University of Alberta

Library Release Form

Name of Author: Somanath Misra
Title of Thesis: Global optimization with application to Geophysics
Degree: Doctor of Philosophy
Year this Degree Granted: 2008

Somanath Misra

University of Alberta

Global optmization with application to Geophysics

by

Somanath Misra ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**

in

Geophysics

Department of Physics

Edmonton, Alberta
Fall 2008

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Global optimization with application to Geophysics** submitted by Somanath Misra in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Dr. Mauricio D. Sacchi (Supervisor)

Dr. Jeff Gu

Dr. Marek Reformat

Dr. Robert J. Ferguson (External Examiner)

Dr. Frances Fenrich (Chair)

Date:

To

Suchana, Shruti and Swati

Abstract

The thesis presents an improved global optimization scheme for applications in geophysical optimization problems with large model dimension. The importance of regularization in the geophysical inversion problems is discussed with particular emphasis on enforcement of edge preserving regularization.

In the first part, I am dealing with an optimization problem where the cost function surface is unknown. I have introduced a new method based on simulated annealing optimization to estimate the phase of the embedded wavelet in the seismic data.

The second part of the thesis deals with optimization schemes in the context of different regularization constraints. A new global optimization technique based on model space preconditioning is developed to enforce blockyness in the estimated model. The model space preconditioning is achieved by means of nonlinear edge preserving operators such that the global optimization algorithm, rather than relying completely on the random perturbations, samples a favorably biased model space. The new approach is studied on optimization of one dimensional earth elastic parameters from amplitude variation with offset (AVO) seismic data. The results, with and without applications of model preconditioning operators, are compared. A linearized inversion scheme is also presented for the estimation of elastic parameters from AVO data. The thesis shows a comparison of results when the linearized inversion has failed and global optimization has succeeded in estimating the elastic parameters from AVO data. The results are further validated by a careful comparison of well log data with the estimated elastic parameters from AVO data.

Last part of the thesis deals with the application of model preconditioning based

global optimization in optimizing over a two dimensional model space. Classical global optimization schemes have very slow convergence when the model dimension is large. Model preconditioning based global optimization has paved the way for optimization involving large model space. I have successfully applied the scheme to optimize for the interval velocity and density over a two dimensional grid via waveform inversion.

ACKNOWLEDGEMENTS

I express my sincere gratitude to Prof. Mauricio D. Sacchi for his unflinching support, patience and guidance as I walked the path of research. Dr. Sacchi's in-depth understanding of the subject matter, up-to-date knowledge about geophysical literature and above all, his unique style of simplifying complex problems, have helped me a great deal as I progressed through my PhD program. I would also like to thank Dr. Sacchi for all his good wishes, concerns and helps. Thank you, Mauricio, for all that you have done for me and for everything that you are!

I would also like to thank the faculty members of the University of Alberta who helped me learn so much. I thank Dr. Jeff Gu, who helped me understand the fundamentals of Seismology. I thank Dr. Konstantin Kabin, who helped me learn scientific computing. I thank Dr. Lawrence Lee for teaching me digital signal processing.

I thank Dr. Ulrych Theune for being so forthcoming in all his helps throughout. I thank Christian Escalante for being such a helpful friend. I learnt a lot many things from my association with the Signal Analysis and Imaging Group (SAIG). I have made some wonderful friends during my stint at the SAIG. I take this opportunity to name and thank a few of them in the alphabetical order- Abdolnaser Yousefzadeh, Bin Liu, Cristina Moldoveanu-Constantinescu, Jiang Feng, Juefu Wang, Mingyu Zhang, Mostafa Naghizadeh, Sam Kaplan, Sharareh Karmand and Soner Bekleric.

I worked as a Geophysicist at the Directorate of Geology, Orissa, India, prior to joining the PhD program at the University of Alberta. I take this opportunity to thank Mr. R. C. Moharana, the then Director for all his helps in granting me the necessary permission to pursue PhD program in Canada. I thank Mr. K. C. Mohapatra at the Directorate of Geology for all his helps. I thank Mr. B. K. Sahu at the Directorate of Geology for being such a wonderful company.

I take this opportunity to thank Mr. Satinder Chopra for all his helps. I also thank, in alphabetical order, Angela Truong, Bill Perry, Carter Edie, Danny Bao, Florencia Vignolo, Muyi Kola-Ojo, Rob Pinnegar, Samo Cilensek, Vlada Avramovic and Xu Yong for being such wonderful company.

The path would have been a lot more difficult to tread without the presence of my wife, Suchana. Thanks a lot for being so understanding. Thanks a lot for all the compromises that you made during the course of my long journey. I have two daughters, Shruti and Swati. They are my strength and my weakness. Thanks, Shruti, for being an inseparable part of my life and for your understanding for all

those that you desired and deserved as a kid yet you missed! My second daughter, Swati, was born when I was almost two years into the PhD program. She grew up from a baby to a very loving kid. Thanks, Swati, for coming to my life and brightening it. Thanks a lot for your silent acceptance of what was being offered to you in place of what you had wished. Thanks to my parents for their support from time to time.

Contents

List of Tables

List of Figures

List of symbols and abbreviations

Symbol/Abbrevation	**Name/Description**
$\mathbf{d}, d_t$	Observed seismic data
$\mathbf{G}$	Forward operator (linear model-data relation)
f, J	Cost function
$\mathbf{d}_p$	Predicted data
$\tilde{\mathbf{m}}$	Estimated model
$\mathbf{m}$	Model
$\boldsymbol{\lambda}$	Lagrange multiplier
μ, δ	Trade-off parameters
$\mathbf{g}$	Forward operator (nonlinear model-data relation)
$\mathbf{m}_0, \mathbf{m}^*$	Reference model
Δ	Small perturbation
$\nabla, \mathbf{g}$	Gradient operator, gradient vector
$\mathbf{S}$	Search direction
α, β	Step size
$\mathbf{H}$	Hessian matrix
P	Metropolis probability
T	Cost temperature
M	Model dimension
i, k	SA/VFSA iteration
A_i, B_i	Upper and lower model bounds.
T_i	Model parameter temperature
T_{0i}, T_{fi}	Initial and final model parameter temperature
Q	Quenching parameter
r_t	Reflectivity series
w_t	Mixed phase wavelet
$\tilde{w}_t$	Minimum phase wavelet
f_t	All-pass operator

Symbol/Abbrevation	**Name/Description**
Z	Z-transform
θ	Phase, Incidence angle
C_4^s, C_4^γ, C_4^ν	4^{th} order trace, reflectivity and wavelet cumulant
M_4^w	4^{th} order wavelet moment
τ	Lag
σ	Standard deviation of noise
$\mathbf{D}$	Derivative operator matrix
P()	Probability distribution function
R()	Regularization term
$*$	Convolution
$\mathbf{V}_p$, $\mathbf{V}_s$, $\boldsymbol{\rho}$	P-, S-wave velocities, density
N_{off}	Number of offset
N_t	Number of time samples
$\mathbf{n}$	Noise
AVO	Amplitude Versus Offset
CG	Conjugate Gradient
NLCG	Nonlinear Conjugate Gradient
SA	Simulated Annealing
VFSA	Very Fast Simulated Annealing
GA	Genetic Algorithm
SNR	Signal-to-Noise Ratio
HFR	High Frequency Restoration
AVA	Amplitude Versus Angle
CPU	Central Processing Unit
Hz	Hertz
NMO	Normal Move-Out
EPS	Edge-Preserving Smoothing
EPR	Edge-Preserving Regularization
MP	Model Preconditioning operator
MP-VFSA	Model Preconditioning based Very Fast Simulated Annealing

Chapter 1

Introduction

Science of optimization forms the backbone of any inverse problem. Inversion is a mean of estimating the unknown from what is known or measured. The unknowns are the model parameters. The objective of inversion is to obtain the best acceptable estimation of the model parameters that satisfy the data and the imposed constraints. In order to do so, a criterion is chosen for comparing different available model parameters and then choosing the best one. Such a criterion is known as the cost function or the objective function. In inverse problems, the cost function is a measure between the misfit in the observed data and the estimated data obtained from the model parameters. In most of the inverse problems, there may be more than one criterion to be satisfied to obtain an acceptable estimation of the model parameters. For example, we might want to obtain an estimation of the model parameters that are smooth along the lateral direction. Imposition of such constraints in the model space results in the regularization of the inverse problem that allows to obtain a solution that is physically reasonable and consistent with the *a priori* information.

Depending on the relation between model parameters and data, an inverse problem is broadly classified into two categories, namely, (a) linear and (b) nonlinear. In the following sections, I discuss each categories of inverse problems.

1.1 Linear/linearized inverse problems

A linear function between model and data makes the inverse problem linear and simpler compared to a nonlinear function which complicates inversion and hence optimization. A linear inverse problem is formulated as follows

$$\mathbf{d} = \mathbf{G}\mathbf{m} + \mathbf{n}, \tag{1.1}$$

where $\mathbf{d}$ is the vector of measured data, $\mathbf{m}$ is the vector of unknown model parameters and $\mathbf{n}$ is the additive noise. The operator $\mathbf{G}$ linearly relates the observed data with unknown model parameters. Such an inverse problem has an explicit form of linear relationship between model and data. Depending on the amount of information content in the linear relationship $\mathbf{d} = \mathbf{G}\mathbf{m} + \mathbf{n}$, an inverse problem is classified into two categories namely, (a) overdetermined problem and (b) underdetermined problem (Menke, 1984).

1.1.1 Overdetermined problem

An inverse problem is classified as "overdetermined" if the linear function between the model and data $\mathbf{d} = \mathbf{G}\mathbf{m}+\mathbf{n}$ contains more than required information to exactly determine the solution. In general, such a situation arises when there exist more known parameters (data) than the unknown parameters (model). A simple example of overdetermined problem is the fitting of a straight line to more than two data points, assuming that the data points are not collinear. An overdetermined problem is solved by the least-squares approach.

Least-squares approach is based on minimization of a norm defined as the sum of the square of errors between the observed and the predicted data. Such a norm is commonly referred to as the L_2-norm. The L_2-norm between the observed and the predicted data is given by

$$J = (\mathbf{d_p} - \mathbf{d})^T(\mathbf{d_p} - \mathbf{d}), \tag{1.2}$$

where $\mathbf{d_p} = \mathbf{G}\mathbf{m}$ represents the predicted data for the model $\mathbf{m}$. Equation 1.2 can be written as

$$J = (\mathbf{G}\mathbf{m} - \mathbf{d})^T(\mathbf{G}\mathbf{m} - \mathbf{d}). \tag{1.3}$$

Minimum of the above set of linear equations is obtained by computing the derivatives with respect to the model parameters and equating them to zero. After differentiation and rearrangement of terms we obtain

$$\tilde{\mathbf{m}} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{d}, \tag{1.4}$$

where $\tilde{\mathbf{m}}$ represents the estimated model obtained while assuming that $(\mathbf{G}^T\mathbf{G})^{-1}$ exists. Equation 1.4 represents the well known least-squares solution to the overdetermined linear inverse problem.

1.1.2 Underdetermined problem

An inverse problem is underdetermined when there exists less than required information in the linear model-data relationship $\mathbf{d} = \mathbf{Gm} + \mathbf{n}$ to exactly determine the solution (Menke, 1984). Such a situation usually arises when there are less number of data points compared to the number of unknowns. A simple example is to fit a straight line given that there exists only one datum. Such a problem has infinitely many solutions and can not be solved with the least-squares approach.

An underdetermined inverse problem requires regularization. The idea is to select a particular suite of solutions from infinitely possible solutions that honor the data. For example, in case of fitting a straight line to a single datum, it might be known *a priori* that the straight line passes through a known point, say the origin. Such an *a priori* information is enough to solve the problem. Incorporation of *a priori* knowledge about the model space help in reducing the domain of the solution such that a solution is obtained that belong to a particular class of solutions that fit the data and at the same time honor the *a priori* information. An elegant way of assuming *a priori* information about the model is to enforce a measure of simplicity in the solution. The idea is that among many solutions, the most pertinent solution is the one that is simple. Distance is a measure of simplicity. A preferred measure of the distance and hence simplicity of the model solution is given by the quadratic function $L = \mathbf{m}^T\mathbf{m}$. Solution to the regularized optimization problem where the model solution is simple and at the same time honors the data is solved by the approach involving Lagrange multipliers. The constrained optimization function is given by

$$J = \mathbf{m}^T\mathbf{m} + \boldsymbol{\lambda}^T(\mathbf{Gm} - \mathbf{d}), \tag{1.5}$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier. The first part of the above equation is a measure of the model simplicity and the second part is a measure of the data misfit. Taking the derivative of equation 1.5 with respect to the model and equating the resulting equation to zero yields

$$\mathbf{m} = -\frac{1}{2}\mathbf{G}^T\boldsymbol{\lambda}. \tag{1.6}$$

Substituting for $\mathbf{m}$ by $\boldsymbol{\lambda}$ and satisfying the condition that $\mathbf{d} - \mathbf{Gm} = \mathbf{0}$ yields

$$\boldsymbol{\lambda} = -2(\mathbf{GG}^T)^{-1}\mathbf{d}. \tag{1.7}$$

Thus from equation 1.6, we obtain the estimated model $\tilde{\mathbf{m}}$ given by

$$\tilde{\mathbf{m}} = \mathbf{G}^T(\mathbf{GG}^T)^{-1}\mathbf{d}. \tag{1.8}$$

It is evident from equation 1.8 that for an underdetermined problem the solution fits the data exactly. This is an undesirable consequence when the data are contaminated with noise. Ideally, the solution should fit the data only to the extent that the error between observed and predicted data (data misfit) contained the noise present in the data. Such an ideal solution is hard to obtain. In order to obtain a compromise between the data misfit and the model norm, a trade-off parameter (μ) is chosen that allows for different weights to be applied on the data misfit term and the model norm. When μ is zero, the solution will not represent the information regarding the *a priori* knowledge. On the other hand, when μ is set to a very high value, the solution will heavily tilt towards a model that honors the *a priori* knowledge but not the data misfit. Thus, the value of the trade-off parameter needs to be so chosen that the resulting solution provides a good balance between the model norm and the data misfit. There exists no simple method to know beforehand the value of an ideal trade-off parameter and more often than not, the trade-off parameter should be determined on a trial and error basis (Menke, 1984). The cost function containing the model norm and the data misfit is given by

$$J = \mu\mathbf{m}^T\mathbf{m} + (\mathbf{Gm} - \mathbf{d})^T(\mathbf{Gm} - \mathbf{d}), \tag{1.9}$$

where μ represents the trade-off parameter that regulates the importance of data misfit and the model norm in the optimization process. Taking the derivative of equation 1.9 with respect to the model and setting the resulting equation to zero, we obtain

$$\mu\mathbf{Im} + (\mathbf{G}^T\mathbf{Gm} - \mathbf{G}^T\mathbf{d}) = 0, \tag{1.10}$$

where $\mathbf{I}$ represents the identity matrix. Thus solving for the unknown model $\tilde{\mathbf{m}}$ yields

$$\tilde{\mathbf{m}} = (\mathbf{G}^T\mathbf{G} + \mu\mathbf{I})^{-1}\mathbf{G}^T\mathbf{d}. \tag{1.11}$$

The estimated model is known as the damped least-squares solution.

So far I have discussed inverse problems involving explicit linear relationship between model and data. However, most of the inverse problems are not explicitly linear. In the following discussion, I delve into those inverse problems that are not apparently linear but can be linearized.

1.1.3 Linearized inverse problems

Certain inverse problems that are not explicitly linear, can be linearized in terms of model and data perturbations. Let the forward operator relating the data and model be represented by

$$\mathbf{d} = \mathbf{g}(\mathbf{m}), \tag{1.12}$$

where $\mathbf{g}$ is the nonlinear operator relating the model with the data. Let the measured data $\mathbf{d}$ be obtained from a model $\mathbf{m} = \mathbf{m_0} + \Delta\mathbf{m}$. Where $\Delta\mathbf{m}$ is the model perturbation with respect to a reference model $\mathbf{m_0}$. Thus

$$\mathbf{d} = \mathbf{g}(\mathbf{m_0} + \Delta\mathbf{m}). \tag{1.13}$$

Let the predicted data $\mathbf{d_p}$ for the reference model be given by

$$\mathbf{d_p} = \mathbf{g}(\mathbf{m_0}). \tag{1.14}$$

Expanding equation 1.13 in a Taylor series at $\mathbf{m_0}$, we obtain

$$\mathbf{g}(\mathbf{m_0} + \Delta\mathbf{m}) = \mathbf{g}(\mathbf{m_0}) + \frac{\partial \mathbf{g}(\mathbf{m_0})}{\partial \mathbf{m}}|_{\mathbf{m}=\mathbf{m_0}}\Delta\mathbf{m} + O(|\Delta\mathbf{m}|^2). \tag{1.15}$$

Ignoring the higher-order terms involving model perturbation we obtain

$$\mathbf{d} = \mathbf{d_p} + \mathbf{G_l}\Delta\mathbf{m}, \tag{1.16}$$

where the elements of the matrix $\mathbf{G_l}$ are the partial derivatives of the predicted data computed with respect to the model. The above equation can be written in terms of the model perturbation and data perturbation as follows

$$\Delta\mathbf{d} = \mathbf{G_l}\Delta\mathbf{m}. \tag{1.17}$$

Equation 1.17 is linear in terms of model and data perturbations.

Certain inverse problems are linearizable via approximations of the forward operator. For example, estimation of earth elastic parameters from AVO data using a forward operator that involves computation of reflection coefficients from Zoeppritz equations is a nonlinear problem. However, for a reasonable angle of reflection, the Zoeppritz equations can be approximated to obtain a linear form. Aki-Richards equation (Aki and Richards, 1980) is one of such linear representations of the Zoeppritz equations.

Not all inverse problems can be expressed in a linear or linearizable form. Nonlinear relationship between the model and data lead to a class of inverse problems known as ”nonlinear inverse problem”.

1.2 Nonlinear inverse problems

As mentioned before, nonlinear inverse problems belong to a class of inverse theory where there exists nonlinearity in the model-data relationship. Nonlinear relation between model and data makes the inverse problem more complex compared to the inverse problems with linear model-data relationship. Nonlinear model-data relationship leads to nonquadratic cost function as opposed to the linear model-data relationship where the cost function is quadratic. The cost function topology is likely to be multimodal in case of nonquadratic cost function. Optimization of such cost function is complex because of the presence of several minima in the cost function surface.

In the following sections, I discuss important gradient based methods and global optimization methods to solve for nonlinear inverse problems.

1.2.1 Iterative least-squares method

Certain nonlinear inverse problems have the advantage that the minimum of the cost function can be reached iteratively through local linearization processes. The iterative least-squares technique starts from a suitably chosen initial model. In each iteration, the model is updated by obtaining a solution of the linearized inverse problem (equation 1.17) by means of constrained least-squares approach (equation 1.11). The iterations for updating the model is terminated when the least-squares error is reduced to a value less than or equal to a predefined value (Lines and

Treitel, 1984). At each iteration of the iterative least-squares method, the algorithm approximates the cost function topology to a paraboloid tangent to that point on the cost function surface that corresponds to the current model (Menke, 1984). The updated model in the next iteration is the point that corresponds to the minimum of the approximated paraboloid that corresponded to the model at the previous iteration. Figure 1.1 shows the convergence of iterative least-squares algorithm with respect to a cost function exhibiting multimodality. McAulay(1985) applied the iterative least-squares method to seismic waveform inversion.

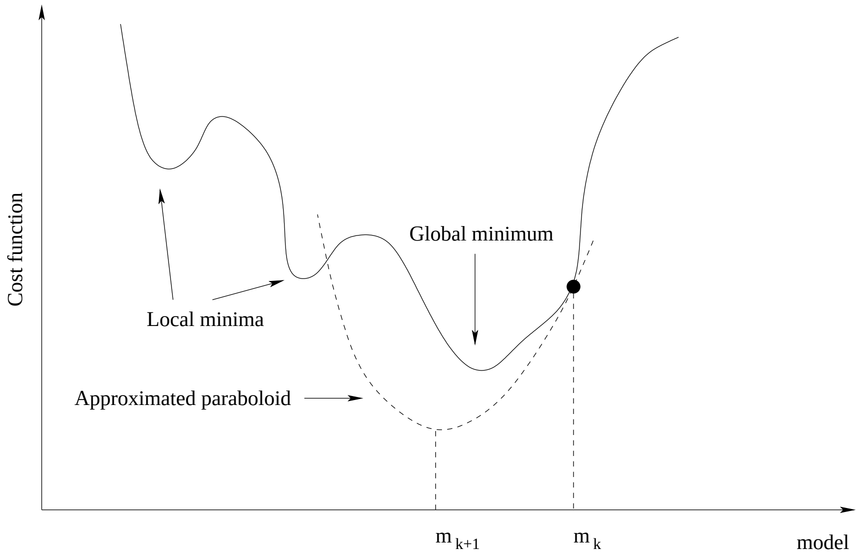

Figure 1.1: The convergence of iterative least-squares algorithm. The model m_k at the iteration k gets updated to the model m_{k+1} at the iteration $k+1$.

Success of iterative least-squares method for nonlinear inverse problems depends on the chosen initial model. If the initial model is close to the global minimum of the cost function then there is greater likelihood that the method will converge to the desired minimum. On the other hand, if the initial model is not close enough to the global minimum, the algorithm may become oscillatory or converge to a local minimum or a maximum depending on the location of the initial model (Menke, 1984).

1.2.2 Nonlinear conjugate gradient method

Another approach to obtain a solution of nonlinear inverse problem is to obtain a minimum of the cost function by means of nonlinear conjugate gradient algorithm.

Nonlinear conjugate gradient algorithm is a generalization of the conjugate gradient algorithm to optimize for nonlinear functions. While conjugate gradient algorithm aims at finding the solution to the system of linear equations $\mathbf{G}^T\mathbf{G}\mathbf{m} = \mathbf{G}^T\mathbf{d}$, the nonlinear conjugate gradient aims at finding a solution to a nonlinear system of equations. The success of nonlinear conjugate gradient algorithm lies in the fact that the initial model be chosen such that the paraboloid approximation of the cost function in the vicinity of the initial model encompasses the global minimum of the cost function. When this condition fails, the nonlinear conjugate gradient algorithm may not necessarily converge to the global minimum of the cost function.

So far I have discussed important concepts involving gradient based optimization schemes for linear, linearized and nonlinear inversion or optimization algorithms. Gradient based optimization schemes form an important part of the theory of inversion or optimization. In spite of their mathematical elegance and convergence properties, the gradient based minimization techniques are not completely foolproof. Optimization schemes based on computation of gradient will always converge to the nearest minimum corresponding to the initial model. Hence, such techniques are often referred to as the ”local optimization” schemes. When the cost function is quadratic, the surface of the cost function is a paraboloid containing a single minimum. The local optimization algorithms, in case of quadratic cost functions, are very efficient in finding the desired model. However, in case of a nonlinear inverse problem, the shape of the cost function becomes complex. The iterative procedures described above are likely to miss the global minimum unless the initial model is chosen so close to the global minimum that the paraboloid approximation of the cost function in the vicinity of the initial model encompasses the global minimum. Optimization of nonlinear problems to estimate the model corresponding to the global minimum thus requires a very carefully chosen initial model. A great deal of *a priori* information about the model is required so as to chose the initial model that corresponds to a point close to the global minimum of the cost function. When the *a priori* information is sparse and the inverse problem is nonlinear, the local optimization schemes are likely to fail. Most of the geophysical inverse problems, unless the forward operator is approximated, are nonlinear and the shape of the cost function surface is highly complex. In such situations global optimization algorithms, as discussed below, are employed to ”hopefully” achieve convergence to the global minimum of the cost function surface. Global optimization algorithms provide means to ”jump out” of a local minimum and allow the convergence to proceed towards the global minimum. Furthermore, the idea of linearizing a nonlinear data-model functionality so as to use a local optimization algorithm, inadvertently brings in inaccuracies to the optimization problem. Such inaccuracies are undesirable and in complex geophysical problems might adversely influence the results.

1.2.3 Global optimization methods

As mentioned in the previous section, global optimization algorithms such as the Monte Carlo method, the genetic algorithm, simulated annealing and neighborhood algorithm are employed in optimization problems where the cost function is non-quadratic and hence likely to exhibit multimodality. Monte Carlo methods are exhaustive search techniques that try to encompass the entire model space. When the model space is large, Monte Carlo technique to obtain the model corresponding to the global minimum becomes computationally very expensive and impracticable. Genetic algorithm and simulated annealing algorithms provide a directed search method to attain the global minimum. The essence of genetic algorithm lies in the process of natural evolution. With the progress of evolution, the current species are more intelligent and better adaptive to the natural circumstances compared to their previous generations. A similar analogy is drawn in the development of genetic algorithm. With the progress of generations, the fitness of the candidate models comprising the population pool increases compared to the previous generations. The population pool is updated in each generation such that only the candidates that have better fitness compared to others have a higher probability to move to the next generations. The algorithm, at the same time, provides a finite probability for the less fit candidate models to move to the next generations. This step in the algorithm provides a mean to climb uphill the cost function surface, thus helping the algorithm to ”jump out” of a local minimum. After progress through a large number of generations, the resulting population pool contains highly efficient candidate models. The solution is obtained by choosing a model from the efficient population pool. The simulated annealing algorithm, on the other hand, mimics the process of thermodynamic annealing. When the temperature of a molten solid is lowered very slowly, the chaotic motion of the molecules of the solid reduces and at a very small temperature the solid attains the minimum energy forming a crystal. The simulated annealing algorithm accepts or rejects a model with a finite probability depending upon the temperature and the cost function. The acceptance/rejection in simulated annealing algorithm constitutes what is known as the "Metropolis criterion". This step in the algorithm provides a finite probability for a model with higher cost function to get accepted for the future iteration, thus helping the algorithm to "jump out" of a local minimum. The Neighborhood Algorithm (NA), introduced by Sambridge (1999a, b), is a global optimization algorithm where simple geometrical structures or cells are used to obtain the convergence to a global minimum. The neighborhood algorithm proceeds in two stages. In the first stage the model space is searched to generate an ensemble of the model that provide good data misfit. The second stage involves exploring the model ensemble to derive information about the convergence properties.

Global optimization schemes become computationally very expensive when the model dimension becomes large. This problem is referred to as "the curse of dimensionality". The primary goal of my thesis is to address this problem effectively in applied seismological optimization problems so that the advantages of global optimization can be harnessed effectively. My approach is to incorporate *a priori* information about the model space into the global optimization algorithm such that the model space is favorably preconditioned. The preconditioned model space helps the global optimization algorithm to preferably sample the model space in consistent with the *a priori* information. I have incorporated the *a priori* information into the simulated annealing algorithm as a second stage of the optimization routine. This approach helped in favorably preconditioning the model space without losing the benefits of exhaustive search. I have discussed, with examples, the application of model preconditioning based global optimization scheme in relatively large model dimension and complex nonlinear geophysical problems.

1.3 Scope and Goals

Most of the geophysical inverse problems involve a nonlinear relationship between the data and the model. Any optimization method with a gradient based algorithm is likely to converge to a minimum that is closest to the starting point in the cost function topology. The way to get around such a shortcoming is to linearize the relation between the data and the model. Every such linearization incorporates approximations to the forward model. This adversely affects the accuracy of the solution. In the past, with limited computing facilities, there was no way but to rely on the optimization schemes based on local optimization procedures. With the advent of modern computing facilities and efficient global optimization algorithms, it is now possible to obtain better results without requiring to introduce linearization schemes to the forward modelling operator. Slow convergence rate is the primary disadvantage of any global optimization scheme. The slow convergence rate becomes more conspicuous as the model dimension increases. In geophysical problems, there are requirements when the optimization need to be performed over a large model space. In such situations, classical global optimization algorithms become prohibitively slow. The major contribution of the thesis is to use the *a priori* information in such a way that the model space is suitably preconditioned for the global optimization involving applied seismological problems. In such a case, the global optimization scheme, instead of relying completely on the random perturbations of the model parameters, optimizes over a suitably conditioned model space that is consistent with the *a priori* information. Incorporation of the *a priori* information as the second stage of the optimization scheme favorably preconditions the model space while preserving the benefits of exhaustive search obtained from the classical simulated annealing technique. I have shown examples where the new technique has operated successfully over a large model space to optimize for the subsurface model parameters via waveform inversion.

1.4 Organization of the thesis

The thesis is organized as follows. In Chapter 2, I am reviewing the local and global optimization methods while posing inversion as an optimization problem. I discuss important gradient based optimization techniques such as the steepest descent algorithm, linear and nonlinear conjugate gradient algorithms, Newton's and Marquardt methods followed by the global optimization methods such as the Monte Carlo technique, genetic algorithm and simulated annealing. Since, I am using simulated annealing as the global optimization algorithm, I will discus in greater details the simulated annealing technique along with its other variants. In Chapter 3, I discuss a new application of global optimization scheme in a blind deconvolution problem. I have used classical simulated annealing technique to optimize for the coefficients of an all-pass operator so as to estimate the mixed phase wavelet from seismic data. In Chapter 4, I am reviewing the theories involved in regularization of the inverse problem. *A priori* information about the model is incorporated into the optimization algorithm by adding regularization terms to the cost function. I review two important regularization constraints, namely, sparseness constraint and blockyness constraint. I show an example where sparseness constraint is applied to real seismic data to recover the high frequency components. I have also made a brief review of blockyness constraint. In Chapter 5, I discuss application of nonlinear conjugate gradient (NLCG) based local optimization technique to estimate the earth elastic parameters ($\mathbf{V}_p$, $\mathbf{V}_s$ and $\boldsymbol{\rho}$) from amplitude variation with offset data. In this chapter, I discuss Zoeppritz equations and their various approximations followed by ray-tracing based forward model to obtain angle and offset dependent seismic gathers. I show an example of application of NLCG algorithm in AVO inversion with the cost function supplemented with blockyness constraint. I further show that NLCG based optimization fails when the forward model involves greater degree of nonlinearity. The example shows the convergence of NLCG based optimization to a local minimum. The purpose of this example is to show the necessity of global optimization in AVO inversion problems. In Chapter 6, I discuss global optimization based AVO inversion for estimation of earth elastic parameters. The optimization involved a relatively large model space. The thesis introduces a new concept of model preconditioning based global optimization where optimization over a larger model space can be effectively achieved. I have used nonlinear edge preservation and smoothing operators to suitably precondition the model space. The model preconditioner is applied as a second stage of the optimization algorithm. In Chapter 7, I have carried forward the concept of model preconditioning based global optimization to optimize for model parameters over a two dimensional grid. This example shows the potential of applying global optimization in inverse problems where the model dimension is very large.

1.5 Contributions

The thesis proposes new method that combines *a priori* information with the classic global optimization schemes to alleviate the adverse effects due to increased model dimension. I have shown a new application of global optimization scheme in a blind deconvolution problem where the model dimension is small. The algorithm estimated the coefficients of all-pass operators efficiently. However, in the subsequent applications of global optimization, the rate of convergence suffered drastically due to the increased model dimension. In the problem of estimating the earth elastic parameters via waveform inversion of AVO data, the model parameterization necessitated introduction of pseudo-layers. This led to a large increase in the model dimension. Hence, the convergence of the algorithm became slow to the extent that the global optimization scheme became computationally prohibitive. Incorporation of *a priori* information in the global optimization scheme favorably modifies the model space. The modified model space is a subset of the entire model space that can be generated by random perturbations. A smaller model space, consistent with the *a priori* information reduces nonuniqueness in the problem and helps the algorithm converge faster. In the last part of the thesis, the algorithm was successfully tested on a very large model dimensional problem.

The thesis opens up a new approach that allows application of global optimization algorithms in applied seismological problems involving large model dimension which have so far been dealt with local optimization techniques.

Chapter 2

Inversion: Local and Global optimization

2.1 Introduction

Computers operate in bits and bytes. Hence, the world of scientific computing is discrete. The bits and bytes of computer technology have resulted in the development of an exciting branch of science known as the 'discrete inverse theory'. Theory of inversion falls into two broad categories, namely, direct inversion and indirect inversion. Inverse problems where direct analytical expressions exist for calculation of model parameters from data belong to the former category. However, for majority of geophysical inverse problems no such direct formula exists. Furthermore, geophysical data are mostly incomplete and noisy. In such situations, it is desirable to estimate the model parameters through indirect means that involve optimization of a function relating model and data. A function designed for the optimization to estimate the model parameters from data is referred to as the cost function. The function usually consists of two parts, namely, (a) the data misfit part that relates how closely or distantly an estimated model fits the observed data and (b) the regularization part that augments the cost function in such a way that the optimized solution honors the *a priori* information as well.

Most optimization processes involve iterative approach. The iterations start from an initial model that is either obtained from a random distribution or through some "smart guess". With every iteration the current model is updated. The iterations continue till the optimized model minimizes the cost function within a predefined accuracy. Figure 2.1 shows a schematic flow of an optimization process. Different optimization schemes differ from one another in the way the model is updated in each iteration.

Topology of a cost function generally comprises several local minima and global minimum. The necessary condition that a cost function f($\mathbf{m}$) will have an extremum at $\mathbf{m} = \mathbf{m}^*$ is that the partial derivatives with respect to the variable $\mathbf{m}$ are zero. The sufficient conditions for the extremum to be a minimum is that the matrix of second partial derivative of the cost function f($\mathbf{m}$) with respect to model is positive-definite and negative-definite for the extremum to be a maximum point. Optimization schemes that use gradient information of the cost function to update the model always proceed in the downhill slope of the cost function topology. Such methods are termed as "local optimization" because they always converge to the nearest minimum corresponding to the location of the initial model. In case of a multimodal

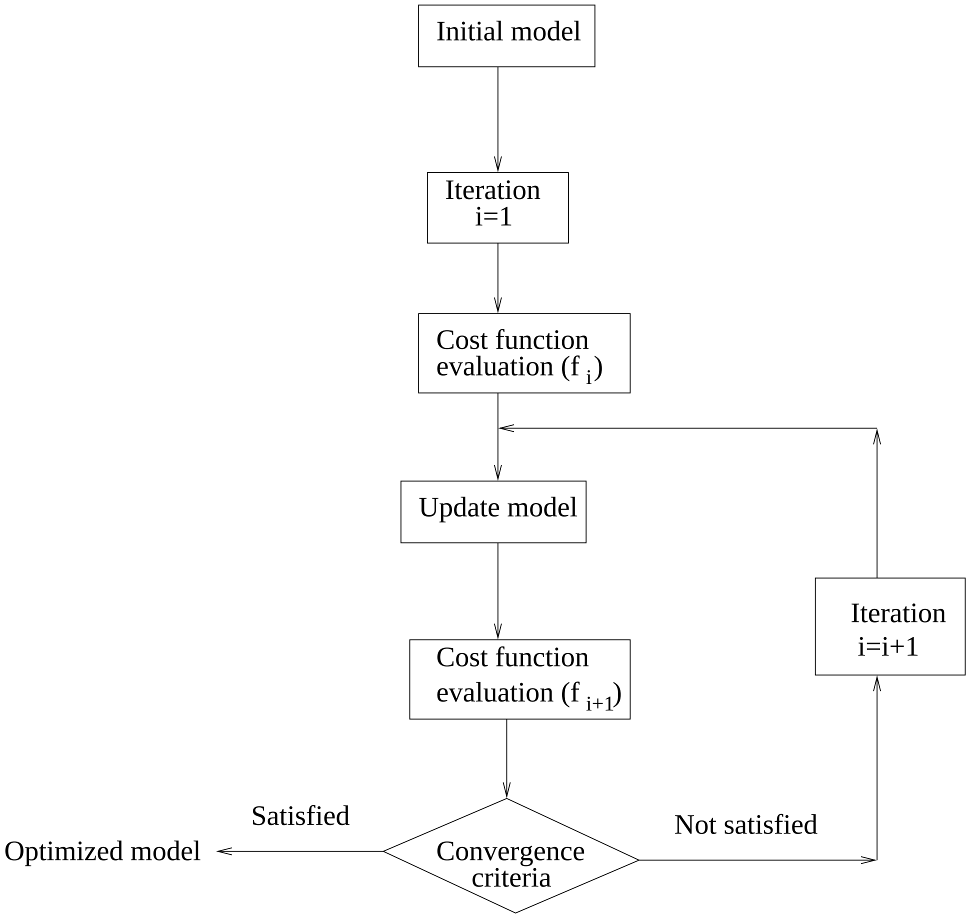

Figure 2.1: Flowchart showing optimization process.

cost function, there exists a fair chance that the convergence will lead to a local minimum unless the initial model lies within the vicinity of the global minimum. The other class of optimization schemes that aim at achieving convergence to the global minimum even in presence of multimodality, are termed as "global optimization" algorithms. Such algorithms rely on random perturbation of the model, instead of the derivative information of the cost function, to update the model. Optimization approaches based on random perturbation of the model provide the means to "jump out" of the local minimum and possibly converge to the global minimum.

Despite the shortcoming of convergence to a local minimum, the local optimization schemes are based on elegant mathematical foundation. Furthermore, the local optimization techniques are highly efficient as they make use of more information about the cost function as opposed to the global optimization schemes. On the other hand, global optimization schemes that rely completely on random perturbations are computationally very expensive. Such schemes are prohibitively slow for optimization over a large model dimension. With the advent of high performance computing facilities, the benefits of global optimization schemes are gradually making their way to more realistic optimization problems.

The table below provides a list of important local and global optimization schemes. Each of these optimization schemes will be discussed in more details in the sections to follow.

Table 2.1: Local and global optimization schemes

Local optimization	Global optimization
Steepest descent method Conjugate gradient method Nonlinear conjugate gradient method Newton's method Marquardt method	Monte Carlo method Genetic algorithm Simulated annealing methods

2.2 Local optimization

Inversion can be posed as an optimization problem. The function to be optimized is called the cost function. In this section, I discuss necessary and sufficient conditions for existence of local extrema and the algorithmic means to investigate the convergence to a possible solution that corresponds to a minimum.

The surface of a cost function may have a single, well-defined minimum or many minima. A cost function surface showing many minima is referred to as the multi-modal surface. Global minimum of a cost function is the lowest minimum among all the minima. Figure 2.2 shows a cost function surface that contains many minima and the global minimum.

The necessary condition for a point $\mathbf{m}^*$ on the cost function surface to lie on an extremum is given by the equation below.

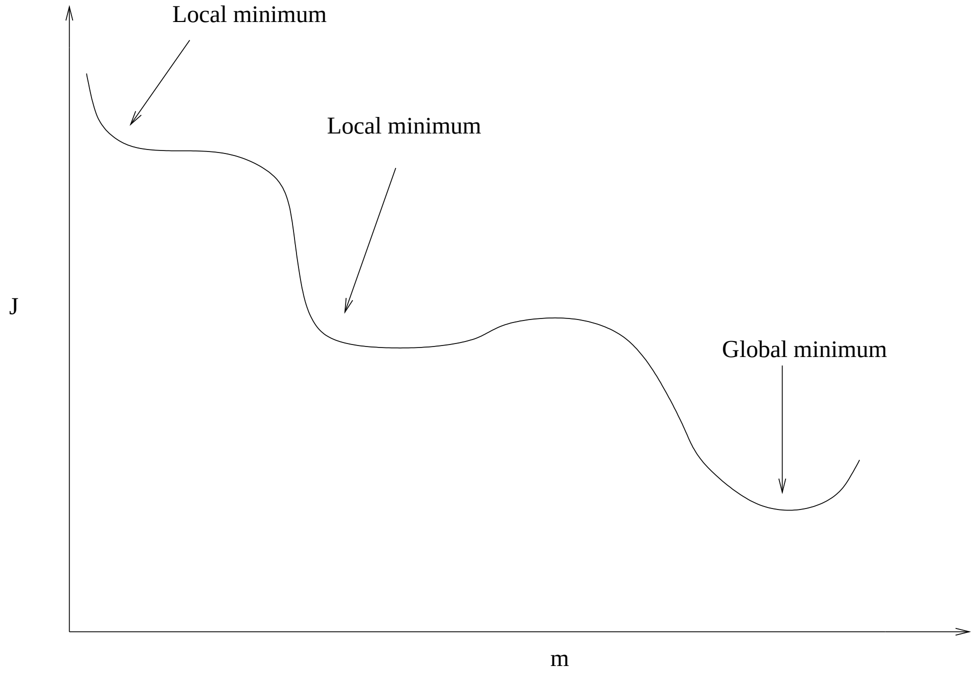

Figure 2.2: Local minima and global minimum of a cost function

$$\frac{\partial f(\mathbf{m}^*)}{\partial m_1} = \frac{\partial f(\mathbf{m}^*)}{\partial m_2} = \ldots = \frac{\partial f(\mathbf{m}^*)}{\partial m_n} = 0. \tag{2.1}$$

The sufficient condition for the point $\mathbf{m}^*$ to be a maximum is that the matrix containing the second partial derivative of the cost function with respect to the model is negative-definite at $\mathbf{m}^*$. Similarly, the sufficient condition that the point $\mathbf{m}^*$ is a minimum is that the matrix containing the second partial derivative of the cost function with respect to the model evaluated at $\mathbf{m} = \mathbf{m}^*$ is positive-definite. The proof for the necessary and sufficient conditions follow from the Taylor series expansion of the cost function $f(\mathbf{m})$ at $\mathbf{m} = \mathbf{m}^*$.

2.2.1 Convergence to a local minimum

The gradient of a function provides information about the direction of maximum increase of a function. Hence to achieve the minimum of a cost function, we proceed in the direction of negative gradient. Gradient of a multivariate function with n variables is given be the following vector representation.

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial m_1} \\ \frac{\partial f}{\partial m_2} \\ \frac{\partial f}{\partial m_3} \\ \vdots \\ \frac{\partial f}{\partial m_n} \end{bmatrix} \tag{2.2}$$

As mentioned before, gradient vector provides the direction of maximum ascent of the cost function. However, the direction of maximum ascent is dependent on the location and hence provides the direction of maximum increase corresponding to that location. Thus the direction of maximum ascent provided by the gradient is a local property rather than a global one. Figure 2.3 shows the direction of maximum local ascent as per the gradient information (Rao, 1996).

It is obvious that negative of the gradient direction provides the direction of maximum descent. Thus, a minimum of a function is obtained in the fastest way if we proceed in the direction of the negative gradient. All local optimization methods use the information provided by the gradient vector to achieve convergence.

Computation of gradient requires evaluation of partial derivatives of the cost function with respect to the model parameters m_i. Partial derivatives of a function can be evaluated analytically. However, in certain situations analytical evaluation of partial derivatives is not possible. Such situations necessitate the numerical computation of partial derivatives. Partial derivatives can be computed by the finite-difference equation as follows

$$\frac{\partial f}{\partial m_i} \simeq \frac{f(m_i + \Delta m_i) - f(m_i)}{\Delta m_i}, \tag{2.3}$$

where m_i is a component of the vector $\mathbf{m}$ and Δm_i is a small increment in m_i. The above finite-difference equation requires one additional calculation of the cost function at $m_i + \Delta m_i$. For better accuracy, the partial derivative can be numerically calculated by the following finite-difference equation.

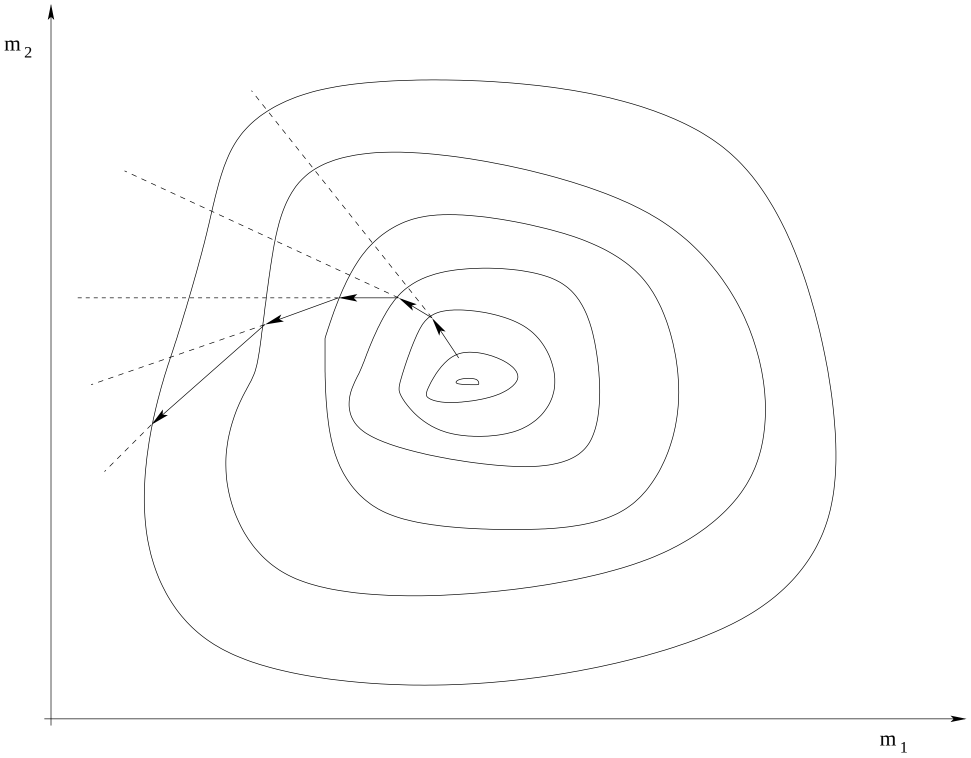

Figure 2.3: Steepest ascent along the gradient direction. The contours show the equal cost function values.

$$\frac{\partial f}{\partial m_i} \simeq \frac{f(m_i + \Delta m_i) - f(m_i - \Delta m_i)}{2\Delta m_i}. \tag{2.4}$$

Equation 2.4 however requires the calculation of the cost function two times thus, making the optimization computationally less economical.

In certain situations, partial derivatives of the cost function do not exist even in numerical formulations. Problems involving cost functions whose partial derivatives are not defined over the search regime require random perturbation based optimization techniques.

In the following section, I will review various widely used techniques to achieve convergence to a local minimum of a function. I will discuss important gradient based optimization algorithms when the cost function is quadratic (linear/linearized inverse problem) and nonquadratic (nonlinear inverse problem).

2.2.2 Gradient based optimization for quadratic cost function

A linear/linearized inverse problem can be posed as an optimization problem where the cost function is quadratic. This means that the surface of the cost function is a paraboloid with a single minimum. The following discussion provides, in detail, the important algorithms employed to optimize a quadratic cost function.

2.2.2.1 Steepest descent algorithm

A quadratic function $f(\mathbf{m}) = \frac{1}{2}\mathbf{m}^T\mathbf{G}\mathbf{m} - \mathbf{d}^T\mathbf{m}$ will have a single minimum with respect to the model parameters if the second-order partial derivative matrix $\mathbf{G}$ is positive-definite. For a two dimensional model space, the quadratic cost function is shown in Figure 2.4.

In Figure 2.5, the contours show the constant cost function values. The arrows show the direction in which the steepest descent algorithm proceeds.

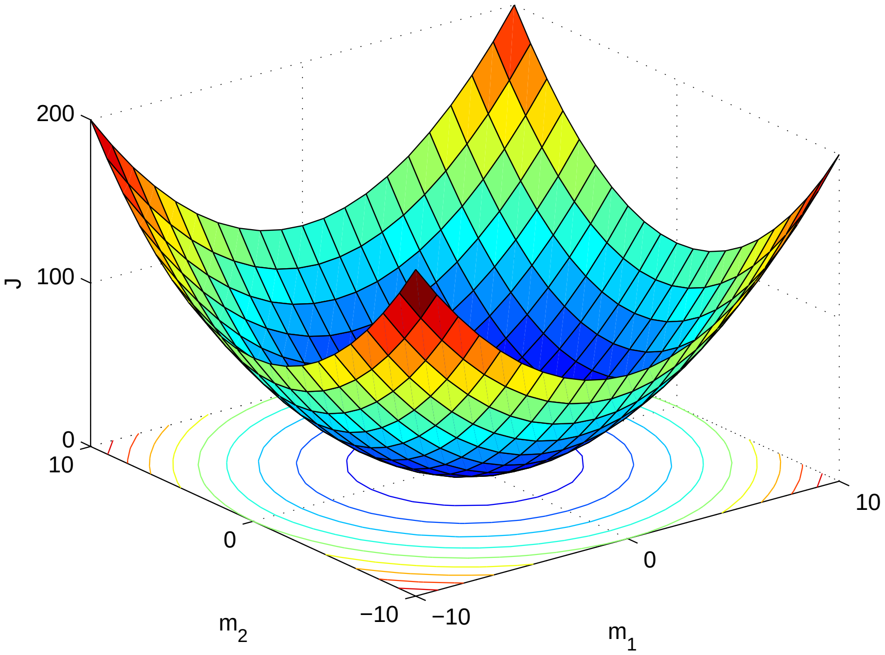

Figure 2.4: Shape of a quadratic cost function when the $\mathbf{G}$ matrix is positive-definite.

The main idea behind the steepest descent algorithm is that given the topology of a function, the calculated gradient at any point gives the direction of maximum

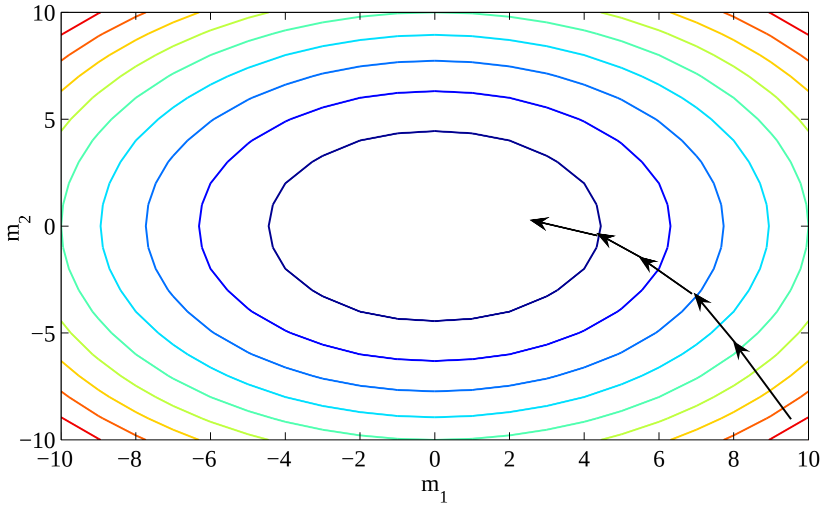

Figure 2.5: Convergence in steepest descent algorithm. Contours show the equal cost function regions.

increase of the function. Thus, if the aim is to find the minimum of the cost function then the algorithm proceeds along a negative gradient direction calculated at each iteration. Though the negative gradient provides the direction of maximum decrease in the cost function, it does not provide the step size. One way to obtain the step size is to keep it constant. However, if the step size is large, there is a possibility that the algorithm will overshoot the minimum point and become oscillatory. In order to avoid such a situation, the step size is calculated by computing the optimum step size at every iteration. This is done by computing the first order derivative with respect to the step size and equating to zero. Figure 2.6 shows the flowchart for steepest descent algorithm. The pseudo-code for the steepest descent algorithm is provided in Algorithm 2.1.

2.2.2.2 Conjugate gradient algorithm

Conjugate gradient is one of the most useful algorithms to solve high dimensional linear and nonlinear equations. The linear conjugate gradient algorithm was proposed by Hestenes and Stiefel (1952) as a means to solve large linear systems of equations involving positive-definite coefficient matrix. The method became popular because it provided faster convergence to a solution compared to the steepest descent method and also it did not require large computer memory to store large

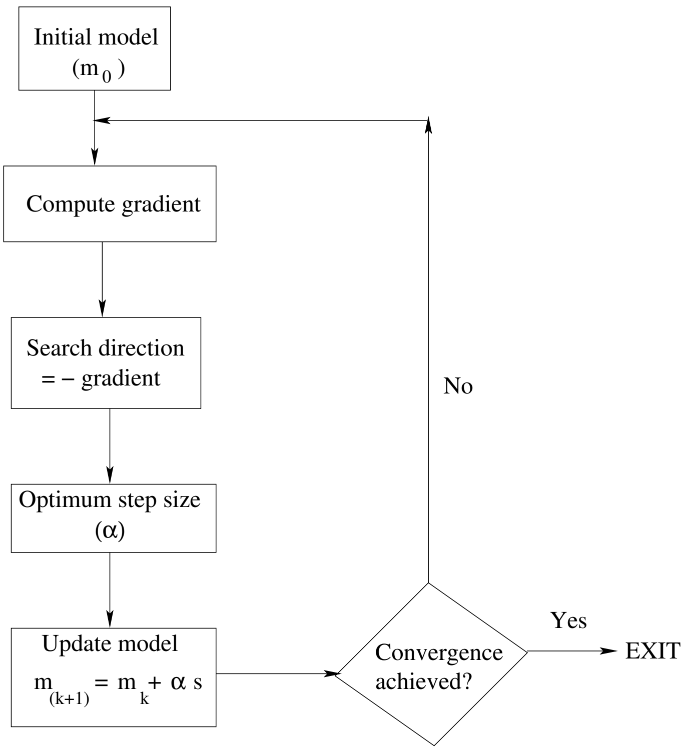

Figure 2.6: Flowchart for steepest descent algorithm.

matrices. In the following discussion, I derive the expressions for the conjugate gradient algorithm with Fletcher-Reeves method (Rao, 1996).

Let us minimize the quadratic function $f(\mathbf{m}) = \frac{1}{2}\mathbf{m}^T\mathbf{G}\mathbf{m} - \mathbf{d}^T\mathbf{m}$. Let $\mathbf{m}_0$ be the initial model. The negative gradient of the function $f(\mathbf{m})$ provides the maximum descent at $\mathbf{m} = \mathbf{m}_0$. Following the direction of steepest descent for the first iteration, we obtain the expression for the search direction

$$\mathbf{S}_0 = -\nabla f_0 = -\mathbf{G}\mathbf{m}_0 + \mathbf{d}. \tag{2.5}$$

Let α_0 be the step length to update the model along the steepest descent direction $\mathbf{S}_0$. To calculate the optimum step length α_0, we take the derivative of $f(\mathbf{m}_0+\alpha_0\mathbf{m}_0)$ with respect to α_0 and equate to zero. Thus, we have

$$\mathbf{S}_0^T \nabla f|_{\mathbf{m}_1=\mathbf{m}_0+\alpha_0\mathbf{S}_0} = 0. \tag{2.6}$$

After a little algebraic manipulation of terms, the optimum step length for updating the model $\mathbf{m}_0$ to $\mathbf{m}_1$ is obtained as

Algorithm 2.1 Pseudo-code for steepest descent algorithm

```
Given $\mathbf{m_0}$;
Set $k \leftarrow 0$;
while (Convergence criteria not satisfied) do
    $\mathbf{S}^k \leftarrow -\nabla f(\mathbf{m}^k)$;
    if $\mathbf{S}^k = 0$ then
        Stop
    end if
    $\alpha^k \leftarrow min_\alpha f(\mathbf{m}^k + \alpha \mathbf{S}^k)$;
    $\mathbf{m}^{k+1} \leftarrow \mathbf{m}^k + \alpha^k \mathbf{S}^k$;
    $k \leftarrow k + 1$
end while
```

$$\alpha_0 = -\frac{\mathbf{S}_0^T \nabla f_0}{\mathbf{S}_0^T \mathbf{G} \mathbf{S}_0}. \tag{2.7}$$

It should be noted here that the above equation for the optimum step length can be generalized for the k^{th} iteration for which the step length α_k is given by

$$\alpha_k = -\frac{\mathbf{S}_k^T \nabla f_k}{\mathbf{S}_k^T \mathbf{G} \mathbf{S}_k}. \tag{2.8}$$

Since the residue vector at the k^{th} iteration is $\mathbf{r}_k = \mathbf{G}\mathbf{m}_k - \mathbf{d} = \nabla f_k$, equation 2.8 can be written as

$$\alpha_k = -\frac{\mathbf{r}_k^T \mathbf{S}_k}{\mathbf{S}_k^T \mathbf{G} \mathbf{S}_k}. \tag{2.9}$$

The current search direction $\mathbf{S}_1$ is given as a linear combination of the previous search direction and the current gradient vector. Thus, the second search direction $\mathbf{S}_1$ is given by

$$\mathbf{S}_1 = -\nabla f(\mathbf{m}_1) + \beta_1 \mathbf{S}_0, \tag{2.10}$$

where β_1 is a scalar chosen in such a way that the current search direction $\mathbf{S}_1$ and the previous search direction $\mathbf{S}_0$ are conjugates, implying $\mathbf{S}_0^T \mathbf{G} \mathbf{S}_1 = 0$. Multiplying equation 2.10 by $\mathbf{S}_0^T \mathbf{G}$ on both sides, we obtain by the conjugacy property

$$\mathbf{S}_0^T \mathbf{G} \mathbf{S}_1 = \mathbf{S}_0^T \mathbf{G}(-\nabla f(\mathbf{m}_1) + \beta_1 \mathbf{S}_0) = 0. \tag{2.11}$$

Since, $\mathbf{m}_1 = \mathbf{m}_0 + \alpha_0 \mathbf{S}_0$, we have $\mathbf{S}_0 = \frac{\mathbf{m}_1 - \mathbf{m}_0}{\alpha_0}$. Thus equation 2.11 can be written as

$$-\frac{(\mathbf{m}_1 - \mathbf{m}_0)^T}{\alpha_0}\mathbf{G}\mathbf{S}_1 = \mathbf{S}_0^T \mathbf{G}(-\nabla f(\mathbf{m}_1) + \beta_1 \mathbf{S}_0) = 0. \tag{2.12}$$

Since, $\nabla f(\mathbf{m}_1) - \nabla f(\mathbf{m}_0) = \mathbf{G}(\mathbf{m}_1 - \mathbf{m}_0)$, equation 2.12 can be written as

$$(\nabla f(\mathbf{m}_1) - \nabla f(\mathbf{m}_0))^T (\nabla f(\mathbf{m}_1) - \beta_1 \mathbf{S}_0) = 0. \tag{2.13}$$

Solving for β_1, we obtain

$$\beta_1 = \frac{\nabla f(\mathbf{m}_1)^T \nabla f(\mathbf{m}_1) - \nabla f(\mathbf{m}_0)^T \nabla f(\mathbf{m}_1)}{\nabla f(\mathbf{m}_1)^T \mathbf{S}_0 + \nabla f(\mathbf{m}_0)^T \mathbf{S}_0}. \tag{2.14}$$

Using equation 2.6, we have

$$\beta_1 = -\frac{\nabla f(\mathbf{m}_1)^T \nabla f(\mathbf{m}_1)}{\nabla f(\mathbf{m}_0)^T \mathbf{S}_0}. \tag{2.15}$$

Since $\mathbf{S}_0 = -\nabla f(\mathbf{m}_0)$, equation 2.15 can be written as

$$\beta_1 = \frac{\nabla f(\mathbf{m}_1)^T \nabla f(\mathbf{m}_1)}{\nabla f(\mathbf{m}_0)^T \nabla f(\mathbf{m}_0)}. \tag{2.16}$$

We express the third search direction as the linear combination of the gradient at $\mathbf{m} = \mathbf{m}_2$ and all the past search directions, namely $\mathbf{S}_0$ and $\mathbf{S}_1$. Thus

$$\mathbf{S}_2 = -\nabla f(\mathbf{m}_2) + \beta_2 \mathbf{S}_1 + \gamma_2 \mathbf{S}_0, \tag{2.17}$$

where $\mathbf{S}_2$ is the current search direction at the updated model $\mathbf{m}_2$, β_2 and γ_2 are two scalars that ensure conjugacy among the current and past search directions. The condition of conjugacy between $\mathbf{S}_0$ and $\mathbf{S}_2$ requires that γ_2 be zero. From the condition of conjugacy between $\mathbf{S}_1$ and $\mathbf{S}_2$ we obtain

$$\beta_2 = \frac{\nabla f(\mathbf{m}_2)^T \nabla f(\mathbf{m}_2)}{\nabla f(\mathbf{m}_1)^T \nabla f(\mathbf{m}_1)}. \tag{2.18}$$

Hence, the current search direction is given by

$$\mathbf{S}_2 = -\nabla f(\mathbf{m}_2) + \frac{\nabla f(\mathbf{m}_2)^T \nabla f(\mathbf{m}_2)}{\nabla f(\mathbf{m}_1)^T \nabla f(\mathbf{m}_1)}\mathbf{S}_1. \tag{2.19}$$

Proceeding likewise the expression for the search direction can be generalized and given by the following equation for the k^{th} iteration.

$$\mathbf{S}_k = -\nabla f(\mathbf{m}_k) + \beta_k \mathbf{S}_{k-1}, \tag{2.20}$$

where $\beta_k = \frac{\nabla f(\mathbf{m}_k)^T \nabla f(\mathbf{m}_k)}{\nabla f(\mathbf{m}_{k-1})^T \nabla f(\mathbf{m}_{k-1})}$. The above equation defines the search directions employed in the Fletcher-Reeves method. As mentioned earlier, the residual vector $\mathbf{r}_k$ at the k^{th} iteration is given by $\mathbf{r}_k = \nabla f(\mathbf{m}_k)$. Hence β_k can be written as

$$\beta_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{r}_{k-1}^T \mathbf{r}_{k-1}}. \tag{2.21}$$

Pseudo-code for the conjugate gradient algorithm is provided in Algorithm 2.2.

Algorithm 2.2 Pseudo-code for conjugate gradient algorithm

```
Given $\mathbf{m_0}$;
Set $k \leftarrow 0$, $\mathbf{r}_0 \leftarrow \mathbf{Gm}_0 - \mathbf{d}$, $\mathbf{S}_0 \leftarrow -\mathbf{r}_0$;
while (Convergence criteria not satisfied) do
    $\alpha_k \leftarrow -\frac{\mathbf{r}_k^T \mathbf{S}_k}{\mathbf{S}_k^T \mathbf{G} \mathbf{S}_k}$;
    $\mathbf{m}_{k+1} \leftarrow \mathbf{m_k} + \alpha_k \mathbf{S}_k$;
    $\mathbf{r_{k+1}} \leftarrow \mathbf{Gm_{k+1}} - \mathbf{d}$;
    $\beta_{k+1} \leftarrow \frac{\mathbf{r_{k+1}}^T \mathbf{r_{k+1}}}{\mathbf{r_k}^T \mathbf{r_k}}$;
    $\mathbf{S_{k+1}} = -\mathbf{r_{k+1}} + \beta_{k+1} \mathbf{S_k}$;
    $k \leftarrow k + 1$;
end while
```

Comparison with steepest descent algorithm: A toy example

Conjugate gradient algorithm provides a faster convergence to a local minimum of the function. If n is the dimensionality of the model space then it is expected that the conjugate gradient algorithm will converge to a solution in n steps. In the following toy example I compare the conjugate gradient algorithm with the steepest descent algorithm and I show that the conjugate gradient algorithm indeed converges to a minimum in two steps in a two dimensional model space and it takes less iterations than that of the steepest descent algorithm.

A toy example

I am solving a simple linear system of equation $\mathbf{Gm} = \mathbf{d}$ in two dimensional model space. In this example, I chose

$$\mathbf{G} = \begin{bmatrix} 15.0 & 3.0 \\ 2.0 & 5.0 \end{bmatrix}; \mathbf{d} = \begin{bmatrix} 2.0 \\ 1.5 \end{bmatrix}. \tag{2.22}$$

Equivalently, we can write the above problem as an optimization problem where

$$f(\mathbf{m}) = \frac{1}{2}\mathbf{m}^T\mathbf{Gm} - \mathbf{d}^T\mathbf{m}. \tag{2.23}$$

Thus in a two dimensional model space (m_1, m_2)

$$f(m_1, m_2) = 15m_1^2 + 5m_1m_2 + 5m_2^2 - 2m_1 - 1.5m_2, \tag{2.24}$$

where $\mathbf{m} = [m_1, m_2,]^T$ is the vector of unknown model parameters.

Figure 2.7 shows the convergence steps using both the steepest descent method and the conjugate gradient method. As expected, the conjugate gradient method converged to the solution in two steps whereas the steepest descent method took more than four steps to converge.

2.2.3 Gradient based optimization for nonquadratic cost function

A nonlinear inverse problem, when posed as an optimization problem, leads to nonquadratic cost function. The surface of such a cost function is likely to exhibit multimodality. There exist elegant gradient based optimization techniques to optimize for nonquadratic cost function. However, such techniques are likely to fail when the chosen initial model is not close enough to the global minimum of the cost function. Below I discuss nonlinear conjugate gradient optimization scheme employed in optimization of nonquadratic cost functions.

2.2.3.1 Nonlinear conjugate gradient algorithm

The conjugate gradient algorithm discussed above is a minimization algorithm for a quadratic function. Certain changes are made to make the conjugate gradient

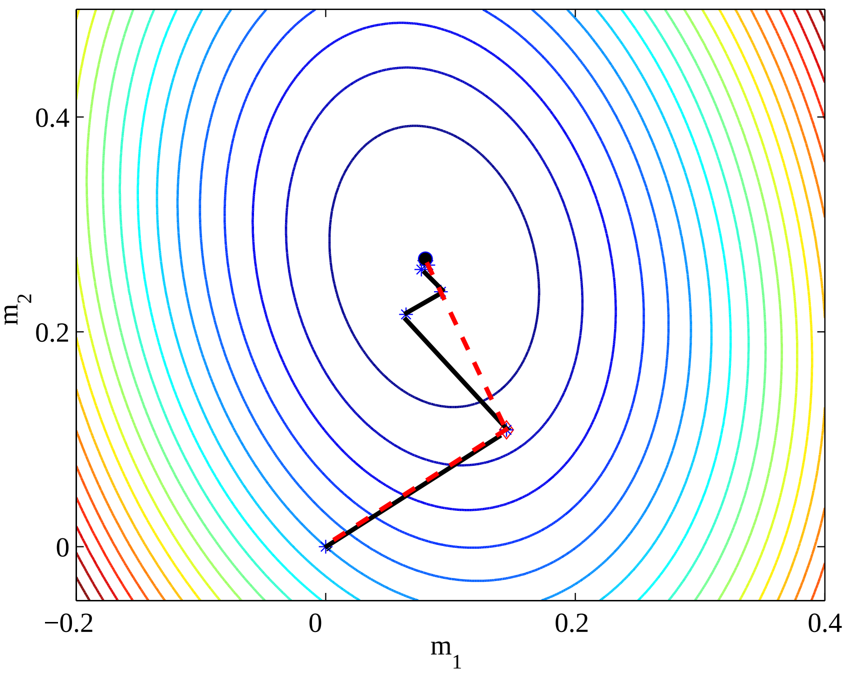

Figure 2.7: Comparison of convergence between steepest descent algorithm and conjugate gradient algorithm. The solid black line shows the steepest descent and the dashed red line shows the conjugate gradient iterations.

algorithm applicable to minimization of nonquadratic functions with nonlinearity expressed in the system of equations that we wish to solve.

As discussed earlier, if the function to be minimized is quadratic e.g. $f(\mathbf{m}) = \frac{1}{2}\mathbf{m}^T\mathbf{G}\mathbf{m} - \mathbf{d}^T\mathbf{m}$, the step length α_k along the direction $\mathbf{S}_k$ for which the function $f(\mathbf{m}_k + \alpha_k\mathbf{S}_k)$ is a minimum, can be analytically calculated by minimizing the function $f(\mathbf{m}_k, \alpha_k)$ with respect α_k. However, for nonlinear functions, in general, there does not exist analytical expression to determine the optimum step length α_k. Thus, a line search technique is used to optimally determine the step size α_k. The nonlinear function is minimized along the direction of $\mathbf{S}_k$ and the residual $\mathbf{r}_k(= \nabla f)$ is replaced by the gradient of the nonlinear function.

The Fletcher-Reeves method of updating the parameter β can be employed in nonlinear conjugate gradient algorithm. A small modification to the Fletcher-Reeves method was suggested by Polak and Ribière. The following equation shows the Polak-Ribière method to update the parameter β.

$$\beta_{(k+1)} = \frac{\nabla f_{(k+1)}^T(\nabla f_{(k+1)} - \nabla f_k)}{\|\nabla f_k\|^2}. \tag{2.25}$$

It should be noted here that when the cost function is perfectly quadratic and the line search for α is exact, $\nabla f_i^T \nabla f_j = 0$, for all $i \neq j$. In such a case, Fletcher-Reeves method and Polak and Ribière method are identical. For a nonquadratic cost function, in general, the line search is not exact. Hence, the Fletcher-Reeves method and the Polak and Ribière method behave differently in nonlinear conjugate gradient algorithms. Numerical studies show that Polak and Ribière method is generally more robust than the Fletcher-Reeves method (Nocedal and Wright, 1999). The pseudo-code for the nonlinear conjugate gradient with the Fletcher-Reeves method and Polak and Ribière method is provided in Algorithm 2.3.

In chapter 5, I shall discuss an application of nonlinear conjugate gradient algorithm with Polak-Ribiere method (Equation 2.25) in the estimation of earth elastic parameters from amplitude variation with offset data.

Algorithm 2.3 Pseudo-code for nonlinear conjugate gradient algorithm

```
Given $\mathbf{m_0}$;
Compute $f_0 \leftarrow f(\mathbf{m}_0)$, $\nabla f_0 \leftarrow \nabla f(\mathbf{m}_0)$;
Set $k \leftarrow 0$, $\mathbf{S}_0 \leftarrow -\nabla f_0$;
while (Convergence criteria not satisfied) do
    Calculate $\alpha_k$ by line search;
    $\mathbf{m}_{(k+1)} \leftarrow \mathbf{m}_k + \alpha_k \mathbf{S}_k$;
    Calculate $\nabla f_{(k+1)}$;
    if (Fletcher-Reeves) then
        $\beta_{(k+1)} \leftarrow \frac{\nabla f_{(k+1)}^T \nabla f_{(k+1)}}{\nabla f_k^T \nabla f_k}$;
    end if
    if (Polak-Ribière) then
        $\beta_{(k+1)} \leftarrow \frac{\nabla f_{(k+1)}^T(\nabla f_{(k+1)} - \nabla f_k)}{\|\nabla f_k\|^2}$;
    end if
    $\mathbf{S}_{(k+1)} \leftarrow -\nabla f_{(k+1)} + \beta_{(k+1)} \mathbf{S}_k$;
    $k \leftarrow k + 1$;
end while
```

2.2.3.2 Newton's optimization method

The idea of well known Newton's method of root finding for a univariate function can be used for optimization of a multivariate function. Let $f(\mathbf{m})$ be a multivariate function whose Taylor series expansion at $\mathbf{m} = \mathbf{m}_i$ is given by

$$f(\mathbf{m}) = f(\mathbf{m}_i) + \nabla f(\mathbf{m}_i)(\mathbf{m} - \mathbf{m}_i) + \frac{1}{2}(\mathbf{m} - \mathbf{m}_i)^T \mathbf{H}_i(\mathbf{m} - \mathbf{m}_i), \tag{2.26}$$

where $\mathbf{H}_i = \nabla^2 f(\mathbf{m})|_{\mathbf{m}_i}$ is the Hessian matrix evaluated at $\mathbf{m} = \mathbf{m}_i$. The first derivative is zero at the point where the function is minimum. So equating $\frac{\partial f(\mathbf{m})}{\partial \mathbf{m}}$ to zero, we obtain from equation 2.26

$$\nabla f(\mathbf{m}) = \nabla f(\mathbf{m}_i) + \mathbf{H}_i(\mathbf{m} - \mathbf{m}_i) = \mathbf{0}. \tag{2.27}$$

Thus $\mathbf{m}_{i+1}$ for the updated model is given as

$$\mathbf{m}_{i+1} = \mathbf{m}_i - \mathbf{H}_i^{-1} \nabla f(\mathbf{m}_i). \tag{2.28}$$

Equation 2.28 is analogous to the expression for the Newton's root finding method for a univariate function which is given by

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}, \tag{2.29}$$

where $f(x)$ is a univariate function of x. Convergence is achieved by the Newton's method provided that the Hessian matrix is nonsingular.

It should be noted here that for a quadratic cost function, Newton's method will find the minimum in one step.

2.2.3.3 Marquardt optimization method

Marquardt's method for optimization (Marquardt, 1963) uses the benefits of both Newton's technique and gradient descent technique to obtain a faster convergence. Marquardt's method modifies the diagonal terms of the hessian matrix by adding a variable scaling factor. The modified hessian matrix is given by

$$\tilde{\mathbf{H}}_i = \mathbf{H}_i + \lambda_i \mathbf{I}, \tag{2.30}$$

where $\mathbf{I}$ is an identity matrix and λ_i is a scalar to modify the diagonal elements of the hessian matrix. It is evident from equation 2.30 that for very large λ_i the term $\lambda_i \mathbf{I}$ dominates the hessian matrix $\mathbf{H}_i$. In such a case

$$\tilde{\mathbf{H}}_i^{-1} = (\mathbf{H}_i + \lambda_i \mathbf{I})^{-1} \simeq \frac{1}{\lambda_i}\mathbf{I}. \tag{2.31}$$

It follows from equation 2.28 that when λ_i is too large, the model is updated in a gradient descent approach. It is obvious that when λ_i is reduced to a small number, the model is updated with a Newton's method approach.

Newton's method provides a faster convergence when the model is close to the minimum point whereas gradient descent algorithm provides a faster approach to bring the initial model closer to the minimum point. Thus the Marquardt algorithm provides a faster convergence when λ_i is set to a large parameter during the initial iterations and then gradually reduced to a small number during the later iterations as the updated model approaches the optimum point.

All gradient based optimization schemes converge to a minimum nearest to the initial model, hence they are referred to as the local optimization algorithms. When the inverse problem is nonlinear, the cost function is nonquadratic and likely to exhibit multimodality. In such situations, gradient based optimization schemes will not converge to the global minimum unless the initial model is close enough to the global minimum. Global optimization algorithms such as the Monte Carlo methods, genetic algorithm or simulated annealing provide the means to achieve possible convergence to the global minimum in presence of multimodality. In the following section, I discuss important concepts in various global optimization algorithms.

2.3 Global optimization

In the thesis, I pose inversion as an optimization problem. Solution of a linear system of equations is same as the minimization of a quadratic function. Local minimization procedures discussed above are suitable when the cost function exhibits a single minimum. However, geophysical inverse problems are in general nonlinear and highly complex. In such cases, the cost function is likely to show several minima. Local optimization techniques are expected to fail in such situations as they have the tendency to converge to the nearest local minimum. Unless there is sufficient *a priori* information so as to make a "smart guess" about the initial model, local minimization algorithms are expected to converge to a false solution and yield poor results. Gradient based optimization schemes do not provide the means to "jump out" of a local minimum and hence possibly converge to a global solution. In most situations however, the global optimization schemes are computationally more expensive compared to the local optimization schemes. With the availability of high computing modern day computers, the cost effectiveness of the global optimization schemes have grown many folds and provide a promising way to find more accurate solutions in highly complex optimization/inversion problems. Figure 2.8 shows two types of cost function with the topology having (a) one minimum and (b) several minima.

2.3.1 Monte Carlo optimization

Global optimization schemes during the early stages of development were based on pure random generation to generate new model for simulations. The main idea was to exhaustively sample the model space with a uniform random number generator and evaluate the cost function at each model. If a model, sampled randomly within a model space, reduces the cost function then the model is accepted otherwise rejected. Global optimization based on such exhaustive search technique is referred to as ”Monte Carlo optimization”. Monte Carlo optimization, though applied to complex optimization problems (Press, 1968; Wiggins, 1969), have proven to be computationally expensive. When the model space becomes large, as is the case with the inversion of seismic data, Monte Carlo methods become impracticable.

The primary motivation to develop efficient global optimization technique lies in the fact that the technique should be practicable in obtaining better results compared to the local optimization techniques in situations where the problem is complex and the model dimensionality is large. Two important global optimization algorithms, (a) Simulated Annealing and (b) Genetic Algorithm, have been developed to deal with situations where local optimization algorithms are likely to fail and pure Monte Carlo technique to obtain a global solution is computationally prohibitive.

In the next sections, I will review the efficient global optimization algorithms. Since I am using simulated annealing optimization scheme in my thesis, I will discuss this technique in more details. Selection of simulated annealing as an optimization algorithm is just a matter of choice. The global optimization algorithm that I am proposing in the thesis could have used genetic algorithm as the basis optimization algorithm and such applications would be straightforward.

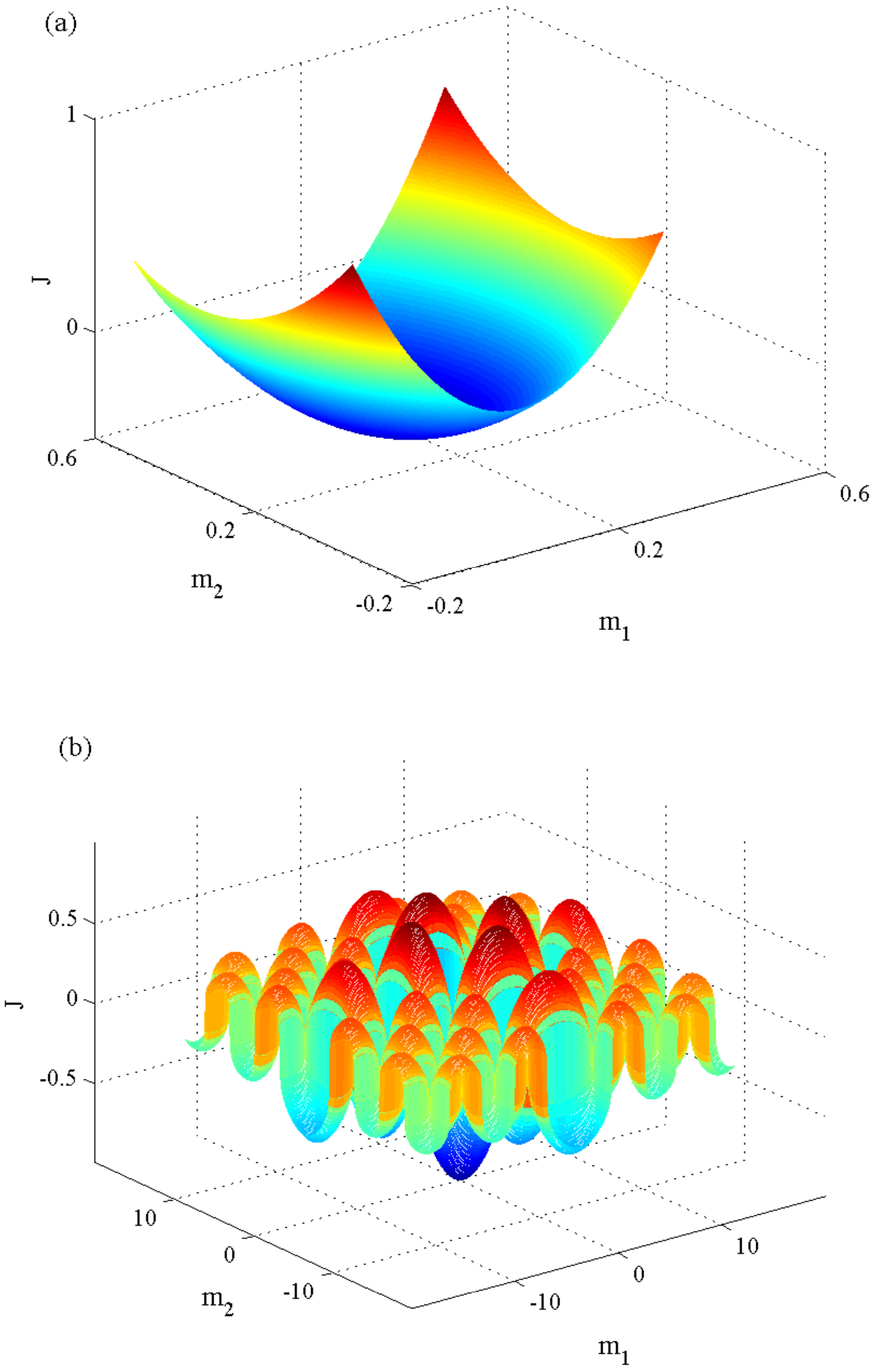

Figure 2.8: Cost function topology. (a) Single minimum and (b) multiple minima.

2.3.2 Simulated Annealing

An inversion problem when posed as an optimization problem, essentially aims at finding a model that best fits the data such that the cost function $f(\mathbf{m})$ attains a global minimum. Local optimization always moves in the downhill direction, thus leaving no scope for the algorithm to move towards a global minimum when stuck in a local minimum. When a cost function has a single minimum, local optimization schemes are efficient and reasonable solutions are obtained. However, in many cases, and more so in geophysical inverse problems, the topology of the cost function is unknown.

Simulated annealing is a procedure analogous to thermodynamic annealing where the chaotic motions of atoms of a molten solid settle down to form a crystal under certain suitable conditions. In thermodynamical processes, the fractional distribution of particles at a particular energy state (E_i) at a temperature T, is given by the Boltzmann or Gibbs distribution function as mentioned below (Sen and Stoffa, 1995)

$$\frac{N_i}{N} = \frac{g_i e^{-\frac{E_i}{k_B T}}}{Z(T)}, \tag{2.32}$$

where N_i is the number of particles in the energy state E_i. The total number of particles $N = \sum_i N_i$. The degeneracy, number of states with energy E_i, is given by g_i. The Boltzmann constant is given by k_B and Z is the partition function given by the equation

$$Z(T) = \sum_i g_i e^{-\frac{E_i}{k_B T}}. \tag{2.33}$$

When the temperature T is reduced gradually, at $T \to 0$, the probability of the state corresponding to the minimum energy becomes increasingly large. If the cooling schedule is too fast, the particles do not attain the minimum energy state, instead they get trapped in a local minimum energy state forming glass.

With a similar analogy, the unknown model parameters constitute the molecules of a molten solid. When the temperature is reduced, the chaotic motion of the molecules gradually ceases and the state corresponding to the global minimum energy (global minimum of the cost function) becomes highly probable at a very low temperature. This is evident in Figure 2.9. Figure 2.9 shows the evaluation of the multimodal cost function given by the equation 2.34 (Sen and Stoffa, 1995) at different temperature T and subsequent calculation of the Gibbs probability distribution function (pdf).

$$J(m_1, m_2) = sgn(\frac{\sin m_1}{m_1})(|\frac{\sin m_1}{m_1}|)^{\frac{1}{4}} sgn(\frac{\sin m_2}{m_2})(|\frac{\sin m_2}{m_2}|)^{\frac{1}{4}}. \tag{2.34}$$

Calculation of Gibbs pdf requires that the cost function be evaluated at all

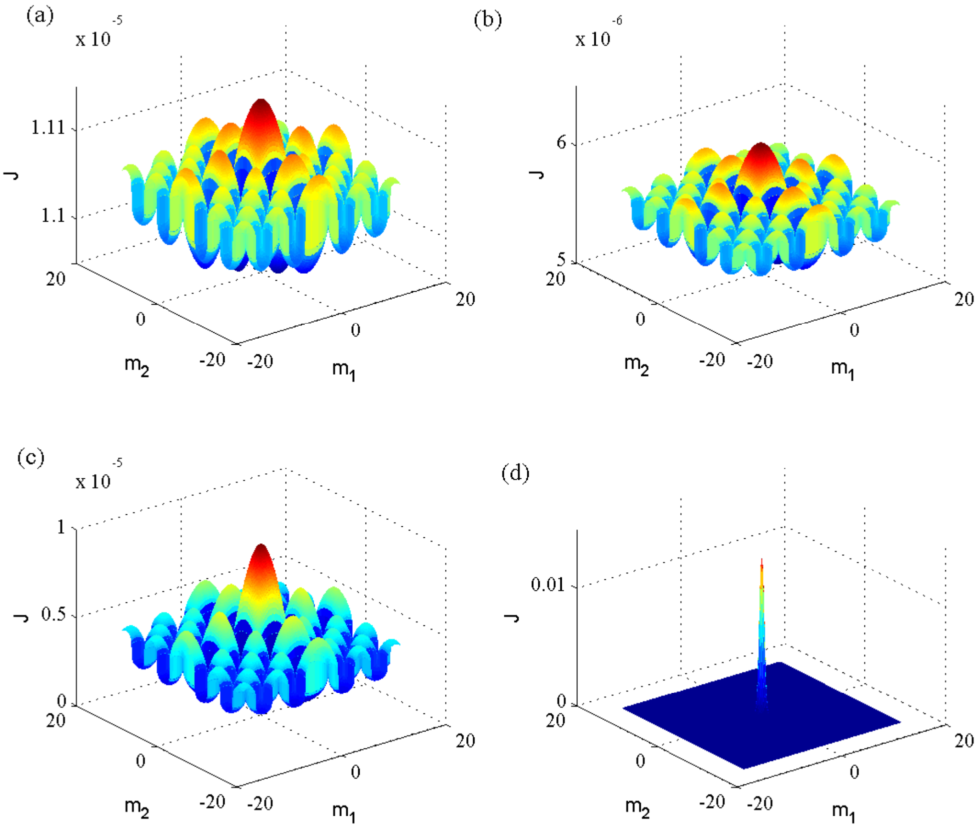

Figure 2.9: Dependence of Gibbs probability density function on temperature T. The Gibbs pdf at (a) $T = 100$. (b) $T = 10$. (c) $T = 1$. (d) $T = 0.01$.

model parameter positions. Such an evaluation makes optimization unnecessary. It is possible to gain insight into the posterior distribution of the model without calculating the Gibbs pdf by the algorithms discussed in the following sections.

2.3.2.1 Metropolis algorithm

Metropolis et al. (1953) proposed the early criterion to obtain a condition for the acceptance/rejection of a model depending upon the evaluated cost function and the cost temperature. Kirkpatrick et al. (1983) brought out a wider applicability of Metropolis algorithm. Metropolis algorithm requires an initial model to be chosen and the cost function be evaluated at the initial model. A random perturbation is applied to the initial model and the cost function is evaluated and compared to the cost function obtained at the previous iteration. If there is a lowering of the cost function then the new model is unconditionally accepted. If the cost function shows an increase then the new model is accepted with the probability of $P = e^{-\frac{\Delta E}{T}}$; where ΔE is the difference of the cost function evaluated at the current and previous iteration. Such an accpetance/rejection condition is known as the "Metropolis criterion".

A toy example

I apply the Metropolis algorithm to find the global minimum of a function that exhibits highly multimodal character. The test function is given by the equation 2.34. The function has a global maximum of 1.0 at (0,0). I redefine the function as follows and minimize the function following Metropolis algorithm.

$$f(m_1, m_2) = (1 - J(m_1, m_2))^2. \tag{2.35}$$

The function $J(m_1, m_2)$ has a minimum at (0,0). I chose an upper and lower bounds of ±10 for the model space. I chose a temperature scale $T =$[100 50 10 5 1 0.5 0.1 0.05 0.01 0.005 0.001 0.0005 0.0001] and number of random moves per temperature was set to 1000. Figure 2.10 shows the estimated solution obtained from the Metropolis algorithm. Figure 2.11 shows the cost function evaluated at each iteration. The plot shows that the value of the cost function decreases and the final estimated model agrees quite well with the global minimum. Figure 2.12 shows a flowchart for the Metropolis algorithm.

In chapter 3, I am presenting a new method based on the Metropolis algorithm to estimate non-minimum phase wavelet from seismic data.

2.3.2.2 Very Fast Simulated Annealing Algorithm

The following functions form the basis of fast and very fast simulated annealing algorithm.

1. Model generating probability distribution function (pdf).

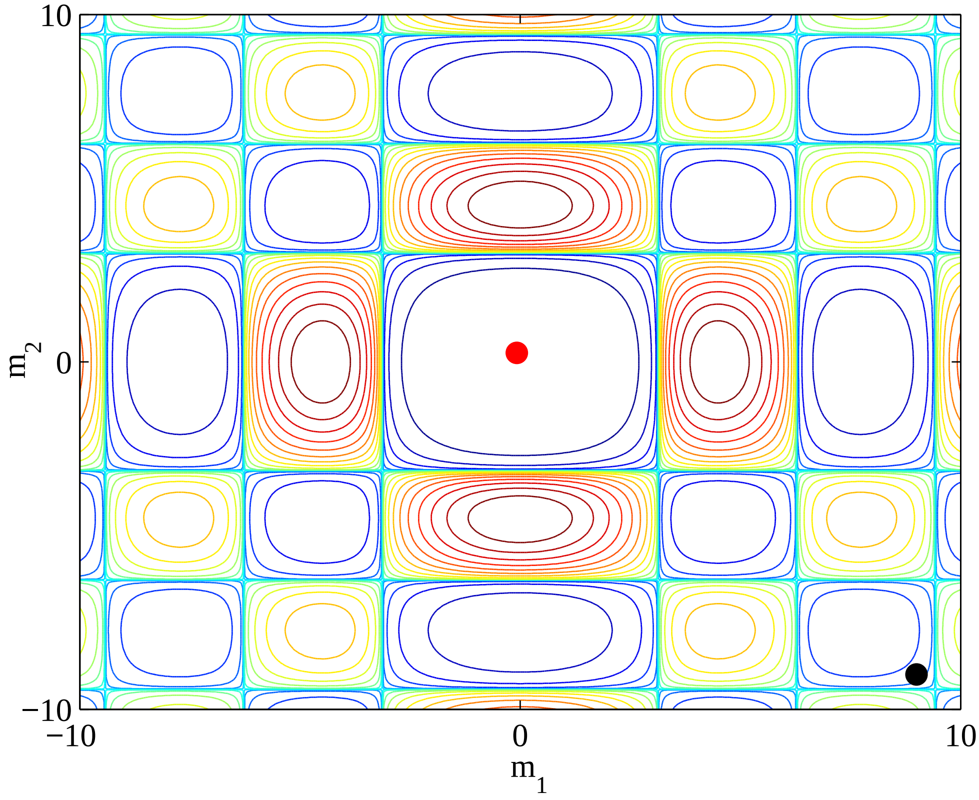

Figure 2.10: The cost function and the location of the estimated model obtained from the Metropolis algorithm. The estimated model agrees well with the global minimum. The black dot shows the initial model and the red dot shows the estimated model. Initial model is at (9.0,-9.0) and the estimated model is at (-0.0760,0.2622). Global minimum is at (0,0).

2. The pdf for the acceptance/rejection criterion (Metropolis condition).
3. Decay of temperature (T) or the cooling/annealing schedule.

Metropolis algorithm, discussed in the previous section, followed a uniform model generating function. Such a uniform pdf provided very slow convergence which could be improved by introducing a multidimensional Gaussian pdf such that as the annealing progressed, the probability of generating a model with lower energy state became higher (Szu and Hartley, 1987). The multidimensional Gaussian pdf as used in the Metropolis algorithm as a model generating distribution function is given by

$$f(\Delta\mathbf{m}) = (2\pi T)^{-\frac{M}{2}} exp(-\frac{\Delta\mathbf{m}^2}{2T}), \tag{2.36}$$

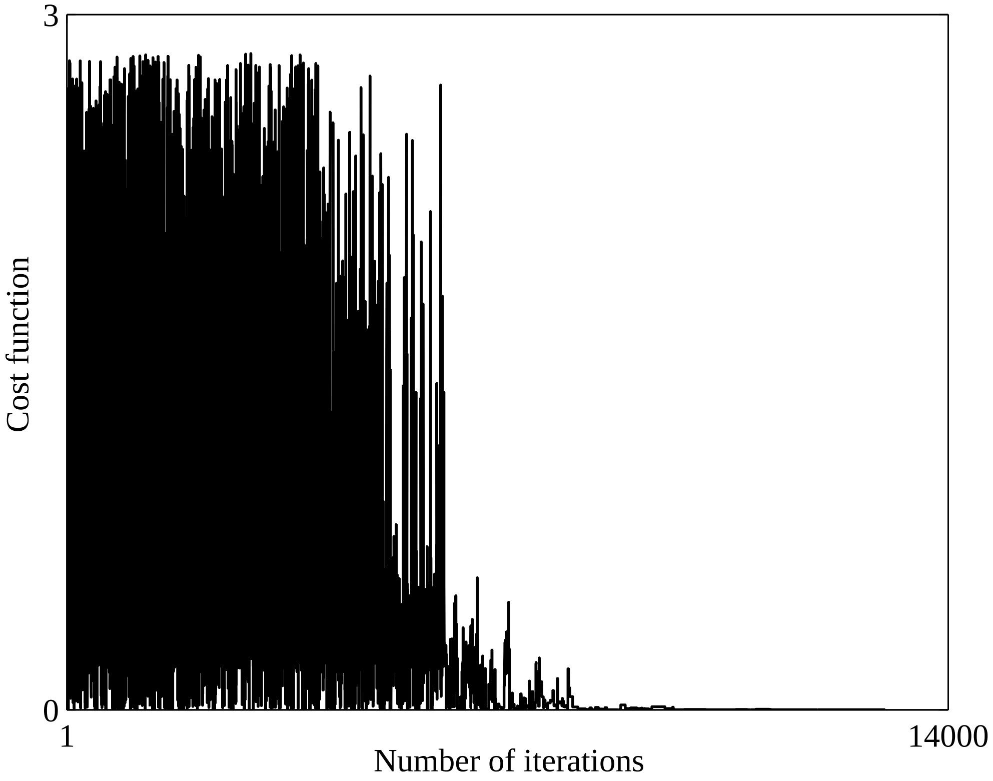

Figure 2.11: The cost function evaluated at each iteration of the Metropolis algorithm. The cost function shows a decrease and attains a value close to the global minimum.

where **m** is the model generated at a particular iteration at a temperature T. The model dimension is given by M and $\Delta\mathbf{m}$ is the perturbation with respect to the previous model. Figure 2.13 shows the variations in the Gaussian model generating pdf as the temperature is annealed.

The Metropolis algorithm provides the following temperature schedule that guarantees global convergence.

$$T_k = \frac{T_0}{\ln(1+k)}, \tag{2.37}$$

where T_k is the temperature at the k^{th} iteration and T_0 is the initial temperature. As noted before, speed of the algorithm is controlled by the rate at which the temperature is decreased. There exists a trade-off in terms of faster convergence and attainment of the global minimum. If the temperature is decreased too rapidly, the

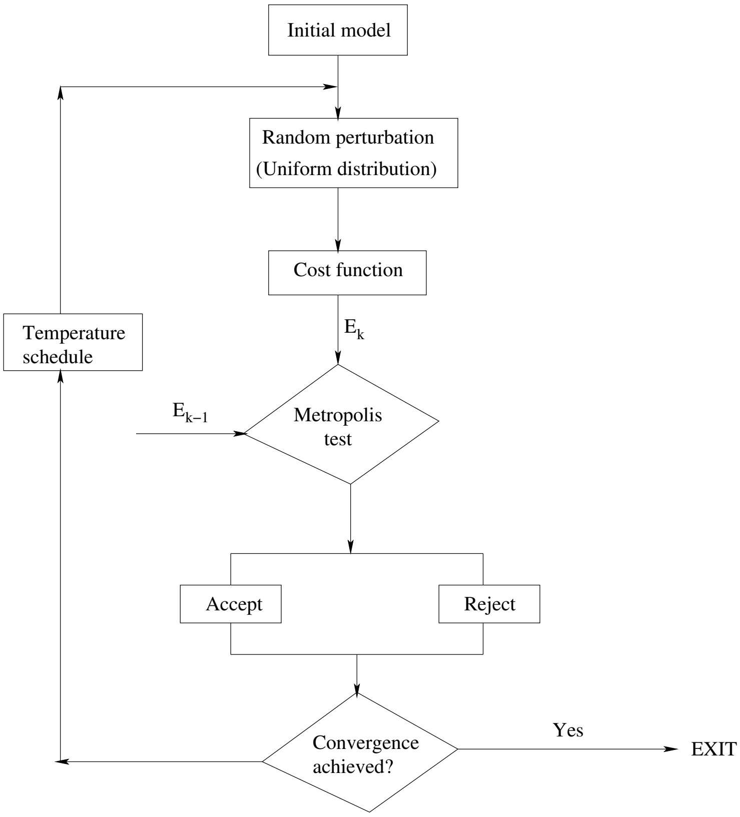

Figure 2.12: Flowchart for Metropolis algorithm

algorithm, instead of converging to the global minimum, is likely to get stuck in a local minimum. If the temperature is decreased too slowly, the algorithm becomes prohibitively expensive. The annealing schedule (the rate at which the temperature is decreased) can be made faster by introducing model generating pdf with the following characteristics

1. The pdf has a sharper peak at low temperature.
2. The pdf has a slightly fatter tail at low temperature.

A faster cooling schedule compared to the Metropolis algorithm can be achieved without affecting the global convergence by introducing a Cauchy pdf for the model generating function. Simulated annealing where the Cauchy pdf is used as the model generating function, is known as the "Fast Simulated Annealing". The Cauchy pdf is given by

$$f(\Delta m) = \frac{1}{\pi} \frac{T}{(\Delta m^2 + T^2)^{(M+1)/2}}. \tag{2.38}$$

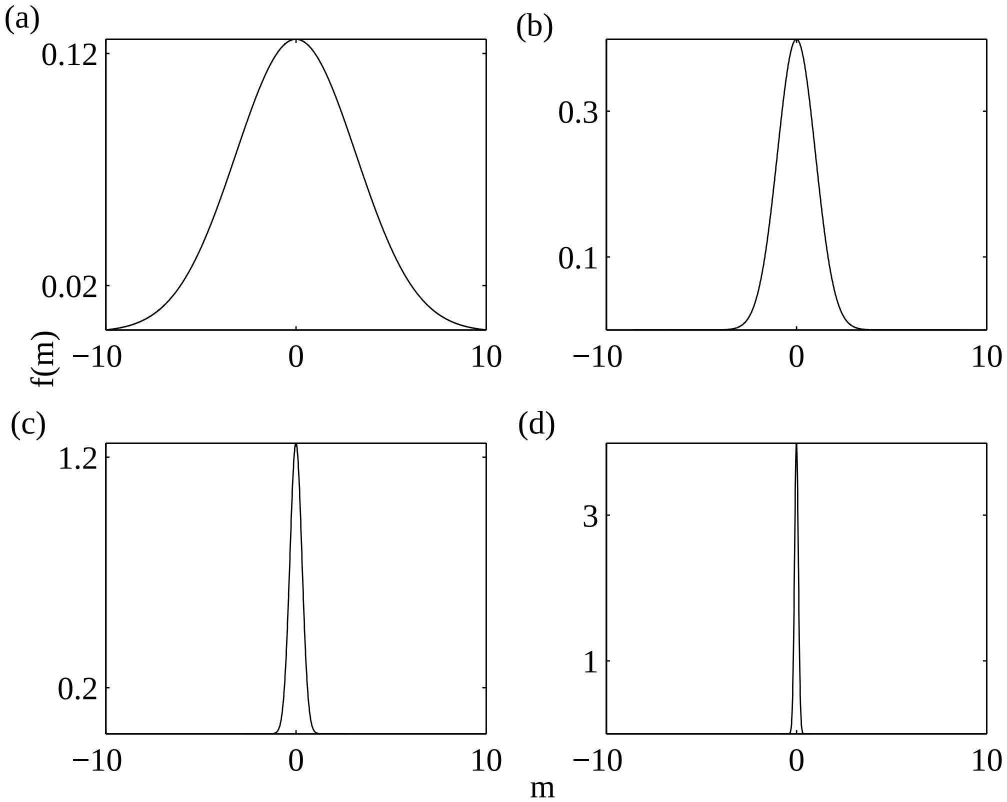

Figure 2.13: Gaussian model generating function at temperature (a) $T = 10$ (b) $T = 1$ (c) $T = 0.1$ (d) $T = 0.01$. The sharpness of the pdf grows as the temperature is decreased.

With a Cauchy pdf, the temperature cooling schedule can be made faster by setting to

$$T_k = \frac{T_0}{k}, \tag{2.39}$$

where T_k is the temperature at the k^{th} iteration and T_0 is the initial temperature. It can be proved that for a model generating function given by the equation 2.38, the temperature cooling schedule as indicated by the equation 2.39, samples the entire model states with a finite probability. The sampled model states have a finite probability of acceptance and rejection as given by the Metropolis criterion. Hence, for the model generating function given by the Cauchy pdf, the convergence to global minimum is statistically guaranteed. Figure 2.14 shows the behavior of Cauchy pdf as the temperature is decreased.

Ingber (1989) introduced the concept of model parameter temperature and a

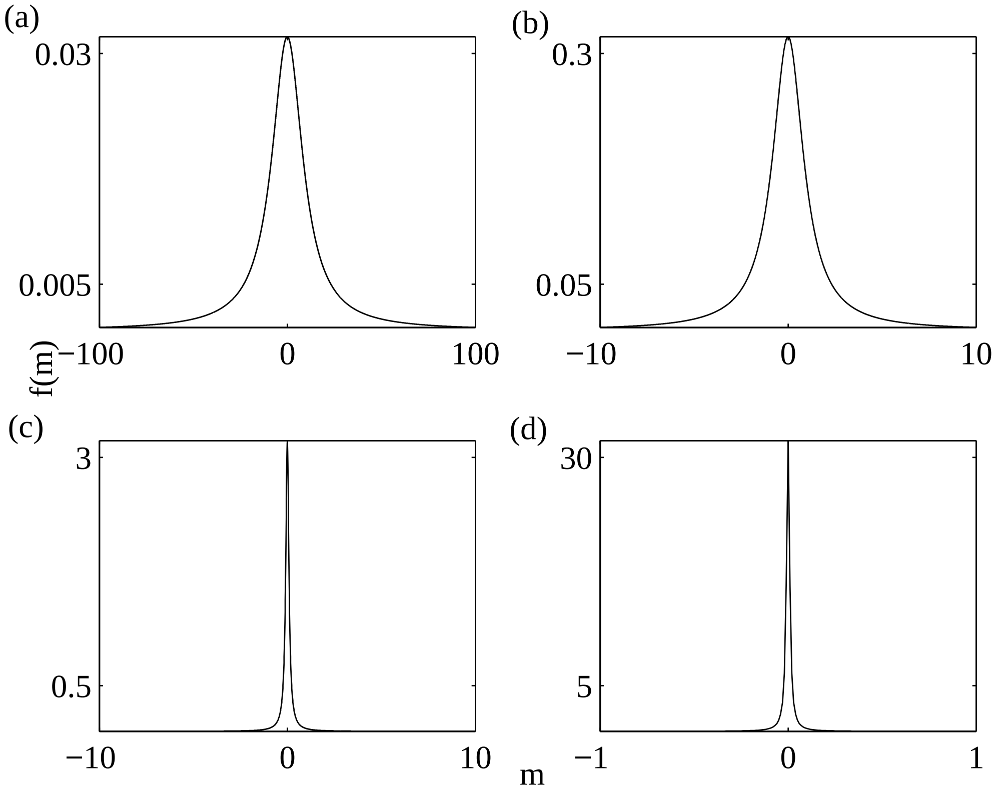

Figure 2.14: The Cauchy model generating function at temperature (a) $T = 10$, (b) $T = 1$, (c) $T = 0.1$, and (d) $T = 0.01$. The sharpness of the pdf grows as the temperature is decreased.

Cauchy-like model generation scheme for each of the model parameters. This resulted in a much faster cooling scale for the individual model parameters which can be accelerated or retarded by including a quenching term to the cooling schedule. The algorithm is known as the "Very Fast Simulated Annealing (VFSA)". The VFSA algorithm provides a means to regulate the expansion or contraction of the model generating pdf depending on the sensitivity of the model parameters to the cost function. Figure 2.15 shows the Cauchy-like model generating pdf and its behavior when the temperature is varied. As seen in the figure, the expansion or contraction of the model generating pdf can be regulated by computing the sensitivity of each model parameter with respect to the cost function. This will allow for a larger or smaller search space for the individual model parameter. The expansion or contraction of the model generating pdf is referred to as "reannealing".

Figure 2.16 shows a comparison of different model generating pdfs. It is ob-

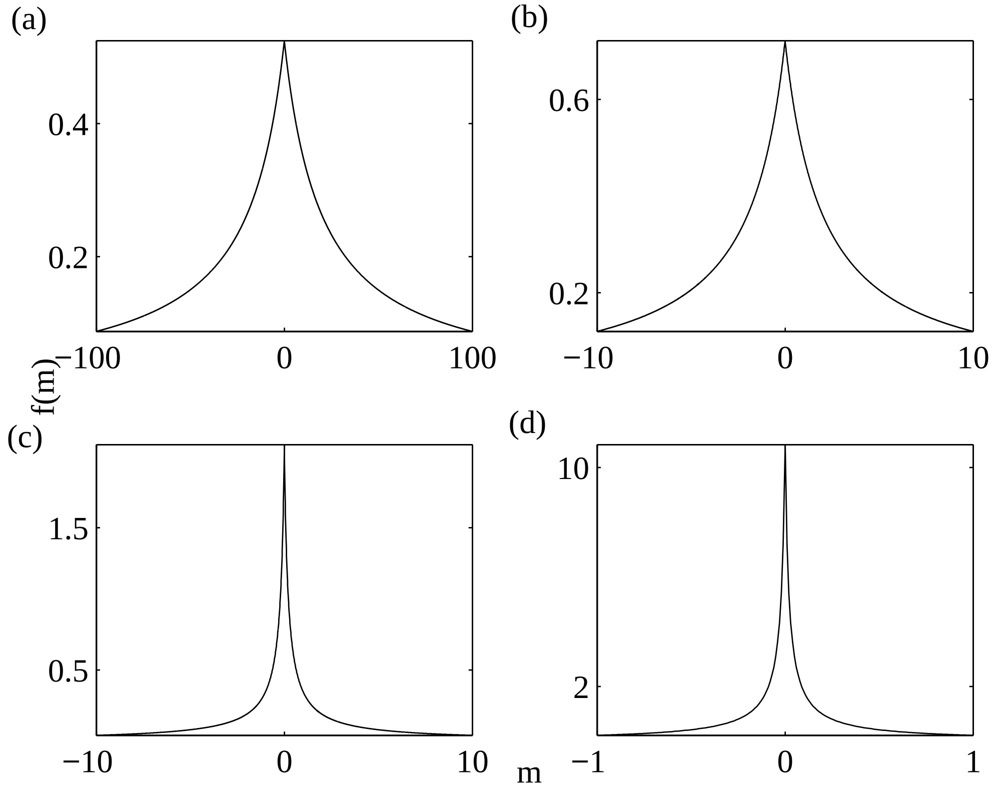

Figure 2.15: The VFSA Cauchy-like model generating function at temperature (a) $T = 10$, (b) $T = 1$, (c) $T = 0.1$, and (d) $T = 0.01$. The sharpness of the pdf grows as the temperature is decreased.

served that the Cauchy and VFSA Cauchy-like distribution functions provide a sharper peak at lower temperature. This means that at low temperature, when the model is close to the global minimum, a smaller perturbation to the model parameter is more likely at the later iterations. This is a desirable effect. We also note that at lower temperature, the VFSA Cauchy-like distribution provides a fatter tail. This suggests that at lower temperature, there also exists a relatively higher probability to generate a model from a slightly larger search space. Such a feature in the model generating function provides the algorithm the means to escape from a local minimum even at a lower temperature.

The VFSA approach provides for two kinds of temperatures, (a) the acceptance/rejection temperature and (b) the model parameter temperature. The acceptance/rejection temperature plays the same role as in the Metropolis algorithm. The model parameter temperature controls the search space of individual model

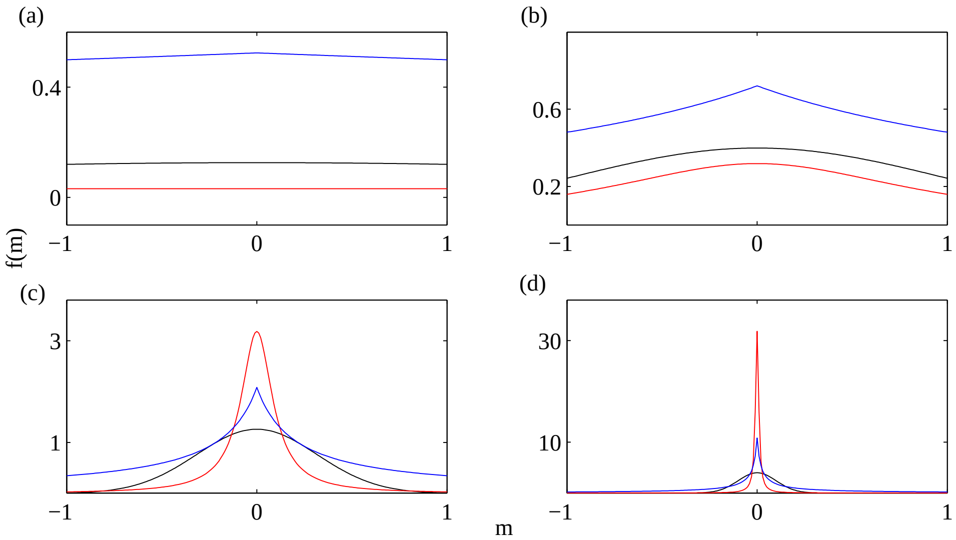

Figure 2.16: Model generating functions- Gauss (black line), Cauchy (red line) and VFSA Cauchy-like (blue line) distribution functions at (a) $T = 10$, (b) $T = 1$, (c) $T = 0.1$, and (d) $T = 0.01$.

parameters by expanding or contracting the model generating pdf.

The algorithm

Let us consider a particular model parameter m_i^k in the model vector $\mathbf{m}$ at an iteration k. Let the upper and lower bounds in the model parameter search space be B_i and A_i such that $A_i \leq m_i^k \leq B_i$. The new model parameter generated in the $(k+1)^{th}$ iteration is given by

$$m_i^{(k+1)} = m_i^k + y_i(B_i - A_i), \tag{2.40}$$

where y_i is a random number between $[-1, 1]$. The model parameter $m_i^{(k+1)}$ is generated such that $A_i \leq m_i^{(k+1)} \leq B_i$. The model generating function for the k^{th} iteration is given by the expression

$$g_T(y) = \prod_{i=1}^{M} \frac{1}{(2|y_i| + T_i)\ln(1 + \frac{1}{T_i})}, \tag{2.41}$$

where M is the model dimension and T_i is the model parameter temperature for the i^{th} model. The cumulative probability is given by

$$G_{Ti} = \frac{1}{2} + \frac{sgn(y_i)}{2} \frac{\ln(1 + \frac{|y_i|}{T_i})}{\ln(1 + \frac{1}{T_i})}. \tag{2.42}$$

Thus, as the algorithm proceeds, new model parameters in the subsequent iterations are generated according to the above mentioned pdf by generating a random number from a uniform distribution $u_i \in U[0, 1]$ and computing the parameter y_i from the following equation

$$y_i = sgn(u_i - \frac{1}{2})T_i[(1 + \frac{1}{T_i})^{|2u_i - 1|} - 1]. \tag{2.43}$$

With the model generating pdf as defined above, the cooling schedule for the model parameter temperature is given by

$$T_i = T_{0i} e^{-c_i k^{\frac{1}{M}}}, \tag{2.44}$$

where T_i is the model parameter temperature for the i^{th} model at the k^{th} iteration and c_i is a constant that is used to attain a particular final model temperature at a given final iteration. The initial model parameter temperature is given by T_{0i}. The value of c_i is computed from the known values of initial and final model temperature by the following equation

$$c_i = k_{fi}^{-\frac{1}{M}} \log \frac{T_{0i}}{T_{fi}}, \tag{2.45}$$

where k_{fi} is the final iteration number for the i^{th} model parameter.

Sensitivity, reannealing and quenching

As mentioned earlier, individual model parameter temperature provides the option to expand or contract the individual search space by changing the shape of $g_T(y)$. Depending on the sensitivity of a model parameter to the cost function, it is desired to have different search regimes for different model parameters. The sensitivity of the model parameter is calculated by computing the derivative of the cost function with respect to the model parameter in consideration. Thus the sensitivity for the model parameter m_i is given by

$$S_i = \frac{\partial f(\mathbf{m})}{\partial m_i}, \tag{2.46}$$

where $f(\mathbf{m})$ is the cost function evaluated at a particular iteration k for a model parameter m_i. The sensitivity is preferably calculated when the algorithm appears to be trapped in a local minimum and the model parameter temperature is scaled by the sensitivity analysis so that a wider search regime is incorporated in the algorithm with the hope that the algorithm will be able to escape the local minimum. The process of scaling the model temperature depending on the sensitivity is referred to as "reannealing" which is shown in the following equation.

$$T_i^r = T_i(\frac{S_m}{S_i}), \tag{2.47}$$

where T_i^r is the reannealed model temperature at the iteration k, T_i is the model temperature at the iteration k before reannealing and S_m is the largest sensitivity among all the model parameters. Since, $\frac{S_m}{S_i} > 1$, reannealing causes the model temperature to increase, resulting in an expansion of the model generating pdf $g_T(y)$ for the model parameter under consideration.

Quenching is a process that results in an accelerated model temperature cooling schedule. This is desired when the temperature has dropped down slowly and the algorithm is close to the global minimum. Under this situation, a faster cooling schedule will provide a faster convergence to the global minimum without being in the risk of getting stuck in a local minimum. An accelerated cooling schedule is obtained by including a quenching term to the cooling equation.

$$T_i = T_{0i} e^{-c_i k^{\frac{Q}{M}}}, \tag{2.48}$$

where Q is the quenching term such that $1 \leq Q \leq M$. Though quenching provides a faster convergence, there is no guarantee that this will provide a global convergence unless $Q = 1$ (no quenching).

A toy example

I applied the VFSA algorithm to find the global minimum of a function that exhibits highly multimodal character. The test function is given by the equation 2.34. The function has a global maximum at (0,0). I redefined the function as shown in equation 2.35. The function $f(m_1, m_2)$ has a minimum at (0,0). VFSA algorithm requires two different temperatures to be defined. For the acceptance/rejection temperature, I define T_a=[100 50 10 5 1 0.5 0.1 0.05 0.01 0.005 0.001 0.0005 0.0001]. The model parameter temperature followed the cooling schedule as defined in equation 2.44. Since, this is a toy problem with only 2 model dimensions, I have set the quenching term $Q = 1$ (no quenching). I performed 1000 iterations at each acceptance/rejection temperature. Figure 2.17 shows the VFSA optimization results. The error plot in the Figure 2.18 shows the convergence of VFSA algorithm as it proceeds through the iterations. Figure 2.19 shows the flowchart for VFSA algorithm.

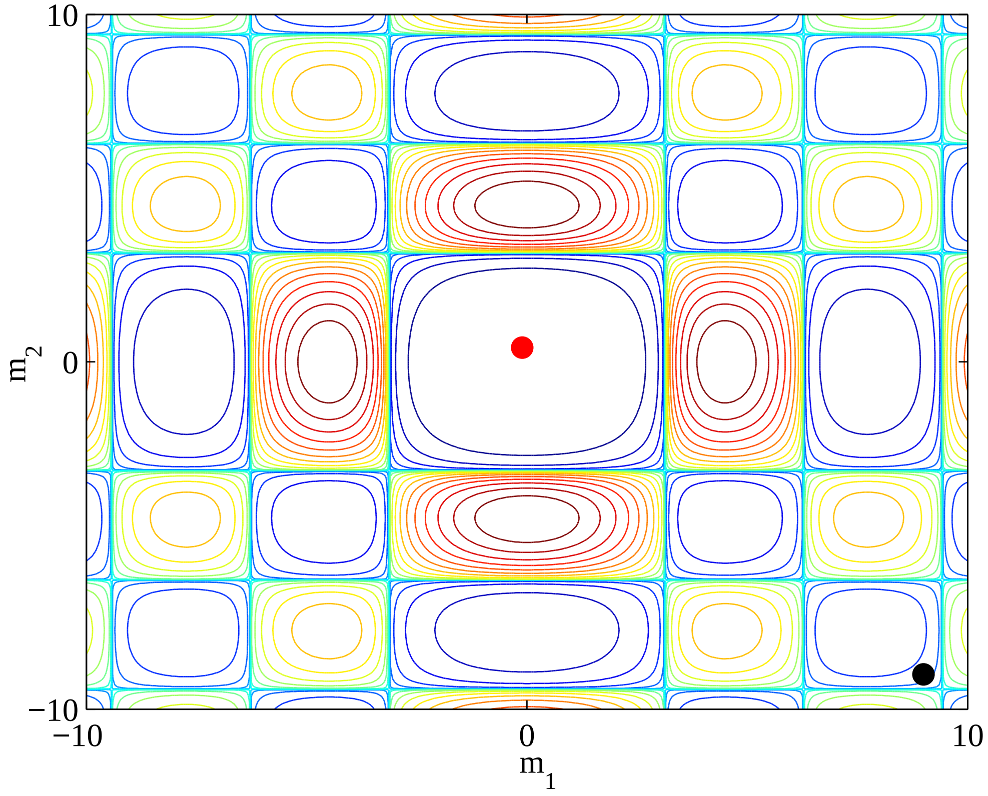

Figure 2.17: The VFSA optimization result for a multimodal cost function. The black dot shows the initial model and the red dot shows the estimated model. Initial model is at (9.0,-9.0) and the estimated model is at (-0.1089,0.4066). Global minimum is at (0,0).

Comparison with Metropolis algorithm

Very Fast Simulated Annealing algorithm provides temperature annealing control over the individual model parameters. In addition, VFSA algorithm provides faster cost temperature annealing schedule compared to the Metropolis algorithm. In the

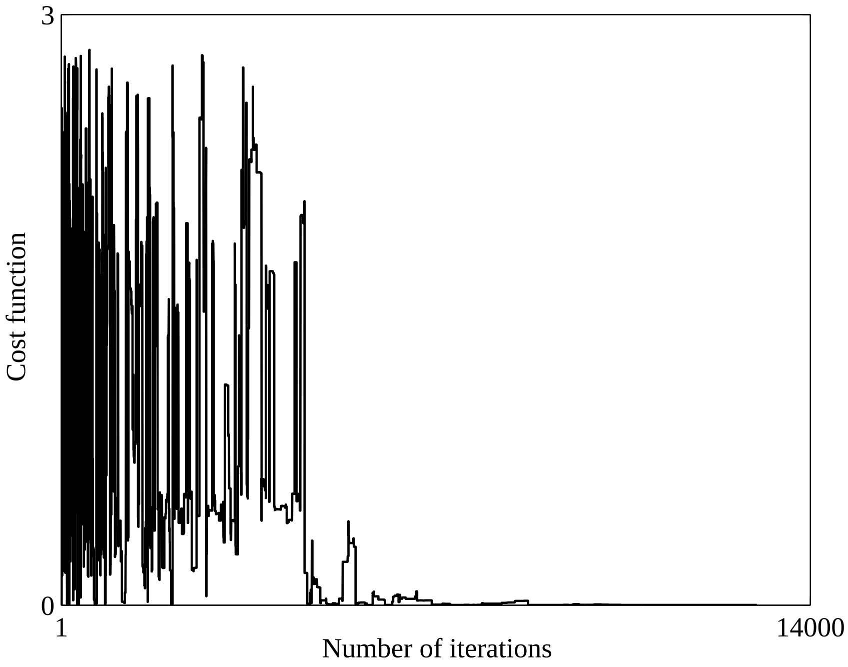

Figure 2.18: The cost function evaluated during VFSA optimization.

example below, I am showing the faster convergence achieved with respect to the Metropolis algorithm by only regulating the model parameter temperature of the VFSA algorithm. I have kept the cost temperature annealing schedule same for both the algorithms. For an unbiased comparison, I am starting both the algorithms at the same initial model (9.0,-9.0). The cost function to be optimized is given by the equation 2.34 and 2.35. The cost temperature schedule T_a=[100 50 10 5 1 0.5 0.1 0.05 0.01 0.005 0.001 0.0005 0.0001] is maintained same for both the algorithms. For the VFSA algorithm, the model temperature annealing schedule is given by the equation 2.44 where the parameters c_i and T_{0i} were set to 1.8 and 20.0 respectively.

Figure 2.20 shows the cost function evaluated at each iterations as both the algorithms converged to the optimum model. It is observed that by regulating the model parameter temperature annealing schedule in the VFSA algorithm, the convergence can be made faster with respect to the Metropolis algorithm.

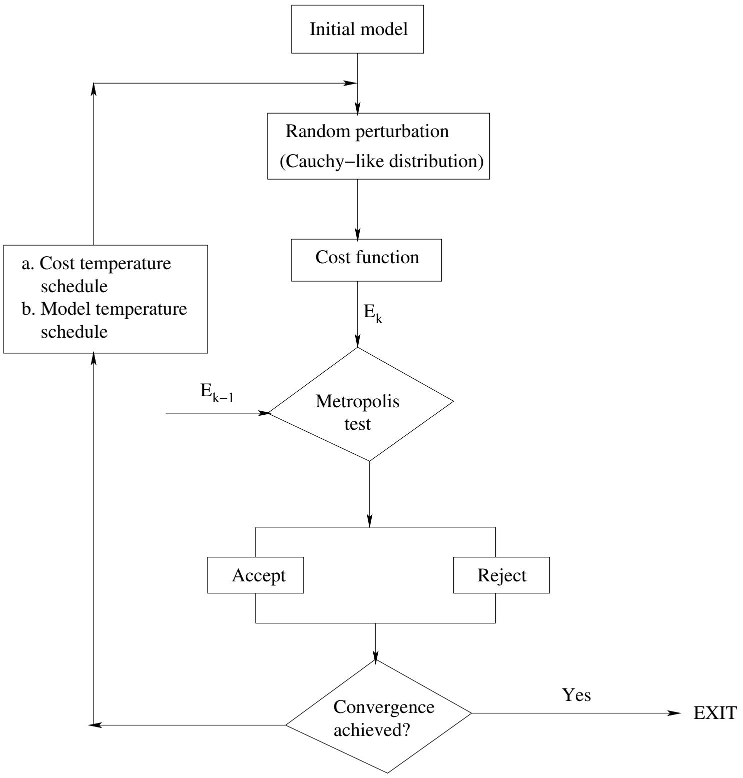

Figure 2.19: Flowchart for VFSA algorithm.

2.3.3 Genetic algorithm

Genetic algorithm (GA), as the name implies, is based on the analogy that the genetic modifications that take place in the living species work towards making the species more intelligent and adaptive to the changing natural surroundings. The process of biological evolution is mimicked in genetic algorithm where a pool of possible solutions is updated every iteration so that the pool contains better candidate models as the iterations proceed. Genetic algorithm was first proposed by Holland (1975) and discussions on GA are found in books published as early as 1989 (Goldberg, 1989). Subsequently, several modifications to the classical GA approaches were proposed (Stoffa and Sen, 1991; Sen and Stoffa, 1992). The modifications to the classical GA involved a hybrid approach between simulated annealing and GA.

In the following discussion, I describe the salient features of genetic algorithm namely, coding, fitness function, selection, crossover and mutation.

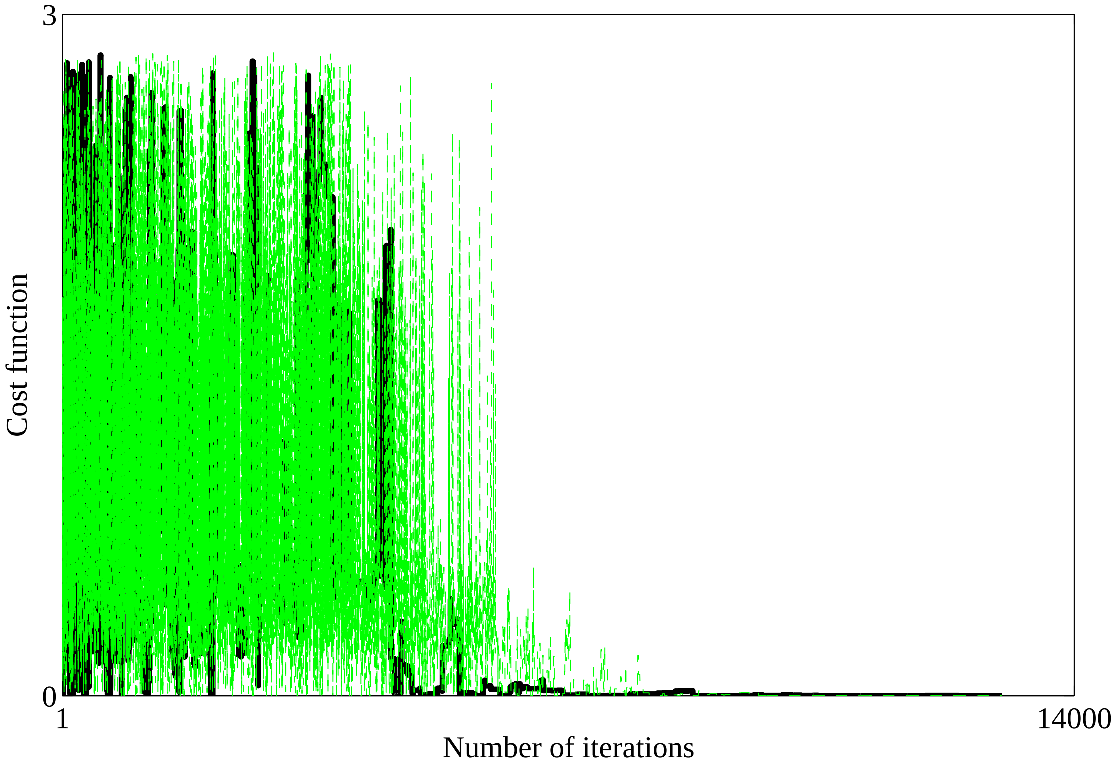

Figure 2.20: Comparison of cost function evaluated for Metropolis algorithm and VFSA algorithm. For the same starting model and same cost temperature annealing schedule the VFSA algorithm shows faster convergence compared to the Metropolis algorithm. Green dashed line and the solid thick black line show the cost function evaluated for the Metropolis algorithm and the VFSA algorithm respectively.

Coding

Coding is a method to digitally represent the model space. A commonly used coding method is the conversion of numerical model parameters to binary strings. In GA literatures such individual binary strings representing individual model parameters are referred to as the *chromosomes*. Every bit in such a binary string is referred to as *gene*. Thus a gene can either be one or zero. Coding method can also involve decimal system in which the model parameters are decimal numbers. Coding defines the resolution and the search space of the optimization algorithm.

Fitness function

In GA literatures, the function that defines how closely a model fits the data and constraints is referred to as the fitness. As opposed to other optimization algorithms, the aim of the genetic algorithm is to increase the fitness function so that the optimum model is the one that has the maximum fitness. Thus the usual practice is to use either a negative or inverse of a cost function to define the fitness function.

Selection

The model parameters are defined in terms of chromosomes. If the chromosomes take the form of binary strings then the model parameters are defined by binary strings. Initially the model parameters are randomly selected within a user defined upper and lower bounds. Each random selection is then converted to chromosomes within a user defined resolution. A pool of such possible solutions for the model is created. Such a pool is referred to as the *population pool*. Once the population pool is created the fitness criterion for each candidate of the population pool is evaluated. The next step is *selection*. Selection is the procedure to update the population pool with fitter models as the algorithm proceeds through *generations* (the updating of the population pool through iterations is termed as *generations*). Usually the selection is done by two procedures, (a) proportional selection and (b) tournament selection. In proportional selection method, probability of selection is evaluated for each individual candidate models comprising the population pool. The probability of selection is computed on the basis of fitness values of each candidate models. The next generation population pool is created by replicating the candidate models of the current pool in direct proportion to their fitness probabilities. In the tournament selection method, the models comprising the population pool are randomly paired and their fitness values are compared. One, out of the two pairing candidate models, is selected depending on a user defined probability value. For example, a probability of greater than 0.4 can be set to accept a candidate model with a higher fitness value among the two pairing models. This allows for a finite probability to populate the next generation population pool with less fit models, thus providing the scope to the algorithm to ”jump out” of the local minimum.

Crossover

Crossover is an analogous operation that mimics transfer and sharing of genetic information among chromosomes. In genetic algorithm, crossover operation involves two chromosomes in a population pool. Crossover can be single-point or multi-point. In single-point crossover, one point is randomly selected and a pair of concatenated binary strings representing chromosomes is also randomly selected. All binary bits to the right of the randomly chosen point are swapped between the two chromosome pairs. In case of multi-point crossover, several crossover points are chosen representing each of the concatenated model parameters. All the bits to the right of the chosen crossover points but within the binary string representing a model parameter are swapped between the pairing chromosomes. Number of crossover operations inside a population pool is controlled by the user defined crossover probability. Crossover operation results in a new population pool.

Mutation

Mutation mimics the process of genetic mutation where a particular gene undergoes a change. In genetic algorithm, mutation operation is performed by randomly picking a bit from a chromosome (binary string) and changing the binary value. For example, if the bit contained a 1, it is changed to a 0 and vice-versa. Number of mutation operations carried out in a population pool is controlled by the mutation probability. A higher mutation probability means greater diversity in the population pool and hence slower convergence to the optimum point. However, greater diversity in the population pool helps the algorithm to escape from a local minimum. After several generations of updating the population pool there is an overall increase in the fitness of the candidate models in the population pool. When the convergence is achieved as per the user defined criteria, one candidate model is selected from the optimum population pool. The candidate model is decoded to obtain the numerical values for the unknown model parameters. Figure 2.21 shows a flowchart diagram for a basic genetic algorithm.

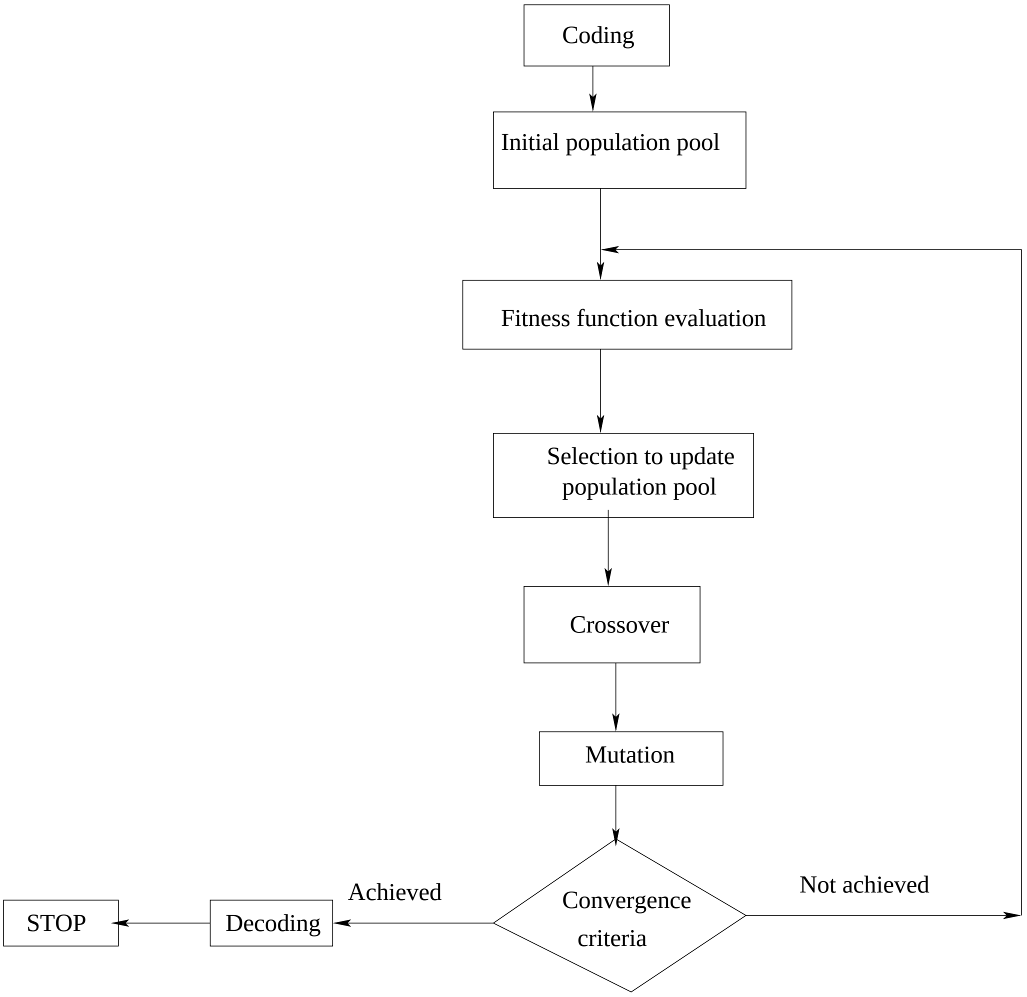

Figure 2.21: Flowchart for Genetic algorithm

2.4 Summary

In this chapter, I reviewed the theories involved in local and global optimization schemes. I discussed steepest-descent, linear and nonlinear conjugate gradient algorithms with the help of toy examples. I discussed global optimization schemes with particular emphasis on simulated annealing. I reviewed the theories on classical Metropolis algorithm and showed an example where the minimum of a multimodal cost function is obtained by the algorithm. I reviewed the theory of very fast simulated annealing (VFSA). I showed an example of VFSA minimization of a multimodal cost function. Both Metropolis algorithm and VFSA algorithm minimized the same cost function starting with the same initial model. On comparison with the Metropolis algorithm, we observe that VFSA optimization resulted in a faster convergence and better estimation of the model. I have also provided flowcharts for local optimization algorithms as well as for global optimization algorithms such as the Metropolis, the VFSA and genetic algorithms.

In the next chapter, I am proposing a new application of simulated annealing algorithm to estimate the mixed phase wavelet embedded in the seismic data. Estimation of the phase characteristics of the wavelet is performed by suitable estimation of the coefficients of an all-pass operator. This is a blind problem because the structure of the cost function is unknown. This necessitates the use of global optimization scheme. I have used the Metropolis algorithm to estimate the coefficients of the all-pass operators required to estimate the phase characteristics of the embedded wavelet in the seismic data.

Chapter 3

Global optimization: Application in blind deconvolution problem[1]

3.1 Introduction

The process of deconvolution requires a proper estimation of the wavelet so as to obtain a more accurate estimate of the underlying reflectivity series. The reflectivity series is commonly assumed to be white, even though it is a well known fact that the reflectivity series is not white in the majority of the cases (Rosa and Ulrych, 1991; Saggaf and Robinson, 2000; Walden and Nunn 1988). The issues with non-white behavior of the reflectivity series are beyond the scope of this chapter. I make the assumption that the reflectivity series is a stationary, non-Gaussian, and statistically independent random process (Walden, 1985). Convolution of the reflectivity with the source wavelet makes the reflectivity series lose high frequency components. Deconvolution, with the assumption that the wavelet is minimum phase, removes from the data the wavelet amplitude signature quite effectively. However, it leaves behind a spurious phase signature in the data. In order that the wavelet phase response is also effectively removed, it is necessary to deconvolve the data with an optimum mixed phase wavelet. Classical approaches such as the Wiener-Levinson predictive deconvolution are intended to estimate the inverse minimum phase wavelet in the data. These methods are based on second order statistical assumptions (Robinson and Treitel, 1980; Robinson, 1967; Peacock and Treitel, 1969). In other words, by making the assumption that the reflectivity is white, one can use the autocorre-

[1] Misra, S. and Sacchi, M. D., 2007, Non-minimum phase wavelet estimation by non-linear optimization of all-pass operators, Geophysical Prospecting, **55** , 223-234.

lation of the seismic trace as an estimator of the autocorrelation of the wavelet. Since the autocorrelation of the wavelet does not contain any phase information, an additional assumption about the wavelet is required. In general, the additional assumption is that the wavelet is minimum phase. With these two assumptions of white reflectivity and minimum phase wavelet, it is possible to recover the seismic wavelet by measuring the autocorrelation of the trace. It is clear that if the wavelet contains non-minimum phase components, the classical procedure outlined above is sure to fail. Fortunately, it is possible to design wavelet estimation strategies based on higher order statistical estimators such as the third order and fourth order cumulants. Unlike the autocorrelation function which is a second order cumulant, the third and fourth order cumulants do preserve the phase of the wavelet when the reflectivity consists of a non-Gaussian white process (Lazear, 1993; Mendel, 1991). Following Lazear (1993) and Velis and Ulrych (1996), I preferred to use the fourth order cumulants rather than the third order cumulants. The latter is preferred due to the fact that the third order cumulant vanishes for symmetric distributions and also for zero-mean wavelets such as seismic wavelets. Since there is no evidence that suggests that reflection coefficients should be modeled via a non-symmetric distribution, I have preferred not to utilize the third order cumulant. The fourth order cumulants, on the other hand, do not suffer from the aforementioned shortcoming. Many approaches have been made to bypass the minimum phase assumption and estimate wavelets that show mixed phase character. Such methods include, homomorphic deconvolution (Oppenheim et al., 1968; Ulrych, 1971; Ulrych et al., 1995), minimum entropy deconvolution (Wiggins, 1978), fourth order cumulants matching (Lazear, 1993), and others.

Tugnait (1987) proposed a fourth order cumulant matching technique to estimate a mixed phase moving average wavelet. Lazear (1993) applied this technique to the real seismic data. Velis and Ulrych (1996) applied the technique with a non-linear optimization approach. They estimated the mixed phase wavelet from the fourth-order cumulant of the trace by means of Very Fast Simulated Annealing (VFSA) optimization method. They showed the dependence of the cumulant matching technique on the ratio of bandwidth to central frequency of the data.

A new global optimization method using the cumulant matching approaches to mixed phase deconvolution is proposed here. Parameterization of the mixed phase wavelet as a convolution of a minimum phase wavelet with an all-pass wavelet (Porsani and Ursin, 1998; Porsani and Ursin, 2000) can significantly simplify the problem.

Deconvolving the seismic trace by the estimated minimum phase wavelet helps in broadening the bandwidth of the deconvolved data. This is a desirable effect. As pointed out earlier by Velis and Ulrych (1996), a proper estimation of the mixed phase wavelet by cumulant matching technique is possible when the ratio of the bandwidth to central frequency is greater than 1 and preferably close to 2. Hence, deconvolution by the estimated minimum phase wavelet works favorably for the cumulant matching technique. Optimization for the all-pass wavelet is performed by means of the technique of Simulated Annealing (Sen and Stoffa, 1995). In blind deconvolution problems like the one I am trying to solve, the topology of the cost function is unknown and could have multiple minima. This prompted me to prefer a global optimization approach so that the problems associated with a possible multimodal cost function could be reduced.

3.2 Theory

With the assumptions that the reflectivity series is non-Gaussian, stationary and a statistically independent random process, the fourth order cumulant of the trace is equal to, within a scale factor, the fourth order moment of the wavelet (Lazear, 1993; Liang, Cai and Li, 2002; Velis and Ulrych, 1996). The next section discusses this further. I have already mentioned that the very assumption that the reflectivity is a white process is questionable as far as the true nature of the reflectivity series is concerned (Saggaf and Robinson, 2000). However, in the present context of the paper, this assumption is considered to be valid and the algorithm is purely based upon the validity of the white reflectivity assumption. The estimation of wavelet phase when the reflectivity series is coloured is not addressed here and is considered beyond the scope of this chapter. When the wavelet is parameterized as a convolution of a minimum phase wavelet and an all-pass wavelet, the fourth-order cumulant of the whitened trace (deconvolved by the minimum phase wavelet) is equal to, within a scale factor, the fourth-order moment of the all-pass wavelet. The minimum phase wavelet estimated from the autocorrelation of the trace has the same amplitude spectrum as that of the corresponding mixed phase wavelet. Thus, deconvolving the trace with the minimum phase wavelet not only removes the wavelet amplitude spectrum from the data but also increases the bandwidth. This is a favorable result as a higher ratio of the data bandwidth to central frequency is desired for more reliable wavelet estimation. The whitened trace now contains only the phase information of the wavelet. Hence, an all-pass wavelet remains to be optimized from the whitened data.

Ignoring seismic noise, seismic data can be expressed as

$$d_t = r_t * w_t, \tag{3.1}$$

where d_t is the seismic reflection signal, r_t is the reflectivity series and w_t is the mixed phase wavelet. The '*' indicates a convolution process between the reflectivity sequence and the wavelet. As mentioned earlier, w_t can be parameterized as the convolution of a minimum phase wavelet and an all-pass wavelet (Porsani and Ursin, 1998). Thus,

$$w_t = \tilde{w_t} * f_t, \tag{3.2}$$

where $\tilde{w_t}$ is the minimum phase wavelet estimated from the trace and f_t is the all-pass wavelet. The Z-transform of the all-pass wavelet can be written as (Porsani and Ursin, 1998)

$$F(Z) = Z^p \frac{B(Z^{-1})}{B(Z)}, \tag{3.3}$$

where $B(Z) = b_0 + b_1 Z + b_2 Z^2 + \ldots + b_p Z^p$ and the term Z^p accounts for the time shift required to make the all-pass wavelet causal. It is important to mention here that the time series $b_t = b_0, b_1, \cdots, b_p$ is minimum phase. This is a very simple parameterization with $b_t = b_0, b_1, \cdots, b_p$, and p as unknowns. For this problem, the term Z^p in the equation 3.3 is not important because it only accounts for the time shift in the final estimation of the wavelet.

Substituting for w_t in equation 3.1,

$$d_t = r_t * \tilde{w_t} * f_t. \tag{3.4}$$

Using the Z-transform, the above equation can be represented as

$$D(Z) = R(Z)\tilde{W}(Z)F(Z). \tag{3.5}$$

Deconvolution by the minimum phase wavelet yields

$$\tilde{D}(Z) = R(Z)F(Z), \tag{3.6}$$

where $\tilde{D}(Z)$ is the deconvolved trace that has been whitened by the removal of the minimum phase wavelet. Thus, ideally the reflectivity sequence can be obtained by deconvolving the whitened trace by an optimum all-pass wavelet.
Taking the Z-transform of both sides of equation 3.2,

$$W(Z) = \tilde{W}(Z)F(Z). \tag{3.7}$$

This can further be written as

$$|W(Z)|e^{i\theta(Z)} = |\tilde{W}(Z)|e^{i\theta_{min}(Z)}|F(Z)|e^{i\theta_F(Z)}, \tag{3.8}$$

where $|W(Z)|$ is the amplitude spectrum of the mixed phase wavelet, $|\tilde{W}(Z)|$ is the amplitude spectrum of the estimated minimum phase wavelet and $F(Z)$ is the amplitude spectrum of the all-pass wavelet, which is equal to 1. Also, $\theta(Z)$ is the phase of the mixed phase wavelet, $\theta_{min}(Z)$ is the phase of the minimum phase wavelet, and $\theta_F(Z)$ is the phase of the all-pass wavelet.
Since, $|W(Z)| = |\tilde{W}(Z)|$,

$$\theta(Z) = \theta_{min}(Z) + \theta_F(Z). \tag{3.9}$$

The problem of estimating the mixed phase wavelet can now be posed as a problem of estimating the optimum phase of the all-pass wavelet from the data whitened by an estimated minimum phase wavelet. Whitening the data with an estimated minimum phase inverse has a trade-off in terms of offering a wider bandwidth and enhanced noise level. It is observed that the practice of whitening deconvolution, in general, helps enhance the bandwidth in a frequency zone where the signal is stronger than the noise. In problems of wavelet phase estimation, the accuracy of the estimated phase depends to a large extent on the bandwidth of the data. Any amount of enhanced bandwidth in a frequency zone where the signal is stronger than the noise would greatly help in the phase estimation of the wavelet. Even though prewhitening of the data does not bring in any new information into the data, it surely helps in shaping the data in such a way that the phase estimation is performed in a more conducive data environment (White, 2006, personal communication; White, 1988; White and Simm, 2003).

3.3 Development of the algorithm

Phase estimation by cumulant matching technique is performed by using the Simulated Annealing (SA) algorithm. Simulated annealing has been applied quite successfully in many geophysical inversion problems. Many variants of simulated annealing technique such as the Metropolis algorithm (Metropolis et al., 1953; Kirkpatrick et al., 1983), Heat bath algorithm (Rebbi, 1984; Creutz, 1984; Geeman and Geeman, 1984; Rothman, 1986), Simulated annealing (Greene and Supowit, 1986), Fast simulated annealing (Szu and Hartley, 1987), Very Fast Simulated Annealing (Ingber, 1989), Mean field annealing (Peterson and Anderson, 1987; Peterson and Anderson, 1988; Peterson and Soderberg, 1989) have been developed due to their wide and successful applications.

I used the Metropolis algorithm to optimize for the model parameters b_t , the Z-transform of which forms the denominator term of the all-pass wavelet (equation 3.3). The unknowns here are the length of b_t and its coefficients. The optimization is performed by fixing p to 4. This is the minimum length of b_t that can produce a combination of real and imaginary roots on the Z-plane. Also it is noted that b_t is minimum phase in character. A minimum phase sequence is obtained by the Kolmogoroff technique (Claerbout, 1992) applied to a randomly generated sequence. The cost function for the optimization is obtained from the Bartlett-Brillinger-Rosenblatt formula (Lazear, 1993; Mendel, 1991). The formula is given by the following convolutional equation

$$C_4^s(\tau_1,\tau_2,\tau_3) = C_4^\gamma(\tau_1,\tau_2,\tau_3) * M_4^w(\tau_1,\tau_2,\tau_3) + C_4^\nu(\tau_1,\tau_2,\tau_3), \tag{3.10}$$

where $C_4^s(\tau_1,\tau_2,\tau_3)$ is the 4^{th} order cumulant of the seismic trace, $C_4^\gamma(\tau_1,\tau_2,\tau_3)$ is the 4^{th} order cumulant of a non-Gaussian, statistically independent, and identically distributed reflectivity sequence, $M_4^w(\tau_1,\tau_2,\tau_3)$ is the 4^{th} order moment of the wavelet and $C_4^\nu(\tau_1,\tau_2,\tau_3)$ is the 4^{th} order cumulant of a Gaussian additive noise. The cumulant lags are represented by τ_1, τ_2 and τ_3. Under the assumption that there exists an infinite number of data along with the other assumptions about the reflectivity sequence mentioned earlier, the term $C_4^\gamma(\tau_1,\tau_2,\tau_3)$ reduces to an impulse at the central lag, scaled by the kurtosis (γ^γ) of the reflectivity series. The term $C_4^\nu(\tau_1,\tau_2,\tau_3)$ involving the Gaussian noise reduces to zero. Hence, under the above assumptions, equation 3.10 becomes

$$C_4^s(\tau_1, \tau_2, \tau_3) = \gamma^\gamma M_4^w(\tau_1, \tau_2, \tau_3), \tag{3.11}$$

where γ^γ is the kurtosis of the reflectivity sequence which is a constant. The kurtosis is zero for a purely Gaussian reflectivity series (Lazear, 1993). Thus, the cost function for the optimization is given by

$$J = \sum_{\tau_1}\sum_{\tau_2}\sum_{\tau_3}[\tilde{C_4}^s(\tau_1, \tau_2, \tau_3) - \tilde{M_4}^w(\tau_1, \tau_2, \tau_3)]^2, \tag{3.12}$$

where $\tilde{C_4}^s(\tau_1, \tau_2, \tau_3)$ is the fourth-order trace cumulant (normalized by the central lag cumulant) and $\tilde{M_4}^w(\tau_1, \tau_2, \tau_3)$ is the fourth-order wavelet moment (normalized by the central lag moment). Figure 3.1 show the flowcharts for the estimation of the mixed phase wavelet from the whitened data.

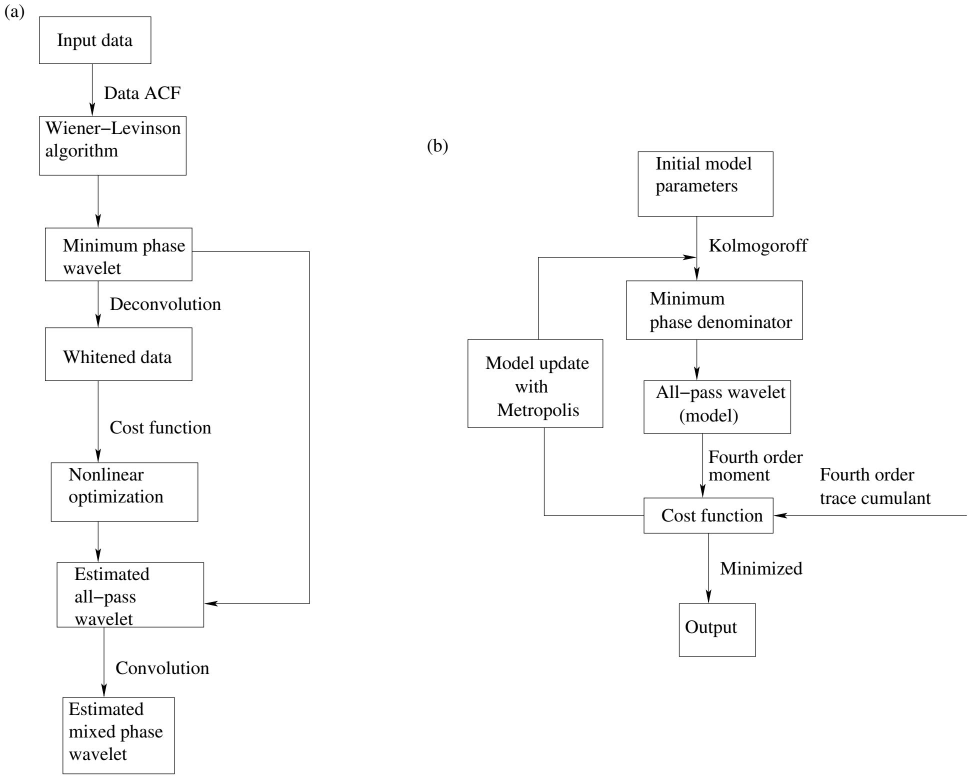

Figure 3.1: Flowcharts for the algorithm. (a) Estimation of minimum phase and mixed phase wavelets. (b) Cumulant matching and model updating with Metropolis algorithm.

Table 3.1: The roots of the synthetic wavelet.

No.	1	2	3	4	5	6	7	8	9
Magnitude	$\frac{1}{1.3}$	$\frac{1}{1.5}$	$\frac{1}{1.11}$	1.2	1.3	1.8	1.95	1.25	1.8
Phase (degree)	0	0	+45 -45	+5 -5	+60 -60	0	180	+120 -120	+160 -160

3.4 Synthetic data example

The proposed algorithm for estimating the mixed phase wavelet is tested by designing a synthetic mixed phase wavelet and a synthetic trace. Table 3.1 shows the roots of the Z-transform of the wavelet coefficients of the synthetic mixed phase wavelet. A similar wavelet was used by Porsani and Ursin to test their algorithm (Porsani and Ursin, 2000). Figure 3.2a shows the synthetic trace. The synthetic trace was generated by convolving a Laplacian mixture distribution of reflectivity sequence (length of the data points $N = 250$) with the true mixed phase wavelet. This particular distribution of the reflectivity series was chosen so as to obtain a better approximation of the true reflectivity distribution (Walden and Hosken, 1986). The Laplacian mixture distribution was obtained by generating two separate Laplacian random deviates and mixing them together by means of a mixing parameter. The first Laplacian deviates was generated by using a Laplace parameter (σ_1) equal to 0.007. The second Laplacian deviate was generated by using a Laplace parameter (σ_2) equal to 0.017. The mixing parameter in this case was chosen to be 23% of the deviates generated by the smaller Laplace parameter. The trace does not contain any noise component. Figure 3.2b shows the data after deconvolution with the estimated minimum phase wavelet. The whitened data so obtained have a larger bandwidth than the original data and contain only the phase information of the wavelet as the amplitude information has been effectively removed by the deconvolution. Hence, the technique of cumulant matching reduces to the matching of the fourth-order moment of the all-pass wavelet and the fourth-order cumulant of the whitened data. Figure 3.2c shows the true mixed phase wavelet. Figure 3.2d shows the estimated minimum phase wavelet obtained from the data by the Wiener-Levinson algorithm. Figure 3.2e shows the estimated mixed phase wavelet for a model length $p = 4$. The correlation measure between the true wavelet and the estimated mixed phase wavelet is 0.99.

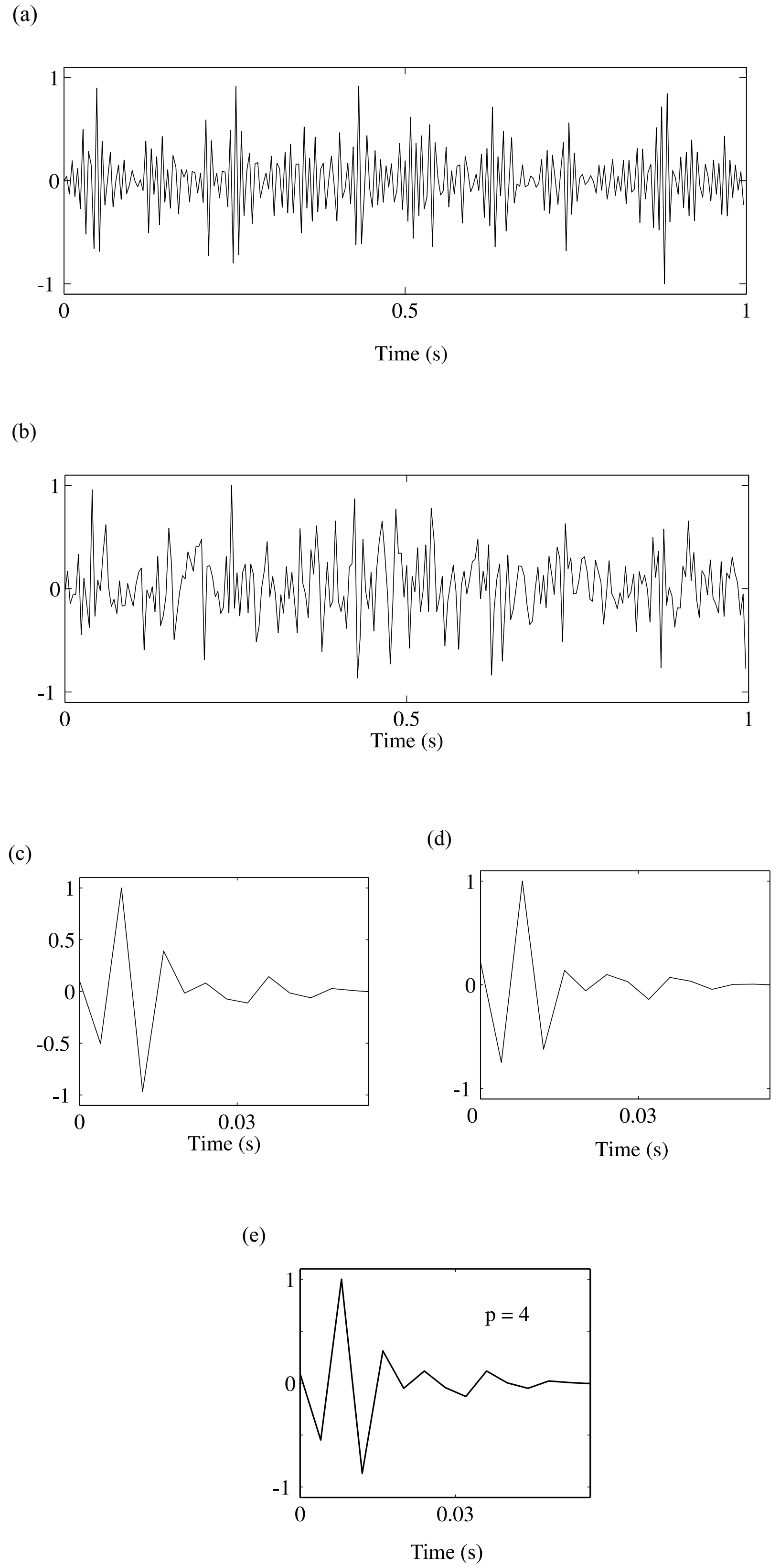

Figure 3.2: Synthetic data example. (a) Synthetic data. (b) Whitened data. (c) True wavelet. (d) Minimum phase wavelet. (e) Mixed phase wavelet.

3.5 Comparison of the results with and without prewhitening

A comparison is called for between the estimation of the mixed phase wavelet with the proposed algorithm and that directly obtained from the data. The cumulant matching technique is not sensitive to phase when the data bandwidth to central frequency ratio is less than 1. Hence, it is expected that cumulant matching technique will not be able to perform well when the mixed phase wavelet is estimated from severely bandlimited data that have the ratio of bandwidth to central frequency less than 1. The proposed technique has the advantage of removing the wavelet amplitude spectrum from the data within a frequency zone where the signal is stronger than the noise, thus resulting in a wider bandwidth of the whitened data. This allows the cumulant matching technique to work in a favorable domain and should improve the results.

A mixed phase wavelet with a bandwidth to central frequency ratio of 0.5 was chosen for the purpose of illustration. Figure 3.3a shows the synthetic trace that was generated by convolving a bandlimited wavelet with a reflectivity series of length N = 250. The reflectivity series has a Laplacian mixture distribution generated by the same parameters mentioned earlier. The synthetic trace does not have any noise component. Figure 3.3b shows the whitened data after minimum phase deconvolution. Figure 3.3c shows the true bandlimited wavelet with the ratio of bandwidth to central frequency equal to 0.5. Figure 3.3d shows the corresponding minimum phase wavelet estimated from the data. Figure 3.4a shows the estimated mixed phase wavelet for a model length p = 4. The estimation is performed over a whitened trace that is obtained by deconvolving the trace with the estimated minimum phase wavelet. The correlation measure between the estimated wavelet and the true wavelet is calculated to be 0.99. The algorithm was further applied on the same data and a mixed phase wavelet was estimated without deconvolving with the estimated minimum phase wavelet. Figure 3.4b shows the estimated mixed phase wavelet obtained from the unwhitened data. The correlation measure obtained for this estimation dropped to 0.89. In order to further substantiate the above test, I conducted 200 Monte Carlo simulations with different realizations of the synthetic data for different number of data points. I used a zero phase bandlimited Ricker wavelet (central frequency = 30 Hz, time sample interval = 0.004s) in the simulations. The simulations for estimations from whitened and unwhitened data were performed for number of data points N = 250 and N = 500. Figure 3.5a shows the correlation measures between the estimated and the true wavelets for both pre-whitened and unwhitened data. The dashed line represents the estimation from unwhitened data and the solid line represents the estimation from the pre-whitened data. Figure 3.5b shows the error bars for the normalized root-mean-square error between the estimated and the true wavelets for 200 simulations for N = 250. Figure 3.5c shows the correlation between the estimated and the true wavelets for 200 simulations with N = 500. Figure 3.5d shows the normalized root-mean-square error between the estimated and true wavelets for N = 500. The mean correlation is high with low variance when estimating from pre-whitened data compared with the values when estimating from unwhitened data. It is also observed that the normalized root-mean-square error is low with smaller variance when estimating from pre-whitened data. These observations corroborate the fact that the cumulant matching technique is sensitive to the ratio of the bandwidth to central frequency. The wavelet was estimated more accurately when this ratio was improved by minimum phase deconvolution.

3.6 Effect of noise and number of data points

The algorithm was tested on different levels of noise in the synthetic data. The data were synthetically generated by convolving a zero phase Ricker wavelet (central frequency = 30 Hz and sampling interval = 0.004s) with a randomly generated reflectivity sequence that followed a Laplacian mixture distribution. The distribution of the data was obtained using the Laplace parameters and the mixing parameter mentioned earlier. The test was carried out over a number of data samples N = 500. A total number of 200 Monte Carlo simulations were performed for each noise level defined in terms of the signal-to-noise ratio given by the following equation

$$SNR = \frac{max(|\mathbf{d}|)}{\sigma}, \tag{3.13}$$

where $\mathbf{d}$ is the data and σ is the standard deviation of the noise. Signal-to-noise ratio values of SNR =[4 20 50 100] were used to test the stability of the algorithm.

Figure 3.6a shows the error bars for the correlation measure between the true wavelet and the estimated wavelet. As anticipated, the correlation measure between the true wavelet and the estimated wavelet shows an increase as the SNR is increased

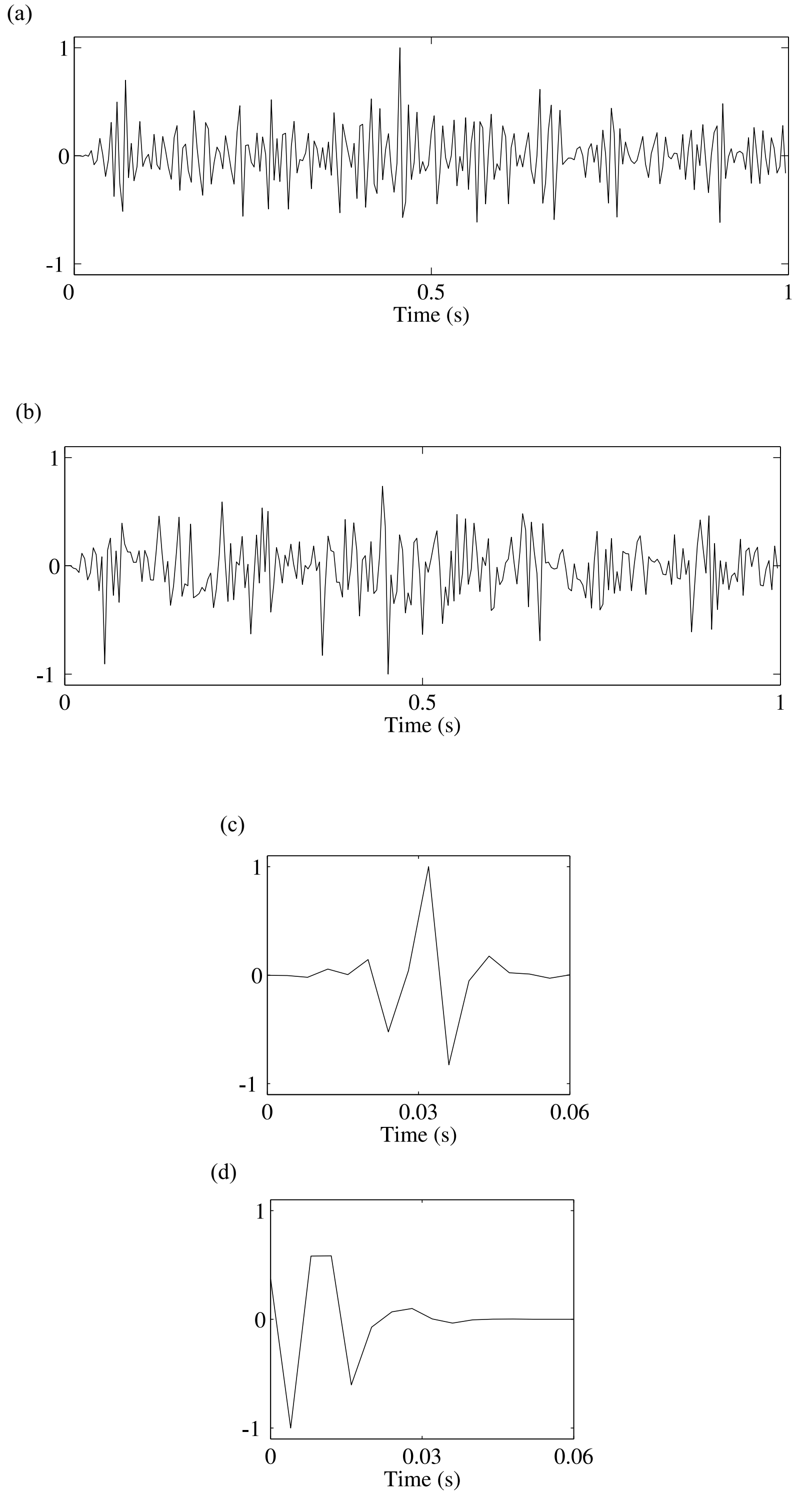

Figure 3.3: (a) Synthetic trace. (b) Whitened trace. (c) True bandlimited wavelet. (d) Estimated minimum phase wavelet.

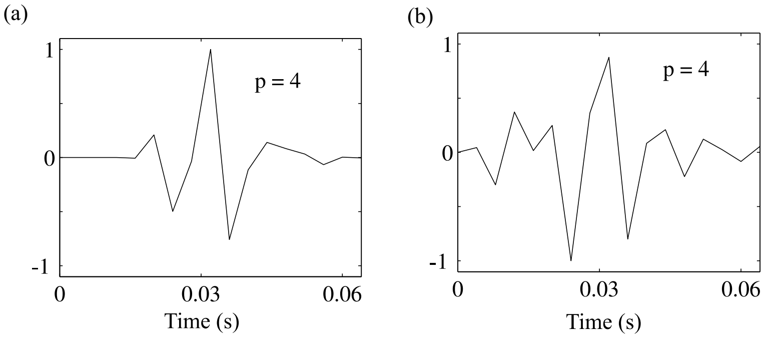

Figure 3.4: (a) Estimated mixed phase wavelet obtained from whitened data. (b) Estimated mixed phase wavelet obtained from nonwhitened data.

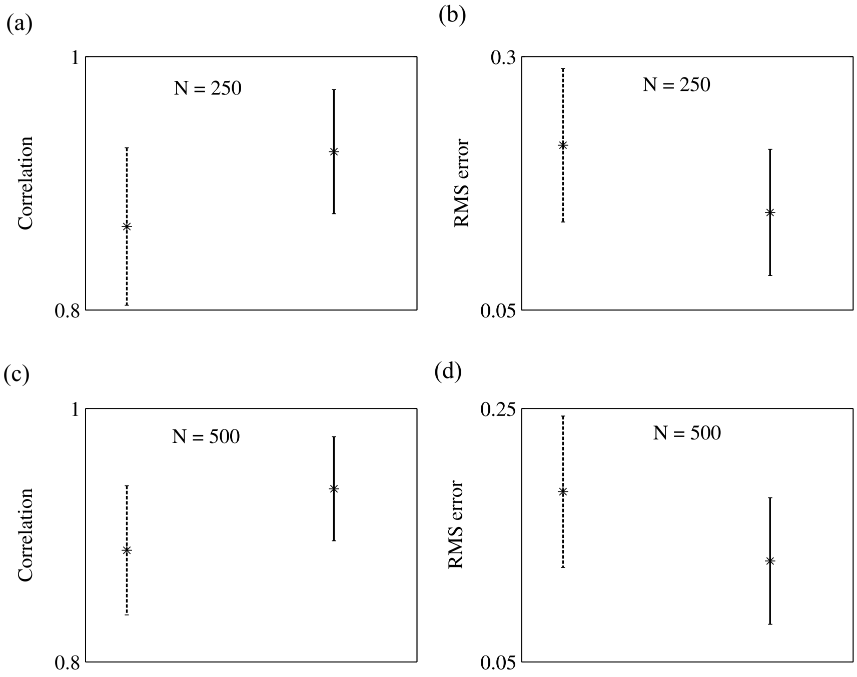

Figure 3.5: Comparison between the estimations from pre-whitened and non-whitened data. N = number of data points considered for estimation. The dashed line is for the estimations from non-whitened data and the solid line is for the estimations from the pre-whitened data. (a) The correlation measure between the estimated and true wavelets for N = 250. (b) The normalized root-mean-square error between the estimated and the true wavelets for N = 250. (c) The correlation measure between the estimated and true wavelets for N = 500. (d) The normalized root-mean-square error between the estimated and the true wavelets for N = 500. I have used 200 Monte Carlo simulations.

from 4 to 20 and then remains almost flat as the algorithm enters into a more stable regime of SNR. Figure 3.6b shows the normalized root-mean-square (rms) error plotted against the SNR. The normalized rms error between the true wavelet and the estimated wavelet goes down as the SNR is increased from 4 to 20 and then remains almost flat. There exists a trade-off between the degree of prewhitening and the amplification of noise (White 1984; White 1988; White and Simm 2003). The Monte Carlo simulations show that the algorithm operates in a more stable domain when the SNR is close to 20 and above. Figure 3.7 shows the phase spectrum of the estimated wavelet for the 200 Monte Carlo simulations. The phase spectrum has been de-trended to remove the linear trend in the phase that has been brought in by the constant time shift. Since the true wavelet is zero phase, the recovered phase (after de-trending) should be close to zero for all frequencies. I considered 200 numbers of Monte Carlo simulations for a total number of data points N=500 with a signal-to-noise ratio $SNR =$[4 20 50 100] . The SNR was defined previously (equation 3.13). Figure 3.7a shows the error bars for the de-trended phase spectra for SNR = 4. Figure 3.7b shows the error bars for the de-trended phase spectra for $SNR = 20$. Figure 3.7c shows the de-trended phase spectra for SNR = 50. Figure 3.7d shows the de-trended phase spectra for SNR = 100. It is observed that as the SNR increases, the accuracy in the phase estimation also increases.

It is known that the statistical methods of wavelet estimation greatly depend on the data volume. As discussed previously, the cost function (equation 3.12) is obtained under the assumption (along with the assumptions on the statistical properties of the reflectivity sequence) that there exists infinite number of data. This makes it necessary to run the proposed algorithm on varying number of data so as to obtain a measure of the stability of the algorithm. The following numbers of data points $N =$[100, 250, 500] were considered to test the algorithm. A total number of 200 Monte Carlo simulations were performed for a given number of data points. The SNR was kept fixed at 40 during the whole simulations. Figure 3.8a shows the correlation measure for different number of data points. As anticipated, the correlation measure between the true wavelet and the estimated wavelet increased as the number of data points increased from 100 to 250 and then to 500. Figure 3.8b shows the normalized root-mean-square (rms) error plotted against the number of data points. The normalized root-mean-square error is calculated between the true wavelet and the estimated wavelet. The normalized rms error decreased as the number of data points is increased. It appears that for all practical purposes,

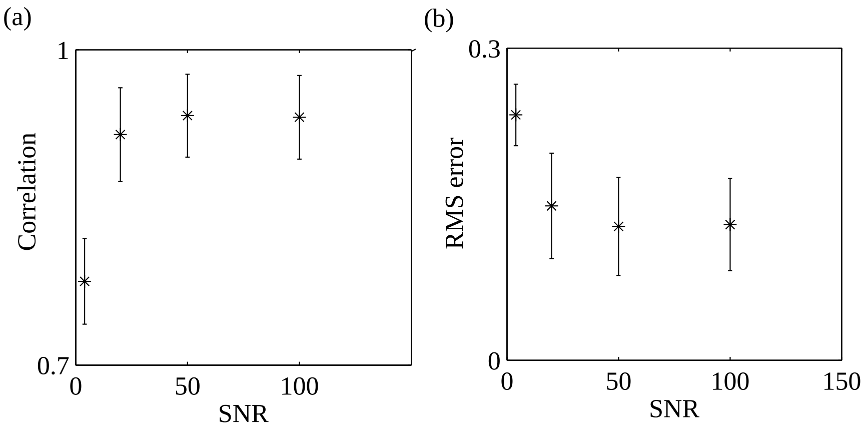

Figure 3.6: Test of the stability of the algorithm with different signal-to-noise ratio. (a) The correlation measure plotted against the SNR. (b) The normalized root-mean-square error plotted against $SNR = [4, 20, 50, 100]$. I have used 200 Monte Carlo simulations.

with a reasonable number of data points, the algorithm is capable of estimating the wavelet with reasonable accuracy. Estimation of the wavelet from a given set of data also depends on the wavelet being stationary. In the context of this chapter, it is assumed that the wavelet is stationary in both the spatial and temporal axes and the issue of non-stationarity is beyond the scope of the chapter.

3.7 Real data example

A stacked seismic section was considered for testing the algorithm. Seismic data with 77 traces and 200 time samples were windowed from the stacked section. The average cumulant was calculated for the data window and incorporated in the cost function (equation 3.12) for the estimation of the all-pass operator. An average minimum phase wavelet was estimated from the data by the Wiener-Levinson algorithm. The data were pre-whitened by minimum phase deconvolution.

Figure 3.9a shows the estimated minimum phase wavelet obtained from the data by the Wiener-Levinson algorithm. Figure 3.9b shows the estimated mixed phase wavelet obtained from the data with the proposed algorithm. Figure 3.10a shows the true stacked section. Figure 3.10b shows the section after being deconvolved with the estimated minimum phase wavelet. Figure 3.10c shows the de-phased stacked section after subtracting the phase of the estimated all-pass operator from

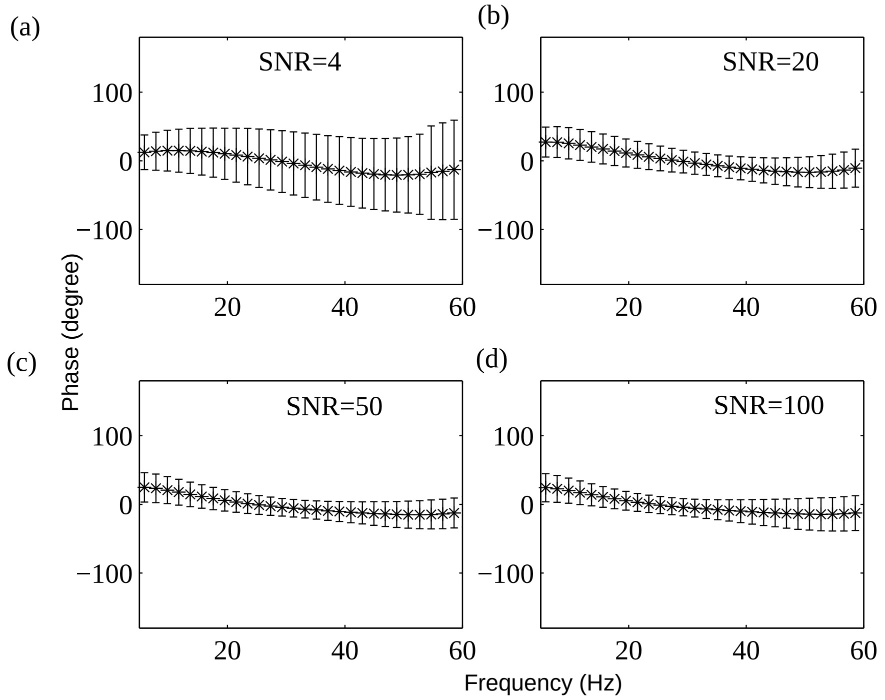

Figure 3.7: The error bars for the de-trended phase spectrum of the estimated wavelet for different noise levels in the data defined in terms of the signal-to-noise ratio (SNR). (a) $SNR = 4$. (b) $SNR = 20$. (c) $SNR = 50$. (d) $SNR = 100$. Number of data points $N = 500$. I have used 200 Monte Carlo simulations.

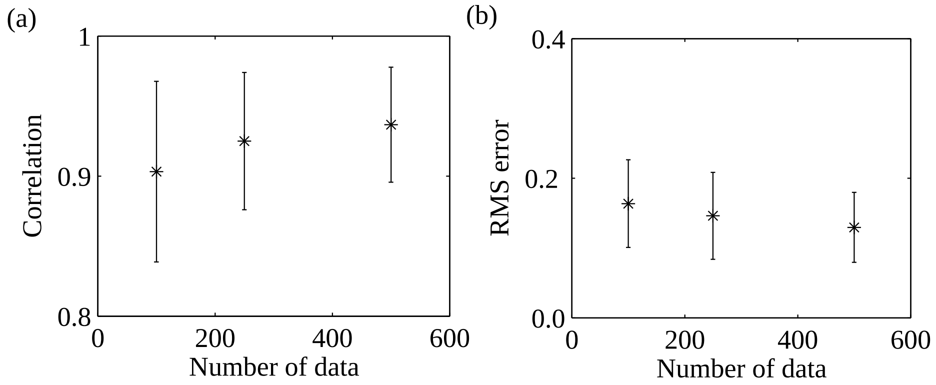

Figure 3.8: Test of the stability of the algorithm with different number of data. (a) The correlation measure plotted against the number of data. (b) The normalized root-mean-square (rms) error plotted against the number of data. I have used 200 Monte Carlo simulations. SNR of the data is 40.

the minimum phase deconvolved data shown in the figure 3.10b. In order to test the algorithm, the deconvolved data were further convolved with a 90^0 constant phase rotated synthetic wavelet. The algorithm was applied to this data to check if the phase rotated wavelet could be effectively recovered (Hargreaves, 1994). Figure 3.11a shows the extracted wavelet after a 90^0 phase rotation. Figure 3.11b shows the wavelet recovered by the proposed algorithm. The algorithm does effectively estimate the phase rotated wavelet.

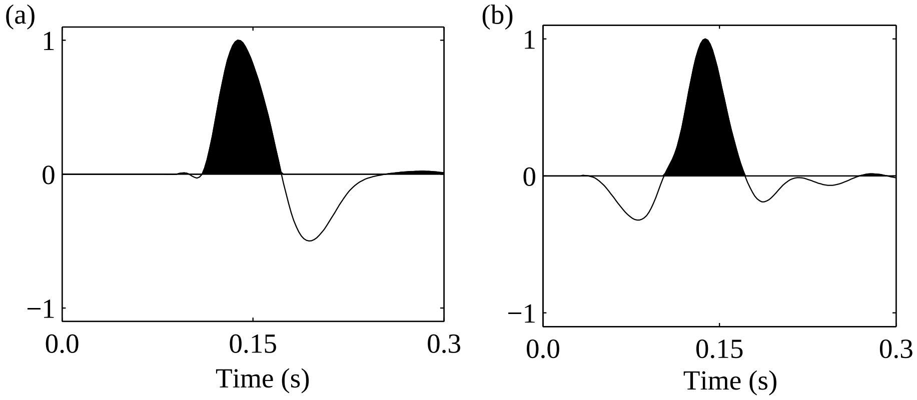

Figure 3.9: (a) The estimated minimum phase wavelet from the real data. (b) The estimated mixed phase wavelet.

3.8 Summary

Minimum phase deconvolution enhances the bandwidth of the data. Since the mixed phase wavelet and its corresponding minimum phase wavelet have the same amplitude spectrum, minimum phase deconvolution effectively removes the amplitude spectrum of the wavelet. This leaves the data requiring only a phase correction. The required phase correction is attainable by means of a simple and short parameterization of the mixed phase wavelet. This is the main advantage of the proposed algorithm. The optimization algorithm matches the 4^{th} order cumulant of the whitened data with the 4^{th} order moment of the all-pass operator. This cumulant matching technique works well when the bandwidth to central frequency ratio in the data is greater than 1. The technique is most suitable when this ratio is close to 2. The proposed technique separates the minimum phase part of the wavelet from the data by deconvolving the data with an estimated minimum phase

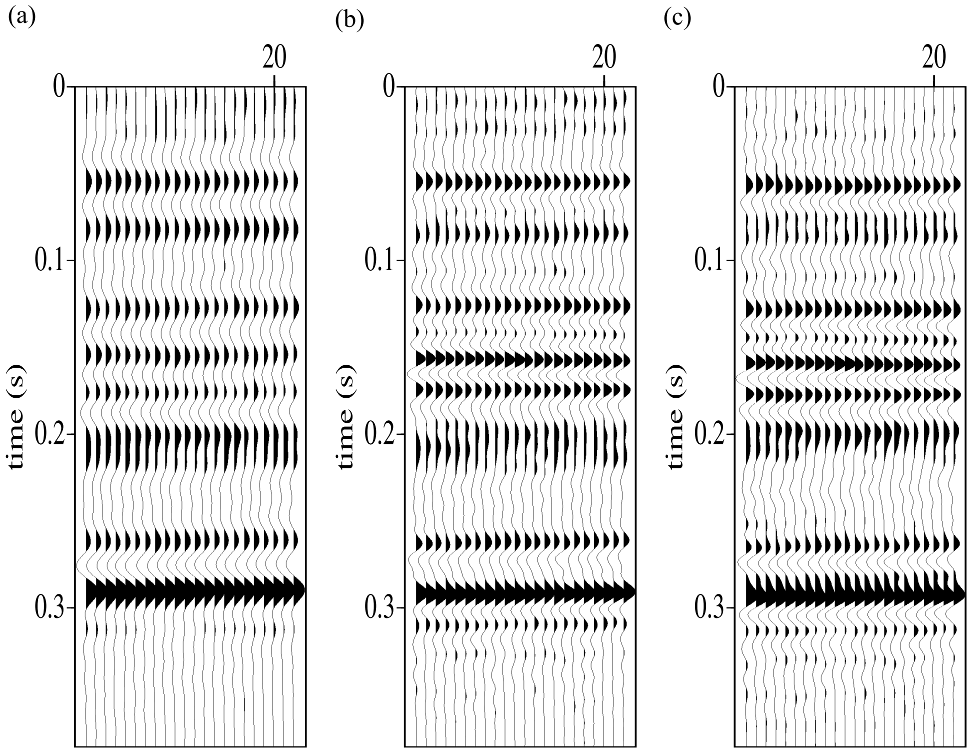

Figure 3.10: Real data example. (a) A window of the data. (b) Minimum phase deconvolution of the data. (c) Mixed phase deconvolution of the data. Result of the minimum phase deconvolution is illustrated here for a comparison with the result obtained from the mixed phase deconvolution. The average fourth-order cumulant is calculated over 77 traces and 200 time samples.

inverse. As a result, the deconvolved data contain only the phase signature of the mixed phase wavelet. This also allows the cumulant matching technique to work in a favorable regime of the bandwidth to central frequency ratio. The synthetic data examples showed that Metropolis algorithm can be used quite effectively to estimate the all-pass wavelet and hence the mixed phase wavelet. The chapter also presented a comparison between wavelet estimation from the whitened data and from unwhitened and severely bandlimited data. When the data were bandlimited, the estimated mixed phase wavelet had a relatively poor correlation with the true wavelet. Suitable parameterization of the wavelet and subsequent whitening of the data improved the estimation of the mixed phase wavelet. The algorithm was also tested on real data. A test with an artificial 90^0 phase rotation subsequently recovered the correctly phase rotated wavelet.

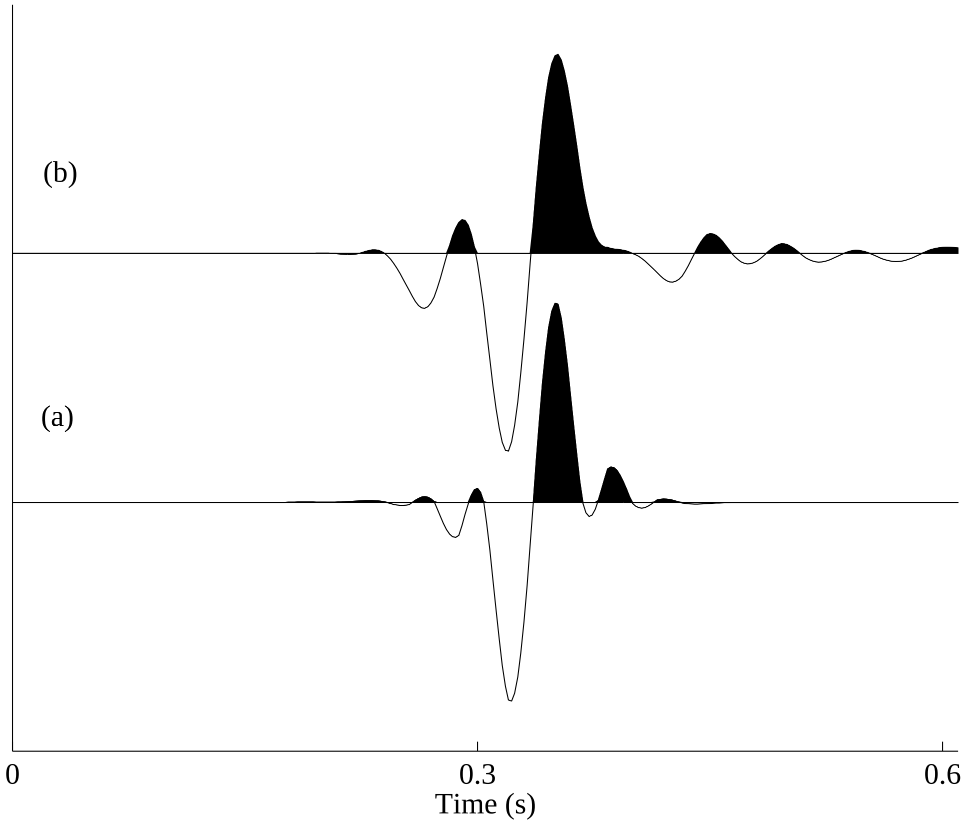

Figure 3.11: (a) The true 90^0 constant phase rotated wavelet. (b) The recovered wavelet.

Chapter 4

Regularizations

4.1 Introduction

Inverse problems are classified into three categories depending on whether or not the model can be uniquely determined. An even-determined problem is one where the measured data provide enough information to uniquely determine the model. The cost function containing the data misfit term alone will be zero for the uniquely determined solution. An even-determined problem is rare. In practice, the measured data are inadequate and contain noise, thus preventing any unique solution to the problem. When the information contained in the data are inadequate to obtain a unique solution, the inverse problem is classified as under-determined. Likewise, if the information contained in the data are more than what is required to obtain a unique solution then the problem is classified as the over-determined problem.

In problems where data do not contain enough information to uniquely estimate the solution, it is necessary to incorporate prior knowledge about the model space so as to constrain the solution to a meaningful regime. Incorporation of *a priori* knowledge to constrain the cost function for a meaningful solution is termed as "regularization". One simple regularization is to constrain the length of the model parameters.

$$J = \mathbf{e}^T\mathbf{e} + \mu\mathbf{m}^T\mathbf{m}, \tag{4.1}$$

where the data misfit $\mathbf{e} = \mathbf{d} - \mathbf{Gm}$ and the model norm is given by $\mathbf{m}^T\mathbf{m}$. The trade-off parameter is given by μ. Larger trade-off parameter makes the model norm dominant over the data misfit term and vice-versa. Following the least-squares minimization procedure for the equation 4.1, we obtain

$$\tilde{\mathbf{m}} = (\mathbf{G}^T\mathbf{G} + \mu\mathbf{I})^{-1}\mathbf{G}^T\mathbf{d}, \tag{4.2}$$

where $\tilde{\mathbf{m}}$ represents the estimated model. The solution is known as the "damped least-squares solution". The minimum norm solution imposes a constraint in the solution space such that the solution is clustered around zero. In many instances, this is an undesirable effect. The minimum norm constraint can be easily modified to include the *a priori* knowledge about the mean of the model. The following cost function imposes a constraint on the model space such that the solution is favored to cluster around the known average model.

$$J = \mathbf{e}^T\mathbf{e} + \mu(\mathbf{m} - \bar{\mathbf{m}})^T(\mathbf{m} - \bar{\mathbf{m}}), \tag{4.3}$$

where $\bar{\mathbf{m}}$ represent the average model.

A smooth solution can be interpreted as a simple solution. Thus, in certain circumstances it is desired that the solution be smooth. Smoothness constraint is imposed on the model space by penalizing larger deviation among the adjacent model parameters. A derivative operator when acts on the model imposes smoothness in the solution. The following equation shows inclusion of smooth regularization term in the cost function.

$$J = \mathbf{e}^T\mathbf{e} + \mu(\mathbf{D}(\mathbf{m} - \bar{\mathbf{m}}))^T\mathbf{D}(\mathbf{m} - \bar{\mathbf{m}}). \tag{4.4}$$

Equation 4.4 is usually represented in terms of an weighting matrix W_m as shown in the following equation.

$$J = \mathbf{e}^T\mathbf{e} + \mu(\mathbf{m} - \bar{\mathbf{m}})^T\mathbf{W}_m(\mathbf{m} - \bar{\mathbf{m}}), \tag{4.5}$$

where the weighting matrix $\mathbf{W_m} = \mathbf{D}^T\mathbf{D}$. Finite difference approximations of first and second order derivative operators are given by

$$\mathbf{D}_1 = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$\mathbf{D}_2 = \begin{pmatrix} 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Minimization of the cost function given by the equation 4.4 results in the estimated model $\tilde{\mathbf{m}}$ given by

$$\tilde{\mathbf{m}} = (\mathbf{G}^T\mathbf{G} + \mu\mathbf{D}_1^T\mathbf{D}_1)^{-1}\mathbf{G}^T\mathbf{d}. \tag{4.6}$$

Regularizaton terms described above are quadratic functions. Hence, they pose as linear problems. However, many situations require different *a priori* constraints to be imposed. Certain *a priori* constraints are non-quadratic and hence result in nonlinear inverse problems. In the following sections, I discus two important nonquadratic regularization functions to impose (a) sparseness constraint and (b) blockyness constraint in the model space.

4.1.1 Sparseness constraint

A sparse solution is characterized by a sequence of model parameters that has many zeros and few non-zero values (Youzwishen, 2001). Figure 4.1 shows an example of sparse solution.

A long-tailed prior probability distribution function with zero mean is a suitable probability function for imposing sparseness constraint on the solution. Two types of prior probability distribution functions, namely, the exponential and the Cauchy distribution functions are usually used to impose sparseness constraint in the model. The exponential probability distribution function is given by

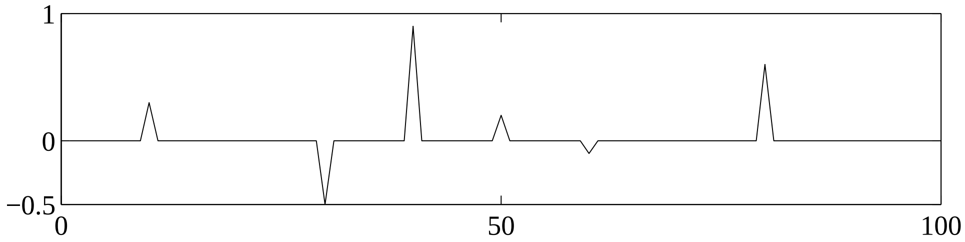

Figure 4.1: An example of sparse solution.

$$P_E(\mathbf{m}) = \frac{1}{2\sigma_m} e^{-\frac{1}{\sigma_m}|m_i - <\mathbf{m}>|}, \tag{4.7}$$

where, $P_E(\mathbf{m})$ is the probability, σ_m^2 is the variance of the model parameters, m_i is the i^{th} model parameter and $< \mathbf{m} >$ is the mean of the model vector. The Cauchy probability distribution is given by

$$P_C(\mathbf{m}) = \frac{1}{(\pi\sigma_m)} [\frac{1}{1 + \frac{(m_i - <\mathbf{m}>)^2}{\sigma_m^2}}], \tag{4.8}$$

where $P_C(\mathbf{m})$ is the probability, σ_m^2 is the variance of the model parameters, m_i is the i^{th} model parameter and $< \mathbf{m} >$ is the mean of the model vector.

Figure 4.2 shows the probability distribution functions for the Gaussian, exponential and Cauchy distributions. It is noticed that compared to the Gaussian distribution, the exponential and the Cauchy distribution functions offer greater freedom for the non-zero values to deviate from zero. Also these distributions are more tightly centered around zero. Such long-tailed distributions around zero are suitable distribution functions for imposition of sparseness constraint on the solution.

Let $\mathbf{m}$ be the model, $\mathbf{G}$ be the forward operator, $\mathbf{d}$ be the observed data and $\mathbf{n}$ be the noise vector. The linear problem is represented as

$$\mathbf{d} = \mathbf{Gm} + \mathbf{n}, \tag{4.9}$$

where $\mathbf{m} = [m_1, m_2, ..., m_M]$ are the model parameters, $\mathbf{d} = [d_1, d_2, ..., d_N]$ are the observed data, $\mathbf{n} = [n_1, n_2, ..., n_N]$ are the noise and $\mathbf{G}$ the forward operator relating the model with the data.

In the Bayesian framework, the posterior probability distribution is given by

$$P(\mathbf{m}|\mathbf{d}) = K[\prod_{i=1}^{M} \frac{1}{1 + m_i^2 \sigma_m^2}] e^{-\frac{1}{2\sigma^2}(\mathbf{Gm}-\mathbf{d})^T(\mathbf{Gm}-\mathbf{d})}, \tag{4.10}$$

where K is a constant and σ is the variance of the noise.
The cost function such that the posteriori distribution is maximized is given by

$$J(\mathbf{m}) = (\mathbf{Gm} - \mathbf{d})^T(\mathbf{Gm} - \mathbf{d}) + \mu R(\mathbf{m}), \tag{4.11}$$

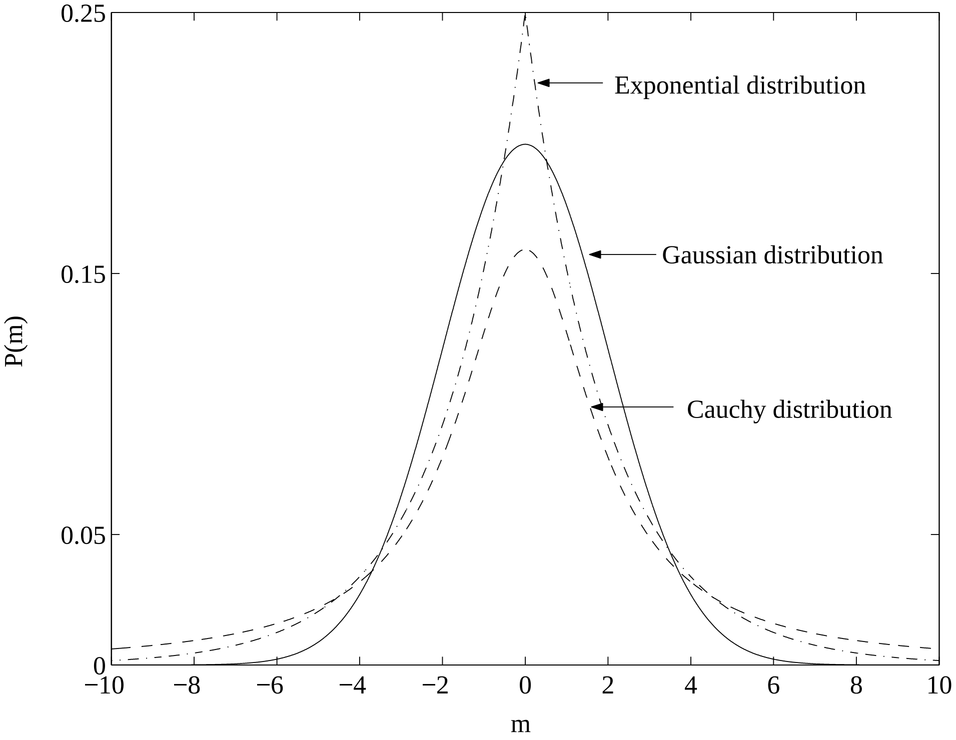

Figure 4.2: The probability distribution functions.

where μ is the trade-off parameter and the regularization term $R(\mathbf{m})$ is given by

$$R(\mathbf{m}) = \sum_{i=1}^{M} ln(1 + \frac{m_i^2}{\sigma_m^2}). \tag{4.12}$$

The cost function J is minimized by taking the derivative with respect to the model parameter m_i and equating to zero. The solution is given by the nonlinear equation

$$\mathbf{G}^T\mathbf{G}\mathbf{m} - \mathbf{G}^T\mathbf{d} + \mu\mathbf{Q}\mathbf{m} = 0, \tag{4.13}$$

where $\mathbf{Q}$ is given by

$$\mathbf{Q} = \begin{pmatrix} \frac{1}{1+\frac{m_1^2}{\sigma_m^2}} & 0 & \dots & 0 \\ 0 & \frac{1}{1+\frac{m_2^2}{\sigma_m^2}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{1+\frac{m_M^2}{\sigma_m^2}} \end{pmatrix}.$$

Equation 4.13 is solved by the method of Iterative Re-weighted Least-Squares (IRLS) method. The estimated model is given by

$$\tilde{\mathbf{m}} = (\mathbf{G}^T\mathbf{G} + \mu\mathbf{Q})^{-1}\mathbf{G}^T\mathbf{d}. \tag{4.14}$$

The solution at the k^{th} IRLS iteration is given by

$$\tilde{\mathbf{m}}_k = (\mathbf{G}^T\mathbf{G} + \mu\mathbf{Q}_{k-1})^{-1}\mathbf{G}^T\mathbf{d}. \tag{4.15}$$

The iteration is terminated when the following condition is met (Sacchi, 1997)

$$\frac{|J_k - J_{k-1}|}{(|J_k| + |J_{k-1}|)/2} \leq \epsilon, \tag{4.16}$$

where ϵ represents the user defined tolerance. In the next section, an example is discussed where sparseness is imposed on the solution by including the Cauchy regularization function in the cost function.

4.1.2 Real data example

A zero-offset seismogram without noise is modeled as

$$d_t = w_t * r_t, \tag{4.17}$$

where d_t represents the recorded data at time t, w_t represents the unknown wavelet and r_t represents the unknown reflectivity. The '*' represents the convolution operation. The aim of inversion is to recover r_t from the measured data d_t.

Since, earth acts as a filter to the propagating seismic waves, the recorded data are bandlimited. Data contain no information about the high frequency and also the low frequency components of underlying reflectivity series. Thus, estimation of fullband reflectivity from data modeled as convolution of reflectivity and wavelet, is a nonunique linear inversion process. The high frequency components lost due to the filtering effects of the wavelet belong to the null space. Hence, to estimate information about the fullband reflectivity requires regularization of the cost function so that a parsimonious solution honoring the data is preferred during the optimization process. The following cost function is optimized for estimation of reflectivity from data.

$$J = \|(\mathbf{w} * \mathbf{r} - \mathbf{d})\|^2 + \mu R(\mathbf{r}), \tag{4.18}$$

where data are represented by $\mathbf{d}$, $\mathbf{r}$ is the reflectivity model that needs to be estimated and $\mathbf{w}$ is the unknown wavelet, $R(\mathbf{r})$ represents the Cauchy regularization term and μ is the trade-off parameter. The Cauchy regularization term, $R(\mathbf{r})$ is given by the equation 4.12. The cost function (equation 4.18) is minimized with a conjugate gradient algorithm.

The sparse inversion algorithm is tested on an inline post-stack data from a 3D survey. It is assumed that the data have been properly de-phased by deconvolution with an estimated wavelet. The algorithm proceeds with estimating a zero phase wavelet for each trace and deconvolving the trace with the estimated wavelet. Figure 4.3a shows data before sparse deconvolution. It is noticed that, due to the wavelet, data lack high frequency components. Figure 4.3b shows the data after sparse inversion. Sparse inversion has resulted in enhancing the resolution of the data. Figure 4.4a shows the spectrum of data before sparse spike deconvolution. Figure 4.4b shows the spectrum of data after sparse spike deconvolution. On comparison, it is noticed that the sparse spike inversion resulted in enhancing the bandwidth of the data by restoring the high frequency components. Figure 4.5 shows a time slice of the seismic cube. Figure 4.5a shows the time slice of the data before applying the high frequency restoration algorithm. Figure 4.5b shows the same time slice after applying the high frequency restoration algorithm. It is observed that there is marked improvement in the spatial resolution in the data by incorporating the sparseness constraint defined in terms of a Cauchy distribution.

4.2 Blockyness constraint

Blocky constraints are imposed in inversion algorithms where the *a priori* information dictates that the solution should be piecewise constant. Such situations occur in regions where geological formations have sharp boundaries resulting in edges in the model. Blocky constraints have the effects of sharpening the image while preserving the edges consistent with the *a priori* information.

Blocky solution will result when the first order derivative of the solution is sparse. For example, the inversion for impedance (I) is blocky ($I = \int_i r_i di$) since the reflec-

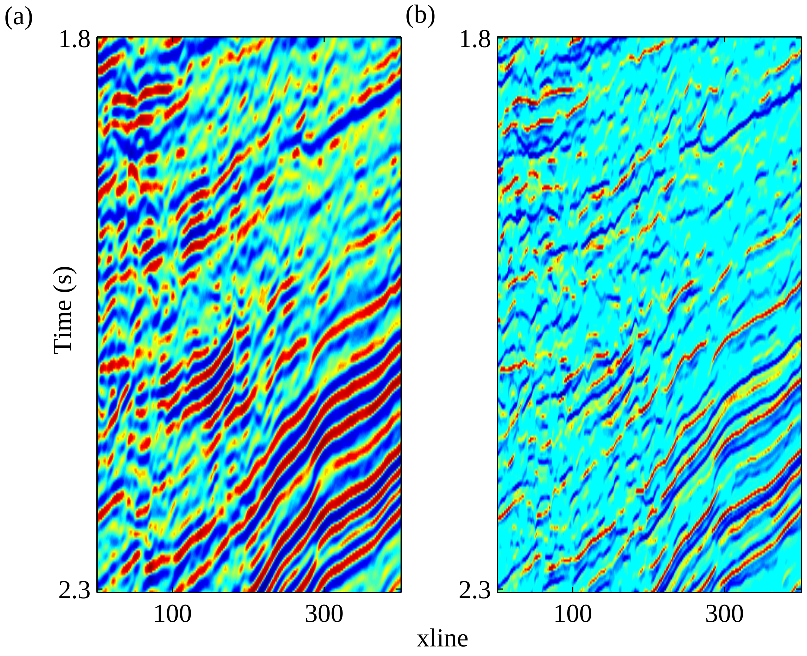

Figure 4.3: (a) Data without high frequency restoration. (b) Data with high frequency restoration obtained by sparse spike inversion.

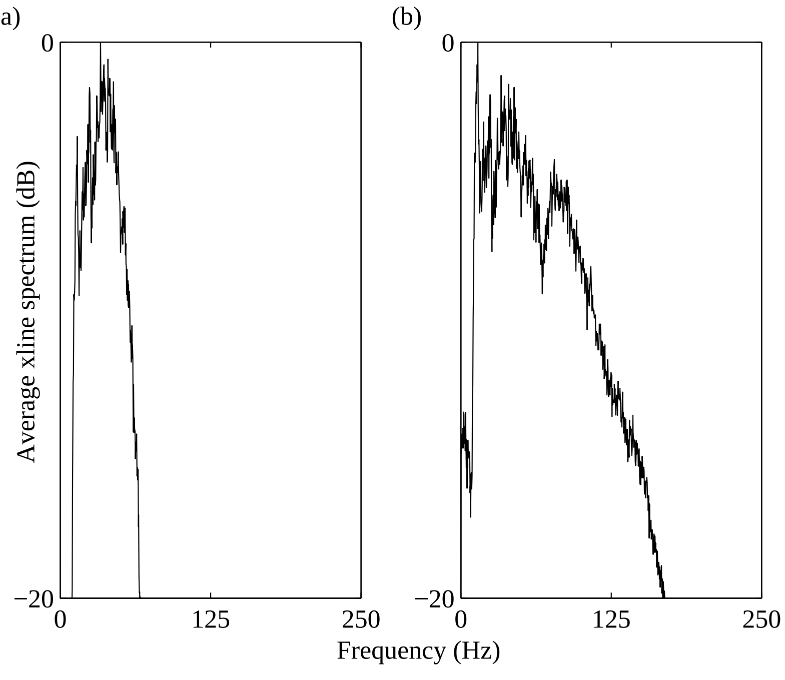

Figure 4.4: (a) Average amplitude spectrum without high frequency restoration. (b) Average amplitude spectrum with high frequency restoration.

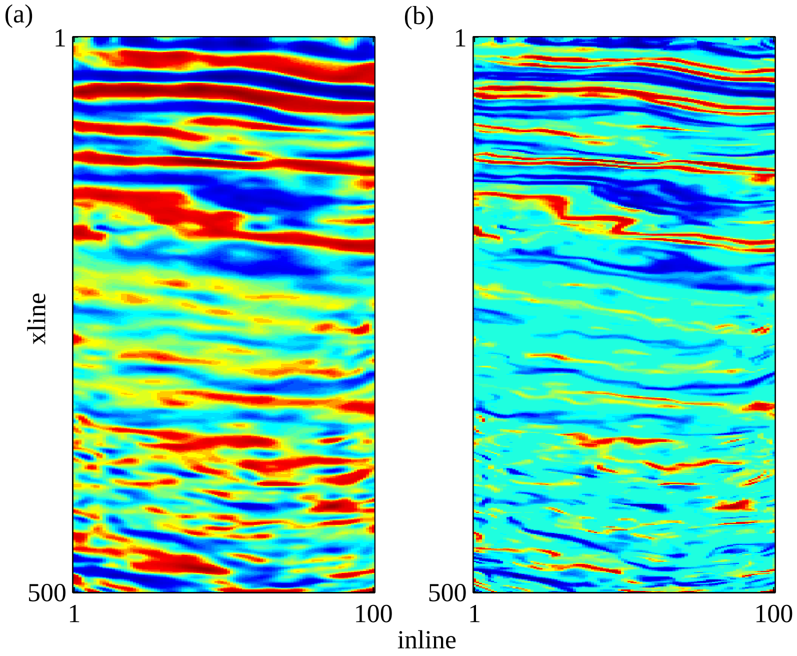

Figure 4.5: (a) Time slice of a seismic volume without high frequency restoration. (b) Time slice with high frequency restoration.

tivity (r_i) is sparse. Due to the property that the blocky regularization preserves the edges in the image, such regularization is also referred to as the Edge Preserving Regularization (EPR) function. The EPR technique has been applied successfully to obtain blocky solutions in the inversion of gravity data (Portniaguine and Zhdanov, 1999), magnetic resonance imaging (Charbonnier et al., 1997; Barone, 1999) and radio astronomy (Molina et al., 2001).

As mentioned earlier, a function becomes piecewise constant or blocky if its first order derivative is sparse. In the previous section, I discussed ways to enforce sparseness in the solution. The same reasoning of sparseness constraint can be extended to blockyness constrain provided that the regularization function enforcing sparseness operates on the first order derivative of the model parameters.

Considering the Cauchy regularization function for the sparseness constraint and operating it on the first order derivative of the model parameters, we obtain a form of the cost function as follows

$$J(\mathbf{m}) = (\mathbf{Gm} - \mathbf{d})^T(\mathbf{Gm} - \mathbf{d}) + \mu_x R(\mathbf{D}_x\mathbf{m}) + \mu_z R(\mathbf{D}_z\mathbf{m}), \tag{4.19}$$

where $\mathbf{G}$ is the forward operator, $\mathbf{m}$ is the model, $\mathbf{d}$ is the data and μ_x and μ_z are the trade-off parameters applied to the regularization terms $R(\mathbf{D}_x\mathbf{m})$ and $R(\mathbf{D}_z\mathbf{m})$ respectively. The subscripts in the trade-off parameters denote the direction in which the regularization terms are applied on a two dimensional model space. The horizontal and vertical derivative operators are denoted by the matrix $\mathbf{D}_x$ and $\mathbf{D}_z$ respectively. The elements in the derivative operator matrices $\mathbf{D}_x$ and $\mathbf{D}_z$ are given by the following equations

$$\begin{aligned}(D_xm)_{ij} &= \frac{m_{i+1,j} - m_{i,j}}{\delta}, \\ (D_zm)_{ij} &= \frac{m_{i,j+1} - m_{i,j}}{\delta}. \end{aligned} \tag{4.20}$$

The two dimensional model space encompasses a matrix of size $\mathbf{M}_x$ by $\mathbf{M}_z$. The indices $i = 1, 2, ..., M_x$ and $j = 1, 2, ..., M_z$ define the position of the derivative operators in the derivative matrices. The regularization function operating on the model derivative results in the enforcement of blocky solution in the inversion algorithm. The Cauchy regularization function applied to the first order model parameter derivative is given by

$$\begin{aligned}R(\mathbf{D}_x\mathbf{m}) &= \sum_{i=1}^{Mx*Mz} \phi([D_xm]_i), \\ R(\mathbf{D}_z\mathbf{m}) &= \sum_{i=1}^{Mx*Mz} \phi([D_zm]_i). \end{aligned} \tag{4.21}$$

For a modified Cauchy prior distribution the function $\phi(t)$ is defined as

$$\phi(t) = \frac{t^2}{1+t^2}. \tag{4.22}$$

In the following chapters, I am applying the edge preserving regularization (EPR) function to estimate the subsurface earth elastic parameters ($\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$) from the amplitude variation with offset (AVO) data. I have approached the problem of simultaneous estimation of the earth elastic parameters such as the P-wave velocity, S-wave velocity and the density from two different perspectives, (a) estimation is obtained from the angle-dependent reflectivity calculated from the Aki and Richards approximation equation and (b) estimation is obtained by the waveform inversion of seismic data exhibiting AVO. The first approach involves linearized inversion with nonlinear conjugate gradient (NLCG) optimization algorithm and the second approach is based on global optimization algorithm where the blockyness in the solution is enforced through a new technique of model preconditioning operators.

4.3 Summary

In this chapter, I discussed regularization in inverse problems. When the problem is ill-posed, there exists no unique solution to the inverse problem. A meaningful solution can be obtained by incorporating *a priori* information to the optimization algorithm. I discussed commonly used quadratic regularization such as the model norm and smoothing regularization. When the *a priori* information dictates parsimony in the solution space, the sparseness regularization function is used. Sparseness regularization function is nonquadratic which lead to nonlinear problem. Such problems are solved iteratively. I discussed an example to recover the high frequency components in the seismic data lost due to the bandlimited wavelet. The example showed an iterative solution with conjugate gradient solver with the cost function containing the data misfit term and Cauchy regularization term to enforce sparseness in the solution. I also discussed blockyness constraint with the Cauchy regularization function acting on the first order derivative of the model. Application of blockyness constraint in the estimation of the earth elastic parameters from AVO data using a linearized optimization approach is presented in the next chapter.

Chapter 5

Local Optimization: Application to AVO Inversion

5.1 Introduction

In the last chapters, I reviewed the theory of optimization procedures as applied in highly complex situations. By highly complex, I mean that the model dimension is large and the model and data are related in a rather complex form. In this chapter, I will discuss optimization procedure for inversion of petrophysical properties from seismic data exhibiting amplitude variation with offset (AVO). I will present an optimization routine based upon nonlinear conjugate gradient algorithm with a cost function that includes data misfit and edge preserving regularization function.

Inversion of prestack amplitude to derive AVO attributes is an important seismic data analysis procedure. AVO attributes include petrophysical properties of the underlying reservoir. In exploration seismic methods, most of the information is carried by the compressional (P-) and shear (S-) waves. As opposed to the surface waves, such as the Love and Rayleigh waves, the P- and the S- waves travel in the interior of the earth and thus carry information about the deeper structures that help in understanding the underlying reservoir characteristics. In case of compressional or P-waves, the particle motion is along the direction of motion of the propagating wave. While the wave is in motion, the particle volume changes due to the compressional stress. In case of shear waves, the particle motion is perpendicular to the direction of the wave propagation. P-wave velocities vary within a very wide range because they depend on a large number of physical parameters such as pore pressure, fluid saturation and porosity etc. Thus, knowledge about P-wave velocity alone is not sufficient to obtain accurate estimation about the petrophysical properties of the underlying earth structures. The S-wave velocity information is used to substantiate the information gained from the P-wave velocities. Knowledge about P-wave and S-wave velocities together provide information about the Poisson's ratio. Estimation of elastic parameters from the reflection seismic data showing variation of amplitude with offset/angle is known as AVO/AVA inversion. Estimation of the elastic parameters/ rock properties from the reflected and transmitted seismic waves that carry the signatures of the angle dependent reflectivities have been studied in great details. The earliest works in this field are attributed to Knott (1899). Wang (1999) used the Zoeppritz equations, a set of four equations derived by Zoeppritz (1919) to compute the amplitudes of reflected and transmitted waves, to estimate the reflection and transmission coefficients from tomographic inversion. However, the relationship between the rock properties and the angle dependent reflection coefficients as established by the Zoeppritz equations is complex and nonlinear. Such complexity makes the inversion of elastic parameters a rather difficult task. Koefoed (1955) predicted a relationship between the angle dependent reflection coefficients and the elastic parameters that can be inverted to obtain the unknown elastic parameters. Subsequently, the works of Bortfeld (1961), Aki and Richards (1980), Shuey (1985), Fatti et al. (1994), Wang (1999) resulted in several approximation equations of the Zoeppritz equations. The underlying assumptions in obtaining the approximations to Zoeppritz equations are (Rüger, 2002)

1. The difference between the P-, S- wave velocities and density across the reflecting boundary is small.

2. The angle of incidence of the impinging seismic wave is much smaller than the critical angle.

Among all the approximations, the Aki and Richards approximation equation for the PP reflection coefficient is most widely used. In the following section, I will discuss Zoeppritz equations and their various approximations with special emphasis on the Aki and Richards approximation equation. I give special attention to Aki and Richards approximation equation because the forward model, used in the local optimization scheme that I describe in the following sections, is based on this approximation.

5.2 Zoeppritz equations and approximation

The reflected and transmitted P- and S-wave components of a non-normal incident P-wave are given by the Zoeppritz equations. The derivation of Zoeppritz equations is obtained by following the approach of Officer (1958) for a simple 2D medium with an elastic half-space underlying an accoustic layer. Figure 5.1 shows the 2D earth model. Following the derivation proposed by Officer (1958), I define the compressional and shear wave potentials in terms of the displacement vector $\mathbf{u}(u, w)$ as follows

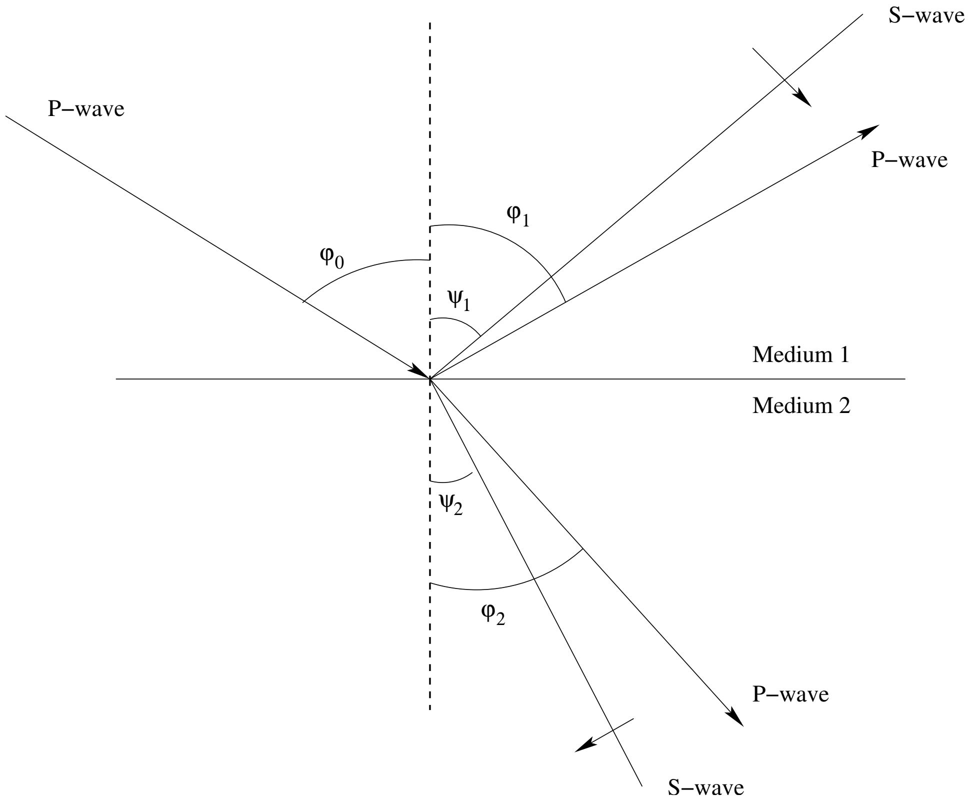

Figure 5.1: A 2D earth model separating two different media. The angle of incidence for P-wave is given by ϕ_0, the angle of reflection for P-wave and S-wave are ϕ_1 and ψ_1 respectively. The angle of refraction for P-wave and S-wave are ϕ_2 and ψ_2 respectively.

$$\begin{aligned} u &= \frac{\partial \phi_P}{\partial x} + \frac{\partial \phi_S}{\partial z}, \\ w &= \frac{\partial \phi_P}{\partial z} - \frac{\partial \phi_S}{\partial z}, \end{aligned} \tag{5.1}$$

where the compressional wave and shear wave potentials are given by ϕ_P and ϕ_S respectively. It can be shown that the potentials defined as above satisfy the compressional and shear wave equations shown below.

$$
\begin{aligned}
\frac{\partial^2 \phi_P}{\partial x^2} + \frac{\partial^2 \phi_P}{\partial z^2} &= \frac{1}{V_p^2} \frac{\partial \phi_P}{\partial t^2}, \\
\frac{\partial^2 \phi_S}{\partial x^2} + \frac{\partial^2 \phi_S}{\partial z^2} &= \frac{1}{V_s^2} \frac{\partial \phi_S}{\partial t^2},
\end{aligned}
\tag{5.2}
$$

where V_p and V_s are the P-wave and S-wave velocities respectively. The plane wave solutions of the two displacement potentials namely, ϕ and ψ, in the cartesian coordinate system of (x,z) are given by the following equations.

$$
\begin{aligned}
\phi_1 &= A_0 e^{i \frac{\omega}{V_{p1}} \sin \phi_0 x + i \frac{\omega}{V_{p1}} \cos \phi_0 z - i\omega t} + A_1^{i \frac{\omega}{V_{p1}} \sin \phi_1 x + i \frac{\omega}{V_{p1}} \cos \phi_1 z - i\omega t}, \\
\psi_1 &= B_1 e^{i \frac{\omega}{V_{s1}} \sin \psi_1 x - i \frac{\omega}{V_{s1}} \cos \psi_1 z - i\omega t}, \\
\phi_2 &= A_2 e^{i \frac{\omega}{V_{p2}} \sin \phi_2 x + i \frac{\omega}{V_{p2}} \cos \phi_2 z - i\omega t}, \\
\psi_2 &= B_2 e^{i \frac{\omega}{V_{s2}} \sin \psi_2 x - i \frac{\omega}{V_{s2}} \cos \psi_2 z - i\omega t},
\end{aligned}
\tag{5.3}
$$

where A_0, A_1, B_1, A_2 and B_2 are the wave amplitudes for the incident P-wave, reflected P-wave, reflected S-wave, transmitted P-wave and transmitted S-wave respectively. The P-wave and S-wave velocities for the medium-1 are given by V_{p_1} and V_{s_1} respectively. The P-wave and S-wave velocities for the medium-2 are given by V_{p_2} and V_{s_2} respectively. The P-wave incident, reflected and transmitted angles are given by ϕ_0, ϕ_1 and ϕ_2 respectively. The S-wave reflected and transmitted angles are given by ψ_1 and ψ_2 respectively.

At the layer boundary $z = 0$, the following four boundary conditions are satisfied.

1. Tangential displacement component is continuous across the boundary $[u_1 = u_2]$.

2. Normal displacement component is continuous across the boundary $[w_1 = w_2]$.

3. Normal stress component is continuous across the boundary $[(P_{zz})_1 = (P_{zz})_2]$.

4. Tangential stress component is continuous across the boundary $[(P_{xz})_1 = (P_{xz})_2]$.

Considering the first boundary condition, we obtain

$$\frac{\partial \phi_1}{\partial x} + \frac{\partial \psi_1}{\partial z} = \frac{\partial \phi_2}{\partial x} + \frac{\partial \psi_2}{\partial z}. \tag{5.4}$$

Differentiating the displacement potentials ϕ and ψ with respect to x and z and using Snell's law we obtain,

$$\frac{\sin\phi_1}{V_{p_1}}A_0 + \frac{\sin\phi_1}{V_{p_1}}A_1 - \frac{\cos\psi_1}{V_{s_1}}B_1 = \frac{\sin\phi_2}{V_{p_2}}A_2 + \frac{\cos\psi_2}{V_{s_2}}B_2. \tag{5.5}$$

This is the first Zoeppritz equation.

Similarly, by imposing the second boundary condition and using the Snell's law we obtain,

$$\frac{\cos\phi_1}{V_{p_1}}A_0 - \frac{\cos\phi_1}{V_{p_1}}A_1 - \frac{\sin\psi_1}{V_{s_1}}B_1 = \frac{\cos\phi_2}{V_{p_2}}A_2 - \frac{\sin\psi_2}{V_{s_2}}B_2. \tag{5.6}$$

This is the second Zoeppritz equation.

In order to impose the third boundary condition, we need to obtain an expression for the normal stress (P_{zz}) in terms of the displacement potentials ϕ and ψ. We know that,

$$P_{zz} = \lambda\Delta + 2\mu e_{zz}, \tag{5.7}$$

where λ and μ are the Lame's constants and Δ represents the dilatation or the fractional change in volume. The normal strain is represented by e_{zz}. When the volume becomes infinitesimally small, the dilataion (Δ) is given by the sum of the three principal strain components (e_{xx}, e_{yy} and e_{zz}). In the xz-plane, $e_{yy} = 0$. Thus,

$$\begin{aligned}\Delta &= e_{xx} + e_{zz}, \\ &= \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}.\end{aligned} \tag{5.8}$$

Substituting the expressions for Δ and e_{zz} in the expression for P_{zz} and rearranging the terms,

$$P_{zz} = (\lambda + 2\mu)\frac{\partial w}{\partial z} + \lambda\frac{\partial u}{\partial x}. \tag{5.9}$$

Replacing the displacement potentials u and w by the compressional and shear wave potentials ϕ and ψ respectively and taking the spatial derivatives and rearranging the terms we obtain,

$$P_{zz} = (\lambda + 2\mu)(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial z^2}) - 2\mu(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\psi}{\partial z\partial x}). \tag{5.10}$$

Substituting the expressions for the compressional wave equation and the S-wave velocity in terms of μ and density ρ and rearranging the terms we obtain,

$$P_{zz} = \rho\frac{\partial^2\phi}{\partial t^2} - 2\rho V_s^2(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\psi}{\partial z\partial x}). \tag{5.11}$$

As per the third boundary condition, $(P_{zz})_1 = (P_{zz})_2$. Thus,

$$\rho_1\frac{\partial^2\phi_1}{\partial t^2} - 2\rho_1 V_{s_1}^2(\frac{\partial^2\phi_1}{\partial x^2} + \frac{\partial^2\psi_1}{\partial z\partial x}) = \rho_2\frac{\partial^2\phi_2}{\partial t^2} - 2\rho_2 V_{s_2}^2(\frac{\partial^2\phi_2}{\partial x^2} + \frac{\partial^2\psi_2}{\partial z\partial x}). \tag{5.12}$$

Differentiating the compressional and shear wave potentials ϕ and ψ and substituting in the above equation for $z = 0$, after rearrangement of terms we obtain,

$$C_1A_0 + C_1A_1 + \sin 2\psi_1 B_1 = \frac{\rho_2}{\rho_1}C_2A_2 - \frac{\rho_2}{\rho_1}\sin 2\psi_2 B_2, \tag{5.13}$$

where $C_1 = (1 - 2\frac{V_{s1}^2}{V_{p1}^2}\sin^2\phi_1)$ and $C_2 = (1 - 2\frac{V_{s2}^2}{V_{p2}^2}\sin^2\phi_2)$. Using Snell's law, we reformat the above equation as follows.

$$-\cos 2\psi_1 A_0 - \cos 2\psi_1 A_1 - \sin 2\psi_1 B_1 = -\frac{\rho_2}{\rho_1}\cos 2\psi_2 A_2 + \frac{\rho_2}{\rho_1}\sin 2\psi_2 B_2. \tag{5.14}$$

This is the third Zoeppritz equation.

It can be shown that the shear stress component P_{xz} is related to the shear strain e_{xz} by the following relation

$$\begin{aligned} P_{xz} &= 2\mu e_{xz}, \\ &= \mu(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}). \end{aligned} \tag{5.15}$$

Substituting the compressional- and shear- wave potentials ϕ and ψ in place of the displacement potentials u and w and taking the spatial derivatives and rearranging the terms we obtain,

$$P_{xz} = \mu(2\frac{\partial^2 \phi}{\partial z \partial x} + \frac{\partial \psi}{\partial z^2} - \frac{\partial^2 \psi}{\partial x^2}). \tag{5.16}$$

Thus, using the fourth boundary condition $(P_{xz})_1 = (P_{xz})_2$, we obtain,

$$\mu_1(2\frac{\partial^2 \phi_1}{\partial z \partial x} + \frac{\partial \psi_1}{\partial z^2} - \frac{\partial^2 \psi_1}{\partial x^2}) = \mu_2(2\frac{\partial^2 \phi_2}{\partial z \partial x} + \frac{\partial \psi_2}{\partial z^2} - \frac{\partial^2 \psi_2}{\partial x^2}). \tag{5.17}$$

Differentiating the wave potentials ϕ and ψ and rearranging the terms for $z = 0$, we obtain,

$$-\frac{\mu_1}{V_{p_1}^2} \sin 2\phi_1 A_0 + \frac{\mu_1}{V_{p_1}^2} \sin 2\phi_1 A_1 - \frac{\mu_1}{\mathbf{V}_{s_1}^2} C_3 B_1 = -\frac{\mu_2}{V_{p_2}^2} \sin 2\phi_2 A_2 - \frac{\mu_2}{\mathbf{V}_{s_2}^2} C_4 B_2, \tag{5.18}$$

where $C_3 = (\cos^2\psi_1 - \sin^2\psi_1)$ and $C_4 = (\cos^2\psi_2 - \sin^2\psi_2)$. Substituting the expression for the S-wave velocity in terms of μ and density ρ, we obtain,

$$-\sin 2\phi_1 A_0 + \sin 2\phi_1 A_1 - \frac{V_{p_1}^2}{V_{s_1}^2} \cos 2\psi_1 B_1 = -\frac{\rho_2 V_{s_2}^2 V_{p_1}^2}{\rho_1 V_{s_1}^2 V_{p_2}^2} \sin 2\phi_2 A_2 - \frac{\rho_2 V_{p_1}^2}{\rho_1 V_{s_1}^2} \cos 2\psi_2 B_2. \tag{5.19}$$

This is the fourth Zoeppritz equation. In the matrix form, for a normalized incident P-wave amplitude $A_0 = 1$, Zoeppritz equations can be written as

$$\mathbf{ZX} = \mathbf{Y}, \tag{5.20}$$

where

$$\mathbf{Z} = \begin{pmatrix} \cos\phi_1 & \frac{V_{p_1}}{V_{s_1}} \sin\psi_1 & \frac{V_{p_1}}{V_{p_2}} \cos\phi_2 & -\frac{V_{p_1}}{V_{s_2}} \sin\psi_2 \\ -\sin\phi_1 & \frac{V_{p_1}}{V_{s_1}} \cos\psi_1 & \frac{V_{p_1}}{V_{p_2}} \sin\phi_2 & \frac{V_{p_1}}{V_{s_2}} \cos\psi_2 \\ -\cos 2\psi_1 & -\sin 2\psi_1 & \frac{\rho_2}{\rho_1} \cos 2\psi_2 & -\frac{\rho_2}{\rho_1} \sin 2\psi_2 \\ \sin 2\phi_2 & -\frac{V_{p_1}^2}{V_{s_1}^2} \cos 2\psi_1 & \frac{\rho_2 V_{s_2}^2 V_{p_1}^2}{\rho_1 V_{s_1}^2 V_{p_2}^2} \sin 2\phi_2 & \frac{\rho_2 V_{p_1}^2}{\rho_1 V_{s_1}^2} \cos 2\psi_2 \end{pmatrix},$$

$$\mathbf{X} = \begin{pmatrix} A_1 \\ B_1 \\ A_2 \\ B_2 \end{pmatrix},$$

$$\mathbf{Y} = \begin{pmatrix} \cos\phi_1 \\ \sin\phi_1 \\ \cos 2\psi_1 \\ \sin 2\phi_1 \end{pmatrix}.$$

Due to the complex relationship between the rock elastic parameters such as the P-wave velocity (V_p), S-wave velocity (V_s) and density (ρ), inversion of these parameters from Zoeppritz equations is nearly impossible. In order to alleviate such problems, approximate equations to the above set of equations for the $\grave{P}\acute{P}$ reflection coefficients are suggested (Bortfeld, 1961; Aki and Richards, 1980), where incident and reflected P-waves are represented as $\grave{P}$ and $\acute{P}$ respectively. The following equation represents approximation proposed by Bortfeld (1961) for the $\grave{P}\acute{P}$ reflection amplitude.

$$R(\phi_2) = \frac{1}{2}\ln(\frac{V_{p_2}\rho_2\cos\phi_2}{V_{p_1}\rho_1\cos\phi_1}) + (2 + \frac{\ln(\frac{\rho_2}{\rho_1})}{\ln(\frac{V_{p2}}{V_{p_1}}) - \ln(\frac{V_{p_2}V_{s_1}}{V_{p_1}V_{s_2}})})\frac{V_{s_1}^2 - V_{s_2}^2}{V_{p_1}^2}\sin^2\phi_1. \quad (5.21)$$

Bortfeld equation, though much simpler compared to the Zoeppritz equations, does not provide direct relationship between angle or offset and the dependent $\grave{P}\acute{P}$ reflectivity. Thus it provides little help in AVO analysis. Aki and Richards (1980) provided an approximation equation for the $\grave{P}\acute{P}$ reflection coefficient which found immediate application in AVO analysis. The Aki and Richards approximation equation is given by the following equation.

$$R(\theta) = [\frac{1}{2}(1 + \tan^2\theta)]\frac{\Delta V_p}{V_p} - [4\frac{V_s^2}{V_p^2}\sin^2\theta]\frac{\Delta V_s}{V_s} + [\frac{1}{2}(1 - 4\frac{V_s^2}{V_p^2}\sin^2\theta)]\frac{\Delta\rho}{\rho}, \quad (5.22)$$

where

$$\begin{aligned} V_p &= \tfrac{V_{p_1}+V_{p_2}}{2} \\ V_s &= \tfrac{V_{s_1}+V_{s_2}}{2} \\ \rho &= \tfrac{\rho_1+\rho_2}{2} \\ \theta &= \tfrac{\phi_1+\phi_2}{2} \\ \Delta V_p &= V_{p_1} - V_{p_2} \\ \Delta V_s &= V_{s_1} - V_{s_2} \\ \Delta\rho &= \rho_1 - \rho_2. \end{aligned}$$

The approximation equations 5.21 and 5.22 are accurate enough within the range of angles that are ordinarily achieved in seismic reflection data acquisition survey.

I am using Aki and Richards approximation equation in the forward model to estimate the earth elastic parameters from AVO data using a nonlinear conjugate gradient optimization approach.

5.3 Local optimization for earth elastic parameters from AVO data

As discussed in the previous sections, the earth elastic parameters such as V_p, V_s and ρ are related to the $\grave{P}\acute{P}$ angle dependent reflectivity through the Zoeppritz equations. The Aki and Richards equation is used in place of the Zoeppritz equations because of their relative simplicity and acceptable accuracy within the data acquisition aperture. I have used Aki and Richards approximation equation in the forward model to perform the inversion for the earth elastic parameters such as the P-wave velocity (V_p), S-wave velocity (V_s) and the density (ρ). I perform optimization with (a) cost function containing the data misfit and (b) with a cost function containing data misfit along with an edge preserving regularization function to ensure that the solution is stepwise smooth or in other words blocky, representing the layer boundaries. The later approach helps in reducing the inherent nonuniqueness in the problem by restricting the solution space to blocky models.

5.3.1 Model space

The unknowns in the inversion of the earth elastic parameters for different subsurface layers are the P-wave and S-wave velocity, density and the thickness of the individual layers. For a zero offset seismic section, the thickness of the individual layer is related to the P-wave velocity and the vertical travel time of the seismic wave within the layer. For unknown layer thicknesses the vertical travel time is also unknown. We have over-parameterized the model space (Sen and Stoffa, 1991) by incorporating a large number of pseudo-layers such that the seismic travel time within each of the pseudo-layer is constant. I expect that when the convergence is achieved, the layers where there is no reflection event would coalesce together.

In the following synthetic data example, the true model space consists of 3 layers. By following the over-parameterization approach, the 3 layer model is represented in terms of 29 microlayers over a half-space. Each microlayer has a constant two-way time. Thus, the model space for inversion is represented in the matrix form as follows

$$\mathbf{V_p} = \begin{pmatrix} V_{p1} \\ V_{p2} \\ V_{p3} \\ \vdots \\ V_{pn} \end{pmatrix}, \qquad \mathbf{V_s} = \begin{pmatrix} V_{s1} \\ V_{s2} \\ V_{s3} \\ \vdots \\ V_{sn} \end{pmatrix}, \qquad \boldsymbol{\rho} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_n \end{pmatrix}, \tag{5.23}$$

where n represents the number of predefined microlayers in the model space (in this example, $n = 30$). Local optimization techniques do not require bounds in the model space as they always converge to the nearest local minimum with respect to the initial model. Thus a more judiciously chosen initial model is important so that the convergence of the local optimization technique will lead to a solution as close as possible to the global minimum. With this in mind, I have chosen an initial model within a bound of ±15% for $\mathbf{V_p}$ and $\boldsymbol{\rho}$ and ±60% for $\mathbf{V_s}$ with respect to the true model values. The bounds are chosen in such a way that there exists sufficient overlapping at the layer boundaries. This way, though I include a bound for the initial model, I do not provide the *a priori* information about the true layer boundaries to the local optimization algorithm. Figure 5.2 shows the true model with upper and lower bounds within which the initial model is selected.

5.3.2 Forward model

An inversion procedure requires a forward operator to generate synthetic data. I have used the ray-tracing based forward operator to compute the simulated NMO corrected seismic data. I have assumed horizontal subsurface layers of contrasting elastic parameters to compute the forward model.

The source-receiver travel path is calculated for a stack of horizontal layers by using the Snell's law and ray shooting approach (Shearer, 1999). The source point is used to shoot rays with increasing angle till the ray, after undergoing several reflections and refractions at the layer interfaces, reaches the receiver at a given location. The angles of incidence of the ray that reaches a receiver at a given offset

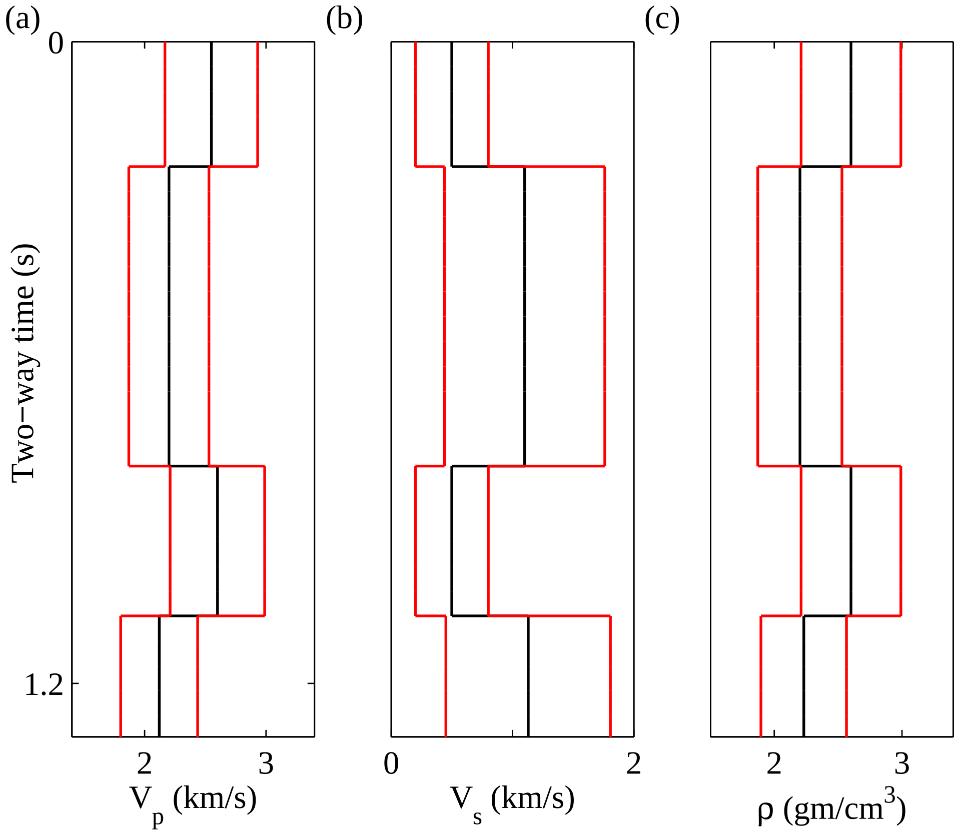

Figure 5.2: The true model, upper and lower bounds for (a) P-wave velocity (b) S-wave velocity and (c) density for the 3 layers. The black line indicates the true model and the red line indicates the upper and lower bounds within which the initial model is chosen.

is recorded. The incidence and the transmission angles of the ray is used to compute the angle-dependent $\grave{P}\acute{P}$ reflection coefficients from the Aki and Richards approximation equation for the stack of layers under consideration. Figure 5.3 shows the ray-tracing method for horizontally stratified earth layers. The reflectors are positioned at the two-way travel time in the corresponding locations of the time-offset axes. The amplitudes of the reflection are computed from the Aki and Richards approximation equation and subsequently convolved with a known wavelet (in the examples, I have used a Ricker wavelet with a central frequency of 15 Hz) to generate synthetic NMO corrected seismic gather. Algorithm 5.1 provides a pseudo-code to compute ray-tracing based forward model. In the pseudo-code, total number of subsurface layers and total number of offsets are given by *nlayer* and *noff* respectively.

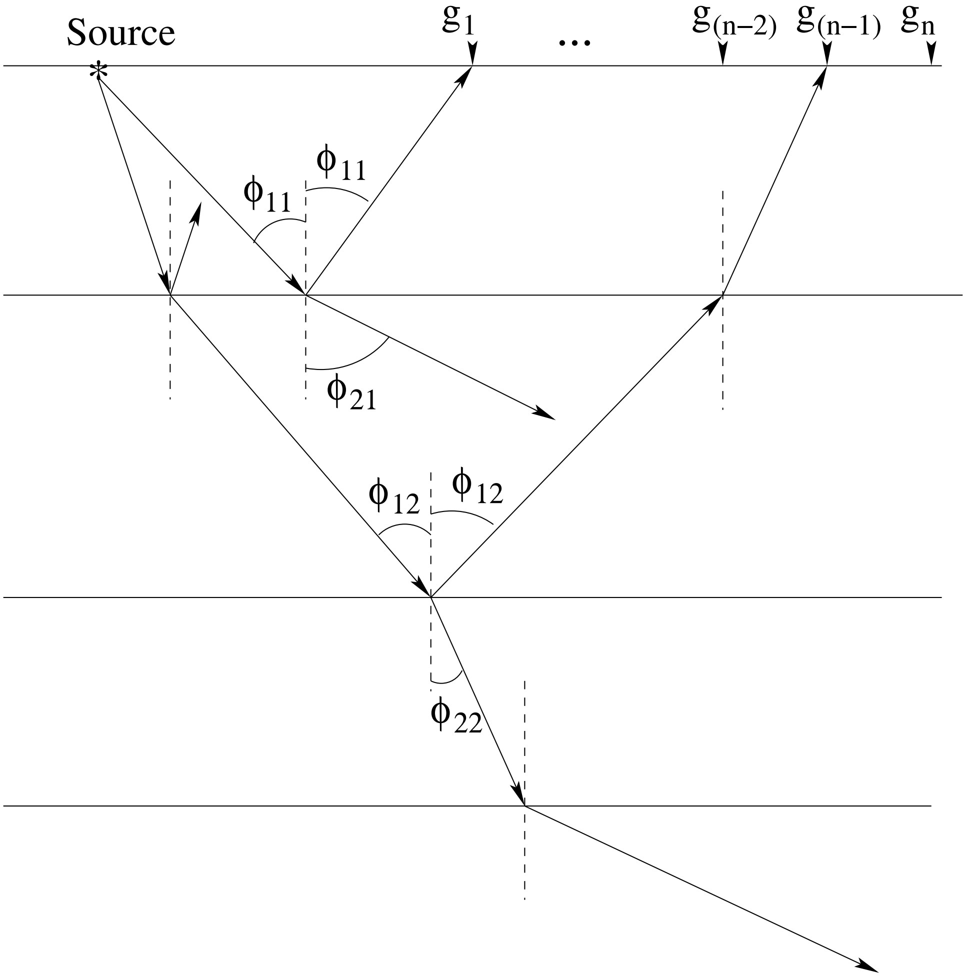

Figure 5.3: The ray-tracing method for horizontally stratified earth layers. The angles ϕ_{11} and ϕ_{21} represent the angle of incidence/reflection and angle of transmission at the first interface respectively. The angles ϕ_{12} and ϕ_{22} represent the angle of incidence/reflection and angle of transmission at the second interface respectively. The position of receivers are marked as g_1, …, $g_{(n-2)}$, $g_{(n-1)}$ and g_n.

I studied the local optimization problem in two different scenarios, (a) the cost function does not contain any *a priori* information and (b) the cost function is supplemented with the *a priori* information that the solution be blocky.

Algorithm 5.1 Pseudo-code for ray-tracing forward model

```
Set $i \leftarrow 0$; $k \leftarrow 0$;
while $i \leq nlayer$ do
  while $k \leq noff$ do
    Compute incidence and transmission angles at layer interfaces;
    Compute angle dependent reflectivities from Aki-Richards equation;
    Set $i \leftarrow i + 1$, $k \leftarrow k + 1$;
  end while
end while
Position angle dependent reflectivities at zero-offset times;
Convolve reflectivities with a known wavelet.
```

5.3.3 Optimization with Nonlinear Conjugate Gradients

The earth elastic parameters (V_p, V_s and ρ) are nonlinearly related to the data through the Aki and Richards approximation equation. The nonlinearity in the model-data relationship stems from the fact that the unknown model parameters are required to compute the angle-dependent reflectivities using the Snell's law. I approach the problem of solving the nonlinear equations with the help of nonlinear conjugate gradient (NLCG) technique. I have followed the Polak-Ribiere approach to update the conjugate direction. The method of NLCG is described in chapter 2.

Success of nonlinear conjugate gradient optimization depends a great deal on the line minimization technique. I have followed the bracketing of the minimum point followed by Brent's parabolic approximation technique to obtain the line minimization (Press et al., 2001) in every NLCG iteration.

In the first approach, I have minimized the data misfit term without incorporating any *a priori* information about the model. The following cost function was minimized by the NLCG method.

$$J = \sum_{i=1}^{Noff} \sum_{j=1}^{N_t} |d(x_i, t_j)^{obs} - d(x_i, t_j)^{est}|^2, \tag{5.24}$$

where J is the cost function for the data misfit between the observed and estimated data at the offset x_i and time t_j. Data are generated by the ray tracing approach based on Aki and Richards approximation equation. Figure 5.4 shows the true model profiles for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ and true data corresponding to the true model. True data contain no noise.

Figure 5.5 shows the initial model which is required as the starting model for

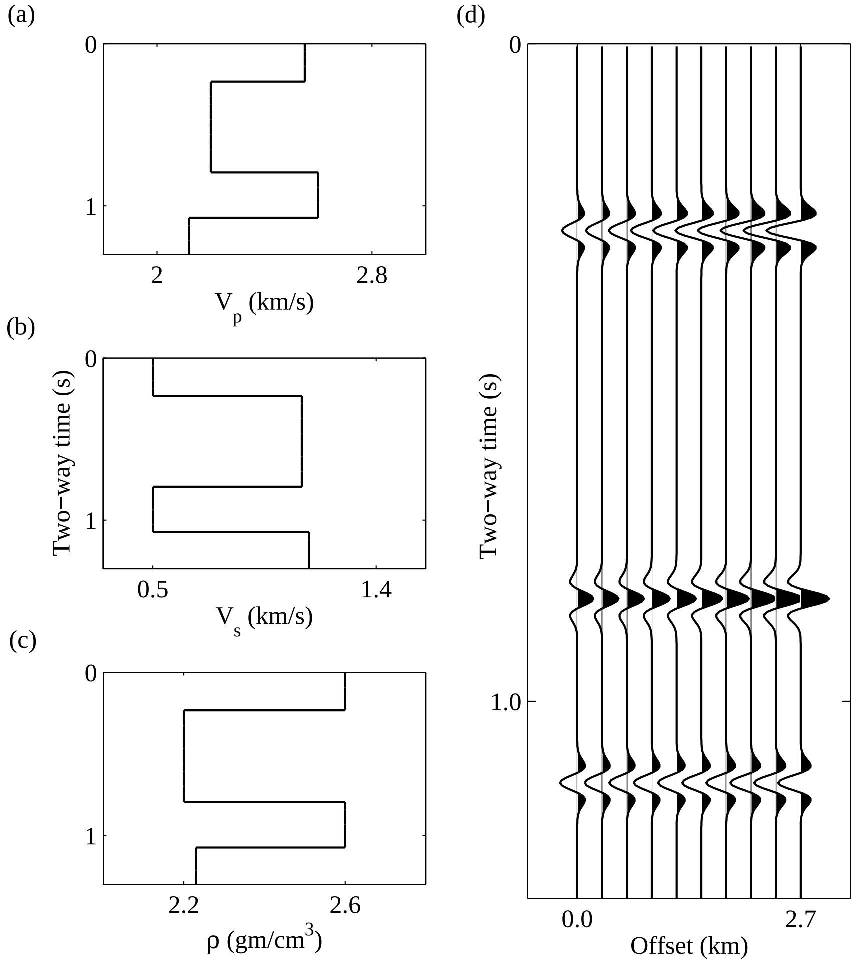

Figure 5.4: True model profiles for (a) $\mathbf{V_p}$, (b) $\mathbf{V_s}$, and (c) $\boldsymbol{\rho}$. (d) True data corresponding to the true model.

the nonlinear conjugate gradient optimization. The red lines show the upper and lower bounds, the blue line shows the true model and the black line shows the initial model. Data computed from the initial model is shown alongside. As observed in the data domain, the initial model is not close to the global minimum which occurs at the true model. Since, the data contain no noise, it is expected that at the global minimum the data misfit should be zero.

I have performed 40 NLCG iterations. The CPU time consumed to perform the 40 NLCG iterations was equal to 0.6717 hours. A 3.2 GHz Pentium 4 machine was used for the computations. Figure 5.6 shows the estimated model (black line) and the true model (blue line). Data computed for the estimated model is shown

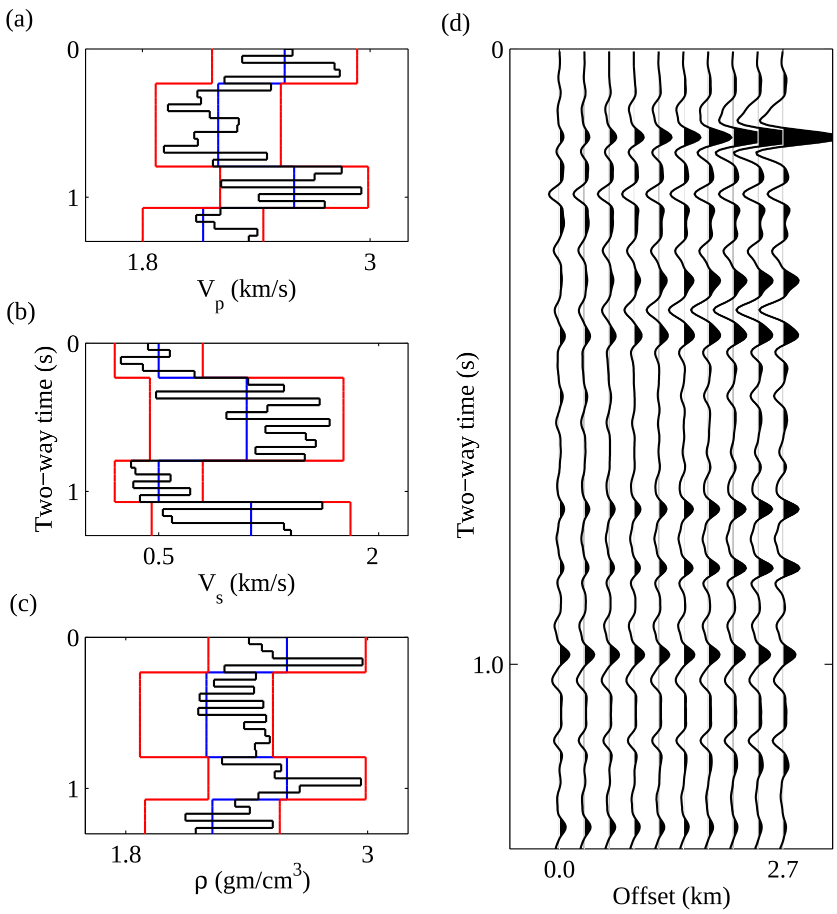

Figure 5.5: Initial model profiles (black line) for (a) $\mathbf{V_p}$ (b) $\mathbf{V_s}$ and (c) $\boldsymbol{\rho}$ are shown superimposed on the true model profiles (blue lines). The red lines show the upper and lower bounds for the individual models. (d) Synthetic data corresponding to the initial model. Data contain no noise.

alongside. Figure 5.7 shows the convergence curve for the NLCG optimization when the cost function contains only the data misfit term.

We observe that the NLCG optimization algorithm converged to a solution that honors the data well. However, the estimated model is far from the true model because the estimated model does not show blocky characteristics of the true model. This is due to the inherent nonuniqueness problem in inversion. The problem of nonuniqueness lies in the null space belonging to the model domain. The null space can be reduced by incorporating *a priori* constraints. For the inversion of earth elastic parameters from the AVO data, the *a priori* constraints can be either smooth or

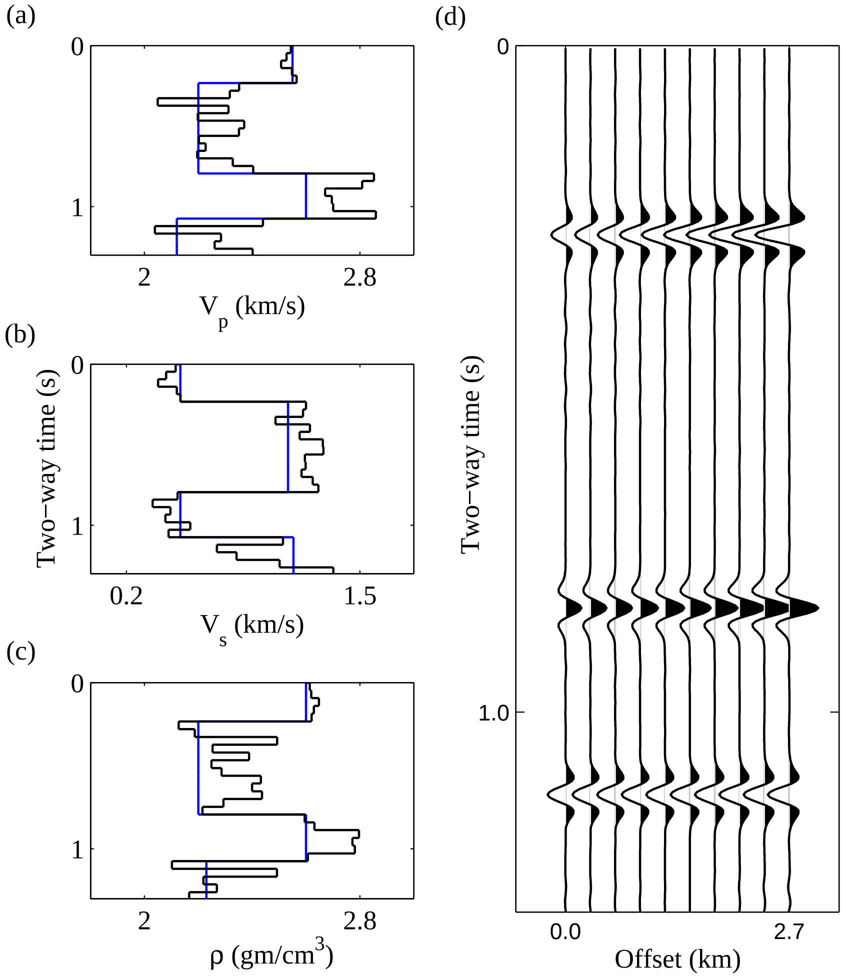

Figure 5.6: Estimated model profiles (black line) for (a) $\mathbf{V_p}$ (b) $\mathbf{V_s}$ and (c) $\boldsymbol{\rho}$ are shown superimposed on the true model profiles (blue lines). (d) Estimated data corresponding to the estimated model.

blocky model solutions. Since, a reflection in the data domain signifies a contrast in the elastic behavior of the subsurface layer, it makes better sense to assume that the model profile be blocky, thus suggesting sparse reflectivity. I include the blocky *a priori* constraint to the cost function given by the equation 5.24 and optimize for the estimated model using NLCG optimizer scheme as discussed above. As discussed in chapter 4, a function becomes blocky or piecewise constant if the first order derivative of the function is sparse. I am applying the Cauchy prior distribution function on the first order derivative of the model to impose blockyness in the estimated

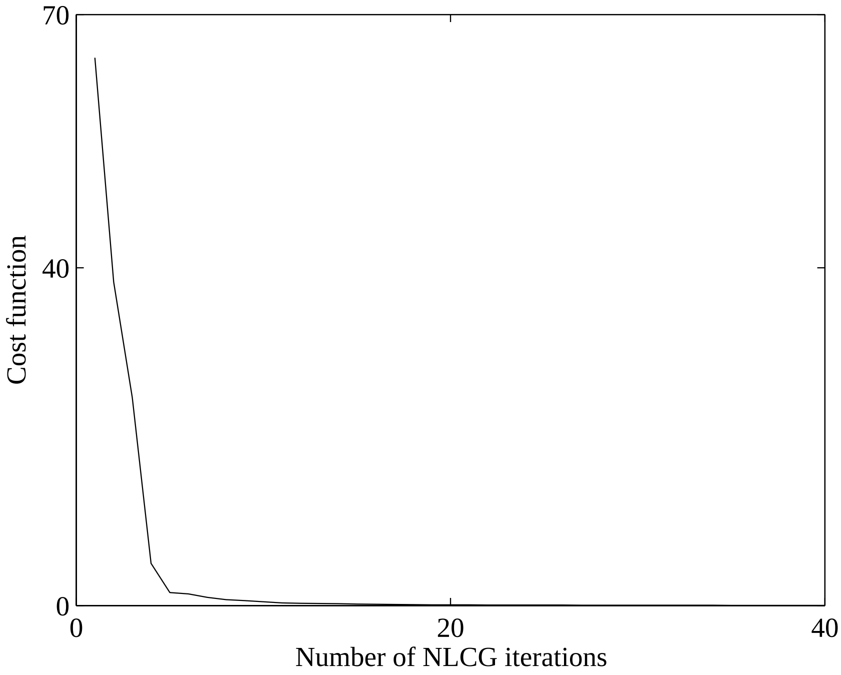

Figure 5.7: The convergence curve for the NLCG optimization. The cost function contains only the data misfit term.

model. The cost function is shown in the equation below.

$$J = \sum_{i=1}^{Noff} \sum_{j=1}^{N_t} |d(x_i, t_j)^{obs} - d(x_i, t_j)^{est}|^2 + \mu R(\mathbf{Dm}), \tag{5.25}$$

where the edge preserving Cauchy regularization function is given by R. As mentioned earlier, a model is piecewise constant when the first order derivative of the model is sparse. Thus, by applying Cauchy prior distribution function (it is discussed in chapter 4 that a Cauchy prior distribution function imposes sparseness on the solution) on the scaled first order derivative of the model vector, blockyness constraint is imposed on the estimated model. The Cauchy edge preserving regularization function is given as below.

$$R(\mathbf{Dm}) = \sum_i \ln(1 + (\frac{Dm_i}{\delta})^2), \tag{5.26}$$

where $Dm_i = (m_{i+1} - m_i)$ and δ is a scaling factor applied to the first order derivative operator. Thus, the edge preserving Cauchy regularization function has two hyperparameters namely, μ and δ. Details about the edge preserving regularization function is described in chapter 4.

I started the NLCG optimization for both cases (with and without the blocky constraint in the cost function) from the same initial model. Figure 5.8 shows the true and estimated model for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$. The blue line shows the true model and the black line shows the estimated model. There are 40 forward model evaluations. The CPU time consumed to perform the 40 NLCG iterations was equal to 0.6770 hours in a Pentium 4, 3.2 GHz machine. The estimated data corresponding to the estimated model is shown alongside. It is observed that the incorporation of the Cauchy edge preserving regularization term in the cost function has guided the NLCG optimization to a blocky solution that is consistent with the *a priori* information and also honors the data. Figure 5.9 shows the convergence curve for the NLCG optimization when the cost function contains both the data misfit and the Cauchy edge preserving regularization function.

Nonlinearity in the problem is greatly reduced when the data are NMO corrected and contain no multiples. Thus, the cost function is nearly quadratic. This argument is well substantiated by the fact that the NLCG optimizer could effectively estimate the model that honors the data and satisfy the *a priori* information. In the subsequent discussion, I will enhance the degree of nonlinearity in the problem by including multiples in non-NMO corrected data. In the examples shown below, I have chosen to use the reflectivity method (Fuchs and Muller, 1971) as the forward model to generate synthetic data from a given model. Figure 5.10 shows the true model and true data generated with the reflectivity method. Data contain no other noise but multiples.

I have used the same upper and lower bounds as the previous example to choose the initial model for the NLCG optimization. Figure 5.11 shows the initial model (black line) superimposed on the true model (blue line) along with the upper and lower bounds (red line). The data generated with the reflectivity method using the initial model is shown alongside.

Figure 5.12 shows the estimated model after 15 NLCG iterations. The total CPU time to perform 15 NLCG iterations with 90 unknown model parameters is 29 hours on a 3.2 GHz (Pentium 4) machine. Figure 5.13 shows the progress of NLCG optimization at each iteration.

It is evident from the Figures 5.12 and 5.13 that when the degree of nonlinearity in the problem is large, the cost function exhibits highly undulating topology.

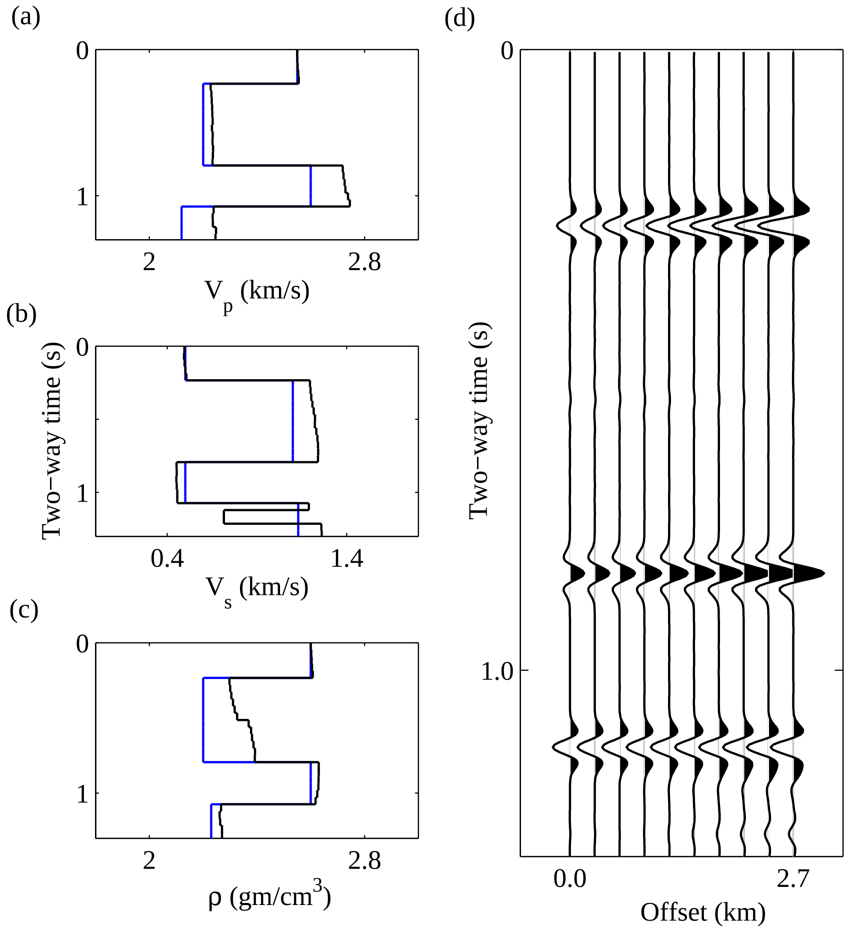

Figure 5.8: Estimated model profiles (black line) for (a) $\mathbf{V_p}$ (b) $\mathbf{V_s}$ and (c) $\boldsymbol{\rho}$ are shown superimposed on the true model profiles (blue lines). (d) Estimated data corresponding to the estimated model.

In such situations the NLCG optimization converged to the nearest local minimum corresponding to the initial model. In the example shown, the nearest local minimum is too far from the global minimum. Hence, the solution obtained with NLCG optimizer is erroneous. In the next chapter, I am solving the highly nonlinear problem with global optimization scheme. The *a priori* information that the solution be blocky is included in the second stage of the global optimization scheme. A new hybrid global optimization algorithm is presented in the next chapter.

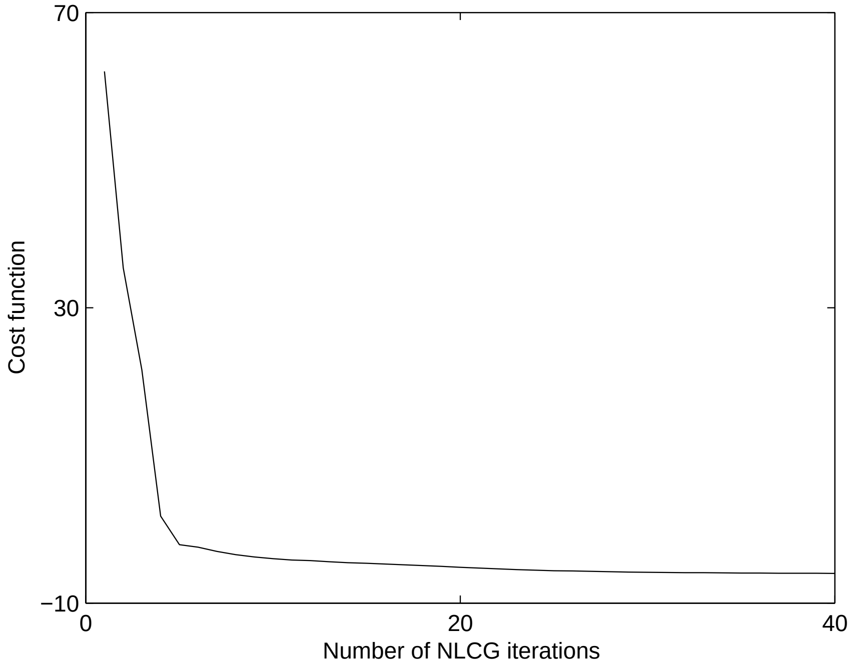

Figure 5.9: The convergence curve for the NLCG optimization. The cost function contains both the data misfit and the Cauchy edge preserving regularization function.

5.4 Summary

In this chapter, I discussed estimation of the earth elastic parameters from amplitude variation with offset data. In the first part, I approached the problem with ray-tracing based forward operator. The simulated NMO corrected data contained no multiples. Hence the inverse problem is weakly nonlinear. I estimated the model parameters using nonlinear conjugate gradient optimizer. The use of Cauchy regularization term on the first order derivative of the model vector resulted in blocky solution that not only honored the data but also consistent with the *a priori* information. In the second part, I increased the degree of nonlinearity in the cost function by replacing the ray-tracing based forward operator with the reflectivity method. The cost function showed a greater degree of nonlinearity as the non-NMO corrected data contained multiples. I showed that the nonlinear conjugate gradient algorithm failed to recover the model as the convergence got stuck in the local minimum close to the initial model. Estimation of model parameters in such situation requires the global optimization techniques. In the next chapter, I show that the model param-

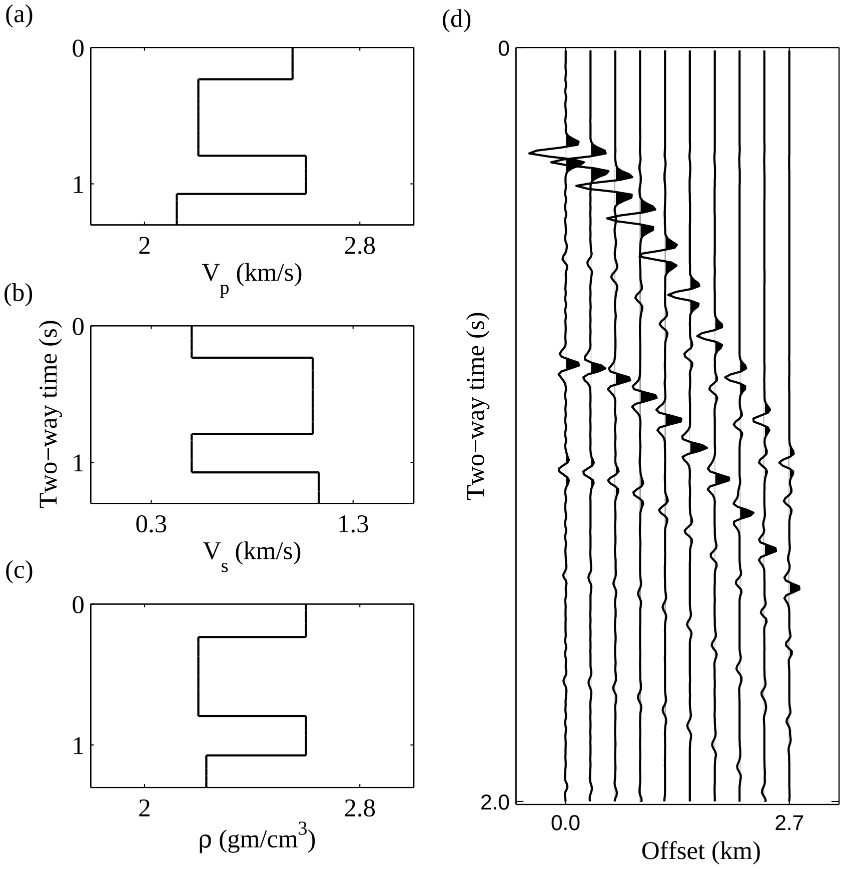

Figure 5.10: The true model for (a) $\mathbf{V_p}$, (b) $\mathbf{V_s}$, (c) $\boldsymbol{\rho}$ considered for the NLCG optimization with the reflectivity forward model. (d) Synthetic data generated with the reflectivity forward model. Data contain multiples and no other noise.

eters are effectively estimated in highly complex and nonlinear problems by using global optimization schemes.

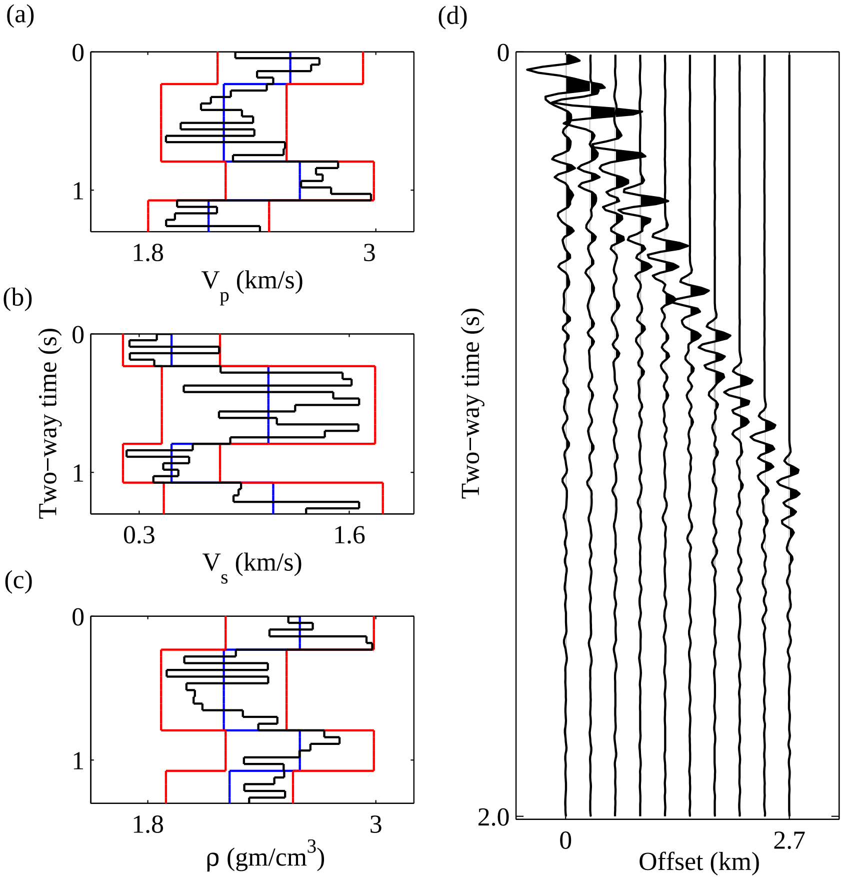

Figure 5.11: The initial model (a) $\mathbf{V_p}$, (b) $\mathbf{V_s}$, (c) $\boldsymbol{\rho}$ and (d) data corresponding to the initial model. The true model is shown in blue line. The upper and lower bounds are shown in red lines and the initial model is shown in black line.

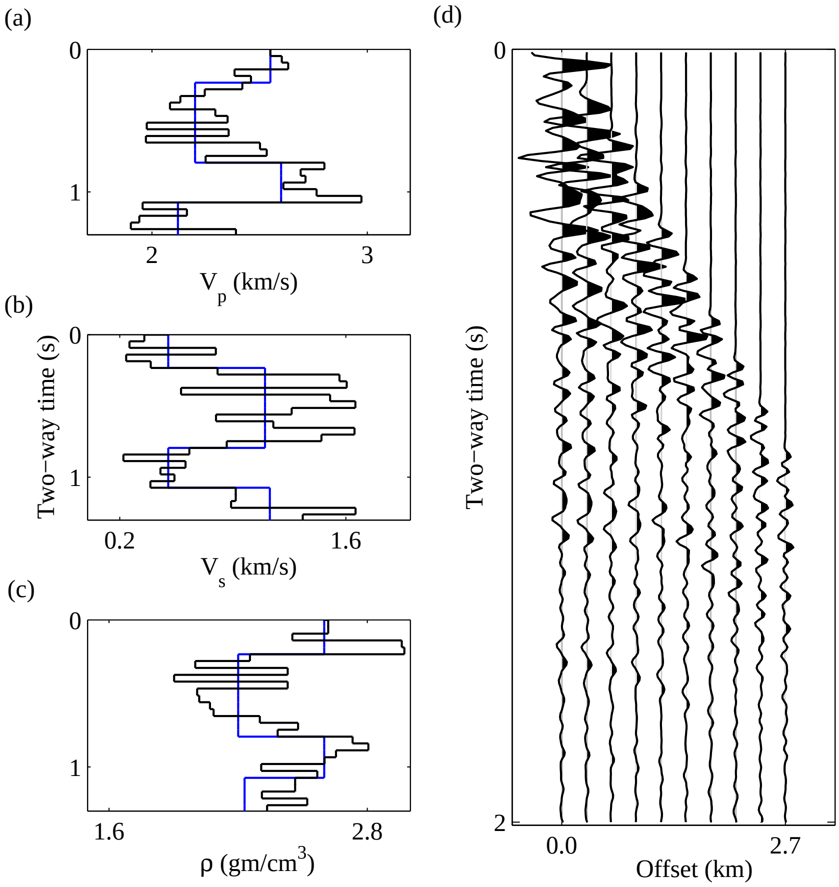

Figure 5.12: The estimated model (a) $\mathbf{V_p}$, (b) $\mathbf{V_s}$, (c) $\boldsymbol{\rho}$ and (d) data corresponding to the estimated model. The true model is shown in blue line and the estimated model in black line.

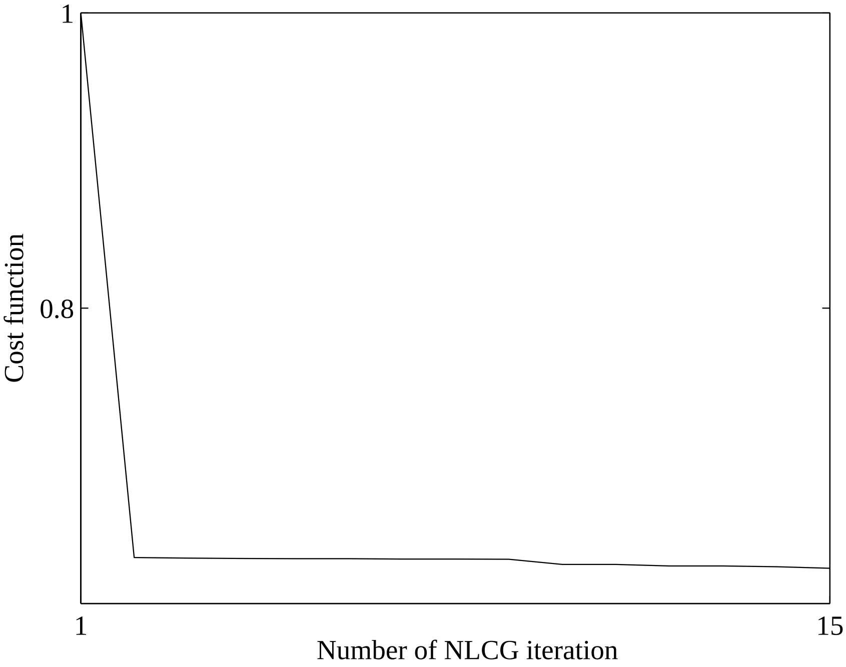

Figure 5.13: The convergence curve for the NLCG optimization. The cost function contains only the data misfit term.

Chapter 6

Global Optimization: Application to AVO Inversion[1]

6.1 Introduction

In geophysical inverse problems, generally data and model are nonlinearly related. A cost function that measures misfit between observed data and predicted data is minimized to estimate the unknown model parameters. Since the operator relating the model and data is often complex and nonlinear, the cost function is likely to have more than one local minimum. Moreover, local minima are also caused by the nonuniqueness of the model and incompleteness of data. In such cases, the linearized inversion approach usually fails to reach the global minimum and hence yields incorrect solutions. In absence of sufficient *a priori* information, better results are obtained through global optimization techniques. Monte Carlo optimization is a global optimization method that generates and accepts models that provide a good match in the data space while satisfying the imposed *a priori* constraints. Press (1968) applied Monte Carlo based inversion to estimate the subsurface elastic parameters from seismological observations and physical parameters such as the mass and moment of inertia of the Earth. Wiggins (1969) applied the Monte-Carlo technique to estimate the upper mantle P-wave velocity from travel time and epicentral distances of the earthquakes. Jin and Madariaga (1994) used the Monte Carlo method to estimate the high frequency velocity variations for the background velocity distribution. The main criticism associated with the Monte Carlo optimization

[1] Misra, S. and Sacchi, M. D., 2008, Global optimization with model space preconditioning: Application to AVO inversion, Geophysics, **73**, R71-R82.

technique is that the method is computationally very expensive. This is because the algorithm has to apply the forward modeling operator at every random generation of the model parameters. In order to overcome this problem, various other stochastic optimization strategies such as the genetic algorithm and the simulated annealing have been proposed. Genetic algorithms mimic the process of evolution with the hope that the models obtained in later generations are better candidates compared to the previous generations (Holland, 1975). Simulated annealing simulates the process of crystallization in a solid. Simulated annealing, with many of its variants such as the Metropolis sampling (Metropolis et al., 1953; Kirkpatrick et al., 1983), fast simulated annealing (Szu and Hartley, 1987), very fast simulated annealing (Ingber, 1989) have been effectively used in various optimization applications. Simulated annealing has been extensively applied in geophysical inversion problems (Sen and Stoffa, 1991; Mosegaard and Vestergaard, 1991; Vestergaard and Mosegaard, 1991; Velis and Ulrych, 1996; Ma, 2001; Ma, 2002; Misra and Sacchi, 2007a). Recently, Varela et al. (2006) applied the simulated annealing optimization scheme to estimate the one-dimensional subsurface layer parameters from prestack seismic data while incorporating smoothness constraints on the cost function.

I present a hybrid optimization technique with very fast simulated annealing. The optimization technique modifies the classical very fast simulated annealing technique at two stages, namely, (a) the generation of model parameters and (b) the acceptance-rejection criteria. The optimization scheme is complimented with a model preconditioner that favors blocky models. I use an edge preserving smoothing (EPS) operator (AlBinHassan et al., 2006) to precondition the model space. The algorithm is applied to the inversion of P-wave velocity, S-wave velocity and density from prestack seismic data showing amplitude variation with offset. I chose such an application because the *a priori* information suggests that the estimated model be blocky.

The chapter is broadly divided into two major sections. In the first section, I discuss the methodology involved in the optimization scheme. In the second section, I discuss the application of the algorithm to the inversion for earth elastic parameters from prestack amplitude variation with offset data. As a test of the feasibility of the algorithm in real data situations, the algorithm is applied to well log simulated data. It is observed that, within a reasonable search bound, the algorithm successfully estimates the well log data from the simulated seismic section.

6.2 Global optimization with SA and VFSA

In this section, I recapitulate the global optimization techniques, namely, the simulated annealing (SA) and the very fast simulated annealing (VFSA) optimization algorithms that I am using in the optimization of earth elastic parameters. Readers familiar with SA and VFSA may skip this section and go directly to the section entitled " VFSA aided with the EPS operators".

Simulated annealing is an optimization procedure that simulates the process of thermodynamic annealing of solids. As the temperature of a solid is lowered, the chaotic motions of the molecules gradually become more organized and with a very slow cooling rate a crystal is formed at a very low temperature. This state corresponds to the global minimum energy state of the solid. In simulated annealing based optimization method, an initial model is generated within a pre-defined upper and lower bound. A forward operator is applied on the model and predicted data are compared with the observed data on the basis of a pre-defined cost function. The initial model is perturbed and a new model is generated. The new model is unconditionally accepted if the cost function with respect to the previous model is less. If the new model results in higher cost function then the new model is still accepted with a certain probability as mentioned in the following equations

$$\Delta E = E_n - E_{n-1}. \tag{6.1}$$

If $\Delta E \leq 0$, then accept. Otherwise, $p = e^{\frac{\Delta E}{T}}$ and $r = U[0,1]$. If $p \geq r$, then accept. Where E_n is the cost function for the current model, E_{n-1} is the cost function for the previous model, T is a parameter (referred to as the acceptance/cost temperature) that controls the rate of acceptance or rejection of the generated models, r is a random number generated from the uniform distribution between 0 and 1. When ΔE is positive, the model is accepted with the probability p . This acceptance/rejection condition is known as the Metropolis criterion.

The method of simulated annealing allows the bad moves to be accepted with a certain finite probability. Thus, there exists a finite probability to search in the uphill direction as well as in the downhill direction, helping the algorithm to "jump out" of a local minimum. This is in contrast to the local optimization schemes where the search direction is always downhill leaving no scope for the algorithm to escape from a local minimum.

The rate at which the acceptance temperature is lowered, often referred to as the cooling schedule, plays an important role in the optimization process. A faster cooling schedule is desired so as to save computational time. However, there exists a trade-off between the rate of cooling and attainment of the global minimum. Geeman and Geeman (1984) showed that the attainment of global minimum can be statistically guaranteed with a cooling rate no faster than the following schedule.

$$T_k = \frac{T_0}{\ln(1+k)}, \tag{6.2}$$

where T_0 is the temperature at the starting point and T_k is the temperature at the iteration k.

Szu and Hartley (1987) proposed a faster cooling schedule with a model generating distribution that followed Cauchy distribution. The distribution function is as follows.

$$F_c(\Delta m_i) = \frac{T}{(\Delta m_i^2 + T^2)^{\frac{1}{2}}}, \tag{6.3}$$

where F_c is the distribution function, T is the temperature and Δm_i is the perturbation in the i^{th} model parameter. The cooling schedule is given by

$$T_k = \frac{T_0}{k}, \tag{6.4}$$

where T_k is the acceptance temperature at the iteration k and T_0 is the starting temperature.

Ingber (1989) proposed a new variant of simulated annealing that allowed for a cooling schedule faster than the ones already mentioned. The very fast simulated annealing (VFSA) algorithm generates new model parameter m_i^{k+1} for the model dimension i and the iteration $(k+1)$ from the corresponding model parameter m_i^k at the previous iteration. The model is updated by the following equation.

$$m_i^{k+1} = m_i^k + y_i[max(m_i) - min(m_i)], \tag{6.5}$$

where $max(m_i)$ and $min(m_i)$ are the pre-defined upper and lower bounds of the model parameter m_i respectively and $y_i \in [-1, 1]$ generated from a distribution function given by

$$y_i = sgn(r_i - \frac{1}{2})T_i[(1 + \frac{1}{T_i})^{|2r_i - 1|} - 1], \tag{6.6}$$

where $r_i = U[0,1]$. When the generation of model parameter follows such a distribution, the global minimum is statistically obtained with the following model parameter cooling schedule

$$T_i^k = T_i^0 e^{-c_i k^{\frac{1}{M}}}, \tag{6.7}$$

where T_i^0 and T_i^k are the temperatures for the i^{th} dimension and the initial and the k^{th} iteration respectively, c_i is a user defined constant that tunes the cooling schedule for different applications and M is the dimension of the model space.

The model parameter temperature allows for different search regimes along different model dimensions. Different model parameters might influence the cost function in different ways. The cost function might be more sensitive to certain parameters and less sensitive to certain other parameters in the model space. Very fast simulated annealing algorithm allows different temperature cooling schedules for model parameters. The cooling schedule can be modified by estimating the sensitivity of the cost function to different model parameters (Ingber, 1989). Sensitivities of each model dimension are tested by calculating the first derivative of the cost function with respect to the model parameters.

When the model dimension is large, it may be useful to accelerate the cooling schedule to save the computational time. This can be achieved by introducing the quenching parameter to the cooling equation.

$$T_i^k = T_i^0 e^{-c_i k^{\frac{Q}{M}}}, \tag{6.8}$$

where Q is the quenching parameter and $1 \leq Q \leq M$.

6.3 VFSA aided with the EPS operators

The purpose of edge preserving smoothing (EPS) operator is to suppress random fluctuations in the model space while preserving the edges. In the optimization algorithm, I follow the *a priori* knowledge that the desired model profile is blocky. I apply the EPS operators as the model space preconditioners so that I incorporate the blocky *a priori* knowledge about the model domain in the optimization scheme. The use of EPS operators in the global optimization routine helps the optimization algorithm in two ways, namely, (a) provides a better convergence and (b) reduces the nonuniqueness in the inversion problem by suitably imposing blocky features in the model space. By applying the EPS operators on the randomly generated model vectors, rather than relying completely on the random perturbations, I impose blocky features on the model space.

The EPS operators have been utilized successfully in reducing noise in the seismic-impedance cubes while preserving the structural and stratigraphic discontinuities (AlBinHassan et al., 2006). The design of EPS operators is simple. I follow the approach of AlBinHassan et al. (2006) to design the EPS operators. Figure 6.1a shows the design of an n-point EPS operator. The figure shows a vector of M parameters. I have chosen n numbers of n-point windows starting from the first element of the model vector. The standard deviation for each window belonging to this array is calculated. The window with least standard deviation is the most homogeneous window among all the windows in the array. The most homogeneous window in the entire array is identified and the mean value of this window is calculated. The middle point of the central window in the array is replaced by the mean of the most homogeneous window to generate the EPS filtered output for that array of windows. The array of windows is run over the entire length of the model vector and the procedure is followed at each location of the array to obtain the EPS filtered output. Figure 6.1b shows the position of the central window in the array of windows for a 5-point EPS operator.

Figure 6.2 compares the actions of a moving average smoothing filter with that of an EPS filter on a model vector containing a sharp edge that has been masked by the embedded noise. Figure 6.2a shows a model vector with a sharp edge. Figure 6.2b shows the same model vector as in Figure 6.2a but embedded with random noise. The effect of noise has masked the sharp edge. Figure 6.2c shows the action of a moving average smoothing filter on the model vector shown in Figure 6.2b. It is observed that the moving average smoothing filter effectively removed the high frequency noise components from the noisy model vector. However, it failed to restore the sharp edge. Figure 6.2d shows the action of EPS filter on the model vector shown in Figure 6.2b. It is noticed that the EPS filter effectively removed the high frequency noisy components and at the same time restored the sharp edge.

Application of edge preserving smoothing (EPS) operators in the very fast simulated annealing based optimization scheme is a two step process. In the first step, the optimization scheme is allowed to follow the classical procedure of model perturbation and the Metropolis acceptance and rejection criterion. The following cost function is evaluated in the first step

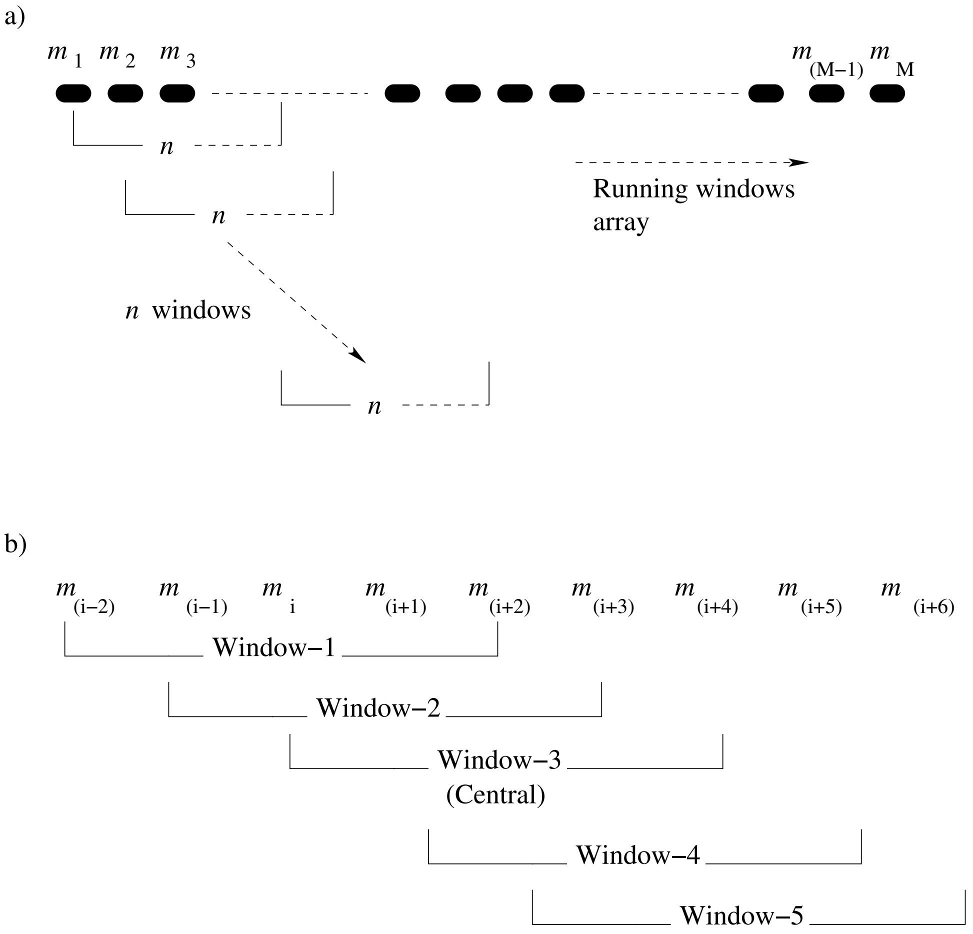

Figure 6.1: The EPS filter. (a) Running windows showing the design of an EPS filter. (b) Location of the central window for the first running window for a 5-point EPS filter.

$$J_1(\mathbf{m}) = \|\mathbf{d}_{obs} - \mathbf{g}(\mathbf{m})\|^2, \tag{6.9}$$

where $J_1(\mathbf{m})$ is the value of the cost function obtained for the model $\mathbf{m}$, $\mathbf{d}_{obs}$ represents observed data and $\mathbf{g}$ is the operator that nonlinearly relates data with the model.

In the second step, the EPS operators are applied on the output model of the classical very fast simulated annealing algorithm. The forward operator is applied to the EPS filtered model and the cost function is evaluated. The following cost function is evaluated in the second step.

$$J_2(EPS(\mathbf{m})) = \|\mathbf{d}_{obs} - \mathbf{g}(EPS(\mathbf{m}))\|^2, \tag{6.10}$$

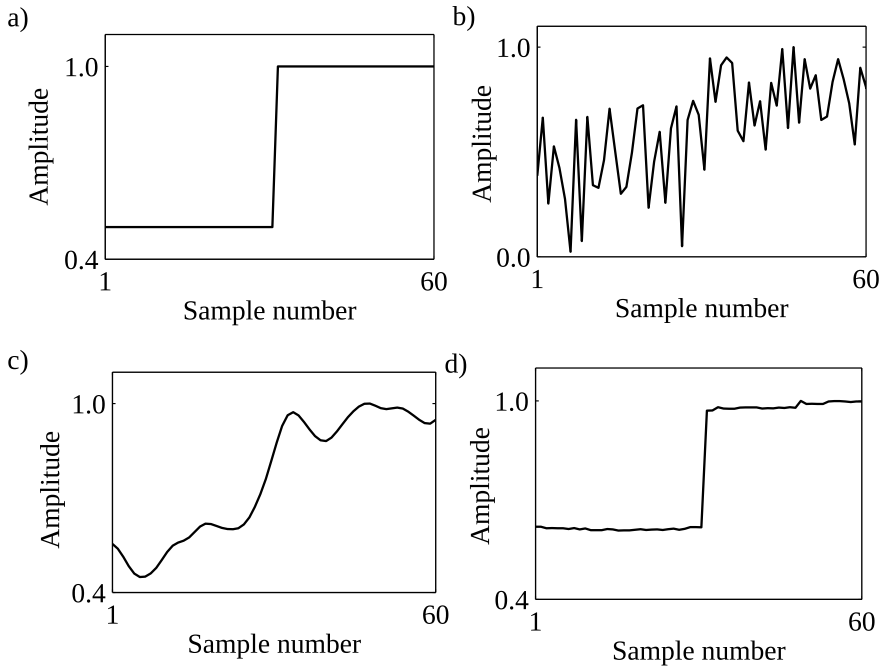

Figure 6.2: Application of edge preserving smoothing (EPS) filter to a model vector. (a) The model vector with a sharp edge. (b) The model vector as (a) with additive random noise. The sharp edge is masked by the random noise. (c) The effect of the moving average filter on the model vector with noise. The smoothing filter has effectively removed the high frequency fluctuations due to the random noise. However, it failed to unmask the sharp edge. (d) The effect of the edge preserving smoothing filter on the model vector with noise. The EPS filter has removed the high frequency oscillations effectively. The sharp edge is well recovered by the EPS filter.

where EPS is the EPS operator acting on the model.

The EPS filtered model is accepted or rejected depending on whether or not there is a lowering of the cost function. If $J_2(EPS(\mathbf{m})) \leq J_1(\mathbf{m})$, the EPS filtered model is accepted and the subsequent model perturbation and Metropolis tests are performed on this accepted model. If the EPS filtered model is not accepted then the subsequent model perturbations are performed on the model obtained from the classical VFSA algorithm. The purpose of incorporating the EPS operators in the second stage of the algorithm is to avoid unduly biasing the optimization routine that favors blocky solutions in places where no sharp physical contrasts exist. Figure 6.3 shows the flowchart of the algorithm.

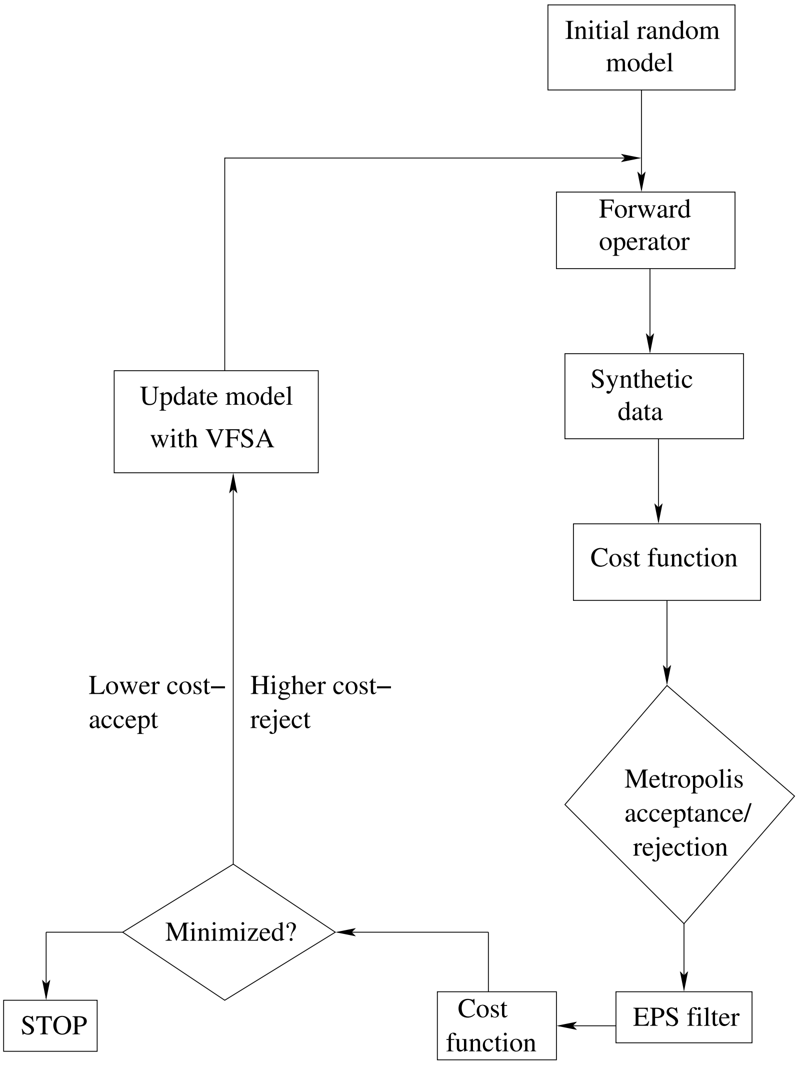

Figure 6.3: The flowchart for the model preconditioning based very fast simulated annealing algorithm.

6.4 Application to AVO inversion

Simultaneous estimation of the elastic parameters of the underlying earth layers are in the forefront of active research because of their importance in oil and gas exploration and reservoir characterization. The techniques to estimate the elastic parameters are broadly classified into two categories (a) the linearized inversion (Smith and Gidlow, 1987; Lörtzer and Berkhout, 1993; Buland and Omre, 2003) techniques and (b) the nonlinear inversion techniques (Mallick, 1995, Varela et al., 2006). The problems with linearized inversion are that they are local minimization processes and dependent on the location of the initial model (Tarantola, 1987). In contrast, the global optimization techniques are independent of the initial model and have the ability to "jump out" of a local minimum and converge towards a global minimum. The inversion of elastic parameters from amplitude variation with offset (AVO) data is nonunique (Mallick, 1995). Furthermore, the model parameters are nonlinearly related to the data which suggest that the cost function topology may be multimodal. These conditions make the problem suitable for a global optimization approach.

Varela et al. (2006) used the simulated annealing based optimization scheme to simultaneously estimate the earth elastic parameters from the prestack amplitude data. They augmented the cost function with the smoothness constraints to enforce smooth solution in the global optimization scheme. I am introducing a model preconditioning based global optimization scheme where flatness and edge preservation in the model domain are simultaneously enforced by means of nonlinear edge preserving smoothing operators. The method is applied to simultaneously estimate the P-wave velocities, the S-wave velocities and the bulk densities for the subsurface layers.

6.4.1 Forward operator

The inversion method requires a forward operator to generate synthetic data. I used the reflectivity method based forward operator (Fuchs and Muller, 1971) in all the examples shown in this chapter except for the section "Assessment of uncertainty" where the ray-tracing approach based forward operator is used. The reflectivity method based forward operator generates synthetic data by computing the full-wave response in the time-offset domain for a stack of homogeneous and isotropic horizontal earth layers. In the ray-tracing approach based forward operator, the amplitudes of prestack seismic data in an angle gather are calculated from the Aki-Richards approximation of Zoeppritz equations. The ray-tracing approach involves calculation of source-receiver travel path from the Snell's law (Shearer, 1999) and positioning the reflectors at the corresponding locations in the time-offset axes. The resulting reflectivity is convolved with a known Ricker wavelet to simulate NMO corrected seismic data. Both, the ray-tracing and reflectivity method based forward operators have inherent nonlinearity (Misra and Sacchi, 2007b).

I used the reflectivity method based forward operator to generate synthetic data.

The synthetic data contain multiples. In order to test the consistency of the algorithm, I performed 100 numbers of Monte Carlo simulations. Since, the reflectivity method based forward operator is computationally expensive, it is impracticable to perform the simulations with the reflectivity method. Thus, I used the ray-tracing approach based forward operator to generate the synthetic data and perform the Monte Carlo simulations to test the consistency of the algorithm.

6.4.2 Model space

I have followed the over-parameterization approach (Sen and Stoffa, 1991; Varela et al., 2006) to parameterize the model space. As a simple example, I represent a 3-layers earth model in terms of a 30- microlayers earth model (29 microlayers over a half-space). Each microlayer has a constant two-way travel time. The model space in the inversion algorithm consists of the P-wave velocities ($\mathbf{V_p}$), the S-wave velocities ($\mathbf{V_s}$) and the bulk density ($\boldsymbol{\rho}$) defined in each of the microlayers of the over-parameterization scheme. The following matrix notation is used to represent the model space

$$\mathbf{V_p} = \begin{pmatrix} v_{p_1} \\ v_{p_2} \\ v_{p3} \\ \vdots \\ v_{p_n} \end{pmatrix}, \quad \mathbf{V_s} = \begin{pmatrix} v_{s_1} \\ v_{s_2} \\ v_{s3} \\ \vdots \\ v_{s_n} \end{pmatrix}, \quad \boldsymbol{\rho} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_n \end{pmatrix}, \qquad (6.11)$$

where n represents the number of pre-defined microlayers in the model space. The search space is $\pm 15\%$ of the true model values for the P-wave velocity and the density parameters. The search space for the S-wave velocity is $\pm 60\%$ of the true values. The search bounds are selected in such a way that the search space provides overlapping of layers at the edges. Figure 6.4 shows the true model and the upper and lower bounds of the model parameters $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$. The black line shows the true profile and the red line shows the upper and lower bounds of the model space.

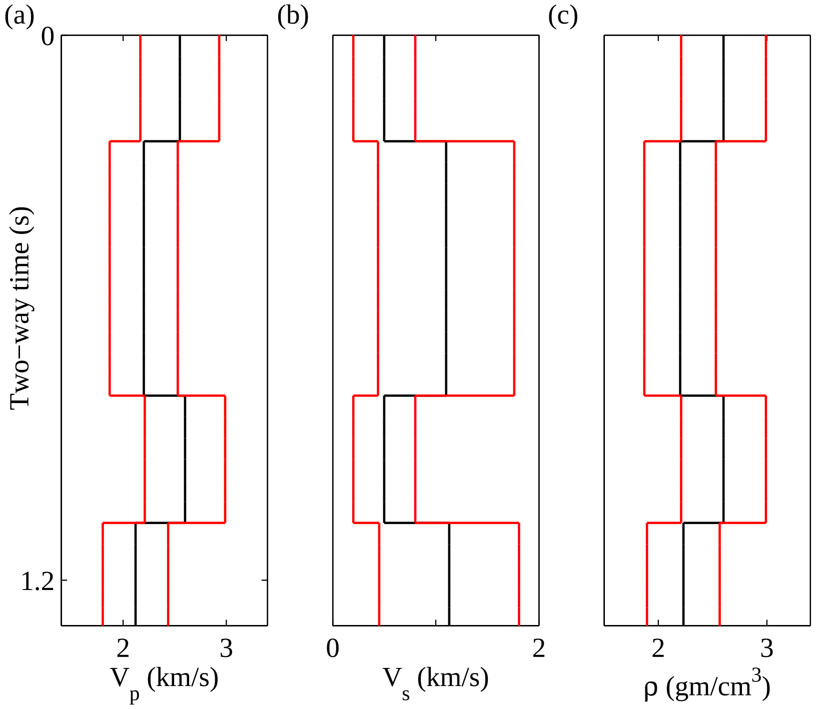

Figure 6.4: The upper and lower bounds of the model space. (a) The bounds for $\mathbf{V_p}$. (b) The bounds for $\mathbf{V_s}$. (c) The bounds for $\boldsymbol{\rho}$. The black line shows the true model value and the red line shows the upper and lower bounds. The bounds are ±15% for $\mathbf{V_p}$ and $\boldsymbol{\rho}$ and ±60% for $\mathbf{V_s}$ with respect to the true model.

6.4.3 Model preconditioning-aided VFSA scheme

The following scheme is used for the very fast simulated annealing optimization for the estimation of the earth layer elastic parameters from prestack amplitude data. The initial cost temperature and the initial model parameter temperature are set to relatively high values. The initial model parameter temperature is kept same for all model dimensions. The cooling schedule for the model parameters is given by the equation 6.7. The parameter c_i is chosen in such a way that the model parameter temperature is sufficiently low after a large number of iterations. The value of c_i is determined by the following equation

$$c_i = R^{-\frac{1}{M}} log(\frac{T_i^0}{T_i^f}), \tag{6.12}$$

where T_i^f is final model parameter temperature, T_i^0 is the initial model parameter temperature, R is the total number of iterations and M is the dimension of the model space.

The cost temperature cooling schedule is given by (Ryden and Park, 2006)

$$T = T^0 a^j, \tag{6.13}$$

where T is the cost temperature after j numbers of accepted transitions. The term a is a positive constant less than 1.

The cost function to be minimized is given by an L_2-norm between the observed and the predicted data.

$$J(m) = \sum_{i=1}^{N_{off}} \sum_{j=1}^{N_t} \left| d\left(x_i, t_j\right)^{obs} - d\left(x_i, t_j\right)^{est} \right|^2 . \tag{6.14}$$

Where $d(x_i, t_j)^{obs}$ and $d(x_i, t_j)^{est}$ are the observed and estimated data respectively for the offset x_i and time sample t_j . The total number of traces is given by N_{off} and the total number of time samples is given by N_t. The data are estimated at each VFSA iteration given by the following equations

$$d(x_i, t_j)_1^{est} = \mathbf{g}(\mathbf{m}), \tag{6.15}$$

$$d(x_i, t_j)_2^{est} = \mathbf{g}(EPS(\mathbf{m})), \tag{6.16}$$

where EPS is the model preconditioning operator and $\mathbf{g}$ is the forward operator. The vector $\mathbf{m}$ contains the unknown model parameters $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$.

6.4.4 Synthetic data example

Synthetic data are generated by using the reflectivity method based forward operator. The data contain multiples. Figure 6.5a, b and c show the true model profiles for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. Figure 6.5d shows the true synthetic data corresponding to the true model profiles shown alongside. The model space is defined over 30 microlayers of constant two-way time. The data space contains 10 traces. This is a very simple example where our *a priori* information is that the model profiles are blocky. Figure 6.6a, b and c show the initial random model for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. Figure 6.6d shows the estimated data corresponding to the initial model. It is clear from Figure 6.6d that the initial random model is not close to the global minimum of the cost function.

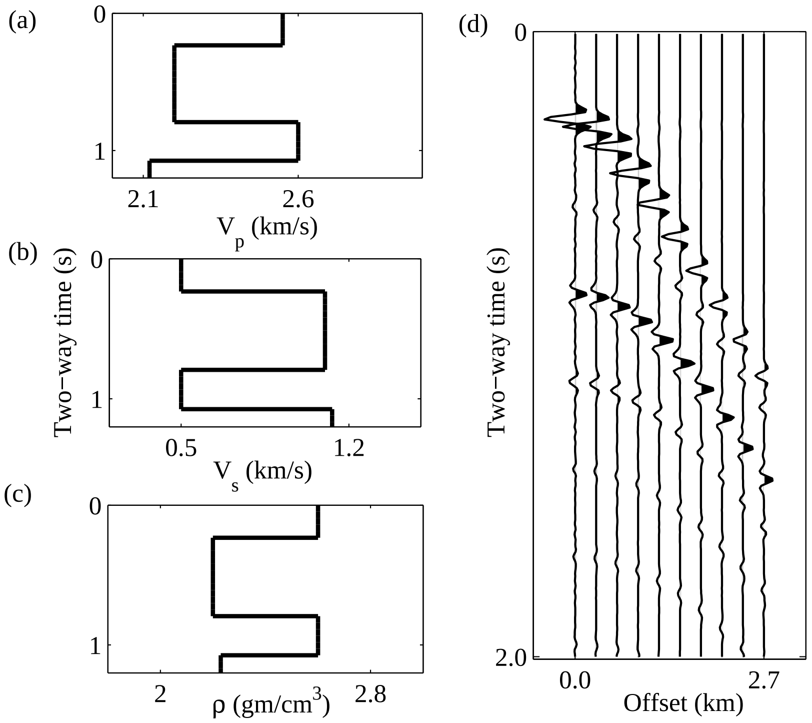

Figure 6.5: The true model and the true synthetic data. (a) The true model for $\mathbf{V_p}$. (b) The true model for $\mathbf{V_s}$. (c) The true model for $\boldsymbol{\rho}$. (d) Synthetic data corresponding to the true model.

Figure 6.7a, b and c show the estimated model for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. Figure 6.7d shows the estimated data corresponding to the estimated model. The optimization scheme incorporates the edge preserving smoothing operators to precondition the model space for blocky solutions. Three point edge preserving smoothing operator window is selected for the three model vectors $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$. The edge preserving smoothing operators were allowed to pass through each of the model vectors only once during each of the very fast simulated annealing iteration. The degree of smoothing is dependent on the length of the edge preserving smoothing operators and the number of times the operators are applied on the model vector. I chose to fix the length of the operator window to three and the number of passes to one. This ensures that the very fast simulated annealing algorithm is not operating in an overly preconditioned model space. The optimization process required a total of 6000 evaluations of the forward model. I observe that there is a good match in the data domain and the estimated model profiles are consistent with the *a priori* information.

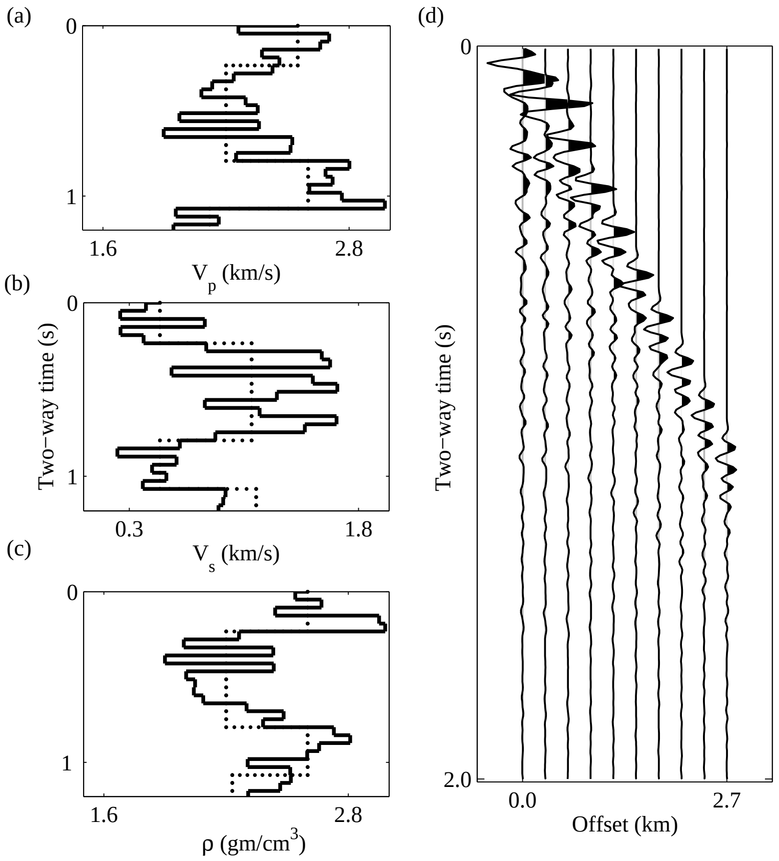

Figure 6.6: The random initial model and the synthetic data corresponding to the initial model. (a) The initial model for $\mathbf{V_p}$. (b) The initial model for $\mathbf{V_s}$. (c) The initial model for $\boldsymbol{\rho}$. The solid line shows the initial model and the dotted line shows the true model. (d) The synthetic data corresponding to the initial model. The figure shows that the starting model is not close to the global minimum of the cost function topology.

I compare the results obtained from the model domain preconditioning operators based global optimization algorithm with results obtained by (a) global optimization without model domain preconditioning and (b) global optimization with the cost function augmented with an edge preserving regularization function. For comparison purpose, I started the algorithms at the same initial model. The algorithms encompassed the same model space. The true model and data are shown in the Figure 6.5. Figure 6.8a, b and c show the final estimated model for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. The estimated model is obtained with very fast simulated annealing optimization without the incorporation of the model domain preconditioning operators. There are 10000 numbers of forward model evaluations. Figure 6.8d shows the

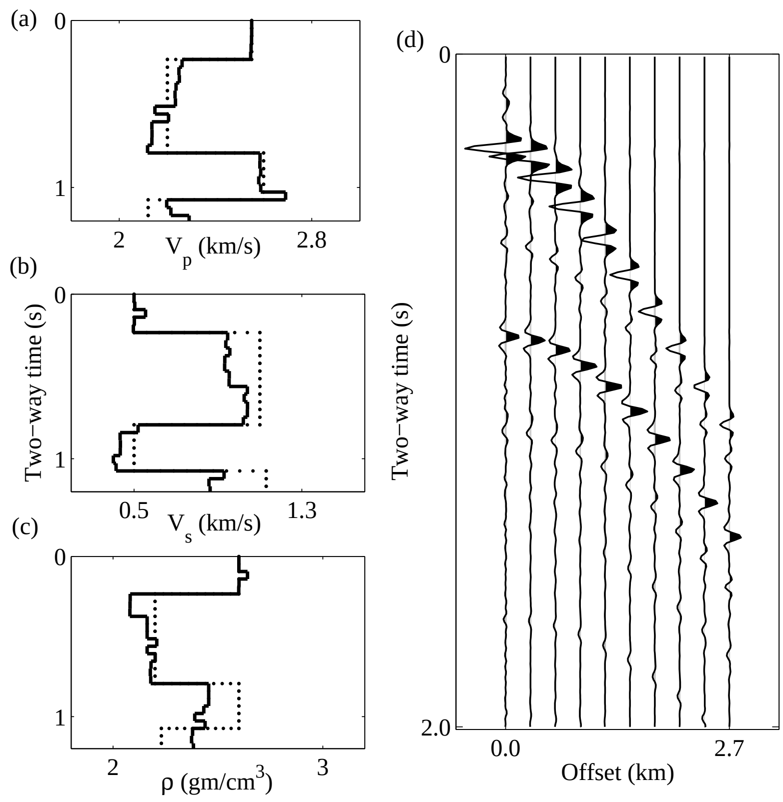

Figure 6.7: The final estimated model and the synthetic data corresponding to the final estimated model. (a) The final estimated model for $\mathbf{V_p}$. (b) The final estimated model for $\mathbf{V_s}$. (c) The final estimated model for $\boldsymbol{\rho}$. The solid line shows the final estimated model and the dotted line shows the true model. (d) The synthetic data corresponding to the final estimated model. The convergence is obtained in 6000 evaluations of the forward model. Model space preconditioning by the EPS filter successfully estimated the blocky profiles for the models.

final estimated data corresponding to the final estimated model. It is clear that the optimization algorithm without the model space preconditioning could not yield the desired blocky solution even though good convergence is achieved in the data domain. The result shows that the algorithm converged to a solution that fit the data well but does not honor the *a priori* information about the model. In contrast, the EPS operator based model space preconditioning effectively estimated the blocky profiles in 6000 evaluations of the forward model (Figure 6.7a, b and c).

Further, I compare the results obtained from the model space preconditioning based global optimization algorithm with the results obtained from a global optimization algorithm when the cost function is augmented with an edge preserving

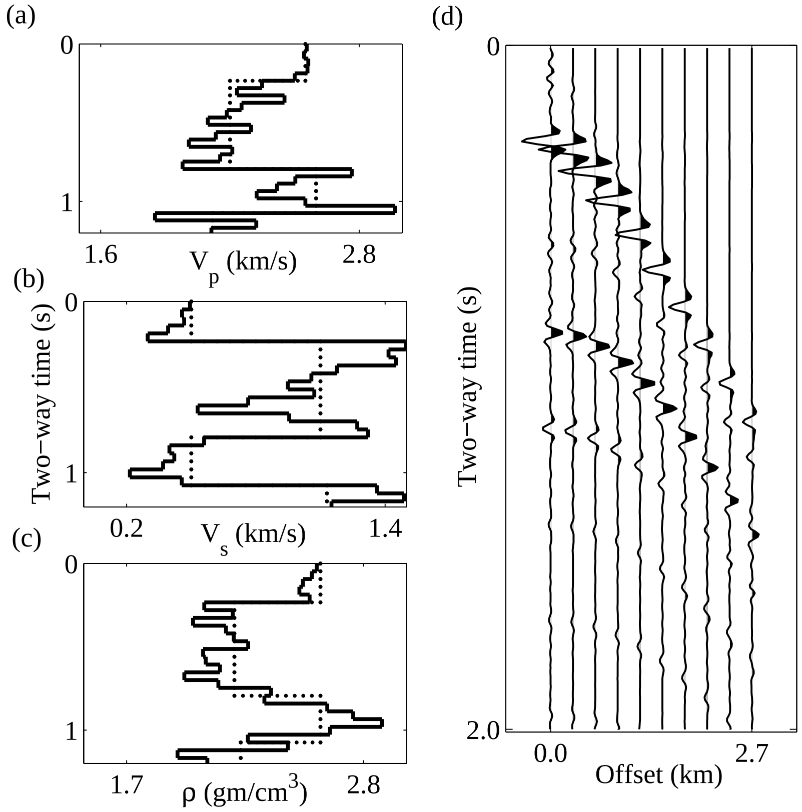

Figure 6.8: The final estimated model and the synthetic data corresponding to the final estimated model without the incorporation of model preconditioning operator. (a) The final estimated model for $\mathbf{V_p}$. (b) The final estimated model for $\mathbf{V_s}$. (c) The final estimated model for $\boldsymbol{\rho}$. The solid line shows the final estimated model and the dotted line shows the true model. (d) The synthetic data corresponding to the final estimated model. There are 10000 evaluations of the forward model. As noticed, the "blocky" *a priori* information is not properly honored by the optimization approach without model space preconditioning even though the algorithm yielded good data fit.

regularization function. The following cost function is minimized to compare the results obtained with the application of model space preconditioning operators.

$$J(\mathbf{m}) = \sum_{i=1}^{N_{off}} \sum_{j=1}^{N_t} \left| d\left(x_i, t_j\right)^{obs} - d\left(x_i, t_j\right)^{est} \right|^2 + \mu R\left(D\mathbf{m}\right), \tag{6.17}$$

where $d(x_i, t_j)^{obs}$ and $d(x_i, t_j)^{est}$ are the observed and estimated data respectively for the offset x_i and time sample t_j. The total number of traces and time samples are given by N_{off} and N_t respectively. The edge preserving regularization (EPR) function is given by R (Youzwishen, 2001, Valencino et al. 2004). The EPR function acts on the scaled first derivatives of the model parameters. The derivative operator is defined as

$$Dm_i = (m_{i+1} - m_i), \tag{6.18}$$

where the index i encompasses the total number of model parameters. The regularization term R is given by

$$R(D\mathbf{m}) = \sum_i \ln(1 + (\frac{Dm_i}{\delta})^2), \tag{6.19}$$

where δ is a scaling factor applied to the first order derivative operator. The edge preserving regularization function has two hyperparameters, namely, μ and δ . Figure 6.9a, b and c show the true and the estimated model. There are 10000 numbers of forward model evaluations. Figure 6.9d shows the synthetic data corresponding to the final estimated model. The hyperparameter δ was fixed at 0.1% of the maximum jump exhibited by the respective model vectors $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$. The other hyperparameter μ was calculated such that data misfit and edge preserving regularization function are properly weighted in the cost function. In the examples shown, the numerical values used for the hyperparameter δ are 0.0004, 0.00048 and 0.00063 for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. The numerical values for the hyperparameter μ were fixed at 10^{-11} for each of the unknown model vectors. The hyperparameters are selected on the basis of several trial runs. Comparison between the Figures 6.7 and 6.9 shows that the models are better estimated with the model space preconditioning approach.

Figure 6.10 shows the log-log plot of the error energy calculated during each cost function evaluation. The dotted line shows the cost function obtained during the very fast simulated annealing optimization when the cost function is augmented with the edge preserving regularization function as given by the equation 6.19. The thick dashed line shows the cost function obtained during the very fast simulated annealing optimization without the incorporation of the edge preserving function and the model preconditioning operators. The cost functions are obtained with 10000 evaluations of the forward model. The solid line shows the cost function obtained by the model preconditioning based very fast simulated annealing optimization. The cost function is obtained with 6000 evaluations of the forward model. It is observed that better convergence of the cost function is obtained when blockyness is enforced

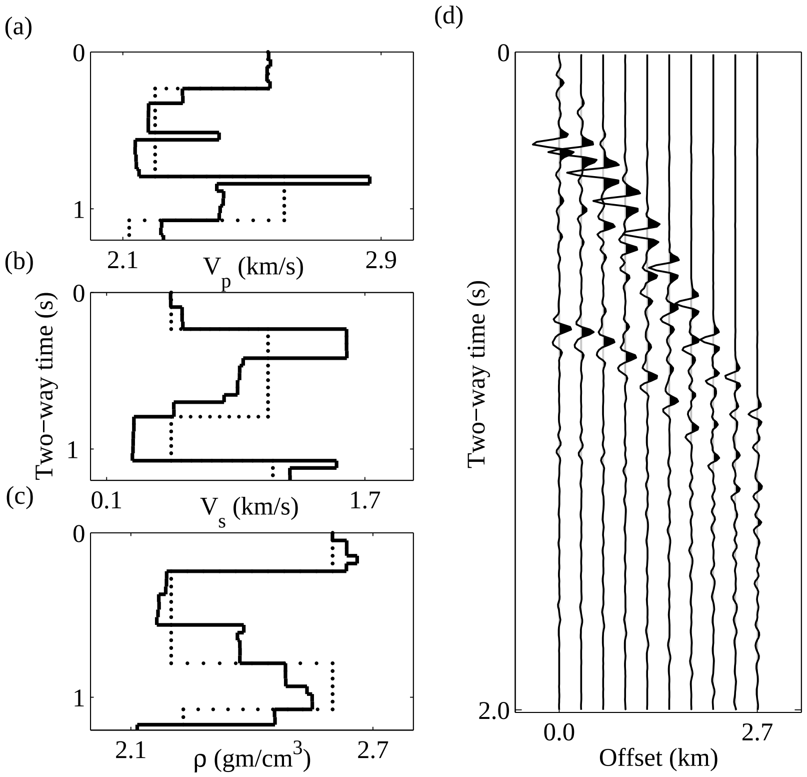

Figure 6.9: The final estimated model and the synthetic data corresponding to the final estimated model when the cost function is augmented with edge preserving regularization function. (a) The final estimated model for $\mathbf{V_p}$. (b) The final estimated model for $\mathbf{V_s}$. (c) The final estimated model for $\boldsymbol{\rho}$. The solid line shows the final estimated model and the dotted line shows the true model. (d) The synthetic data corresponding to the final estimated model. There are 10000 evaluations of the forward model.

in the solution by the model space preconditioning approach.

6.4.5 Stability of the algorithm with signal to noise ratio

The proposed algorithm is tested for stability with regard to the noise level. The signal to noise ratio (SNR) is defined as

$$SNR = \frac{max|\mathbf{d}|}{\sigma}, \tag{6.20}$$

where $\mathbf{d}$ is the observed data and σ is the standard deviation of noise. The signal to noise ratios considered for the stability test are given by $SNR = [20, 10]$.

The reflectivity method based forward operator is used to generate the synthetic

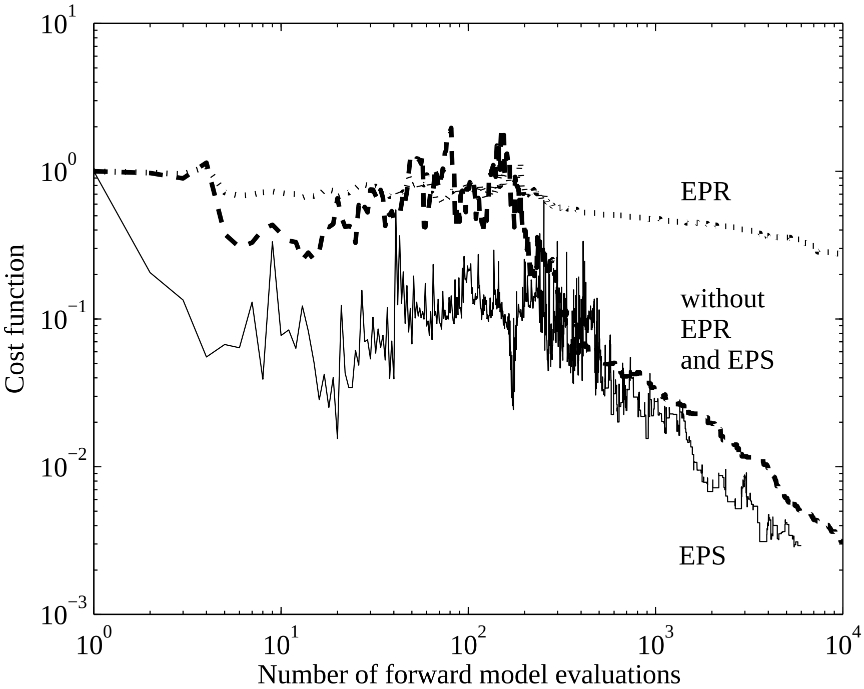

Figure 6.10: The normalized cost function computed at each forward model evaluation for the cases (a) preconditioned model space with EPS operators (solid line), (b) without model space preconditioning and without application of the edge preserving regularization potential (thick dashed line) and (c) edge preserving regularization (EPR) (dotted line). The plot is in the log-log axes. The model space preconditioning for blocky solutions provides faster convergence. The error plot for the case when the model space preconditioning operators are applied shows a greater degree of fluctuation because the cost function is evaluated twice at a particular cost and model parameter temperature.

data. The data contain multiples. Figure 6.11a shows the true synthetic data with additive noise such that $SNR = 20$. The model search bounds are same as the previous examples. Figure 6.11b shows the estimated data obtained after 6000 forward model evaluations with the model preconditioning based very fast simulated annealing optimization. Figure 6.12a, b and c shows the estimated model profiles for $\mathbf{V_p}$,$\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. It is noticed that the algorithm effectively estimates the blocky model profiles consistent with data and *a priori* knowledge. Figure 6.13a shows the true synthetic data with $SNR = 10$. Figure 6.13b shows the estimated data after 6000 forward model evaluations with the model preconditioning based very fast simulated annealing optimization. Figure 6.14a, b and c shows the estimated model profiles for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. The algorithm effectively estimates

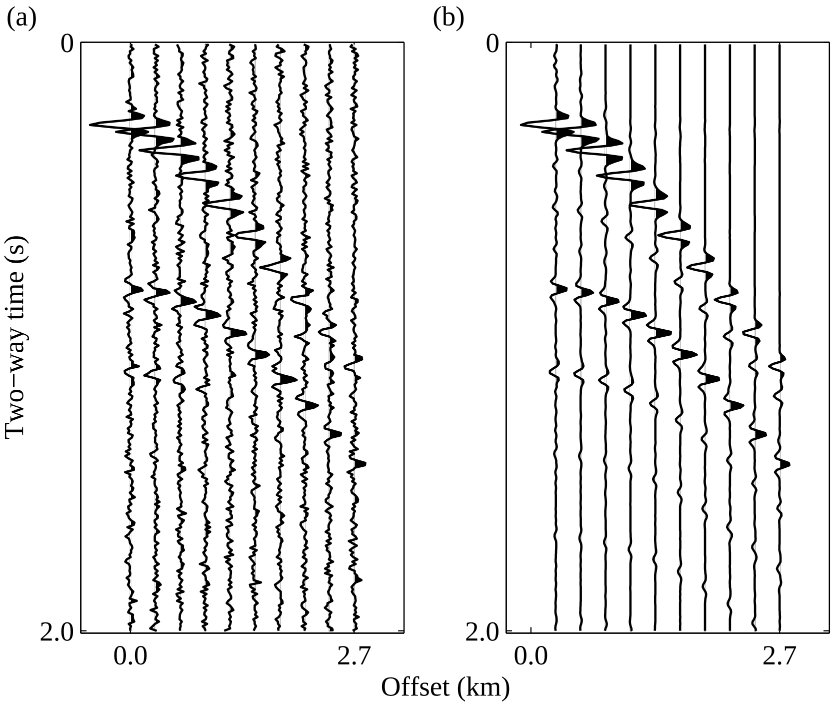

Figure 6.11: (a) The true synthetic data with $SNR = 20$. (b) The estimated data.

the blocky model profiles from noisy data consistent with the data and the *a priori* information.

6.4.6 Assessment of uncertainty

The proposed algorithm is assessed for the degree of uncertainty by successively running the algorithm with different random number generating seeds. The test was performed on the model given by the Figure 6.5a, b and c. In assessing the uncertainty involved in the algorithm, our aim is to observe consistency of the results when the algorithm is run with different random number generating seeds. In order to have any statistical conclusion about the consistency of the algorithm, a large number of Monte Carlo simulations are required to be performed. Since, the reflectivity method based forward operator is computationally expensive, the Monte Carlo test for consistency of the algorithm is impracticable within a reasonable time frame. Thus, I chose to use the ray-tracing based forward model to perform 100 numbers of Monte Carlo simulations to test the uncertainty in the model space preconditioning based very fast simulated annealing algorithm. The forward operator involved calculating the angle dependent PP-reflection coefficients from the Aki-

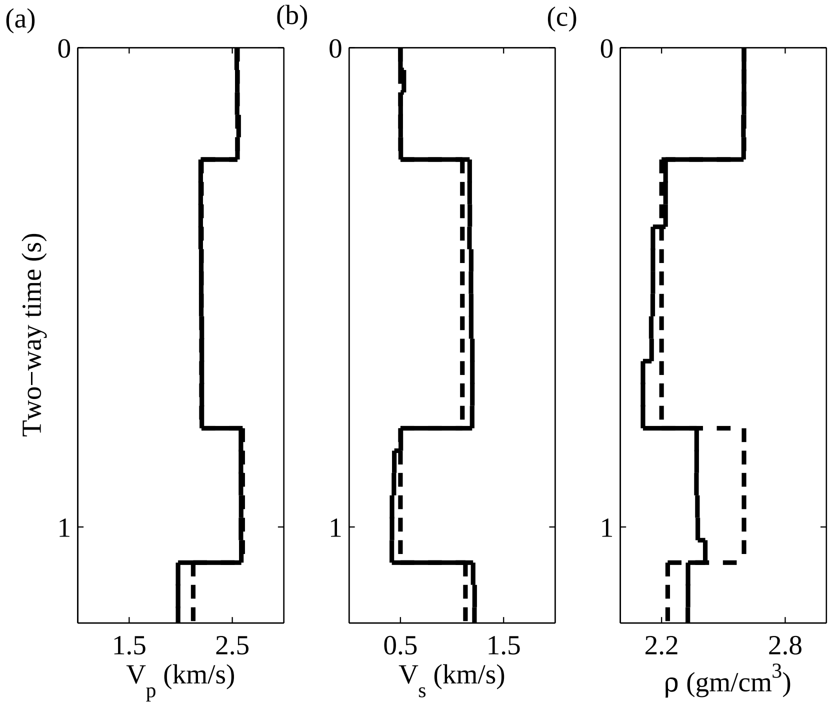

Figure 6.12: (a) Estimated models for $\mathbf{V_p}$.(b) The estimated model for $\mathbf{V_s}$. (c)The estimated model for $\boldsymbol{\rho}$. The solid line shows the estimated model and the dashed line shows the true model. Blocky profiles consistent with the data and *a priori* information are obtained in 6000 evaluations of the forward model in the optimization scheme.

Richards approximations of the Zoeppritz equations and convolving the reflection amplitudes with a known Ricker wavelet. The time-offset section thus generated, simulated the NMO corrected prestack seismic data. Synthetic data are further contaminated with bandlimited additive noise. The signal-to-noise ratio for the synthetic data is 15. The forward operator is nonlinear because the unknown model parameters are required to calculate the incidence angles from the Snell's law. The model space encompassed ±15% bounds of the true model values for $\mathbf{V_p}$ and $\boldsymbol{\rho}$ and ±60% for $\mathbf{V_s}$. Figure 6.15a, b and c show the plots for the mean of the estimated model parameters for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. The mean estimation is obtained from 100 Monte Carlo simulations. Figure 6.16a, b and c show the estimated model parameters obtained within ±50% of search bounds. It is observed that 46 times out of 100 the entire model vector $\mathbf{V_p}$, 46 times out of 100 the entire model vector $\mathbf{V_s}$ and 49 times out of 100 the entire model vector $\boldsymbol{\rho}$ for all the 30 microlayers are estimated within ±50% of the search bounds. It is also observed that the very

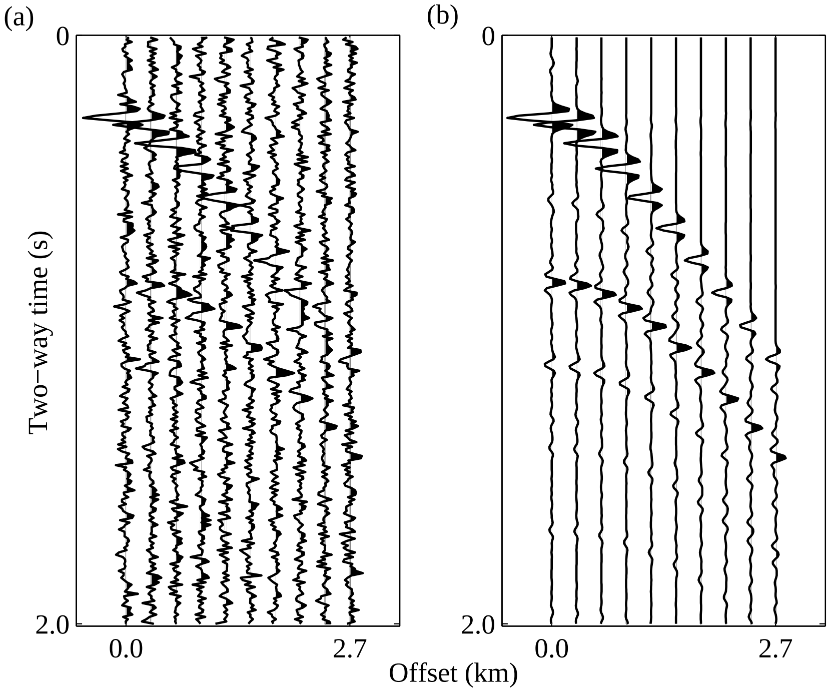

Figure 6.13: (a) The true synthetic data with $SNR = 10$. (b) The estimated data.

fast simulated annealing algorithm aided by the model preconditioning operators, consistently estimated, within reasonable accuracy, the blocky profiles for the model parameters. In Figure 6.16, though the estimated model profiles are blocky, they are not very close to the true profiles of $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$. This is due to the fact that the problem of nonuniqueness also exists within the smaller model space of blocky solutions. Though the model preconditioning operators could reduce the nonuniqueness by preferably conditioning the model space for blocky solutions, the problem of nonuniqueness within the preconditioned model space is evident in the figure.

6.4.7 Inversion of simulated data obtained from real well logs

Feasibility of the algorithm in real data situation is studied by comparing the results of model parameter estimation obtained from the well log simulated data. Due to the unavailability of the well log data for the S-wave velocities, I synthetically generated the S-wave velocity log data from the actual P-wave velocity log data. The S-wave

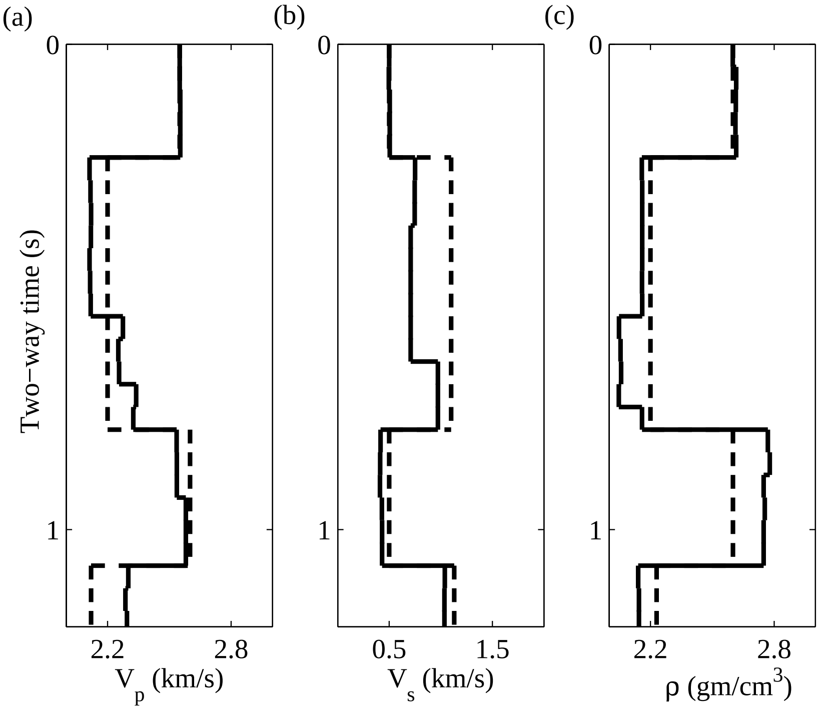

Figure 6.14: (a) Estimated models for $\mathbf{V_p}$. (b) The estimated model for $\mathbf{V_s}$. (c) The estimated model for $\boldsymbol{\rho}$. The solid line shows the estimated model and the dashed line shows the true model. Blocky profiles consistent with the data and *a priori* information are obtained in 6000 evaluations of the forward model in the optimization scheme.

velocity log data are generated by dividing the P-wave velocity log data values by $\sqrt{2}$. Figure 6.17a, b and c shows the true well log data resampled at a constant time interval. I used the P-wave velocity log to transform the log data from depth to vertical seismic travel time. An edge preserving smoothing filter is applied on the true well log data to reduce the high frequency oscillations while enhancing the edgy features at the possible layer boundaries. Figure 6.18a, b and c show the well log data after a single pass application of 5-point EPS filter. It is noticed that the application of EPS filter has enhanced the edgy features in the log profile while reducing the high frequency oscillations that are ordinarily not represented in the data. The reflectivity method based forward operator is used to synthetically generate the well log simulated data. The well log simulated data are further contaminated with bandlimited noise. The signal-to-noise ratio for the simulated well log data is 15. In order to test the stability of the algorithm with regard to the search bounds, I performed the simulations in 3 different search regimes, namely, ±15%, ±25% and

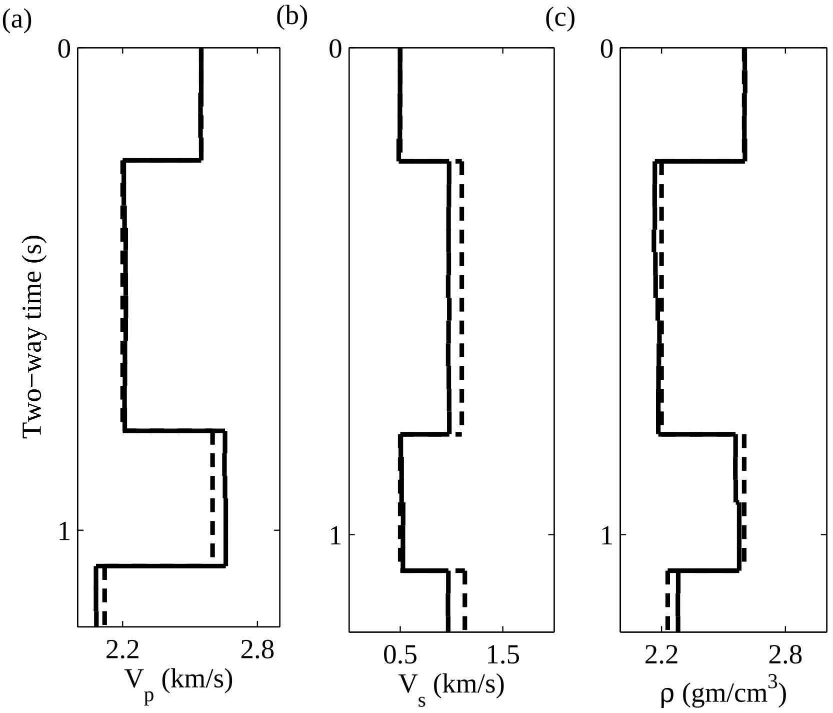

Figure 6.15: Monte Carlo simulations with 100 different random number generating seeds. (a) Mean of 100 Monte Carlo estimations for $\mathbf{V_p}$. (b) Mean of 100 Monte Carlo estimations for $\mathbf{V_s}$. (c) Mean of 100 Monte Carlo estimations for $\boldsymbol{\rho}$. The solid line represents the estimated profile and the dashed line represents the true profile.

±35% of the true well log data. As mentioned earlier, the optimization algorithm is based upon an overparameterized model space. In the optimization of the well log data, I have overparameterized the model space into 50 microlayers of equal time interval. The total number of unknown model parameters is 150. A 3-point model preconditioning operator is used in the global optimization scheme for the well log simulated data. Figure 6.19 and Figure 6.20 show the results obtained for the case when the search space is ±15% of the true well log data. Figure 6.19a shows the true well log simulated data. Figure 6.19b shows the predicted data for the initial random model. I observe that our initial random model is far away from the global minimum in the cost function. Figure 6.19c shows the estimated data after 6000 evaluations of the forward model. Figure 6.20 shows the results of the optimization in the model domain. Figure 6.20a, b and c shows the true well log and estimated well log data for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. The thick solid line shows the estimated well log and the thin solid line shows the true well log data. It is noticed that the proposed algorithm estimated the well log data within a reasonable accuracy. Figure

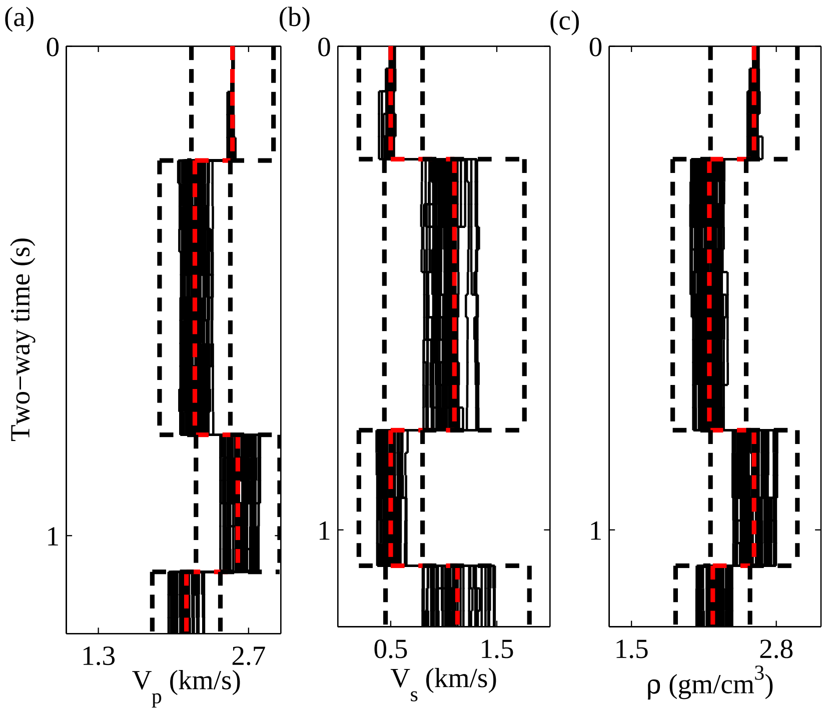

Figure 6.16: Monte Carlo simulations showing estimated models within ±50% of the search bounds. A total of 100 Monte Carlo simulations was performed. (a) Estimations for $\mathbf{V_p}$ (46 numbers). (b) Estimations for $\mathbf{V_s}$ (46 numbers). (c) Estimations for $\boldsymbol{\rho}$ (49 numbers). The outer dashed lines represent the search bounds.

6.21 and Figure 6.22 show the results obtained for the case when the search space is ±25% of the true well log data. Figure 6.21a, b and c show observed data, estimated data for the initial model and estimated data after 6000 evaluations of the forward model respectively. Figure 6.22a, b and c shows the true well log and estimated well log data for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. The thick solid line shows the estimated well log and the thin solid line shows the true well log data. It is noticed that for a search space of ±25%, the proposed algorithm estimated the well log data within a reasonable accuracy. Figure 6.23 and Figure 6.24 show the results obtained for the case when the search space is ±35% of the true well log data. Figure 6.23a, b and c shows observed data, estimated data for the initial model and estimated data after 6000 evaluations of the forward model respectively. Figure 6.24a, b and c show the true well log and estimated well log data for $\mathbf{V_p}$, $\mathbf{V_s}$ and $\boldsymbol{\rho}$ respectively. The thick solid line shows the estimated well log and the thin solid line shows the true well log data. It is noticed that, for a search space of ±35%, the algorithm converged to a solution that estimated the data. However, the model estimations are relatively less accurate. It is a known fact that the aperture limitation adversely affects the estimation of the density parameter from the prestack seismic data. I observe the same effects of aperture limitation in the model preconditioning based global optimization procedure. The estimated model shows that the P-wave and S-wave velocities are better estimated compared to the density parameter. In real data situation, it is important to obtain an accurate estimation of the wavelet. Inaccuracies in the estimation of the phase and frequency content of the wavelet will lead to less accurate estimation of the earth elastic parameters.

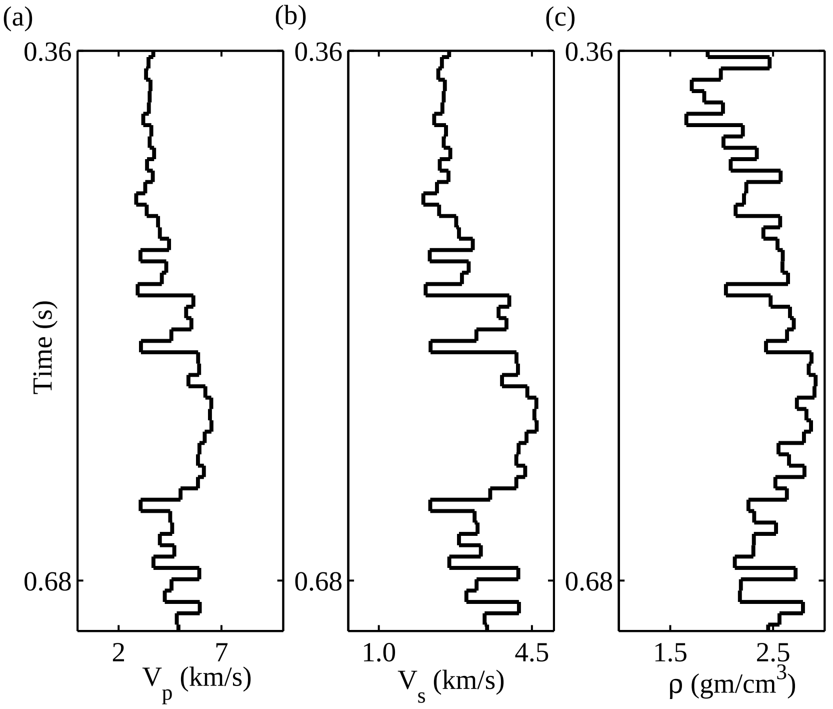

Figure 6.17: The well log data. (a) The well log data for $\mathbf{V_p}$. (b) The well log data for $\mathbf{V_s}$. (c) The well log data for $\boldsymbol{\rho}$. The S-wave well log is synthetically generated from the P-wave well log.

6.5 Summary

A new method to enforce flatness and edge preservation in the model estimation through global optimization algorithm is proposed. The method incorporates edge preserving smoothing operators as the model preconditioners. The EPS operators are nonlinear filters that involve a running array of windows over the entire model

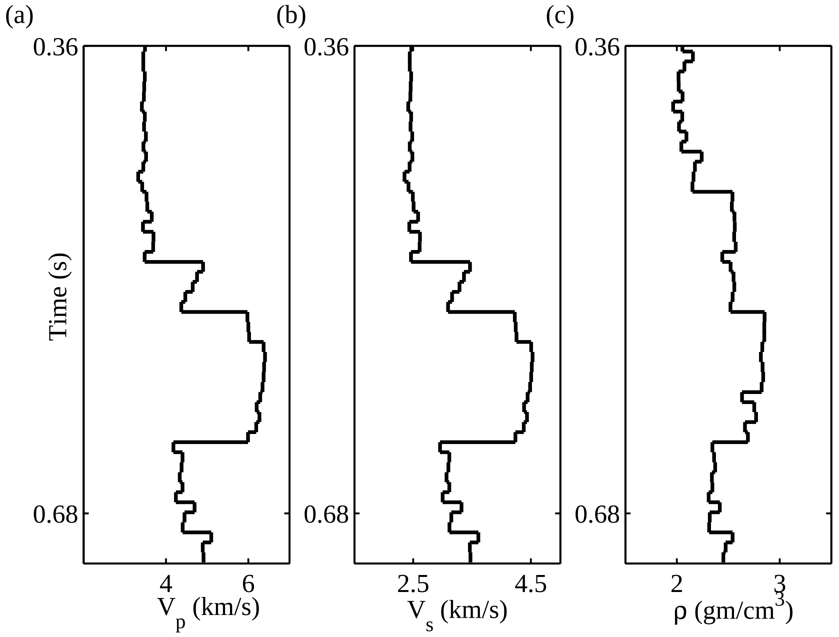

Figure 6.18: Well log data after application of EPS filter. It is observed that the EPS filter enhanced the edgy features in the profiles while suppressing the high frequency fluctuations that are not ordinarily represented in the data.

length. The most homogeneous window in each array is determined by computing the variances for each window. Flatness and edge preservation are enforced in the model domain by replacing the value of the middle point of the central window in the array with the mean value corresponding to the most homogeneous window in that array. The model preconditioner suitably biases the model space as per the blocky *a priori* information. The inclusion of model space preconditioning operators makes the global optimization algorithm converge faster. This is because the algorithm, instead of relying completely on the random perturbation, works in a suitably preconditioned model domain. Also the preconditioning operators reduce the nonuniqueness in the inversion problem by preferentially modifying the model space for a particular suite of solutions consistent with data and *a priori* knowledge. The modified model space is designed to be smaller than the complete suite of solutions consistent with the data. I applied the model space preconditioning based very fast simulated annealing algorithm to estimate the earth elastic parameters from prestack seismic amplitude data showing amplitude variation with offset. Synthetic data containing multiples were generated using the reflectivity method. The tests with synthetic data show that the algorithm effectively estimated blocky

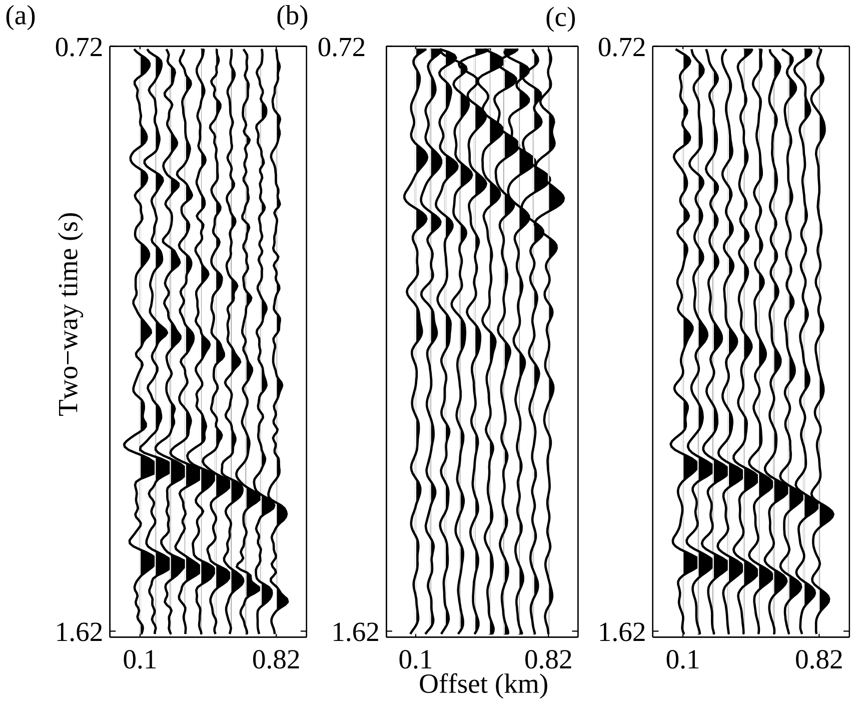

Figure 6.19: (a) The true well log simulated data with bandlimited noise. (b) The data corresponding to the initial model. (c) The estimated data obtained by the model preconditioning based very fast simulated annealing optimization algorithm. The model search space is ±15% of the true well log data.

solutions from data consistent with the *a priori* information. I have compared the results obtained with and without the application of model preconditioning operators. I observe that the results are better and the convergence is faster when the preconditioning operators are applied iteratively in the model domain. Further, the algorithm is compared with the results obtained from global optimization algorithm with a cost function augmented with an edge preserving regularization function. The comparison showed that the model preconditioning based global optimization yields better convergence and more accurate estimation of the model parameters. The algorithm is tested for stability by incorporating various levels of noise in synthetic data. The algorithm is found to be stable over a reasonable noise level. The algorithm was further tested for consistency by running several Monte Carlo simulations with different random number generating seeds. The test showed that the algorithm estimates blocky solutions from data within reasonable accuracy. I applied the algorithm to well log simulated data. I tested the stability of the algorithm over a wide range of model search bounds. I observed that the estimated model parameters are

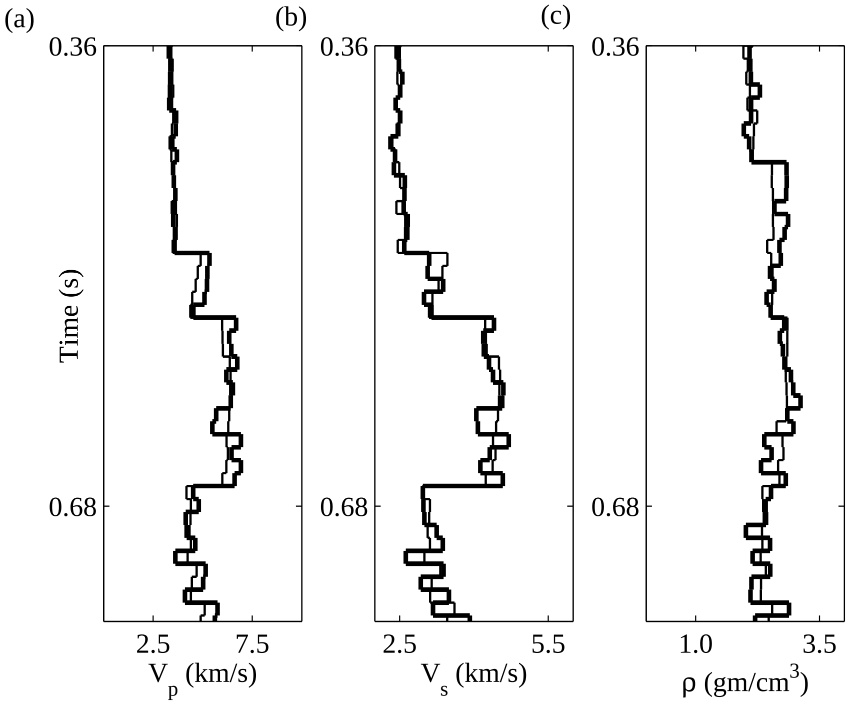

Figure 6.20: The true well log data (thin line) and the estimated well log (thick line). (a) The true and estimated models for $\mathbf{V_p}$. (b) The true and estimated models for $\mathbf{V_s}$. (c) The true and estimated models for $\boldsymbol{\rho}$. The search bound is ±15%. The model consists of 50 microlayers of constant time. Total number of unknowns is 150.

reasonably accurate over a relatively wide search bounds. The accuracy, however, decreased when the search bound reached ±35% of the true model. It is a known fact that when the optimization problem involves too many unknown parameters, reasonable search bounds be used for simulated annealing algorithms to perform well.

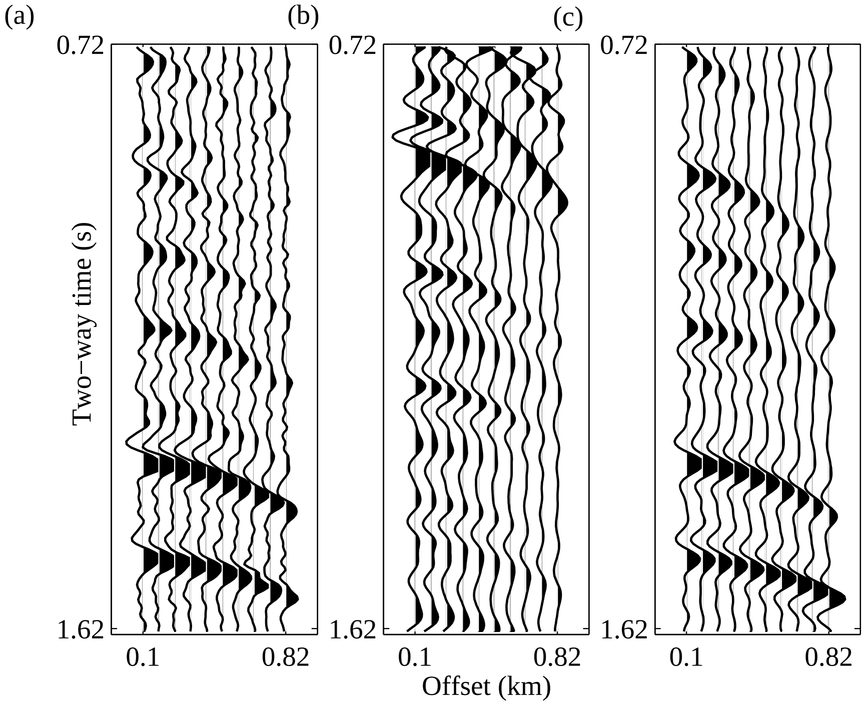

Figure 6.21: (a) The true well log simulated data with bandlimited noise. (b) The data corresponding to the initial model. (c) The estimated data obtained by the model preconditioning based very fast simulated annealing optimization algorithm. The model search space is ±25% of the true well log data.

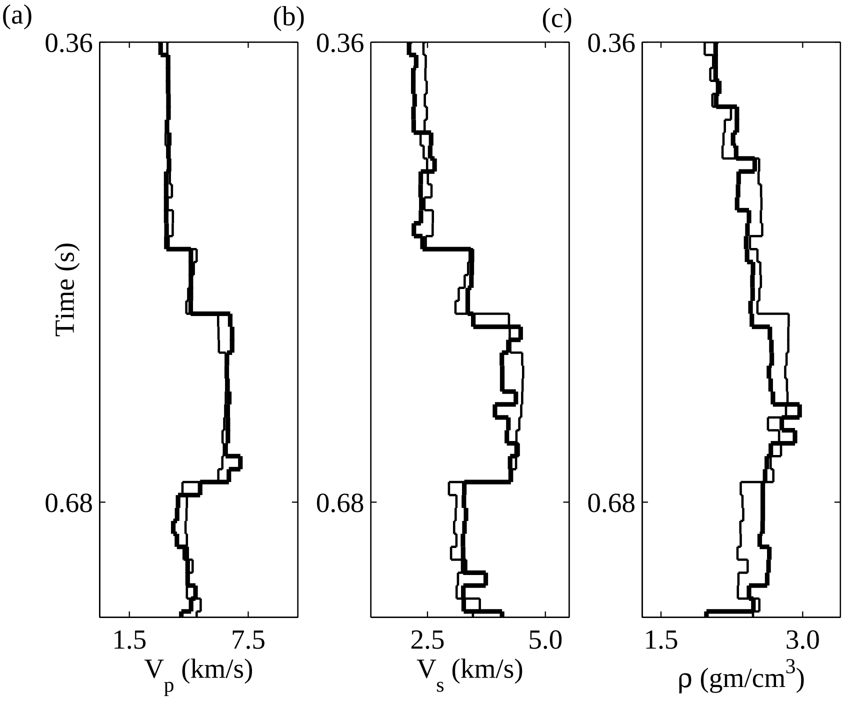

Figure 6.22: The true well log data (thin line) and the estimated well log (thick line). (a) The true and estimated models for $\mathbf{V_p}$. (b) The true and estimated models for $\mathbf{V_s}$. (c) The true and estimated models for $\boldsymbol{\rho}$. The search bound is ±25%. The model consists of 50 microlayers of constant time. Total number of unknowns is 150.

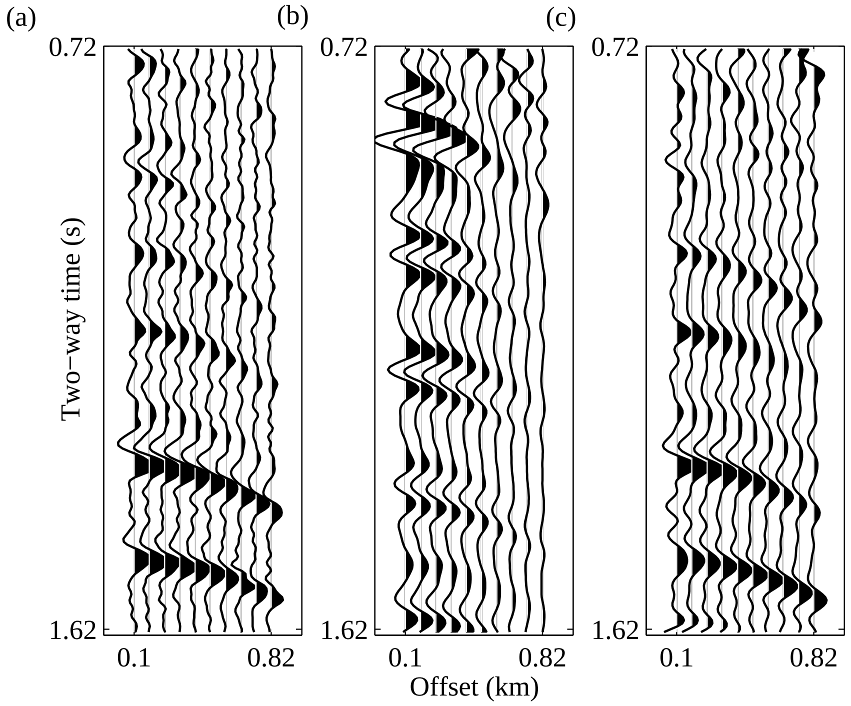

Figure 6.23: (a) The true well log simulated data with bandlimited noise. (b) The data corresponding to the initial model. (c) The estimated data obtained by the model preconditioning based very fast simulated annealing optimization algorithm. The model search space is ±35% of the true well log data.

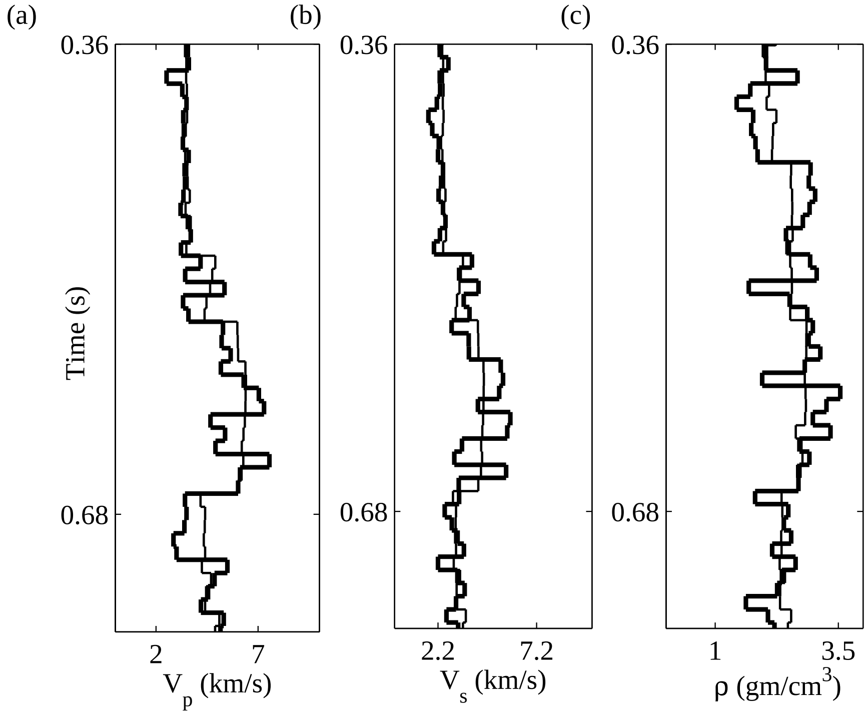

Figure 6.24: The true well log data (thin line) and the estimated well log (thick line). (a) The true and estimated models for $\mathbf{V_p}$. (b) The true and estimated models for $\mathbf{V_s}$. (c) The true and estimated models for $\boldsymbol{\rho}$. The search bound is ±35%. The model consists of 50 microlayers of constant time. Total number of unknowns is 150.

Chapter 7

Global Optimization: Application to 2D Velocity and density Estimation via Waveform Inversion

7.1 Introduction

In the previous chapter, I discussed estimation of the earth elastic parameters such as the P-wave velocity, S-wave velocity and density for the layered earth structure. The underlying assumption was that the earth elastic parameters vary only in the vertical direction. Such an assumption results in one-dimensional earth model. We have seen in chapter 6 that estimation of earth elastic parameters from the seismic data showing amplitude variation with offset is a nonlinear problem with multimodel cost function when the data contain multiples and the model is related to the data through the reflectivity method based forward operator. Multimodality in the cost function necessitated the use of global optimization schemes such as the simulated annealing or genetic algorithm. However, with the microlayer parameterization approach, the model space became quite large which adversely affected the convergence of the algorithm. Also, as noticed in the previous chapter, the *a priori* information that the solution be blocky was difficult to introduce as a regularization term in the cost function. It took a number of hit and trial approaches to find a suitable value for the trade-off parameter for the regularization term. I developed the approach of preconditioning the model space so that the global optimization algorithm worked in a model regime favorably biased according to the *a priori* information. In the example shown in the previous chapter, the *a priori* information was that the solution be blocky. In order to incorporate the *a priori* information in the global optimization algorithm, I favorably preconditioned the model space with the application of edge preserving smoothing operators on the model space generated during the very fast simulated annealing iterations. The preconditioning approach is incorporated in the algorithm as a second stage that follows the Metropolis acceptance/ rejection criterion. The length and the number of passes of the edge preserving smoothing operators are kept at a minimum. This approach ensures that the model space is not unduly biased for a blocky solution. As evident from the results shown in the previous chapter, the proposed optimization method not only made the convergence rate faster, but also resulted in accurate blocky solutions consistent with the *a priori* information (Misra and Sacchi, 2008a).

In this chapter, I discuss possible applications of global optimization in a very large model domain. We have seen that the model preconditioning approach in the global optimization scheme has made it possible to obtain a solution that honors the data and shows consistency with the *a priori* information. I have chosen a two dimensional velocity and density model space to test the viability of such a global optimization algorithm. Shot gathers, encompassing the entire model space are generated with the second-order finite difference acoustic wave equation modeling approach. The velocity and density in the two dimensional grid are estimated from the shot gathers via full waveform inversion. Jervis et al. (1993, 1996) proposed estimation of two dimensional model such as the prestack migration velocity via global optimization schemes. Their approach was based upon parameterization of the model space by cubic B-splines nodes. Such a parameterization helped the classical global optimization algorithm by limiting the model search space to the chosen cubic B-splines nodes. Application of model preconditioning based global optimization has been proposed earlier to estimate the migration velocity from the shot gathers without requiring to use cubic B-spline nodes (Misra and Sacchi, 2008b). Herein I propose model preconditioning based global optimization algorithm where the model consists of individual grid nodes and the model is preconditioned consistent with the *a priori* information. The enforced *a priori* information in the global optimization scheme are (a) smoothness in the model space along the lateral direction and (b) blockyness in the model space along the vertical direction. I have preconditioned the model space by applying smoothing operators along the lateral direction and edge preserving smoothing operators along the vertical direction of the model space. I have followed the approach of introducing the model preconditioning operators at the second stage of the algorithm as described in the previous chapter. It is a known fact that global optimization schemes operate well when the bounds of the model space are chosen judiciously. I have tested the proposed algorithm on a model space with a bound of ±15% and ±25% of the true model. The results show that the model preconditioning based global optimization scheme effectively attains a good convergence within a reasonable time frame when the model space is very large. The results also indicate effective incorporation of *a priori* information in the model space. The examples shown in this chapter are very simple. The aim is to make use of the *a priori* information and apply the global optimization algorithm to obtain a convergence to a solution that honors the data and constraints. The real earth situation is obviously more complex compared to the examples shown here and the algorithm is not tested in real earth situations. However, the ability of the proposed algorithm to obtain a good convergence within a reasonable time frame where the model space is very large is encouraging and further research work will lead to more successful applications of global optimization in complex real earth situations.

7.2 Forward operator

I have used explicit second-order finite difference acoustic wave equation based forward operator (Alford at al., 1974; Kelly et al., 1976) to generate synthetic data in the form of shot gathers. In the explicit finite-difference method the motion at a particular spatial coordinate at a future time due to a source excitation is computed from the knowledge of the motion in the past times. The forward model involved propagation of a Ricker wavelet with peak frequency of 25Hz and maximum frequency of 55Hz. Synthetic seismograms generated by the finite-difference modeling are further filtered with a passband of 1Hz to 40Hz. Synthetic seismograms are sampled at every 20m interval with a minimum and maximum source-receiver offset of 100m and 1100m respectively. I have applied muting to the direct arrivals as they are not important in the waveform inversion. I have used 4 shot gathers with a spacing of 300m between the adjacent source locations. The shot gathers are chosen in such a way that the underlying earth is properly sampled by the source-receiver geometry.

7.3 Model space

I have followed an over-parameterization approach (Sen and Stoffa, 1991; Varela et al., 2006) to define the model space for the 2D velocity and density optimization. I have over-parameterized the model space in the vertical direction. As an example, I have represented a simple 5-layer earth model in terms of 200 horizontal blocks of constant thickness. The over-parameterization is achieved by considering a two dimensional grid with 200 equidistant grid nodes in the vertical direction. The goal of the optimization is to obtain a solution where the velocity values at the grid nodes coalesce together in places that do not contribute to a reflection event and a sharp boundary in places that correspond to a reflection event. Thus, the overall solution is blocky with edges at the layer boundaries corresponding to a reflection event. In the two synthetic examples, the optimization algorithm operates within a search bound of ±15% and ±25% of the true model values respectively. It is true that the global optimization algorithms are independent of the initial model and hence are independent of the search bounds. However, in order to obtain convergence within a reasonable time frame it is imperative to impose judiciously chosen search bounds.

7.4 Model preconditioning aided VFSA scheme

The algorithm of model preconditioning aided very fast simulated annealing (MP-VFSA) has been developed and explained in detail in the previous chapter. In this section, I will discuss detail procedures for applying the MP-VFSA algorithm in estimating velocity and density over a two dimensional grid.

The following very fast simulated annealing scheme is used for the optimization of velocity and density on a two dimensional grid. The initial cost temperature and the model parameter temperature are set to relatively high values. The model parameter temperature is kept same for all the model parameters over the grid. The model parameter temperature cooling schedule is chosen such that the model parameter temperature is sufficiently low after a large number of iterations.

The cost temperature cooling schedule is given by (Ryden and Park, 2006)

$$T = T_0 a^j, \tag{7.1}$$

where T_0 is the initial cost temperature and T is the cost temperature at a given iteration. The parameter j indicates the number of accepted transitions. The term a is a positive constant less than 1. In the examples shown below, I have fixed $j = 5$ and $a = 0.9$.

The cost function to be minimized is given by an L_2-norm between the observed and predicted data. For the examples shown in the following sections, I have computed 4 shot gathers with a source-receiver geometry that encompasses the entire grid. The cost function involves computing an L_2-norm based data misfit between the observed and predicted shot gathers. As mentioned before, the shot gathers are computed by a finite-difference acoustic wave equation solver. The cost function is given by

$$J(\mathbf{m}) = \frac{1}{N_p} \sum_{s} \sum_{i=1}^{N_{off}} \sum_{j=1}^{N_t} \| d_s(x_i, t_j)^{obs} - d_s(x_i, t_j)^{est} \|^2, \tag{7.2}$$

where J is the cost function, s identifies the shot gather, $d_s(x_i, t_j)^{obs}$ and $d_s(x_i, t_j)^{est}$ are the observed and predicted data for the offset x_i and time sample t_j for the shot gather s. The total number of traces in a shot gather is given by N_{off} and the total number of time samples in a trace is given by N_t. The term N_p indicates the number of shot gathers considered in the optimization algorithm. In the following examples, I have used four shot gathers ($N_p = 4$) in the optimization algorithm. The matrix $\mathbf{m}$ contains two unknown models, namely, velocity and density over a two dimensional grid, each of size 201 by 201 cells. The equations 7.3 and 7.4 show the computation of predicted data at each MP-VFSA algorithm iteration.

$$d_s(x_i, t_j)_1^{est} = \mathbf{g}(\mathbf{m}), \tag{7.3}$$

$$d_s(x_i, t_j)_2^{est} = \mathbf{g}(MP(\mathbf{m})), \tag{7.4}$$

where $\mathbf{g}$ and MP are the forward operator and the model preconditioning operator respectively.

The *a priori* constraints I am imposing on the MP-VFSA algorithm are (a) the model is smooth along the lateral direction and (b) the model is blocky along the vertical direction. Hence, I am using smoothing operators along the lateral direction of the model space and edge preserving smoothing operators along the vertical direction of the model space. This means that the operator MP in the equation 7.4 consists of two different operators, namely, the smoothing operator and the edge preserving smoothing(EPS) operator. I have used moving average smoothing operator along the lateral direction of the model space and EPS operator, as defined in the previous chapter, along the vertical direction of the model space.

7.5 Synthetic data example

Synthetic data are generated by solving the second-order finite difference acoustic wave equation. Data contain multiples. Figure 7.1 shows the true velocity and density models used in generating the synthetic data. The synthetic data corresponding to the true model are shown in Figure 7.2.

The model space is defined over two grids, each consisting of 201 by 201 cells.

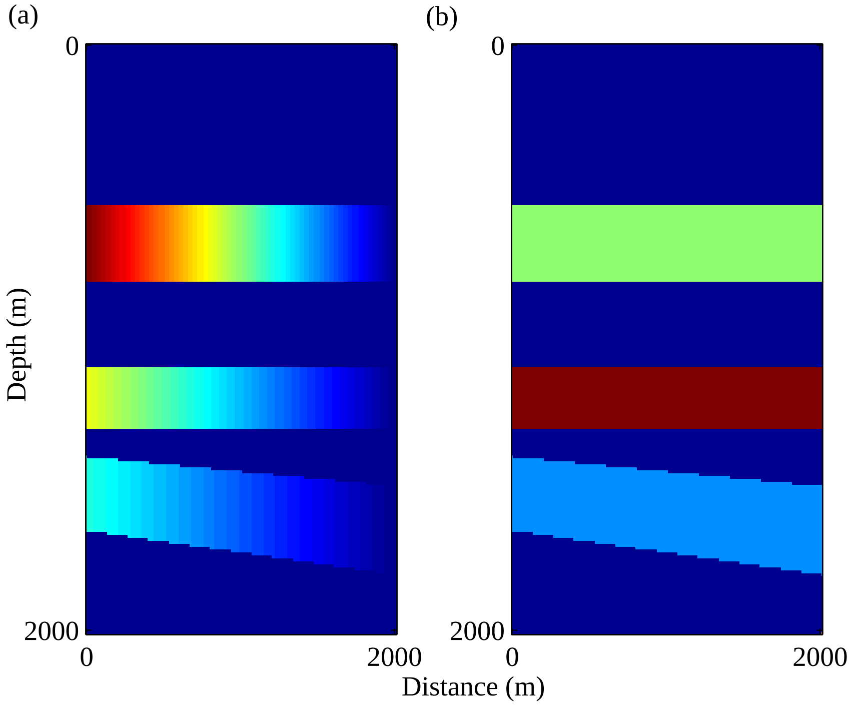

Figure 7.1: The true models. (a) Velocity model and (b) Density model. The velocity model shows a lateral velocity gradient.

The grid nodes are separated by 10m in both lateral and vertical axes. The data space contains 4 shot gathers. The shot locations for each gather is separated by

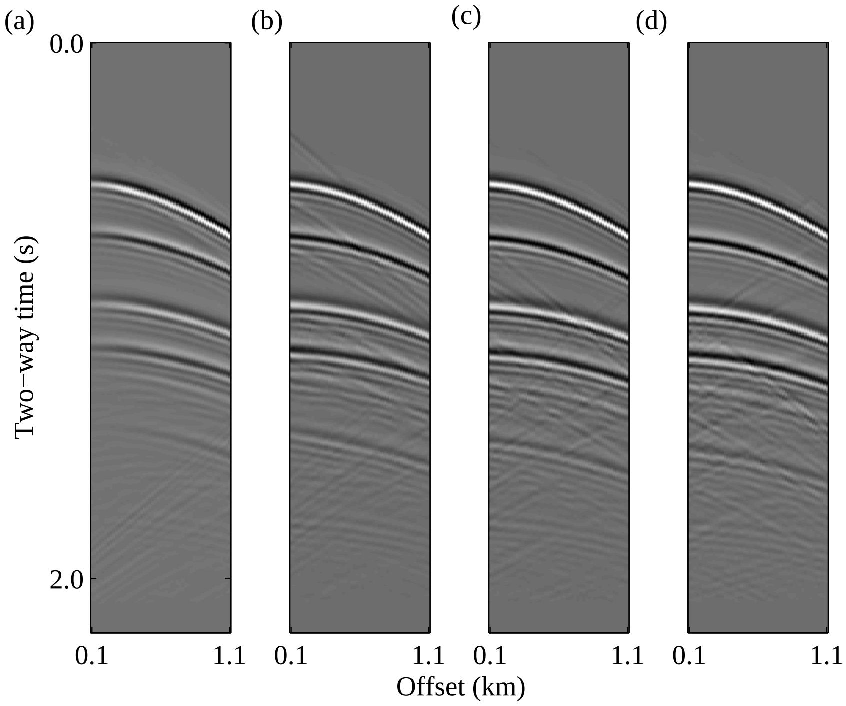

Figure 7.2: The true shot gathers. (a) shot location at the origin 0m (b) Shot location at 300m from the origin (c) Shot location at 600m from the origin and (d) Shot location at 900m from the origin.

300m with respect to the adjacent gather. The shot gathers are chosen in such a way that the entire model grid is covered by the forward model. In the synthetic example, I am imposing smoothness constraint in the lateral direction and blockyness constraint in the vertical direction that is consistent with the *a priori* information. Figures 7.3 and 7.4 show the initial random models generated within ±15% and ±25% bounds respectively. I have chosen a particular grid location (300m,0m) to show the initial model as seen in the depth axis. Figure 7.5 and 7.6 show the initial velocity and density models at the grid location (300m,0m) for search bounds of ±15% and ±25% respectively. The figures also show the true model and the respective upper and lower bounds of the model space.

Figure 7.7 and Figure 7.8 show the shot gathers corresponding to the initial random model generated within a search bound of ±15% and ±25% respectively. Difference between the true data and the data corresponding to the initial random model for the bounds of ±15% and ±25% respectively are shown in Figures 7.9 and 7.10. The figures indicate that the initial models are not close to the global minimum of the cost function.

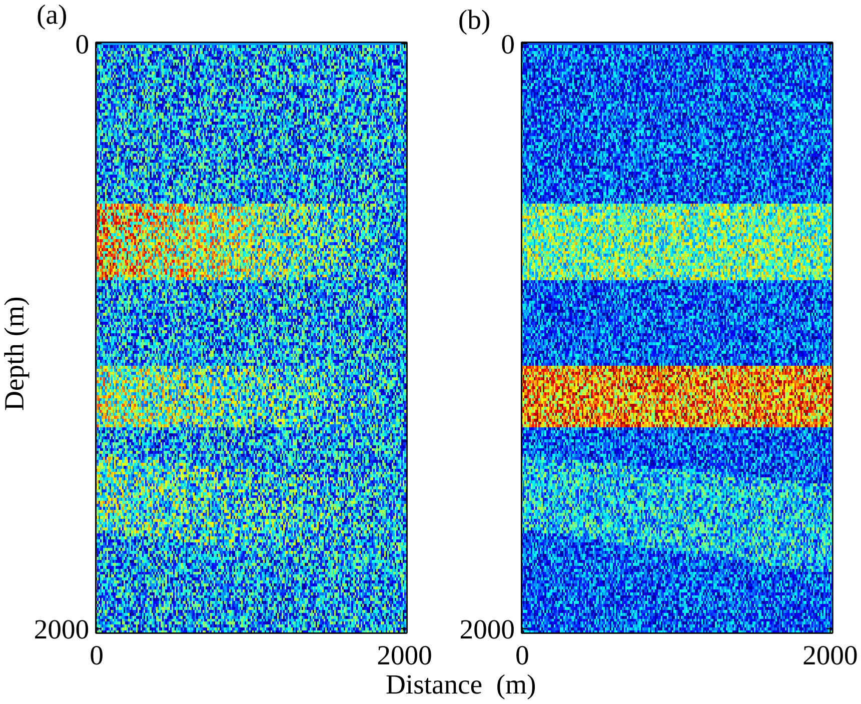

Figure 7.3: The initial velocity and density models. The initial model is chosen with an upper and lower bounds of ±15% of the respective true values.

Figure 7.11 and Figure 7.12 show the estimated velocity and density values for the model preconditioning based VFSA optimization operating within ±15% and ±25% bounds respectively. Results of the model preconditioning based VFSA optimization is better visualized by plotting the vertical distribution of the velocity and density distribution at a particular lateral location. Figure 7.13 and Figure 7.14 show the estimated, initial and true model profiles for velocity and density at the surface location 300m from the origin for the search bounds ±15% and ±25% respectively. It is observed that the model preconditioning based VFSA algorithm optimizes for the velocity and density models within a reasonable accuracy. Figures 7.15 and 7.16 show the estimated data for the model preconditioning based VFSA algorithm operating within search bounds of ±15% and ±25% respectively. Figure 7.17 and Figure 7.18 show the difference between the estimated data and the true data for the model preconditioning based VFSA algorithm operating within search bounds of ±15% and ±25% respectively. The figures show that the model preconditioning based global optimization algorithm is successful in reasonably estimating the data.

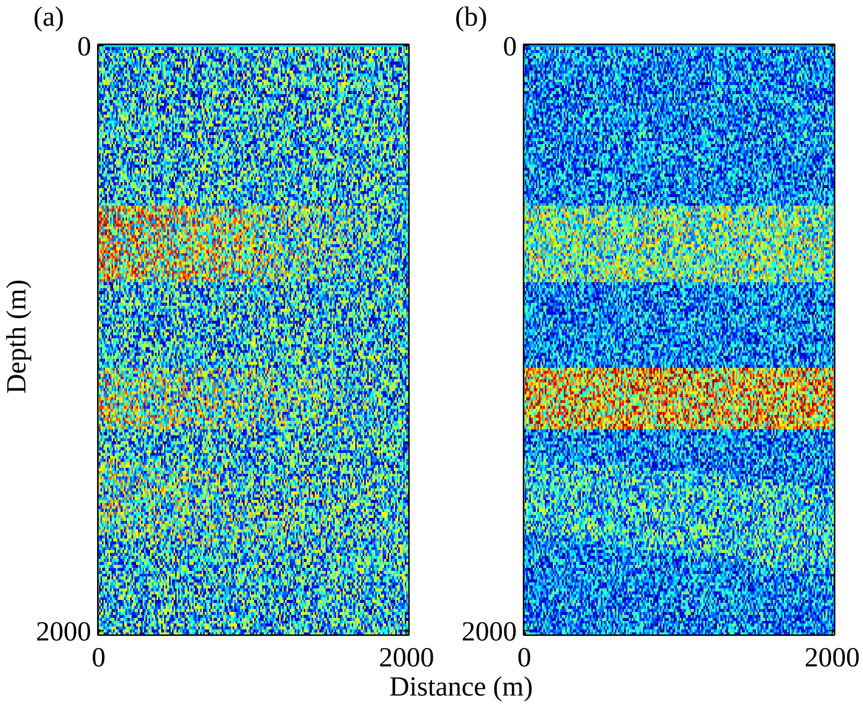

Figure 7.4: The initial velocity and density models. The initial model is chosen with an upper and lower bounds of ±25% of the respective true values.

Progress of the model preconditioning based VFSA algorithm is indicated by the evaluation of the cost function at each MP-VFSA iteration. Figure 7.19 and Figure 7.20 show the evaluation of the cost function as the model preconditioning based VFSA algorithm progressed through iterations operating within the search bounds of ±15% and ±25% respectively.

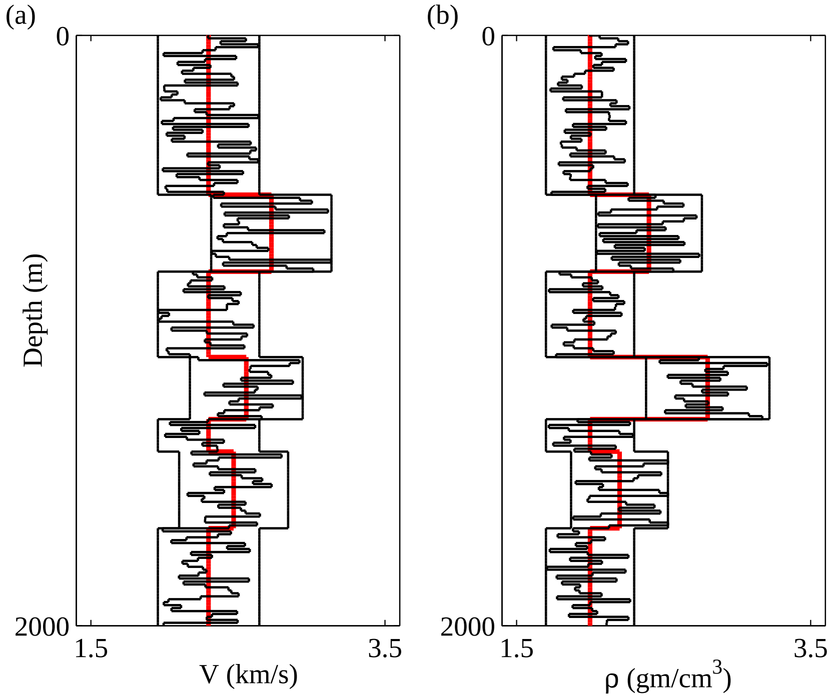

Figure 7.5: The initial velocity and density models in one dimension at the location (300m, 0m). The initial model is chosen with an upper and lower bounds of ±15% of the respective true values. The red line shows the true model, the blocky line shows the upper and lower bounds of the model space and the zigzag line shows the initial model.

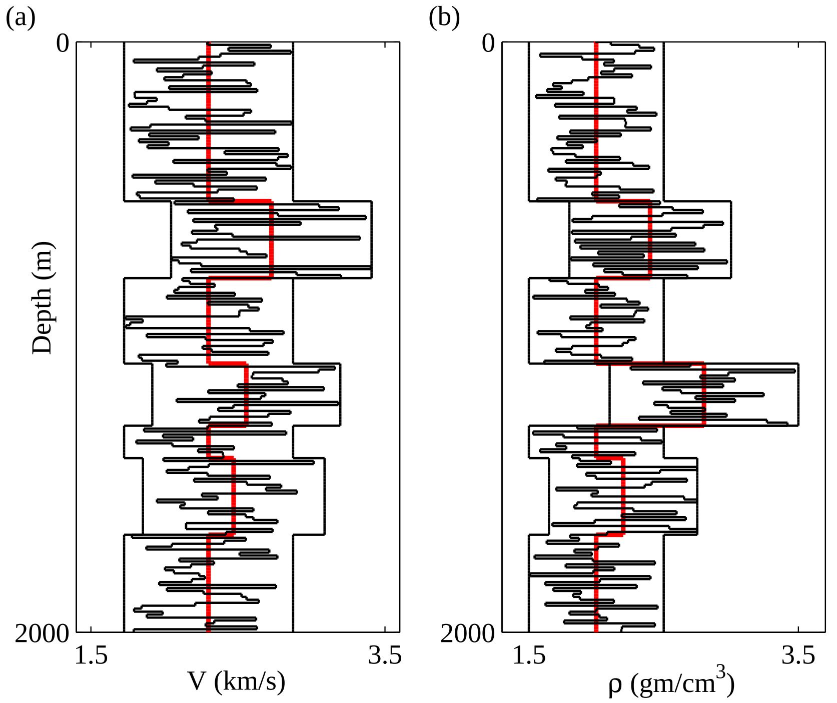

Figure 7.6: The initial velocity and density models in one dimension at the location (300m, 0m). The initial model is chosen with an upper and lower bounds of ±25% of the respective true values. The red line shows the true model, the blocky line shows the upper and lower bounds of the model space and the zigzag line shows the initial model.

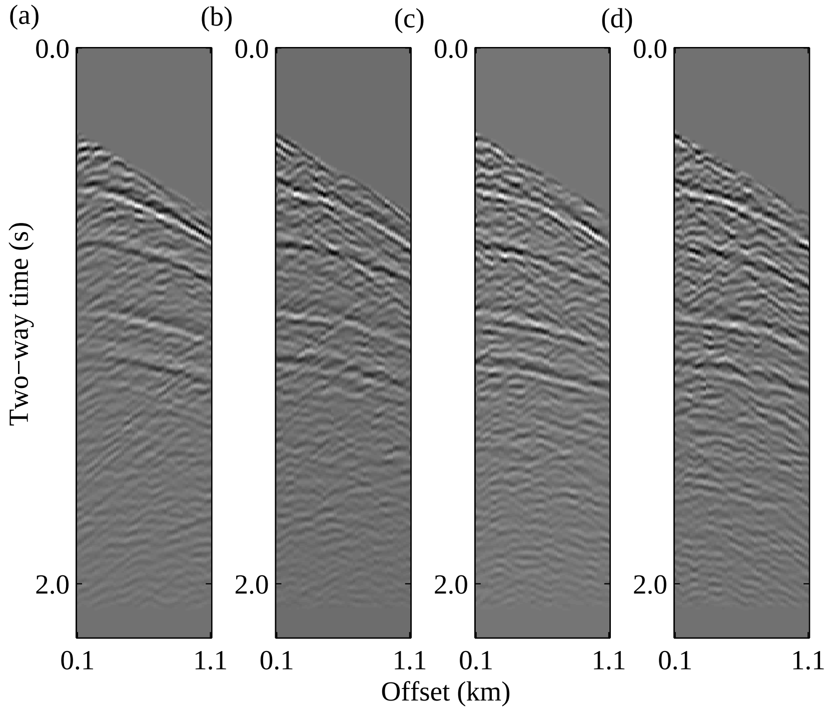

Figure 7.7: Data corresponding to the initial model chosen within ±15% bounds of the true model. (a) Shot location at the origin (0m, 0m). (b) Shot location at 300m from the origin. (c) Shot location at 600m from the origin and (d) Shot location at 900m from the origin.

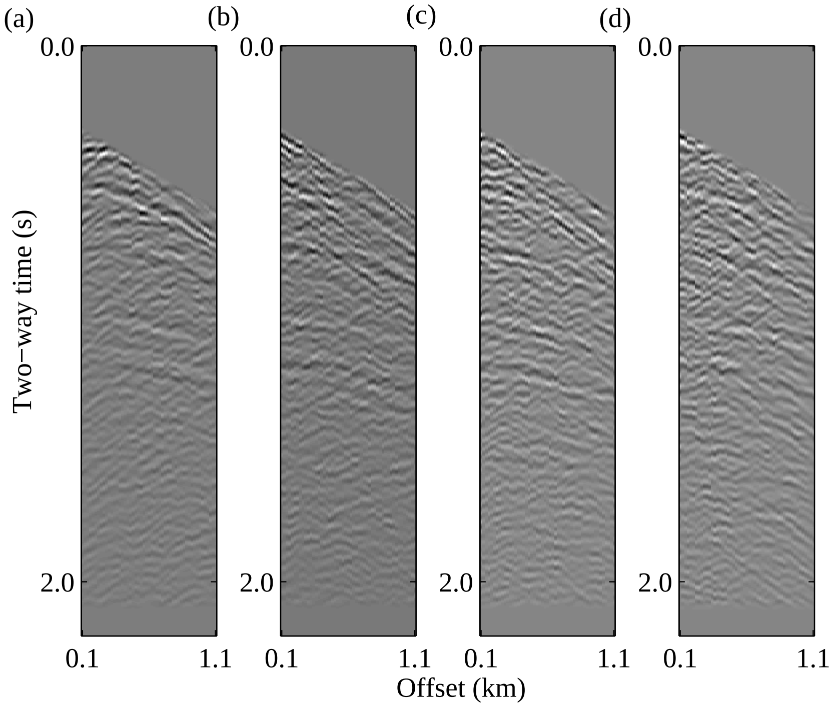

Figure 7.8: Data corresponding to the initial model chosen within ±25% bounds of the true model. (a) Shot location at the origin (0m,0m). (b) Shot location at 300m from the origin. (c) Shot location at 600m from the origin and (d) Shot location at 900m from the origin.

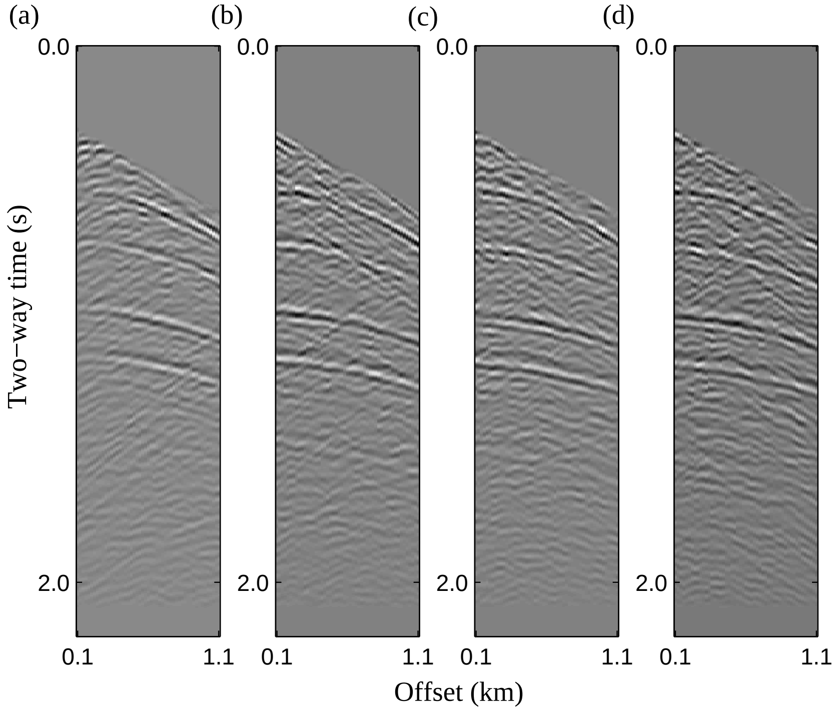

Figure 7.9: Difference between data corresponding to the initial model chosen within ±15% bounds of the true model and the true data. (a) Shot location at the origin (0m, 0m). (b) Shot location at 300m from the origin. (c) Shot location at 600m from the origin and (d) Shot location at 900m from the origin.

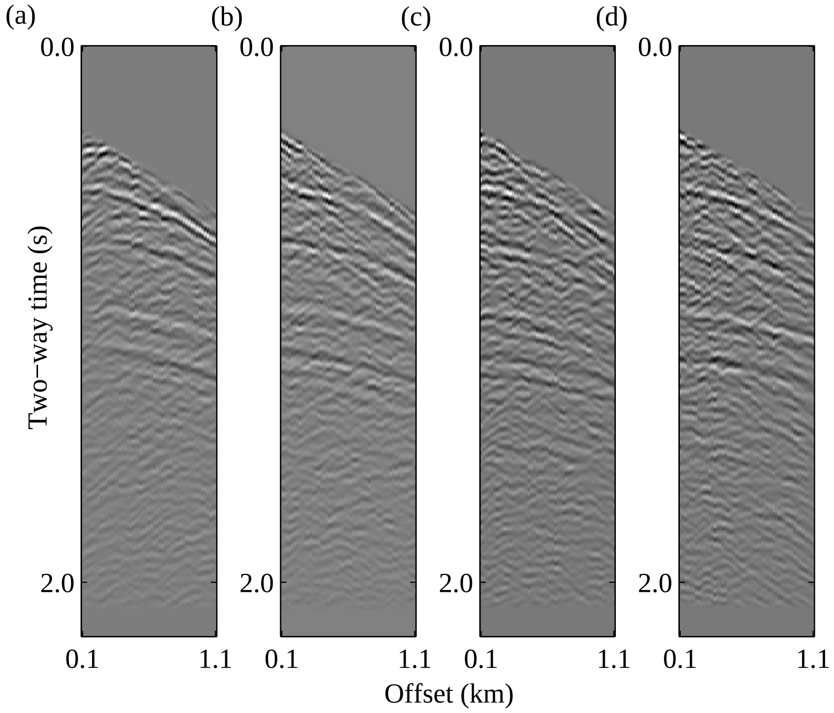

Figure 7.10: Difference between data corresponding to the initial model chosen within ±25% bounds of the true model and the true data. (a) Shot location at the origin (0m, 0m). (b) Shot location at 300m from the origin. (c) Shot location at 600m from the origin and (d) Shot location at 900m from the origin.

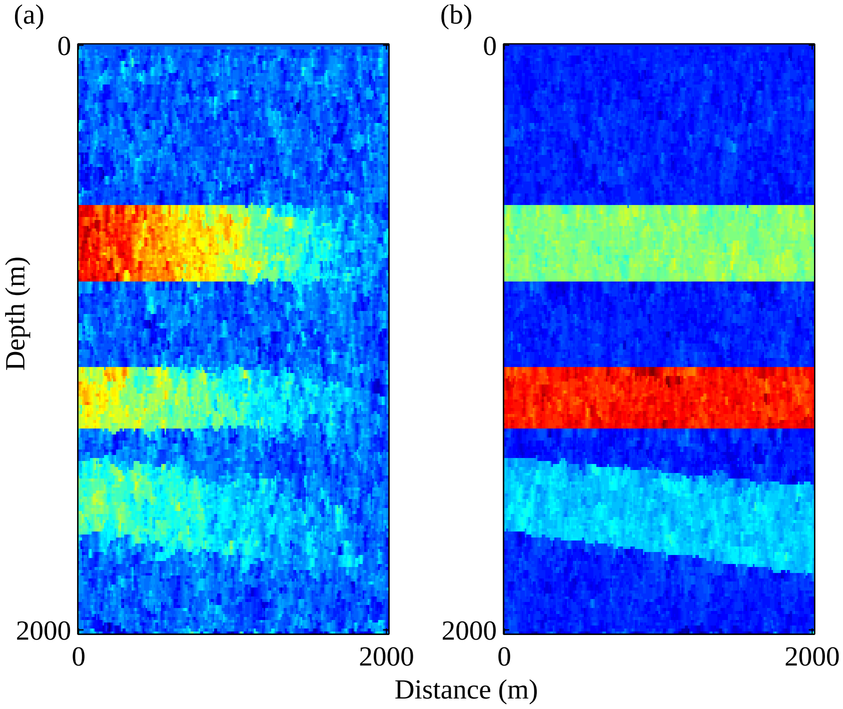

Figure 7.11: Estimated velocity and density model after 400 model preconditioning based VFSA iterations. (a) The estimated velocity model and (b) the estimated density model. The model search bound is ±15% of the true model values.

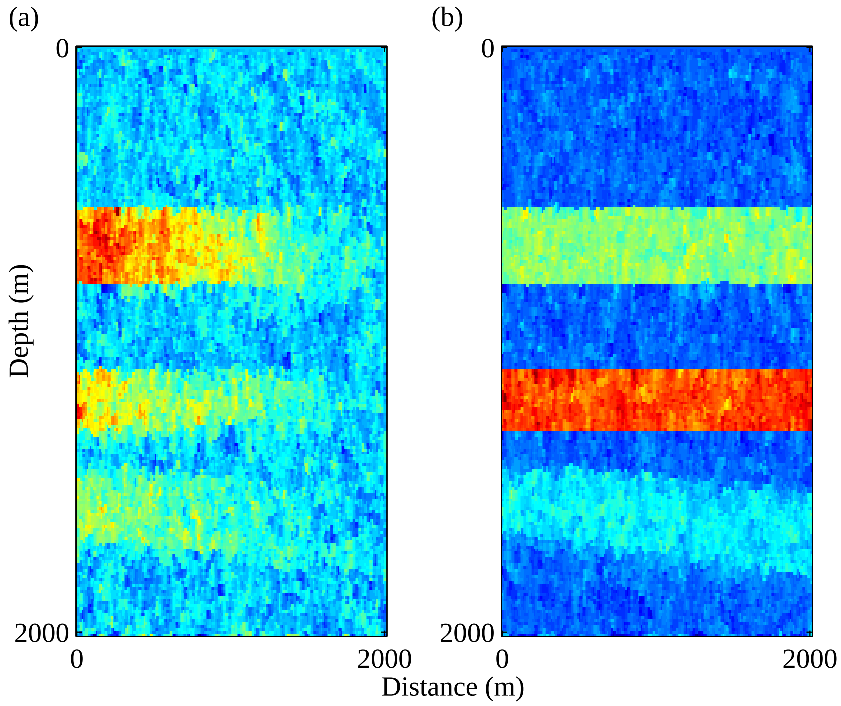

Figure 7.12: Estimated velocity and density model after 400 model preconditioning based VFSA iterations. (a) The estimated velocity model and (b) the estimated density model. The model search bound is ±25% of the true model values.

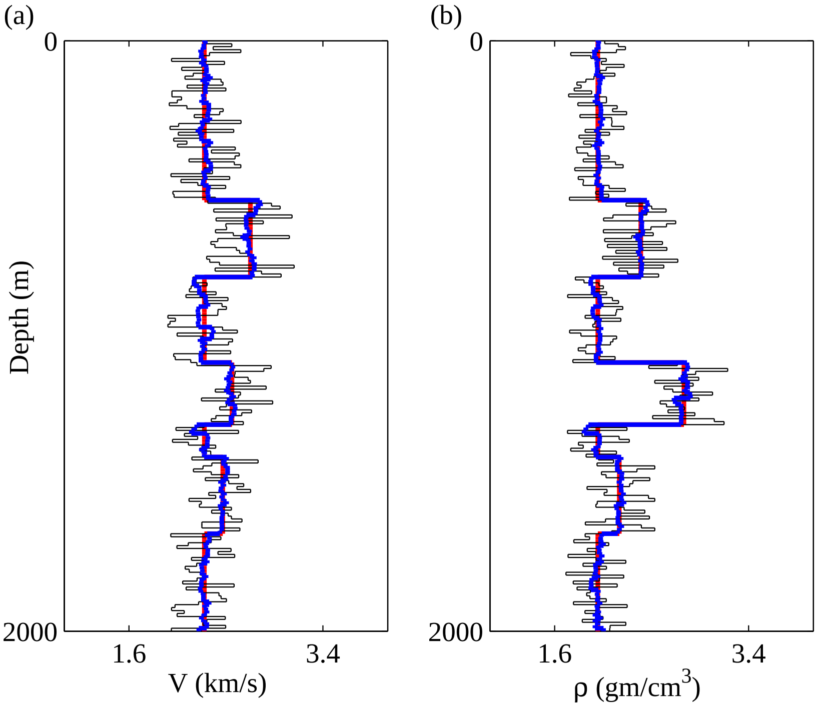

Figure 7.13: Estimated model (blue), true model (red) and initial model (black) for (a) velocity ($\mathbf{V}$) and (b) density ($\boldsymbol{\rho}$). The model search bound is ±15% of the true model values. Surface location of the vertical profile is 300m from the origin.

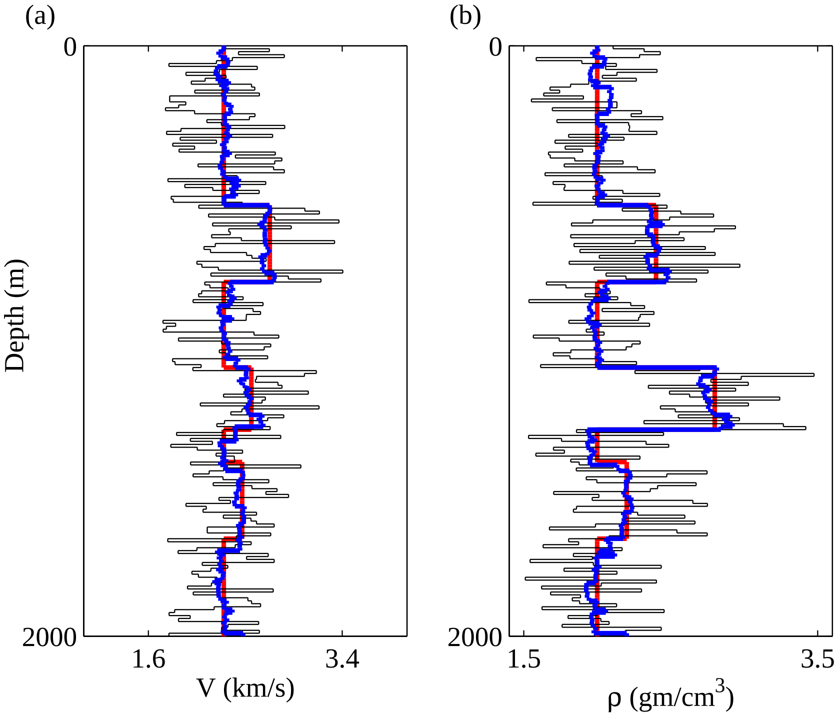

Figure 7.14: Estimated model (blue), true model (red) and initial model (black) for (a) velocity ($\mathbf{V}$) and (b) density ($\boldsymbol{\rho}$). The model search bound is ±25% of the true model values. Surface location of the vertical profile is 300m from the origin.

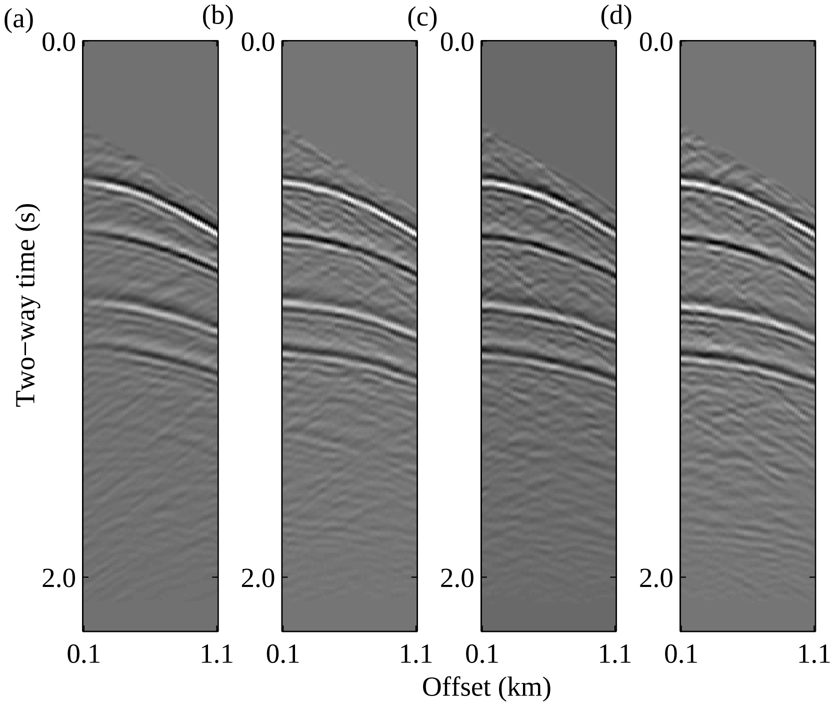

Figure 7.15: Estimated data after 400 model preconditioning based VFSA iterations. (a) Shot location is at the origin (0m, 0m). (b) Shot location is at 300m from the origin. (c) Shot location is at 600m from the origin. (d) Shot location is at 900m from the origin. The search bounds are ±15% of the true models.

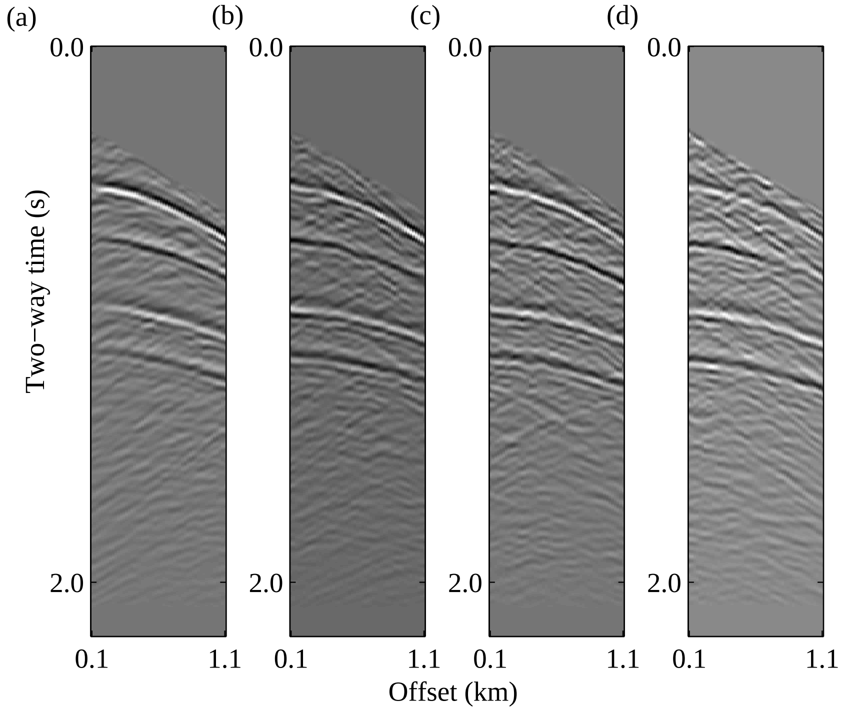

Figure 7.16: Estimated data after 400 model preconditioning based VFSA iterations. (a) Shot location is at the origin (0m,0m). (b) Shot location is at 300m from the origin. (c) Shot location is at 600m from the origin. (d) Shot location is at 900m from the origin. The search bounds are ±25% of the true models.

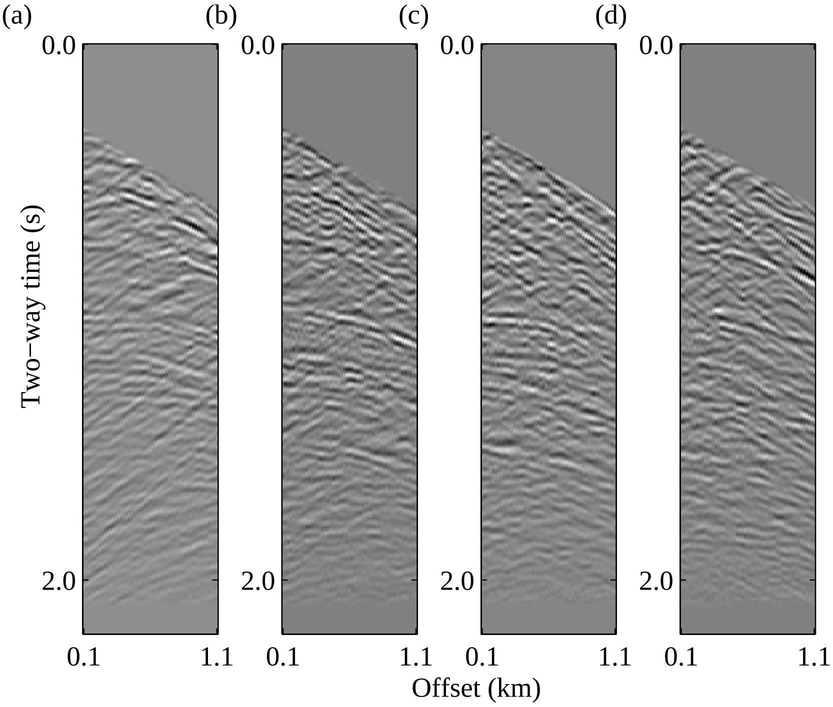

Figure 7.17: Difference between the estimated and the true data. (a) Shot location is at the origin (0m, 0m). (b) Shot location is at 300m from the origin. (c) Shot location is at 600m from the origin. (d) Shot location is at 900m from the origin. The search bounds are ±15% of the true models.

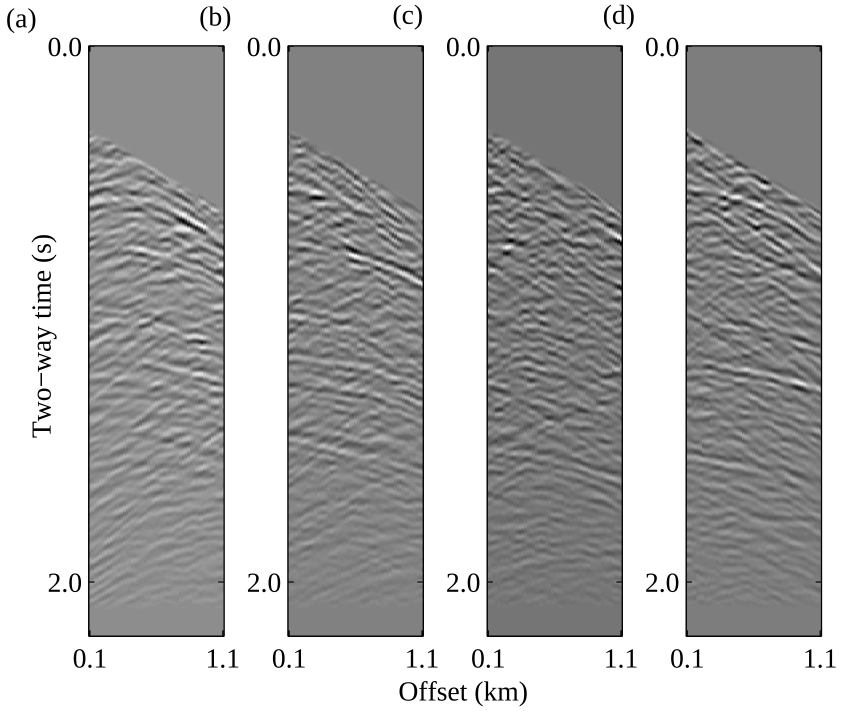

Figure 7.18: Difference between the estimated and the true data. (a) Shot location is at the origin (0m, 0m). (b) Shot location is at 300m from the origin. (c) Shot location is at 600m from the origin. (d) Shot location is at 900m from the origin. The search bounds are ±25% of the true models.

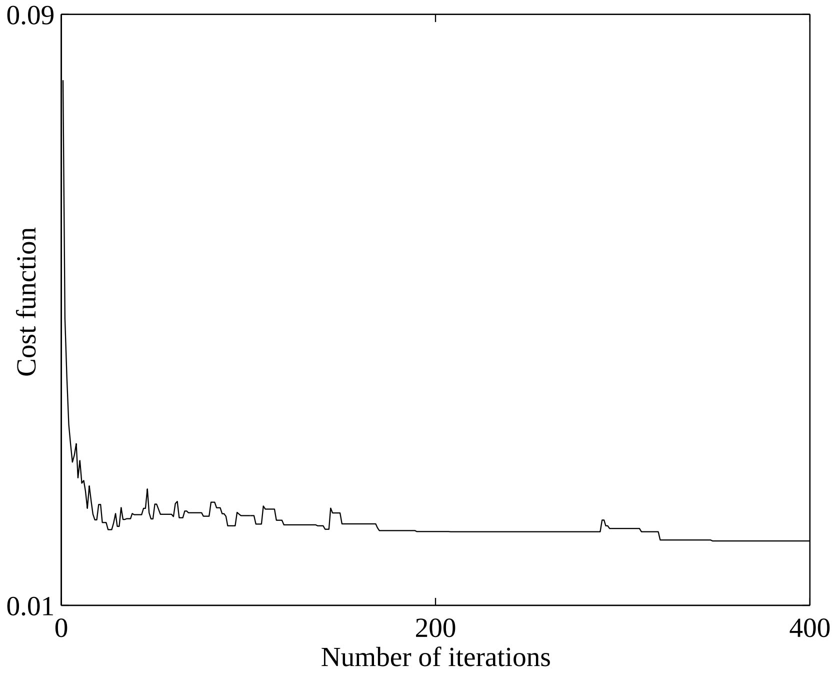

Figure 7.19: The cost function evaluated at each model preconditioning based VFSA optimization. The search bounds are ±15% of the true models.

The *a priori* information that the model space is smooth in the lateral direction and blocky in the vertical direction is imposed in the global optimization scheme by applying smoothing operators along the lateral direction and edge preserving smoothing operators along the vertical direction of the model space. Smoothing along the lateral direction is enforced by applying three-point moving average filter in the lateral direction of the model space. The moving average operator is allowed to pass through the lateral direction of the model only once during each VFSA iteration. The length of the edge preserving smoothing operator is also kept fixed at 3 and the operators are allowed to pass through the vertical direction of the model only once during the VFSA iteration. By keeping the moving average and the edge preserving smoothing operators length to a minimum and the number of passes to one, the model space is prevented from being overly smooth in the lateral direction and overly blocky in the vertical direction. A total of 400 model preconditioning based VFSA iterations were performed to obtain the results shown in the examples.

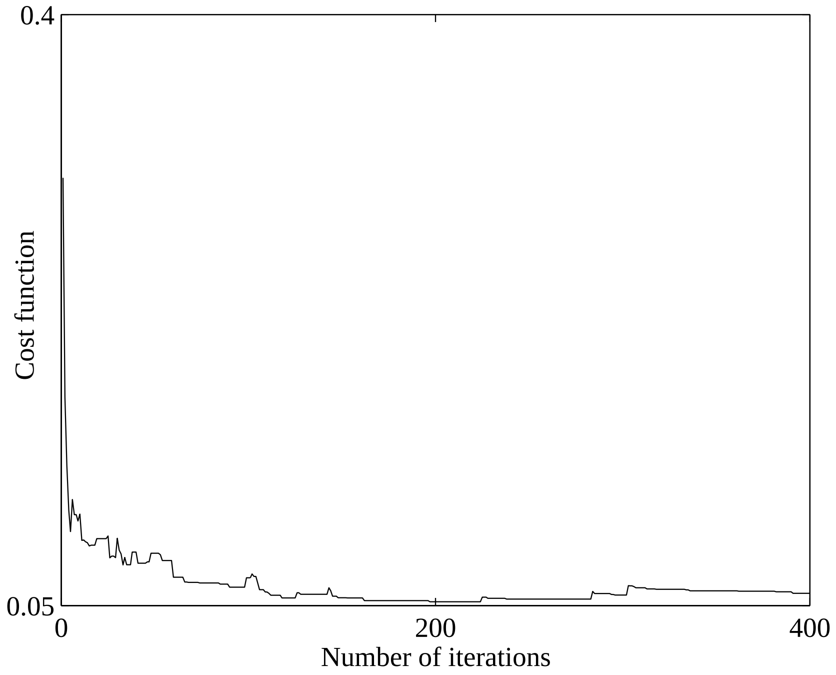

Figure 7.20: The cost function evaluated at each model preconditioning based VFSA optimization. The search bounds are ±25% of the true models.

7.5.1 Stability of the algorithm with SNR

Model preconditioning based VFSA algorithm is tested for stability with respect to different noise levels. I have tested the algorithm operating within two different search bounds, namely, ±15% and ±25% of the true models at $SNR = [20, 10]$. The SNR is defined by the following equation.

$$SNR = \frac{max_{i,j}|d(x_i, t_j)|}{\sigma}, \tag{7.5}$$

where SNR represents the signal-to-noise ratio, $d(x_i, t_j)$ represents the observed data at offset x_i, time t_j and σ is the standard deviation of noise.

Synthetic data generated by the second-order finite difference solution of the acoustic wave equation are further contaminated with bandlimited noise such that the SNR in two data sets are 20 and 10 respectively. The stability of the algorithm is tested within the search bounds of ±15% and ±25% respectively. Figure 7.21 shows the true data with $SNR = 20$, estimated data and the data residue after 400 model preconditioning based VFSA iterations when the search bound is set to

±15% of the true model values.

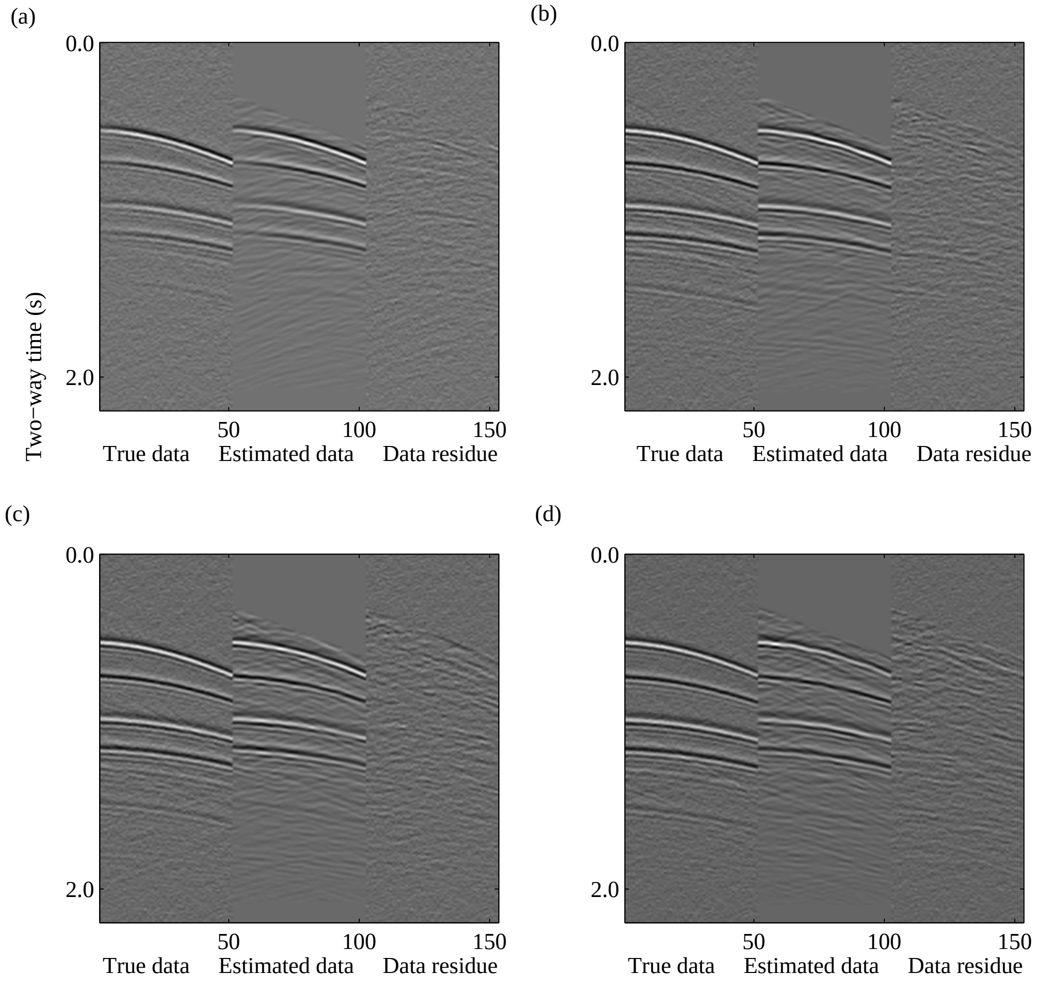

Figure 7.21: True data, estimated data and data residue. (a) Shot location is at the origin (0m, 0m). (b) Shot location is at 300m from the origin. (c) Shot location is at 600m from the origin. (d) Shot location is at 900m from the origin. True data contain bandlimited noise such that $SNR = 20$. Search bound of the model space is ±15%.

Figure 7.22 shows the estimated model after 400 model preconditioning based VFSA iterations. It is observed that the algorithm successfully estimates the model within a search bound of ±15% via waveform inversion of shot gathers with $SNR = 20$.

Figure 7.23 shows the true data with $SNR = 10$, estimated data and the data residue after 400 model preconditioning based VFSA iterations when the search bound is set to ±15% of the true model values. Figure 7.24 shows the estimated model after 400 model precondition based VFSA iterations. It is observed that the algorithm successfully estimated the model within a search bound of ±15% when data are contaminated with noise such that $SNR = 10$.

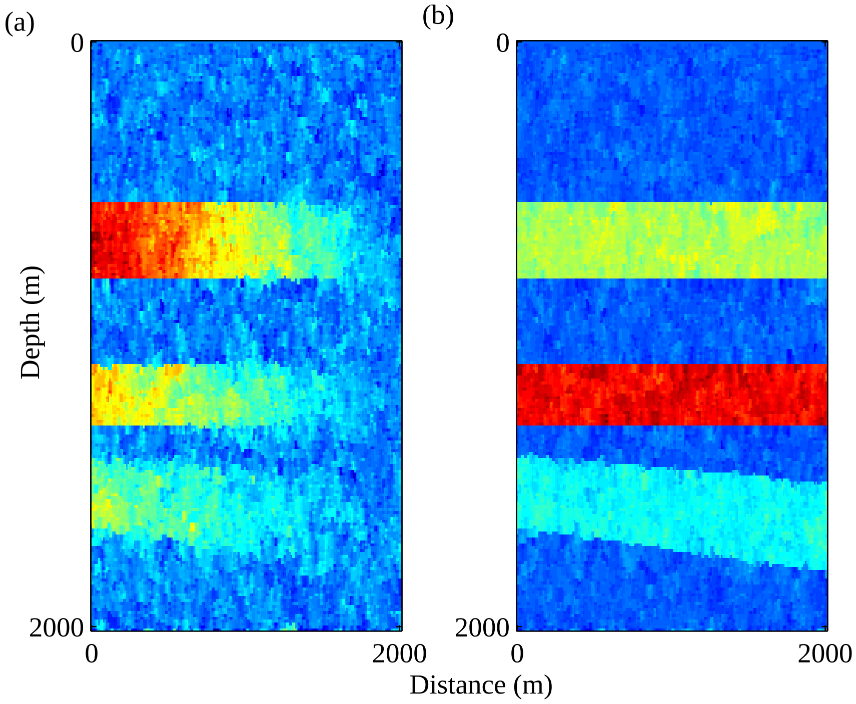

Figure 7.22: Estimated model after 400 model preconditioning based VFSA iterations. (a) Estimated velocity and (b) estimated density. True data contain bandlimited noise such that $SNR = 20$. Search bound of the model space is ±15%.

Further, the algorithm is tested for stability within a search bound of ±25% with $SNR = [20, 10]$. Figure 7.25 shows the true data with $SNR = 20$, estimated data and the data residue after 400 model preconditioning based VFSA iterations when the search bound is set to ±25% of the true model values. Figure 7.26 shows the estimated model after 400 model precondition based VFSA iterations. It is observed that the algorithm successfully estimated the model within a search bound

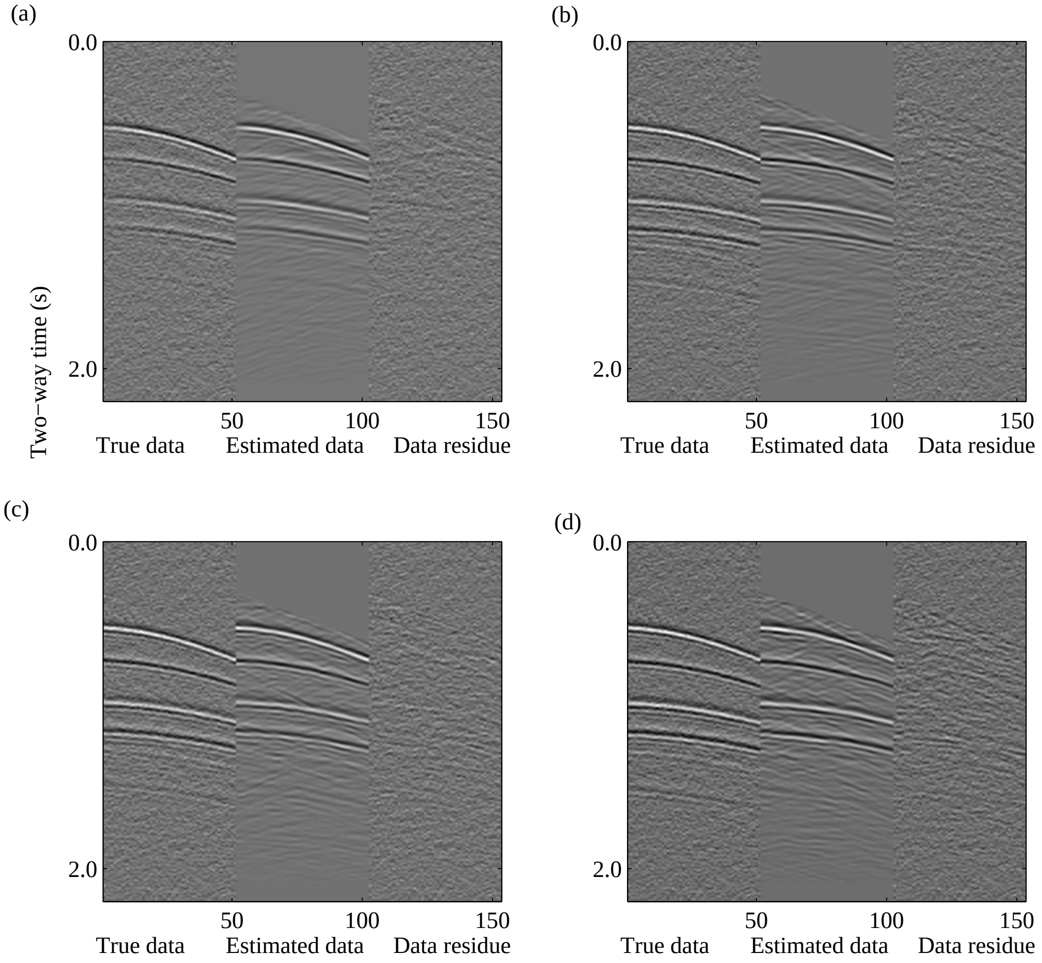

Figure 7.23: True data, estimated data and data residue. (a) Shot location is at the origin (0m,0m). (b) Shot location is at 300m from the origin. (c) Shot location is at 600m from the origin. (d) Shot location is at 900m from the origin. True data contain bandlimited noise such that $SNR = 10$. Search bound of the model space is ±15%.

of ±25% with noisy data such that $SNR = 20$.

Figure 7.27 shows the true data with $SNR = 10$, estimated data and the data residue after 400 model preconditioning based VFSA iterations when the search bound is set to ±25% of the true model values. Figure 7.28 shows the estimated model after 400 model preconditioning based VFSA iterations. It is observed that the algorithm successfully estimated the model within a search bound of ±25% when shot gathers are noisy such that $SNR = 10$.

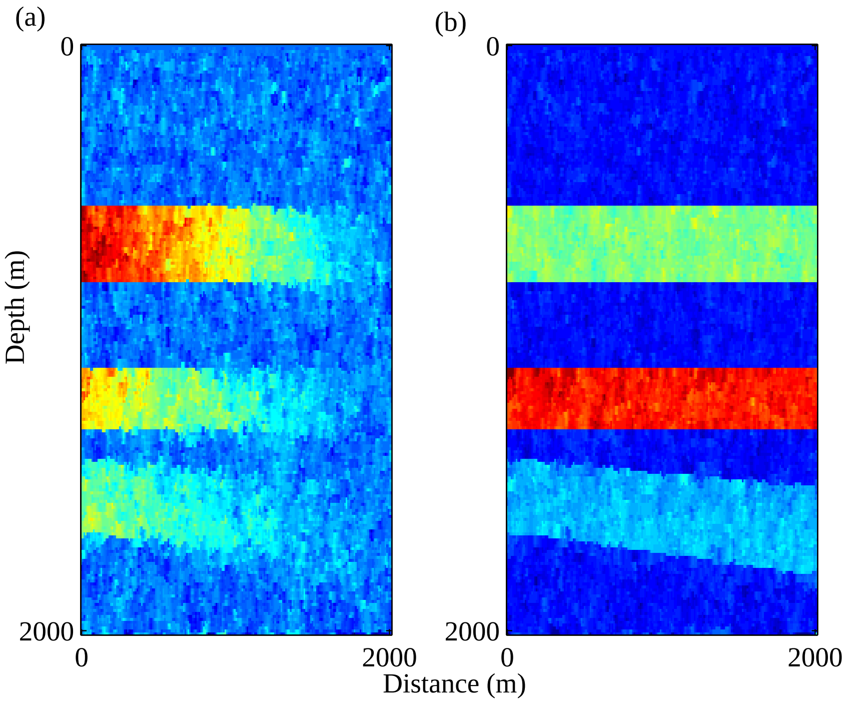

Figure 7.24: Estimated model after 400 model preconditioning based VFSA iterations. (a) Estimated velocity and (b) estimated density. True data contain bandlimited noise such that $SNR = 10$. Search bound of the model space is ±15%.

The results of the algorithm obtained from data contaminated with various degree of noise show that the algorithm is stable over a reasonable noise level in the data. The algorithm operating within the search bounds of ±15% and ±25% successfully recovered the model within reasonable accuracy in 400 model preconditioning based VFSA iterations.

7.6 Summary

In the previous chapter, I applied the algorithm based on model preconditioning and very fast simulated annealing algorithm on a one-dimensional model space to estimate the earth elastic parameters from AVO data. I had shown that the algorithm brings in the possibility of finding applications in very large dimensional model space without being largely affected by the adverse effects caused by the large model dimension. In this chapter, I have shown an application of model preconditioning based global optimization in estimating the subsurface velocity and density on a two

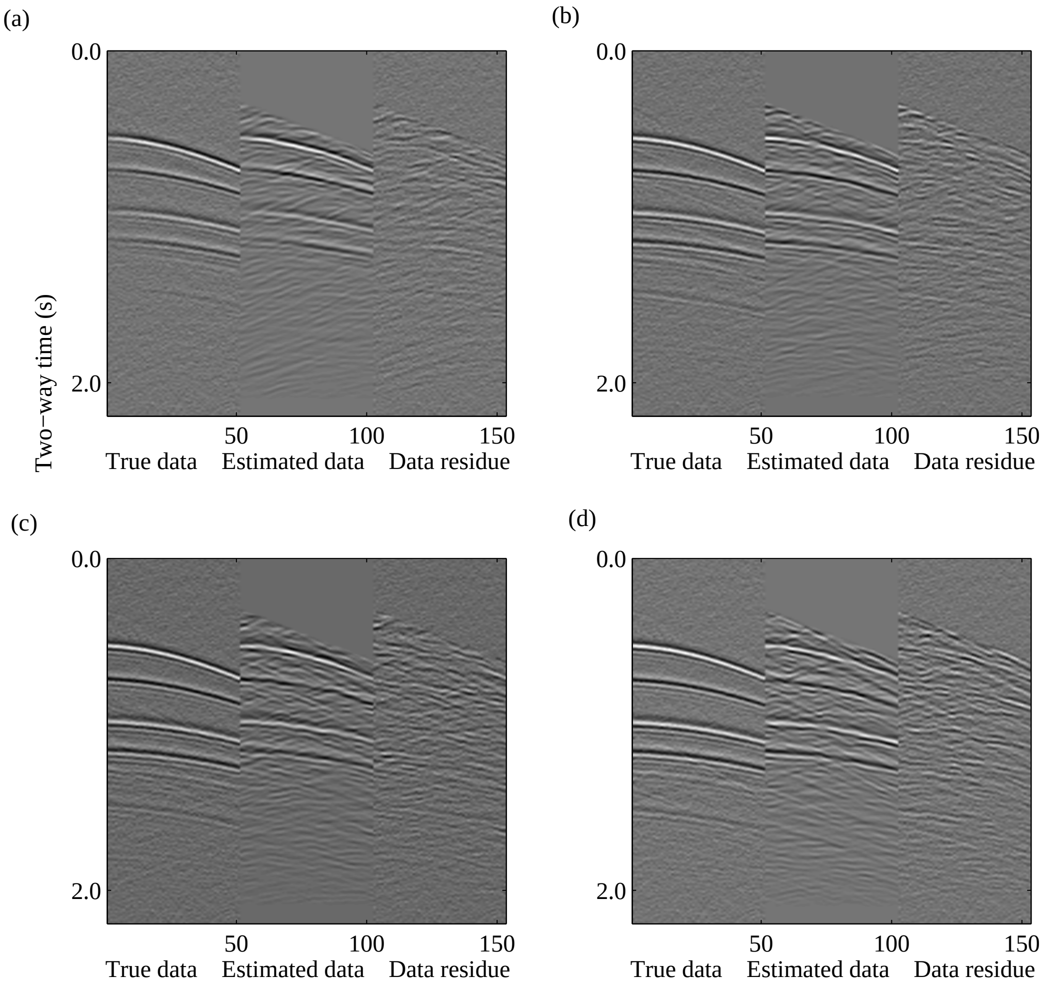

Figure 7.25: True data, estimated data and data residue. (a) Shot location is at the origin (0m,0m). (b) Shot location is at 300m from the origin. (c) Shot location is at 600m from the origin. (d) Shot location is at 900m from the origin. True data contain bandlimited noise such that $SNR = 20$. Search bound of the model space is ±25%.

dimensional grid. The estimation is obtained from shot gathers encompassing the entire model grid. In the examples discussed above, I have used four shot gathers to cover the entire model grid. The shot gathers are computed by solving second-order finite difference acoustic wave equation. The shot-receiver geometry is chosen in such a way that the four shot gathers encompassed the entire model grid. Model preconditioning based VFSA algorithm is used to optimize for the velocity and density parameters over a grid of size 201 by 201 cells each. This is a very large model

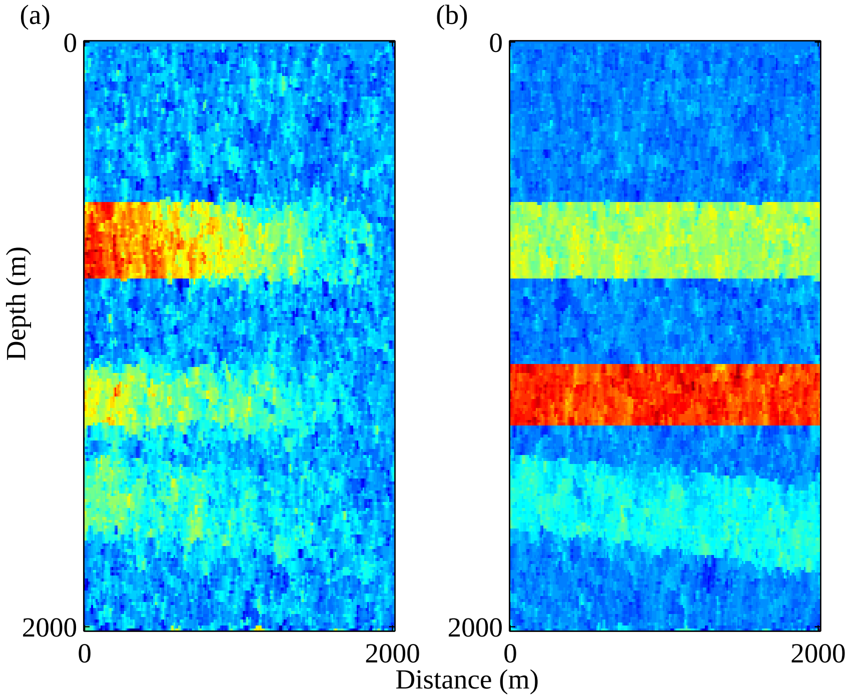

Figure 7.26: Estimated model after 400 model preconditioning based VFSA iterations. (a) Estimated velocity and (b) estimated density. True data contain bandlimited noise such that $SNR = 20$. Search bound of the model space is ±25%.

space for the optimization algorithm to work efficiently. I have incorporated the *a priori* information, namely, smoothness along the lateral direction and blockyness along the vertical direction of the model space in the model preconditioning based VFSA scheme. The results show that the algorithm successfully estimates the velocity and density parameters via full waveform inversion of the four shot gathers. I have shown that the algorithm is stable over a reasonable noise level in the data. It is a known fact that as opposed to gradient based optimization algorithms, the global optimization algorithms provide means to escape from a local minimum. However, for all practical applications of optimization algorithms, the cost of computation is a primary concern. This requires that the global optimization schemes be provided with a reasonable and justified search bounds so that the computational cost can be significantly minimized. I have shown that the model preconditioning based VFSA algorithm successfully estimated the unknown model parameters within the search bounds of ±15% and ±25% of the true model values. I would like to emphasize that the examples shown in this chapter are very simple compared to the real earth situations. However, the examples serve quite well the primary aim to show

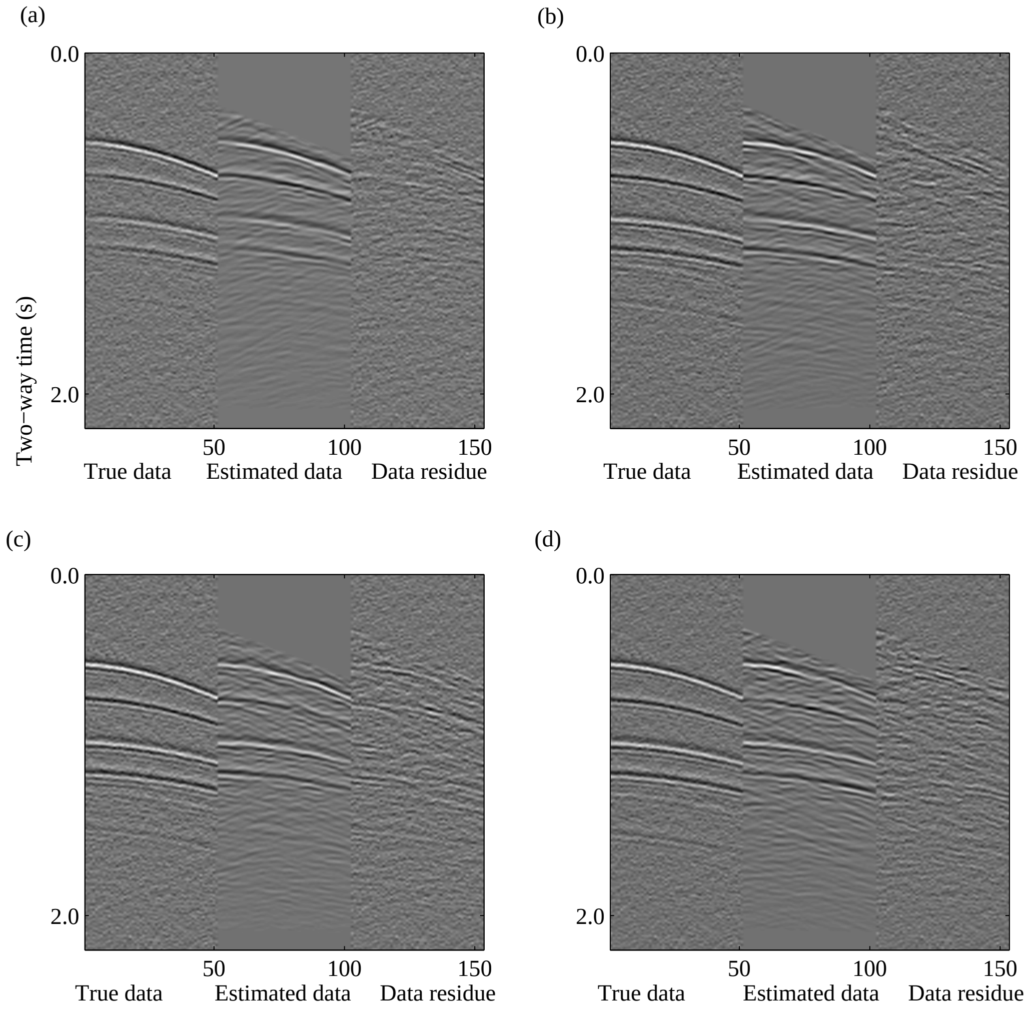

Figure 7.27: True data, estimated data and data residue. (a) Shot location is at the origin (0m,0m). (b) Shot location is at 300m from the origin. (c) Shot location is at 600m from the origin. (d) Shot location is at 900m from the origin. True data contain bandlimited noise such that $SNR = 10$. Search bound of the model space is ±25%.

the practical viability of global optimization schemes in a large dimensional model space. Further research work in this direction would possibly make the benefits of global optimization schemes more accessible in day-to-day optimization problems involving complex real earth situations.

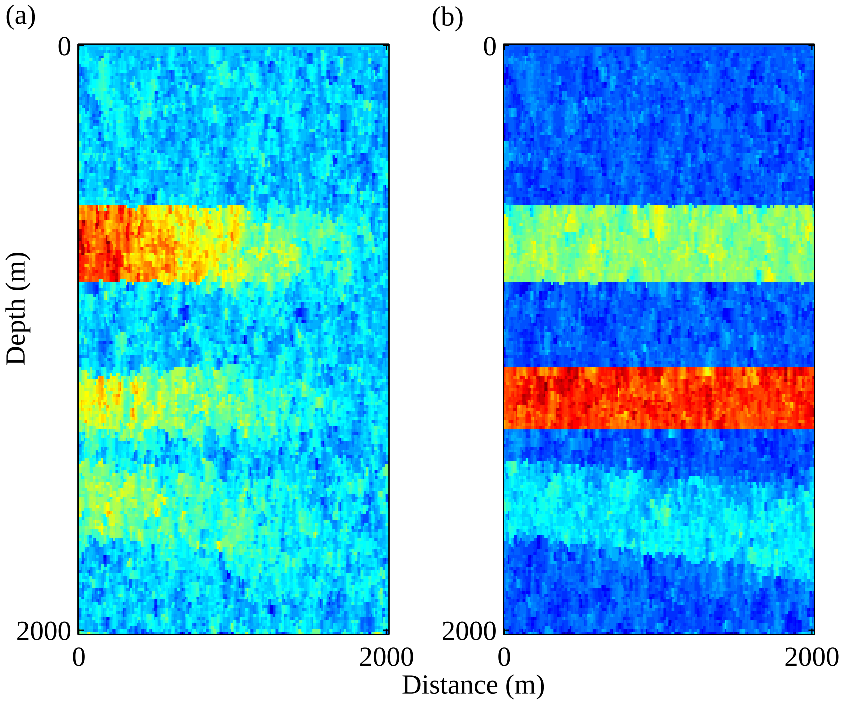

Figure 7.28: Estimated model after 400 model preconditioning based VFSA iterations. (a) Estimated velocity and (b) estimated density. True data contain bandlimited noise such that $SNR = 10$. Search bound of the model space is ±25%.

Chapter 8

Discussion and conclusions

It is nature's prerogative to conceal from us what we want to know. Be it the knowledge of density of the ore body, subsurface conductivity or reservoir volume, we have no direct access to such information. Our quest for knowledge about these parameters is limited to acquisition of data through some experiments. However, data do not provide any direct knowledge about what we want to know. If we want to know the subsurface ore body density then we perform an experiment to acquire gravity data. Likewise, we want to know about the subsurface conductivity or reservoir volume, we perform experiments to acquire resistivity or seismic data. In the examples above, the model is the unknown (density, conductivity etc.), thus the end product of inversion and data serve as the means to the end.

Inversion can be direct or indirect. Direct inversion involves estimation of the model from the data when there exists direct analytical expressions to estimate the model. Geophysical inversion is highly complex and hence more often than not, there exists no direct analytical expressions to estimate the model. Thus, estimation of model parameters is obtained through indirect means. Inversion is posed as an optimization problem and the solution is represented by the model that best optimizes a user defined cost function. A cost function is defined in terms of a data misfit term and regularization term. Data misfit term measures how closely an estimated model fits the data. The regularization term incorporates *a priori* knowledge about the model. Thus the regularization term constrains the optimization to a solution that not only honors the data but also is consistent with what is known about the solution *a priori*. The optimization problem becomes simple when there exists linear or linearizable relationship between the model and the data. In case of a linear or linearizable inverse problems, the cost function can be chosen to be quadratic, hence making it unimodal. Local optimization techniques based on gradient computation are the most elegant means to obtain the minimum of a quadratic cost function. However, when the data and model are related by nonlinear equations, the cost function is no more quadratic. In such a situation, the cost function topology is complex and likely to contain more than one minima. Gradient based optimization techniques always proceed downhill and hence the optimization is local. Depending on the initial model, the convergence invariably proceeds to the nearest minimum. If the nearest minimum is not the global minimum then the obtained solution is erroneous. On the other hand, optimization methods that are based on exhaustive search techniques provide the means to "jump out" of a local minimum by allowing the optimization algorithm to climb uphill with a finite probability. Such methods have the potential to find the global minimum and hence referred to as the global optimization methods. Exhaustive search technique, also known as the Monte Carlo algorithm, in spite of it's ability to find the global minimum, is highly inefficient. The algorithm becomes impracticably slow when the model dimension grows and the rate of convergence suffers drastically. Methods such as the simulated annealing and genetic algorithms were proposed to alleviate the problem of slower convergence as observed in the Monte Carlo algorithm while preserving the benefits of exhaustive search and possible convergence to the global minimum. Further modifications to the classical simulated annealing were proposed in later times to achieve still faster convergence to the global minimum. However, despite the modifications, the global optimization algorithms continued to suffer from slow convergence as the model dimension grew. Such a shortcoming in global optimization algorithms is generally referred to as the "curse of dimensionality". Geophysical inverse problems are highly complex and involve a very large model space, thus limiting the scope for the application of global optimization algorithms. Main thrust of my thesis is to develop an approach to alleviate the issues associated with large model dimension and make global optimization algorithms a viable means to obtain accurate solutions in highly complex and nonlinear inverse problems in applied seismology.

My thesis can be broadly classified in two parts. In the first part, I am applying classical simulated annealing algorithm in a blind deconvolution problem where the optimization involves estimation of a mixed phase wavelet from seismic data. This problem is of particular interest because the global optimization algorithm tries to solve one equation with two unknowns. I am proposing a new approach based on global optimization algorithm to estimate the coefficients of an all-pass operator.

The estimated all-pass operators are such that the phase of an estimated minimum phase wavelet can be optimally varied to obtain a mixed phase wavelet that more accurately represented the data. The optimization is carried over by minimizing a cost function defined as the L_2-norm between the fourth-order cumulant of the data whitened by the removal of the estimated minimum phase wavelet and the fourth-order moment of an all-pass operator. The results show that the proposed global optimization algorithm effectively solves a blind deconvolution problem.

In the second part of my thesis, I am proposing a new technique where the *a priori* information are suitably incorporated into the global optimization algorithm so that the problem of slower convergence due to large model dimension is minimized. In this part of my thesis, I discussed important *a priori* constraints such as the sparseness and blockyness constraints. I showed an example where sparseness constraint is imposed in a deconvolution problem to obtain a high resolution solution through the Iterative Re-weighted Least-Squares (IRLS) approach. In the subsequent chapter, I am solving an AVO inversion problem by means of NonLinear Conjugate Gradient (NLCG) method. I have divided the problem into two parts, namely, (a) forward operator is based on Aki-Richards approximation of the Zoeppritz equations and (b) forward operator is based on the reflectivity method. In the former case, the forward operator though linearized, has a degree of nonlinearity because the model (P-wave velocity, S-wave velocity and density) formed a part of the forward operator. The Aki-Richards equation based forward operator simulated normal move-out corrected data with no multiples. In the latter case, the problem is highly nonlinear. In addition to the inherent nonlinearity in the reflectivity method, the presence of multiples and normal move-out in the data added to the nonlinearity of the problem. In both the cases, I have imposed blockyness constraint on the model space. The results show that accurate solution by means of local optimization algorithm can be obtained when the degree of nonlinearity in the problem is minimal. However, when the problem became highly nonlinear, the local optimization technique failed. Through this example I showed a situation that necessitated the use of global optimization scheme in order to obtain accurate results. In the subsequent chapter, I am solving a nonlinear and complex AVO inversion problem by means of global optimization algorithm. The model space comprised a large number of thin pseudo-layers and each layer contained three unknowns, namely, P-wave velocity, S-wave velocity and density. Such a parameterization of the model space resulted in a large model domain for a global optimization algorithm. I have used Very Fast Simulated Annealing (VFSA) algorithm as the tool to achieve the global optimization. I have incorporated blockyness constraint in the optimization algorithm via nonlinear operators. The nonlinear operators, referred to as the Edge Preserving Smoothing (EPS) operators, act on the model space as a second stage to the VFSA algorithm. Incorporation of *a priori* information as a second stage of the VFSA algorithm resulted in preconditioning the model space in a favorable way. As opposed to relying on completely random perturbations, the proposed approach helped the VFSA algorithm to operate in a favorably biased model domain. Results indicate that the proposed algorithm provides faster convergence and solutions that honor the data and consistent with the *a priori* information.

Further, I carried forward the approach of model preconditioning based global optimization and applied the proposed algorithm on a very large dimensional problem. I optimized for velocity and density over a two dimensional grid via waveform inversion of a number of shot gathers encompassing the entire model space. The example, though trivial from the point of view of seismic data analysis, is highly complex from the point of view of global optimization. The complexity of the optimization problem lies in the fact that the model space is enormous. I have incorporated model preconditioning operators in both the lateral and vertical direction of the model grid. I preconditioned the model space with a smoothing operator in the lateral direction and edge preserving smoothing operator in the vertical direction. As mentioned before, the model preconditioning was applied as a second stage to the classical VFSA algorithm. The algorithm was tested on synthetically generated data. The results show that good convergence to a solution consistent with the *a priori* information and data could be achieved within a reasonable number of forward model evaluations. The problem discussed is simple but serves as a good example to test the proposed global optimization algorithm when the model dimension becomes very large. I have shown by the help of this example that model preconditioning based global optimization schemes can favorably bias the model space according to the *a priori* information to achieve faster convergence without jeopardizing the benefits of exhaustive search.

Selection of very fast simulated annealing algorithm to test the model preconditioning based global optimization was simply a matter of personal choice. I believe that extending the proposed approach of augmenting the global optimization algorithm iterations with the model preconditioning operators to other global optimization algorithms, such as the genetic algorithm, will be straightforward. In the last example, I have shown a successful application of the proposed algorithm on a very large dimensional problem. Though the problem addressed in the example was trivial from the point of view of seismic inversion, it's importance can not be undermined with regard to global optimization. I expect that further research can lead the proposed algorithm to achieve better convergence and accurate solutions in very large dimensional, highly complex and more realistic optimization problems.

Bibliography

Aki, K., and Richards, P. G., 1980, Quantitative Seismology: Theory and Methods, W.H. Freeman and Co.

AlBinHassan, N. M., Luo, Y., and Al-Faraj, M. N., 2006, 3D edge-preserving smoothing and applications, Geophysics, **71**, P5-P11.

Alford, R. M., Kelley, K. R., and Boore, D. M., 1974, Accuracy of finite-difference modeling of the acoustic wave equation, Geophysics, **39**, 834-842.

Barone, P., 1999, Fast deconvolution by a two-step method, SIAM Journal of Sci. Comp., **21**, 883-899.

Bortfeld, R., 1961, Approximation to the reflection coefficients of plane longitudinal and transverse waves, Geophysical Prospecting, **9**, 485-502.

Buland, A., and Omre, H., 2003, Bayesian linearized AVO inversion, Geophysics, **68**, 185-198.

Charbonnier, P., Blanc-Feraud, L., Aubert, G., and Barlaud, M., 1997, Deterministic edge-preserving regularization in computed imaging, IEEE Transaction on Image Processing, **6**, 298-311.

Claerbout, J. F., 1992, Earth sounding analysis. Processing versus inversion, Blackwell Scientific Publications.

Creutz, M., 1984, Quarks, gluons and lattices. Cambridge University Press.

Fatti, J. L., Smith, G. C., Vail, P. J., Strauss, P. J., and Levitt, P. R., 1994, Detection of gas in sandstone reservoirs using AVO analysis: 3D seismic case history using the geostack technique, Geophysics, **59**, 1362-1376.

Fuchs, K, and Muller, G., 1971, Computation of synthetic seismograms with the reflectivity method and comparison with observations, Geophysical Journal of Royal Astronomical Society, **23**, 417-433.

Geman, S., and Geman, D., 1984, Stochastic relaxations, Gibbs distribution and Bayesian restoration of images, IEEE transactions on pattern analysis and machine intelligence PAMI-6, 721-741.

Goldberg, D., 1989, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley Publishing Company, Inc.

Greene, J. W., and Supowit, K. J., 1986, Simulated annealing without rejected moves, IEEE transactions on Computer-Aided Design CAD-5, **1**, 221-228.

Hargreaves, N., 1994, Wavelet estimation via fourth-order cumulants, 64th SEG Meeting, Los Angeles, USA, Expanded abstracts, 1588-1590.

Hestenes, M., and Steifel, E., 1952, Methods of conjugate gradients for solving linear systems, National Bureau of Standards Journal Research, **49**, 403-436.

Holland, J. H., 1975, Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, Michigan.

Ingber, L., 1989, Very fast simulated reannealing, Mathematical and computer modeling, **12**, 8, 967-993.

Jervis, M. A., Stoffa, P. L., and Sen, M. K., 1993, 2-D migration velocity estimation using a genetic algorithm, Geophysical Research Letters, **20**, 1495-1498.

Jervis, M. A., Sen, M. K., and Stoffa, P. L., 1996, Prestack migration velocity estimation using nonlinear methods, Geophysics, **60**, 138-150.

Jin, S., and Madiriaga, R., 1994, Nonlinear velocity inversion by a two-step Monte Carlo method, Geophysics, **59**, 577-590.

Kelly, K. R., Ward, R. W., Treitel, S., and Alford, R. M., 1976, Synthetic seismograms: A finite difference approach, Geophysics, **41**, 2-27.

Kirkpatrick, S., Delatt, C. D. (Jr), and Vecchi, M. P., 1983, Optimization by simulated annealing, Science, **220**, 671-680.

Knott, C., 1899, Reflection and refraction of elastic waves with seismological applications, Philosophical Magazine, **48**, 64-97.

Koefoed, O., 1955, On the effects of Poisson's ratio of rock strata on the reflection coefficients of plane wave, Geophysical Prospecting, **3**, 381-387.

Lazear, G. D., 1993, Mixed-phase wavelet estimation using fourth-order cumulants, Geophysics, **58**, 1042-1051.

Liang, G., Cai, X., and Li, Q., 2002, Using high-order cumulants to extrapolate spatially variant seismic wavelets, Geophysics, **67**, 1869-1876.

Lines, L. R., and Treitel, S., 1984, A review of least-squares inversion and its application to geophysical problems, Geophysical Prospecting, **32**, 159-186.

Lörtzer, G. J. M., and Berkhout, A. J., 1993, Linearized AVO inversion of multicomponent seismic data, *in* Castagna, J., and Backus, M., Eds., Offset-dependent reflectivity- Theory and practice of AVO analysis: Society of Exploration Geophysicists, 317-332.

Ma, X, Q., 2001, A constrained global inversion method using an overparameterized scheme: Application to poststack seismic data, Geophysics, **66**, 613-626.

Ma, X, Q., 2002, Simultaneous inversion of prestack seismic data for rock properties using simulated annealing, Geophysics, **67**, 1877-1885.

Mallick, S., 1995, Model-based inversion of amplitude-variations-with-offset data using a genetic algorithm, Geophysics, **60**, 939-954.

Marquardt, D. T., 1963, An algorithm for least squares estimation of non-linear parameters, Journal of the Society of Industrial and Applied Mathematics, **11**, 431-441.

McAulay, A. D., 1985, Prestack inversion with plane-layer point source modeling, Geophysics, **50**, 77-89.

Mendel, J. M., 1991, Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications, Proceedings of the IEEE, **79**, 278-305.

Menke, W., 1984, Geophysical data analysis: Discrete inverse theory, Academic Press, Inc.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., 1953, Equation of state calculations by fast computing machines, The Journal of Chemical Physics, **21**, 1087-1092.

Misra, S., and Sacchi, M. D., 2007a, Non-minimum phase wavelet estimation by non-linear optimization of all-pass operators, Geophysical Prospecting, **55**, 223-234.

Misra, S., and Sacchi, M. D., 2007b, Nonlinear One dimensional prestack seismic inversion with edge preserving smoothing filter, Extended abstract, EAGE 69^{th} conference and exhibition, London, United Kingdom,

Misra, S., and Sacchi, M. D., 2008a, Global optimization with model space preconditioning: Application to AVO inversion, Geophysics, **73**, R71-R82.

Misra, S., and Sacchi, M. D., 2008b, Model preconditioning based global optimization: Application to prestack migration velocity analysis, Extended abstract, 2008 CSPG CSEG CWLS Convention, Calgary, Canada, 330-333.

Molina, R., Nunez, J., Cortijo, F. J., and Mateos, J., 2001, Image restoration in Astronomy: A Bayesian perspective, IEEE Signal Proc.Mag., **18**, 11-29.

Mosegaard, K., and Vestergaard, P. D., 1991, A simulated annealing approach to seismic model optimization with sparse prior information, Geophysical Prospecting, **39**, 599-611.

Nocedal, J., and Wright, S. J., 1999, Numerical Optimization, Springer.

Officer, C. B., 1958, Introduction to the theory of sound transmission with application to the ocean, McGraw-Hill Book Co.

Oppenheim, A. V., Schafer, R. W., and Stockham, T. G., 1968, Nonlinear filtering of multiplied and convolved signals, Proceedings of the IEEE, **65**, 1264-1291.

Peacock, K. L., and Treitel, S., 1969, Predictive deconvolution: Theory and practice, Geophysics, **34**, 155-169.

Peterson, C., and Anderson, J. R., 1987, A mean field theory learning algorithm for neural networks, Complex Systems, **1**, 995-1019.

Peterson, C., and Anderson, J. R., 1988, Neural networks and NP-complete optimization problems; A performance study on the graph bisection problem, Complex System, **2**, 59-89.

Peterson, C., and Soderberg, B., 1989, A new method for mapping optimization problems onto neural networks, International Journal of Neural Sciences, **1**, 3-22.

Porsani, M. J., and Ursin, B., 1998, Mixed phase deconvolution, Geophysics, **63**, 637-647.

Porsani, M. J., and Ursin, B., 2000, Estimation of an optimal mixed phase inverse filter, Geophysical Prospecting, **48**, 663-676.

Portniaguine, O. and Zhdanov, M. S., 1999, Focussing geophysical inversion images, Geophysics, **64**, 874-887.

Press, F., 1968, Earth models obtained by Monte-Carlo inversion, Journal of Geophysical Research, **73**, 5223-5234.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 2001, Numerical recipes in Fortran 77: The art of scientific computing, Second Edition, Cambridge University Press.

Rao, S. S., 1996, Engineering optimization- Theory and Practice, John Wiley & Sons, Inc.

Rebbi, C., 1984, Monte Carlo calculations in lattice gauge theories, Applications of the Monte Carlo method, 277-298, Springer-Verlag Inc.

Robinson, E. A., 1967, Predictive deconvolution of time series with application to seismic exploration, Geophysics, **32**, 418-484.

Robinson, E. A., and Treitel, S., 1980, Geophysical signal analysis, Prentice Hall, Inc.

Rosa, A. L. R., and Ulrych, T. J., 1991, Processing via spectral modeling, Geophysics, **56**, 1244-1251.

Rothman, D, H., 1986, Automatic estimation of large residual static corrections, Geophysics, **51**, 337-346.

Rüger, A., 2002, Reflection coefficients and azimuthal AVO analysis in anisotropic media, SEG geophysical monograph series number 10, Society of Exploration Geophysicists.

Ryden, N., and Park, C., 2006, Fast simulated annealing inversion of surface waves on pavement using phase-velocity spectra, Geophysics, **56**, 1624-1638.

Sacchi, M. D., 1997, Reweighting strategies in seismic deconvolution, Geophysical Journal International, **129**, 651-656.

Saggaf, M. M., and Robinson, E. A., 2000, A unified framework for the deconvolution of traces of nonwhite reflectivity, Geophysics, **65**, 1660-1676.

Sambridge, M., 1999a, Geophysical inversion with a Neighbourhood Algorithm I: Searching a parameter space, Geophysical Journal International, **138**, 479-494.

Sambridge, M., 1999b, Geophysical inversion with a Neighbourhood Algorithm II: Appraising the ensemble, Geophysical Journal International, **138**, 727-746.

Sen, M. and Stoffa, P. L., 1991, Nonlinear one-dimensional seismic waveform inversion using simulated annealing, Geophysics, **56**, 1624-1638.

Sen, M. and Stoffa, P. L., 1992, Multilayer AVO inversion by genetic algorithms, 62^{nd} annual international meeting and exposition, Society of Exploration Geophysicists.

Sen, M. and Stoffa, P. L., 1995, Global optimization methods in geophysical inversion, Elsevier Science Publishing Co.

Shearer, P. M., 1999, Introduction to seismology, Cambridge University Press.

Shuey, T., 1985, A simplification of the Zoeppritz equations, Geophysics, **50**, 609-614.

Smith, G., and Gidlow, P., 1987, Weighted stacking for rock property estimation and detection of gas, Geophysical Prospecting, **35**, 993-1014.

Szu, H., and Hartley, 1987, Fast simulated annealing, Physics Letters. A., **122**, 157-162.

Tarantola, A., 1987, Inverse problem theory: Methods for data fitting and model parameter estimation, Science Publishing Co.

Tugnait, J. K., 1987, Identification of linear stochastic systems via second- and fourth-order cumulant matching, IEEE Transactions on information theory IT-33, 393-407.

Ulrych, T. J., 1971, Application of homomorphic deconvolution to seismology, Geophysics, **36**, 650-660.

Ulrych, T. J., Velis, D. R., and Sacchi, M. D., 1995, Wavelet estimation revisited, The Leading Edge, **14**, 1139-1143.

Valencino, A. A., Brown, M., Guitton, A., and Sacchi, M. D., 2004, Interval velocity estimation using edge-preserving regularization, SEG Expanded Abstract, **23**, 2431-2434.

Varela, O. J., Torres-Verdin, C., and Sen, M. K., 2006, Enforcing smoothness and assessing uncertainty in non-linear one dimensional prestack seismic inversion, Geophysical Prospecting, **54**, 239-259.

Velis., D. R., and Ulrych, T. J., 1996, Simulated annealing wavelet estimation via fourth-order cumulant matching, Geophysics, **61**, 1939-1948.

Vestergaard, P. D., and Mosegaard, K., 1991, Inversion of post-stack seismic data using simulated annealing, Geophysical Prospecting, **39**, 613-624.

Walden, A. T., 1985, Non-Gaussian reflectivity, entropy, and deconvolution, Geophyscics, **50**, 2862-2888.

Walden, A. T., and Hosken, J. W. J., 1986, The nature of non-Gaussianity of primary reflection coefficients and its significance for deconvolution, Geophysical Prospecting, **34**, 1038-1066.

Walden, A. T., and Nunn, K. R., 1988, Correcting for coloured primary reflectivity in deconvolution, Geophysical Prospecting, **36**, 282-297.

Wang, Y., 1999, Approximations to the Zoeppritz equations and their use in AVO analysis, Geophysics, **64**, 1920-1927.

White, R. E., 1984, Signal and noise estimation from seismic reflection data using spectral coherence methods, Proceedings of the IEEE, **72**, 1340-1356.

White, R. E., 1988, Maximum kurtosis phase correction, Geophysical Journal, **95**, 371-389.

White, R. E., and Simm, R., 2003, The importance of bandwidth in seismic cross-equalization- well tie and other examples, 65th EAGE Conference, Stavanger, Norway, Extended abstracts, P071.

Wiggins, R. A., 1969, Monte Carlo inversion of body-wave observations, Journal of Geophysical Research, **74**, 3171-3181.

Wiggins, R. A., 1978, Minimum entropy deconvolution, Geoexploration, **16**, 21-35.

Youzwishen, C. F., 2001, Non-linear sparse and blocky constraints for seismic inverse problems, M. Sc. Thesis, University of Alberta, Canada.

Zoeppritz, K., 1919, On the reflection and penetration of seismic wave through unstable layers, Göttinger Nachrichten, **1**, 66-84.