mathbf{m}_0 = an initial guess or $\mathbf{m}_0 = \mathbf{0}$

$$\mathbf{r}_0 = \mathbf{G}^T \mathbf{d}$$

$$\mathbf{d} = \mathbf{G}\mathbf{m}_0$$

$$\mathbf{r}_0 = \mathbf{r}_0 - \mathbf{G}^T \mathbf{d}$$

$$\omega_0 = \beta_0 = \alpha = 1$$

$$\mathbf{p}_0 = \mathbf{r}_0$$

$$\mathbf{q}_0 = \mathbf{v}_0 = \mathbf{0}$$

For $i = 1$: iterations limit

$$\beta = \mathbf{p}_{i-1} \cdot \mathbf{r}_{i-1}$$

$$\omega = \frac{\beta}{\beta_0} \times \frac{\omega_0}{\alpha}$$

$$\beta_0 = \beta$$

$$\mathbf{q}_i = \mathbf{r}_{i-1} + \omega(\mathbf{q}_{i-1} - \alpha\mathbf{v}_{i-1})$$

$$\mathbf{d} = \mathbf{G}\mathbf{q}_i$$

$$\mathbf{v}_i = \mathbf{G}^T \mathbf{d}$$

$$\omega = \frac{\beta}{\mathbf{p}_i \cdot \mathbf{v}_i}$$

$$\omega_0 = \omega$$

$$\mathbf{s}_i = \mathbf{r}_{i-1} - \omega\mathbf{v}_i$$

$$\mathbf{d} = \mathbf{G}\mathbf{s}_i$$

$$\mathbf{t}_i = \mathbf{G}^T \mathbf{d}$$

$$\alpha = \frac{\mathbf{t}_i \cdot \mathbf{s}_i}{\mathbf{t}_i \cdot \mathbf{t}_i}$$

$$\mathbf{m}_i = \mathbf{m}_{i-1} + \omega\mathbf{q}_i + \alpha\mathbf{s}_i$$

$$\mathbf{r}_i = \mathbf{s}_i - \alpha\mathbf{t}_i$$

Endfor

**Appendix E: DEMONSTRATING THE POSITIVE DEFINITENESS OF THE SECOND
DIFFERENCE MATRIX**

The second difference matrix, \mathbf{A} , in equation 3.64 is positive definite if, for any non-zero column vector \mathbf{m} with all real entries ($m \in \mathbb{R}$), $\mathbf{m}^T \mathbf{A} \mathbf{m} > 0$. Consider \mathbf{A} be a 4 by 4 matrix as,

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix},$$

and,

$$\mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix}.$$

Then,

$$\mathbf{m}^T \mathbf{A} \mathbf{m} = [m_1 \quad m_2 \quad m_3 \quad m_4] \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix}$$

$$\begin{aligned} &= 2m_1^2 - m_1m_2 - m_1m_2 + 2m_2^2 - m_2m_3 - m_2m_3 + 2m_3^2 - m_3m_4 - m_3m_4 + 2m_4^2 \\ &= m_1^2 + m_1^2 - 2m_1m_2 + m_2^2 + m_2^2 - 2m_2m_3 + m_3^2 + m_3^2 - 2m_3m_4 + m_4^2 + m_4^2 \\ &= m_1^2 + (m_1 - m_2)^2 + (m_2 - m_3)^2 + (m_3 - m_4)^2 + m_4^2. \end{aligned}$$

This is nonnegative and is zero if and only if $m_1 = m_2 = m_3 = m_4 = 0$. Since \mathbf{m} is a nonzero vector, $\mathbf{m}^T \mathbf{A} \mathbf{m}$ is always positive. Consequently, the second difference is a positive definite matrix. Any equation of the form of $\mathbf{A} \mathbf{x} = \mathbf{b}$ is solvable by the CG method.