RICE UNIVERSITY

Joint Inversion Using the Convolutional Model

by

Nathan W. Winslow

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Master of Arts

Approved, Thesis Committee:

William W. Symes, Chairman
Professor of Computational and Applied Mathematics

Liliana Borcea
Assistant Professor of Computational and Applied Mathematics

Colin A. Zelt
Assistant Professor of Geology and Geophysics

Houston, Texas

April, 2000

Abstract

Joint Inversion Using the Convolutional Model

by

Nathan W. Winslow

A detailed imaging of an acoustic medium via a seismic experiment requires an accurate representation of the source. A joint source and reflectivity inversion may provide the means to obtain the desired detail. Joint inversion using the acoustic wave equation is computationally expensive. The convolutional model of the seismogram in the offset-time domain provides a computationally cheaper means of approximating the wave equation.

We rigorously derive the convolutional model of the seismogram as an approximation to the linearized acoustic wave equation where the medium to be imaged is layered and has a constant density. Using the Hilbert Class Library (a mathematical C++ library that among other things provides access to optimization algorithms), we implement the convolutional model and its inverse. Using linear and non-linear optimization methods applied to a least squares data residual objective function we test the robustness of inversion results from the convolutional model in the offset-time domain.

Based on these experiments we make the following statements:

- All optimization algorithms tested successfully reduce the relative error in the data residuals within 5%. I.e.

  $$\frac{||b_{sim} - b_{resim}||}{||b_{sim}||} < .05$$

  where b_{sim} is data simulated by running a predetermined source and reflectivity through the forward map, and b_{resim} is data obtained by running a source and reflectivity from the inversion through the forward map.

- The trust region and limited memory BFGS algorithms are significantly faster at reducing the relative error in the data residual to below 5%

- The shape of the source and reflectivity obtained from reducing the relative error in the data residuals below 5% can be markedly different from the shape of the source and reflectivity used to generate the simulated data.

- When the source is oscillatory and the reflectivity has small support, the objective function can be reduced below 10^{-6}, but the source and reflectivity that is output from the inversion can have markedly different shapes compared to the shapes of the source and reflectivity used to generate the simulated data.

- The above apparent non-uniqueness does not violate the theoretical results of Minkoff and Symes [11]

In addition, numerical tests indicate that the trust region and limited memory BFGS approaches to optimizing the objective function are both faster than alternation and various subspace techniques.

Acknowledgments

This is my second Master's degree. The first was in Geology at the University of Michigan. Changing academic areas is an arduous task. I owe thanks to many people for helping me negotiate that change. Before I mention those people, however, I wish to thank those who encouraged me to make the change. They pointed me in a direction which has brought, and I'm sure will continue to bring, much fulfillment to my life. Dr. Mark Gockenbach gave me a taste of what Applied Mathematics is about in a course at the University of Michigan. He later convinced me that I could actually get an advanced degree in Applied Math, pointed me toward my current advisor, and made a few instrumental phone calls on my behalf. Mark I owe you big time.

Dr. Richard Nolen-Hoeksema was also instrumental in my decision to move to Rice. I wasted many hours of Richard's time discussing the pros and cons of moving from Michigan to Rice. I am indebted to Richard both for his patience and his clear foresight as to what would be best for me intellectually and academically. I also owe Richard about 20 large cups of coffee.

There were many people who helped me change fields from Geology to Applied Mathematics. The big three include my advisor, his sabbatical replacement, and my office mate. Dr. Jianliang Qian, my officemate was important for my survival as a first year Applied Mathematician. Without his help Analysis would have pounded me to an intellectual pulp.

Dr. Mark Gockenbach (my advisor's sabbatical replacement) saved me in my second year. I'm still amazed by the many hours I spent in his office working on homework problems for his classes and others. Mark's talents as a teacher and mentor are among the best that I've seen in over 10 years of higher education. Without his help, I would have been lost. At the risk of sounding redundant, Mark I owe you big time.

Dr. Bill Symes is without a doubt the best advisor I could ever hope to work with. I will always have fond memories of the many hours spent with Bill laughing, deriving, arguing, discovering, planning, compiling and debugging. Leaving Rice and going to work for anybody other than Bill is going to be a great disappointment.

I also wish to thank all the excellent teachers here at the CAAM department. In particular, I want to recognize the teaching talents of Drs. Borcea, Cox, Heinkenschloss, Sorenson and Symes. Their skillful instruction often kept me motivated to push a little bit harder, learn a little bit more and still enjoy the process.

Finally, a big thanks to my family. Heidi, Mom, Dad, Frank, Char, Wendi, and of course Cousin Dave. Thank you for your empathy, intellectual capabilities, curiosity and support.

Contents

Illustrations

Tables

Chapter 1

Introduction

1.1 Overview

Recent research suggests that the source estimate can affect the earth parameter estimates output from seismic inversion [18, 10]. This is due to the fact that model seismograms (generated with model sources) are compared to real seismograms. If the model seismograms are incorrect (due to incorrect sources), the earth parameters may be incorrect as well. All aspects of the source (wave form, strength, and radiation pattern) may influence the success of inversion. The sources used in actual data collection are anisotropic and have a notable radiation pattern [18, 10].

Given that proper inclusion of the source into an imaging algorithm is a worthy goal, another question immediately arises: Is it possible to invert for both the source and earth parameters? In other words, do the source and reflectivity affect the seismogram in unique ways: when including source inversion within a reflectivity inversion are we really obtaining the source and reflectivity or simply over fitting the data? Can the source and reflectivity destructively combine to hide wiggles in the seismograms?

We can partly answer this question via an examination of the convolutional model of the seismogram in an acoustic layered medium

$$b(x,t) = f(t) *_t r(\phi(x,t))$$

where $b(x,t)$ is the set of seismograms, $f(t)$ is the source, $*_t$ is a convolution in time, $r(z)$ is the reflectivity as a function of depth and $z = \phi(x,t)$ is a change of variables that converts depth to horizontal offset and time. The reflectivity has the following form

$$r(z) = \frac{\partial}{\partial z}\left[\frac{\delta c(z)}{c(z)}\right]$$

where $c(z)$ is a slowly varying background P wave velocity, and $\delta c(z)$ represents perturbations to the background velocity.

This equation of the seismogram shows the scale ambiguity inherent in the convolutional model. In other words doubling the amplitude of the source and halving the amplitude of the reflectivity generates no net change in the seismogram. The scale ambiguity is a degree of freedom which cannot be directly constrained. Assuming that the reflecivty is white and that the source is minimum phase allows for a lessening of its ambiguity, but that is not the subject of this paper. For the sake of argument, we acknowledge that the scale ambiguity exists and constrain the size of the source as follows.

$$\|f(t)\|_2 = \int_{-\infty}^{\infty} f^2(t)dt = 1.$$

This allows us to proceed in answering the question of uniqueness. Specifically, do the source and reflectivity have different effects on the seismogram?

This question was answered to varying degrees by previous authors. Specifically, a quasi-impulsive, isotropic point source can be uniquely recovered from reflection seismograms using linear perturbation approximations [1]. In an acoustic constant density, constant background velocity medium, perturbations in the plane wave seismograms uniquely determine perturbations in the source [9]. In addition, if the material velocity does not change too rapidly the source can be uniquely determined within a given pass band [11]. Minkoff and Symes also showed that as the slowness aperture is widened the extent to which the reflectivity and source can be separately determined improves. Further, if the slowness aperture is closed (there exists only one seismogram) there is no chance of recovering the source and reflectivity.

The proofs of the above statements are all fairly involved. They are not repeated in this thesis. Instead, this thesis focuses on giving some legitimacy of the convolutional model as an approximation to the wave equation. This is done via a rigorous derivation of the convolutional model from a linearization of the acoustic wave equation applied to a layered medium.

In addition, aspects of the numerical implementation of the convolutional model and methods of inverting the convolutional model are discussed. We invert for the source and reflectivity using linear and non-linear optimization techniques applied to a data residual least squares objective function.

$$\min_{r(z)\in\Re^m, f(t)\in\Re^n} \|b(x,t) - f(t) * r(\phi(x,t))\|_2^2$$

where m and n are the number of points in the discretization of $r(z)$ and $f(t)$ respectively, $b(x,t)$ is a data set, and $\phi(x,t)$ is a change of variables function that maps offset and time to depth.

The optimization can be treated in two different ways. The first option (using an alternation algorithm) recognizes the fact that this is a bi-linear problem. In other words fixing the source and inverting for the reflectivity is a linear problem. Fixing the reflectivity and inverting for the source is linear problem. Joint inversion for the source and reflectivity, however, is a non-linear problem.

Alternation is the method previously applied to joint inversion of the convolutional model [7] [10] [11]. The algorithm is simple: fix the source while inverting for the reflectivity, update the reflectivity, fix the reflectivity while inverting for the source, update the source, repeat.

The second option for optimization has not been previously attempted. This option requires the use of non-linear optimization techniques. We implemented a variety of non-linear optimization techniques and compared their success/failures to the alternation method.

The remaining sections of this chapter give a brief overview of each of the following chapters in this thesis.

1.2 Foundations

In chapter 2 we provide a basis for using the convolutional model of the seismogram. Initially, we derive the convolutional model of the seismogram using a thought experiment. This exercise is a slight reworking of what was done by Ziolkowski [17]. It is repeated here for two reasons. First, the derivation provides the reader with an excellent intuitive grasp of what aspects of the wave field are captured by the convolutional model of the seismogram. Second, Ziolkowski's text is no longer in print, and has been stolen from more than one academic library.

Following the reworking of Ziolkowski's thought experiment, a complete and finely detailed derivation of the convolutional model is undertaken. The final result of the derivation is a model of the seismogram which is a function of offset and time whose arguments involve take off angle, travel time, and the various partial derivatives of travel time.

1.3 Numerical Methods

In chapter 3 we provide a discussion of the numerical methods used to implement the convolutional model and its inverse. The implementation was done using the object oriented paradigm in C++. More specifically, the implementation was done within the framework of the Hilbert Class Library or HCL (pronounced as hickle). HCL is a mathematical library that among other things provides relatively easy access to some very powerful optimization algorithms.

We describe some of the more important details of implementation and give snippets of code where appropriate. We also provide a sketch of some of the less familiar optimization algorithms.

1.4 Results

In chapter 4 we outline the results from some of the numerical experiments performed. The outputs of various optimization algorithms are also compared. The full range of tests could not be presented in this thesis, so the results of other tests can be viewed at

http://www.caam.rice.edu/ winslow/research

In essence, numerical tests indicate that the trust region and limited memory BFGS approaches to optimizing the objective function are both faster and more accurate (in terms of relative error in the source and reflectivity) than alternation and various subspace techniques.

In addition, under some conditions (when the reflectivity is a spike and the source is oscillatory) we have numerical evidence for families of solutions to the inverse problem which are computationally non-unique but remarkably different from the true solution to the inverse problem. By computationally non-unique we mean that these solutions reduce the objective function to values below 10^{-3}.

Chapter 2

Foundations

2.1 Introduction

This chapter will derive the convolutional model two different ways. The first derivation will use a rigorous thought experiment. The second derivation uses a linearization of the acoustic wave equation. In addition, this chapter will explain the forward map, the adjoint map, and the objective function.

2.2 Experimental Derivation

In this section I will derive the convolution operator within the context of a simple experiment. The derivation is not new [17], but its source is no longer in print. In addition, this derivation is slightly more rigorous than the previous presentation. For a completely rigorous derivation of the convolutional model as a solution to the linearized acoustic wave equation see section 2.3.

Figure 2.1 shows the geometry of the experiment. We have a source of waves, $f(t)$, buried inside of a medium. The waves propagate away from the source. Some waves travel directly to a set of receivers, $b(t, x)$, while others travel further into the medium, reflect off some structure, then propagate to the set of receivers. To simplify the derivation I will assume the source is a point radiator, and the medium is a linear, isotropic, acoustic, and constant velocity.

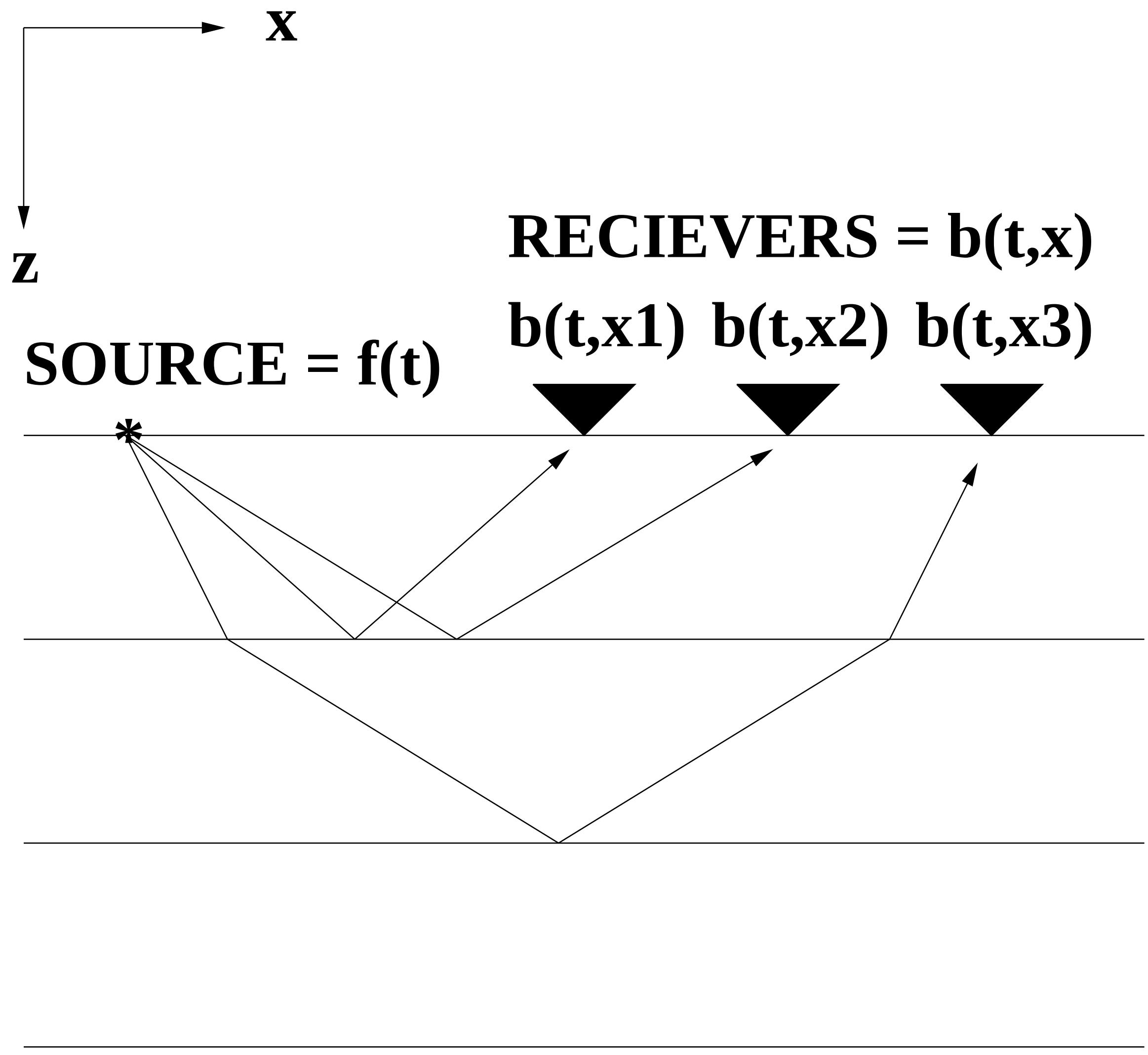

Figure 2.1 Geometry of the experiment. A wave source (asterisk), f(t) emits waves (arrows) which travel to receivers (triangles) that record the waves as data b(t,x). Some waves travel directly to the receivers, some bounce off the lower reflectors before reaching the receivers.

First, let us consider the case of a wave propagating directly from the source to a receiver. Specifically, at time zero allow the source to emit a wave with shape $f(t)$. This wave will propagate away from the source with a velocity c, arriving at the receiver at time t_1

$$t_1 = x_1/c$$

where x_1 is the distance or offset from the source to the receiver. Since our medium is linear, the shape of the wave will not change. Its amplitude will, however, be decreased by divergence. Let us call the amplitude scaling factor R_1. Thus, the data $b(x_1, t)$ recorded at receiver at position x_1 will have the following form

$$b(x_1, t) = R_1 f(t - t_1)$$

In addition, we can define a function $R(x_1, t)$ as follows

$$R(x_1, t) = \begin{cases} R_1 & \text{if } t = t_1 \\ 0 & \text{otherwise} \end{cases}$$

this allows us to write

$$b(x_1, t) = R(x_1, t_1) f(t - t_1)$$

Now we will consider the effects of the reflected wave. As in the previous scenario, at time zero the source emits a wave with shape $f(t)$. This wave will propagate away from the source with a velocity c, arriving at the receiver at time t_2

$$t_2 = x_2/c$$

where x_2 is the length of the path taken by the wave as it travels from the source to the receiver. In the linear approximation the shape of the wave will not change. Its amplitude will, however, be decreased by divergence, and by a reflection coefficient related to the strength of the reflector and the angle at which the wave is incident upon the reflector. Let us call the amplitude scaling factor R_2. Thus, if we consider only the reflected wave, the seismogram $b(x_1, t)$ recorded at position x_1 will have the following form

$$b(x_1, t) = R_2 f(t - t_2)$$

If we modify our function $R(x_1, t)$ as follows

$$R(x_1, t) = \begin{cases} R_1 & \text{if } t = t_1 \\ R_2 & \text{if } t = t_2 \\ 0 & \text{otherwise} \end{cases}$$

this allows us to write

$$b(x_1, t) = R(x_1, t_2) f(t - t_2)$$

For consistency with later formulations let $R(x_1, t)$ be named the reflectivity.

If we wish to consider the effects of both the reflected wave and the direct wave simultaneously, we need only recall that we are dealing with a linear acoustic medium. This assumption implies that waves superpose. Hence, we can write

$$b(x_1, t) = R(x_1, t_1) f(t - t_1) + R(x_1, t_2) f(t - t_2)$$

or

$$b(x_1, t) = \sum_{i=1}^{2} R(x_1, t_i) f(t - t_i)$$

This formulation allows us to consider more complicated models of the medium. Specifically, we can consider a source buried in an infinite whole space. The medium may have infinitely many reflectors. Hence, our receiver will record infinitely many reflected waves. Each of these reflected waves will be recorded at some time t_i, and each will be scaled by some amplitude factor R_i, where $i \in (-\infty, \infty)$. We modify our reflectivity function $R(x_1, t)$ as follows

$$R(x_1,t) = \begin{cases} \vdots & \vdots \\ R_{-1} & \text{if } t = t_{-1} \\ R_0 & \text{if } t = t_1 \\ R_1 & \text{if } t = t_2 \\ \vdots & \vdots \\ 0 & \text{otherwise} \end{cases}$$

this allows us to write

$$b(t) = \sum_{i=-\infty}^{\infty} R(t_i) f(t - t_i)$$

This is the discrete form of the convolution operator.

Alternatively, we can think of the reflectivity $R(x_1, t)$ as a function that varies continuously with time and write

$$b(x_1,t) = \int_{-\infty}^{\infty} R(x_1, t-\tau) f(\tau) d\tau$$

This is the continuous form of the convolution operator. Hence we have

$$b(x_1,t) = R(x_1,t) * f(t) = \int_{-\infty}^{\infty} R(x_1, t-\tau) f(\tau) d\tau$$

where $*$ denotes the convolution operator.

Lastly, when there are multiple receivers with different positions, the reflectivity $R(x_1, t)$ becomes a function of time and position $R(x, t)$. In this form the reflectivity is referred to as the offset dependent reflectivity. This also forces the data $b(x_1, t)$ to become a function of time and receiver position $b(x, t)$.

2.3 Theoretical Derivation

The convolution operator has many applications, and a reasonable amount of mathematical results exist which place it within a theoretical framework [2] [16]. One of the many ways in which the convolution operator is applied is as a model to the wave equation. In this section the convolutional model is rigorously derived as a linearized, high frequency approximation to the linear acoustic wave equation as applied to a layered three dimensional medium with source and receivers in the plane $z = 0$.

Consider the pressure $p(x, t, s)$ field arising from the solution of the three dimensional point source acoustic wave equation where density is considered constant and equal to one, with a point source:

$$\frac{1}{c^2(x)}\frac{\partial^2 p}{\partial t^2}(x, t, s) - \nabla^2 p(x, t, s) = f(t)\delta(x - s)$$

Here x is a three dimensional position vector, s is the position of a point source, and $f(t)$ is the source time function. The Green's function $p_0(x, t, s)$ is then given by the following relationship

$$\frac{1}{c^2(x)}\frac{\partial^2 p_0}{\partial t^2}(x, t, s) - \nabla^2 p_0(x, t, s) = \delta(t)\delta(x - s)$$

We can convolve the above equation with the source to get

$$\int d\hat{t} f(t - \hat{t}) \left(\frac{1}{c^2(x)}\frac{\partial^2}{\partial \hat{t}^2} - \nabla^2 \right) p_0(x, \hat{t}, s) = \int d\hat{t} f(t - \hat{t})\delta(\hat{t})\delta(x - s)$$

Since

$$\frac{d}{dt}(f * g) = \frac{df}{dt} * g = \frac{dg}{dt} * f$$

we can use the reproducing property of the delta function and say

$$\frac{1}{c^2(x)}\frac{\partial^2}{\partial t^2}\left[\int d\hat{t} f(t - \hat{t}) p_0(x, \hat{t}, s)\right] - \nabla^2 \int d\hat{t} f(t - \hat{t}) p_0(x, \hat{t}, s) = f(t)\delta(x - s)$$

or

$$\left(\frac{1}{c^2(x)}\frac{\partial^2}{\partial t^2} - \nabla^2\right)(f * p_0)(x,t,s) = f(t)\delta(x-s)$$

which is the wave equation operating on the source convolved with the Green's function, which is the pressure field. In other words

$$p(x,t,s) = f(t) * p_0(x,t,s)$$

Now, the convolutional model considers only perturbations to the pressure field (direct waves are ignored, only reflections are modeled). Perturbations to the pressure field to be given by

$$\delta p(x,t,s) = f(x,t,s) * \delta p_0(x,t,s)$$

Hence to compute the convolutional model of the seismogram, we need to compute the perturbation to the Green's function and convolve that with the source.

We now set about the task of computing a perturbational form of the Green's function. First, however, we need a linearized form of the wave equation. Recall the three dimensional point source acoustic wave equation where density is considered constant and equal to one, with a point source:

$$\frac{1}{c^2(x)}\frac{\partial^2 p}{\partial t^2}(x,t,s) - \nabla^2 p(x,t,s) = f(t)\delta(x-s)$$

Assuming that the material velocity $c(x)$ is a linear combination of a slowly varying background velocity and a perturbation to the velocity

$$c(x) = c(x) + \delta c(x)$$

and that the pressure field is a linear combination of a slowly varying background pressure with a perturbational component

$$p(x,t,s) = p(x,t,s) + \delta p(x,t,s)$$

the wave equation has the following form

$$\frac{1}{(c+\delta c)^2(x)}\frac{\partial^2(p+\delta p)}{\partial t^2}(x,t,s)-\nabla^2(p+\delta p)(x,t,s)=f(t)\delta(x-s).$$

Multiplication by $(c+\delta c)^2(x)$ and the fact that the derivative is a linear operator yields

$$\frac{\partial^2 p}{\partial t^2}(x,t,s)+\frac{\partial^2 \delta p}{\partial t^2}(x,t,s)-(c+\delta c)^2(x)\nabla^2 p(x,t,s)-(c+\delta c)^2(x)\nabla^2\delta p(x,t,s)$$
$$=(c+\delta c)^2(x)f(t)\delta(x-s).$$

In the linear approximation the δc^2 term is considered negligible. Thus an expansion of $(c+\delta c)^2(x)$ gives

$$\frac{\partial^2 p}{\partial t^2}(x,t,s)+\frac{\partial^2 \delta p}{\partial t^2}(x,t,s)-c^2(x)\nabla^2 p(x,t,s)-2c(x)\delta c(x)\nabla^2 p(x,t,s)-$$
$$c^2(x)\nabla^2\delta p(x,t,s)-2c(x)\delta c(x)\nabla^2\delta p(x,t,s)$$
$$=c^2(x)f(t)\delta(x-s)+2c(x)\delta c(x)f(t)\delta(x-s)$$

but from the wave equation we know that

$$\frac{\partial^2 p}{\partial t^2}(x,t,s)-c^2(x)\nabla^2 p(x,t,s)-c^2(x)f(t)\delta(x-s)=0.$$

This yields the following simplification

$$\frac{\partial^2 \delta p}{\partial t^2}(x,t,s)-2c(x)\delta c(x)\nabla^2 p(x,t,s)-c^2(x)\nabla^2\delta p(x,t,s)-2c(x)\delta c(x)\nabla^2\delta p(x,t,s)$$
$$=2c(x)\delta c(x)f(t)\delta(x-s).$$

Dividing through by $c^2(x)$ and rearranging terms, the above becomes

$$\frac{1}{c^2(x)}\frac{\partial^2 \delta p}{\partial t^2}(x,t,s)-\nabla^2\delta p(x,t,s)=$$
$$\frac{2\delta c(x)}{c(x)}\left(f(t)\delta(x-s)+\nabla^2 p(x,t,s)+\nabla^2\delta p(x,t,s)\right).$$

Again from the wave equation

$$f(t)\delta(x-s) + \nabla^2 p(x,t,s) = \frac{1}{c^2(x)}\frac{\partial^2 p}{\partial t^2}(x,t,s)$$

and substituting this gives

$$\frac{1}{c^2(x)}\frac{\partial^2 \delta p}{\partial t^2}(x,t,s) - \nabla^2 \delta p(x,t,s) = \frac{2\delta c(x)}{c^3(x)}\frac{\partial^2 p}{\partial t^2}(x,t,s) + \frac{2\delta c(x)}{c(x)}\nabla^2 \delta p(x,t,s)$$

Once more, in the linear approximation it is assumed that

$$\frac{2\delta c(x)}{c(x)}\nabla^2 \delta p(x,t,s) = 0$$

which leads to

$$\frac{1}{c^2(x)}\frac{\partial^2 \delta p}{\partial t^2}(x,t,s) - \nabla^2 \delta p(x,t,s) = \frac{2\delta c(x)}{c^3(x)}\frac{\partial^2 p}{\partial t^2}(x,t,s)$$

which is simply the wave equation applied to $\delta p(x,t,s)$ with a source equal to the right hand side. This is the form of the linearized wave equation which is useful to us. Now we can return to the task of deriving the perturbation to the Green's function $\delta p_0(x,t,s)$

Using the reproducing property of the delta function, δp_0 can be written as follows

$$\delta p_0(r,t,s) = \int dx \int d\hat{t}\delta(x-r)\delta(t-\hat{t})\delta p_0(x,\hat{t},s)$$

where r is the position of a receiver. We can substitute the wave equation formulation of the Green's function into the above result to obtain

$$\delta p_0(r,t,s) = \int\int dx d\hat{t} \left(\frac{1}{c^2(x)}\frac{\partial^2}{\partial \hat{t}^2} - \nabla^2\right) p_0(x,t-\hat{t},r)\delta p_0(x,\hat{t},s)$$

Since the boundary terms are zero for very large and very small time, integration by parts yields

$$\delta p_0(r,t,s) = \int\int dx d\hat{t} p_0(x,t-\hat{t},r) \left(\frac{1}{c^2(x)}\frac{\partial^2}{\partial \hat{t}^2} - \nabla^2\right) p_0(x,t-\hat{t},r)\delta p_0(x,\hat{t},s)$$

Substituting the right hand side of the linearized form of the wave equation gives

$$\delta p_0(r,t,s) = 2\int\int dx d\hat{t} p_0(x, t-\hat{t}, r)\frac{\delta c(x)}{c^3(x)}\frac{\partial^2 p_0}{\partial t^2}(x,\hat{t},s)$$

Again, since the boundary terms are zero for very large and very small time, integration by parts yields

$$\delta p_0(r,t,s) = -2\int dx \frac{\delta c(x)}{c^3(x)}\int d\hat{t}\frac{\partial p_0}{\partial \hat{t}}(x,t-\hat{t},r)\frac{\partial p_0}{\partial \hat{t}}(x,\hat{t},s)$$

Now note that a pressure field arising from a delta function source has a high frequency approximation of the following form

$$p(x,t,s) = a(x,s)\delta(t-\tau(x,s))$$

where $a(x,s)$ specifies an amplitude scale factor and $\tau(x,s)$ is the travel time from the source to some point x. Taking partial derivatives gives

$$\frac{\partial p}{\partial \hat{t}}(x,\hat{t},s) = a(x,s)\delta'(x,\hat{t}-\tau(x,s),s)$$

and

$$\frac{\partial p}{\partial \hat{t}}(x,t-\hat{t},r) = -a(x,r)\delta'(x,t-\hat{t}-\tau(x,r),r)$$

Substituting into the integrals gives

$$\delta p_0(r,t,s) = 2\int dx\frac{\delta c(x)}{c^3(x)}\int d\hat{t}a(x,r)\delta'(x,t-\hat{t}-\tau(x,r))a(x,s)\delta'(x,\hat{t}-\tau(x,s))$$

or

$$\delta p_0(r,t,s) = 2\int dx\frac{\delta c(x)}{c^3(x)}a(x,s)a(x,r)\int d\hat{t}\delta'(x,t-\hat{t}-\tau(x,r))\delta'(x,\hat{t}-\tau(x,s))$$

By the properties of the delta function this can be written as

$$\delta p_0(r,t,s) = 2\int dx\frac{\delta c(x)}{c^3(x)}a(x,s)a(x,r)\int d\hat{t}\delta(x,t-\hat{t}-\tau(x,r))\delta''(x,\hat{t}-\tau(x,s))$$

or

$$\delta p_0(r,t,s) = 2\int dx \frac{\delta c(x)}{c^3(x)} a(x,s)a(x,r)\int d\hat{t}\delta''(x,\hat{t}-\tau(x,s)-\tau(x,r))$$

Integration in $\hat{t}$ gives

$$\delta p_0(r,t,s) = 2\int dx \frac{\delta c(x)}{c^3(x)} a(x,s)a(x,r)\frac{\partial^2}{\partial t^2}\delta(t-\tau(x,s)-\tau(x,r))$$

which can also be written as

$$\delta p_0(r,t,s) = 2\frac{\partial^2}{\partial t^2}\int dx \frac{\delta c(x)}{c^3(x)} a(x,s)a(x,r)\delta(t-\tau(x,s)-\tau(x,r))$$

Using the Fourier Transform, the delta function can be written as

$$\delta(t-\tau(x,s)-\tau(x,r)) = \frac{1}{2\pi}\int d\omega e^{i\omega(t-\tau(x,s)-\tau(x,r))}$$

which yields

$$\delta p_0(r,t,s) = \frac{1}{\pi}\frac{\partial^2}{\partial t^2}\int dx \frac{\delta c(x)}{c^3(x)} a(x,s)a(x,r)\int d\omega e^{i\omega(t-\tau(x,s)-\tau(x,r))}$$

For a three dimensional medium, the variable x is really a vector (x,y,z)

$$x \leftarrow (x,y,z)$$

Recalling the assumption that the media is layered, i.e.

$$c = c(z)$$

and

$$\delta c = \delta c(z)$$

allows us to write

$$\delta p_0(r,t,s) = \frac{1}{\pi}\frac{\partial^2}{\partial t^2}\int dz \frac{\delta c(z)}{c^3(z)}$$
$$\int d\omega \int\int dxdy e^{i\omega(t-\tau(x,y,z,s)-\tau(x,y,z,r))} a(x,y,z,s)a(x,y,z,r) \tag{2.1}$$

Now recall the Principle of Stationary Phase [4]

$$\int\int dxdy e^{i\omega f(x,y)} g(x,y) = \left(\sqrt{\frac{2\pi}{\omega}}\right)^2 \sum_{\nabla f(x,y)=0} e^{i\omega f(x,y)} g(x,y) \frac{e^{\frac{i\pi sgn \nabla\nabla f(x,y)}{4}}}{\sqrt{det \nabla\nabla f(x,y)}} + O(\omega^{-2})$$

where $sgn\nabla\nabla f(x,y)$ is the sum of the signs of the eigenvalues of $\nabla\nabla f(x,y)$.

The Principle of Stationary phase can be applied here with the $O(\omega^{-2})$ ignored because in the high frequency approximation ω^{-2} terms are very small. To apply this Principle to the above integral form, it is necessary to know where the (x,y) gradient of the phase is zero. In other words at what (x,y) is it true that

$$\frac{\partial}{\partial x}[t - \tau(x,y,z,s) - \tau(x,y,z,r)] = 0 = \frac{\partial}{\partial y}[t - \tau(x,y,z,s) - \tau(x,y,z,r)]$$

Note that in a layered medium with the source and receiver located on a horizontal plane, there is a translation invariance in the travel time. This means that the travel time can also be considered a function of the distance between the source and the receiver. Hence we wish to know where

$$\frac{\partial}{\partial n}[t - \tau(n,z,s) - \tau(n,z,r)] = 0$$

where the n axis corresponds to the line connecting the source and the receiver. Since t is not a function of n this is equivalent to asking where

$$\frac{\partial}{\partial n}\tau(n,z,s) = -\frac{\partial}{\partial n}\tau(n,z,r)$$

Now, the horizontal translation invariance also implies that

$$\tau(r + \triangle r, z, r) = \tau(r - \triangle r, z, r)$$

and

$$\frac{\partial}{\partial n}\tau(r + \triangle n, z, r) = -\frac{\partial}{\partial n}\tau(r - \triangle n, z, r)$$

In addition, the horizontal translation invariance implies that

$$\frac{\partial}{\partial n}\tau(s+\triangle n,z,s)=\frac{\partial}{\partial n}\tau(r+\triangle n,z,r)$$

taking the above two equations we have

$$\frac{\partial}{\partial n}\tau(s+\triangle n,z,s)=-\frac{\partial}{\partial n}\tau(r-\triangle n,z,r)$$

Hence, we need to know when

$$s+\triangle n=r-\triangle n$$

This equation holds when

$$\triangle n=\frac{r-s}{2}$$

or midway between the source and the receiver.

Again, because of the translation invariance of the problem, we can choose our (x,y) coordinate system so that the direction n corresponds to the x axis, and the source is located at $(x=0,y=0)$. In this case, the receiver is located at $(x=r,y=0)$. Then, the stationary point occurs at $(x=\frac{r}{2},y=0)$.

In the linear approximation, the Principle of Stationary Phase combined with (2.1) results in

$$\delta p_0(r,t,s)|_{s=0}=2\frac{\partial^2}{\partial t^2}\int dz\frac{\delta c(z)}{c^3(z)}\int\frac{d\omega}{\omega}e^{i\omega[t-\tau(x,y,z,s)-\tau(x,y,z,r)]}$$

$$a(x,y,z,s)a(x,y,z,r)\frac{e^{\frac{i\pi sgn\nabla_{xy}\nabla_{xy}[t-\tau(x,y,z,s)-\tau(x,y,z,r)]}{4}}}{\sqrt{det\nabla_{xy}\nabla_{xy}[t-\tau(x,y,z,s)-\tau(x,y,z,r)]}}|_{x=\frac{r}{2},y=0,s=0}$$

In order to simplify this form we make the following definitions

$$\delta p_0(r,t)=\delta p_0(r,t,s)|_{s=0}$$

$$\tau(x,z)|_{x=\frac{r}{2}}=\tau(x,y,z,s)|_{x=\frac{r}{2},y=0,s=0}$$

$$a(x,z) = a(x,y,z,s)|_{y=0,s=0}$$

Note that since the travel time from the source to the mid point is the same as the travel time from the receiver to the mid point

$$\tau(x,y,z,s)|_{x=\frac{r}{2},y=0,s=0} = \tau(x,y,z,r)|_{x=\frac{r}{2},y=0} = \tau(x,z)|_{x=\frac{r}{2}}$$

Also note that because of the horizontal translation invariance, the amplitude of a wave traveling from the source to the mid point is the same as the amplitude of a wave traveling from the receiver to the mid point.

$$a(x,y,z,s)|_{x=\frac{r}{2},y=0,s=0} = a(x,y,z,r)|_{x=\frac{r}{2},y=0} = a(x,z)|_{x=\frac{r}{2}}$$

Thus

$$\delta p_0(r,t) = 2\frac{\partial^2}{\partial t^2}\int dz \frac{\delta c(z)}{c^3(z)}\int \frac{d\omega}{\omega} e^{i\omega[t-2*\tau(x,z)]} a^2(x,z)$$
$$\frac{e^{\frac{i\pi sgn \nabla_{xy}\nabla_{xy}[t-\tau(x,y,z,s)-\tau(x,y,z,r)]}{4}}}{\sqrt{det \nabla_{xy}\nabla_{xy}[t-\tau(x,y,z,s)-\tau(x,y,z,r)]}}|_{x=\frac{r}{2},y=0,s=0} \tag{2.2}$$

It is now useful to compute the form of

$$\nabla_{xy}\nabla_{xy}[t-\tau(x,y,z,s)-\tau(x,y,z,r)]_{x=\frac{r}{2},y=0,s=0}$$

Note that because of the horizontal translation invariance we can make a change of variables

$$n(x,y) = \sqrt{x^2+y^2}$$

such that

$$\tau(x,y,z,r) = \tau(n(x,y),z,r)$$

Hence

$$\frac{\partial \tau}{\partial y} = \frac{\partial \tau}{\partial n}\frac{\partial n}{\partial y} = \frac{y}{n}\frac{\partial \tau}{\partial n}$$

and

$$\frac{\partial^2\tau}{\partial y^2} = \frac{\partial}{\partial y}\left(\frac{y}{n}\frac{\partial\tau}{\partial n}\right) = \left(\frac{1}{n} - \frac{y^2}{n^3}\right)\frac{\partial\tau}{\partial n} + \frac{y}{n}\frac{\partial^2\tau}{\partial n^2}\frac{\partial n}{\partial y}$$

and last but not least

$$\frac{\partial^2\tau}{\partial x\partial y} = \frac{\partial}{\partial x}\left(\frac{y}{n}\frac{\partial\tau}{\partial n}\right) = -\frac{xy}{n^3}\frac{\partial\tau}{\partial n} + \frac{y}{n}\frac{\partial^2\tau}{\partial n^2}\frac{\partial n}{\partial x}$$

Now recall that because of the horizontal translation invariance, we have a coordinate system such that n corresponds with the x axis. This implies

$$\frac{1}{n}\frac{\partial\tau}{\partial n}(x,y,z,s) = \frac{1}{x}\frac{\partial\tau}{\partial x}(x,y,z,s)$$

and

$$\frac{1}{n}\frac{\partial\tau}{\partial n}(x,y,z,r) = \frac{1}{(x-r)}\frac{\partial\tau}{\partial x}(x,y,z,r)$$

Thus

$$\frac{\partial^2\tau}{\partial y^2}(x,y,z,s)|_{x=\frac{r}{2},y=0} = \frac{2}{r}\frac{\partial\tau}{\partial x}(x,y,z,s)|_{x=\frac{r}{2},y=0}$$

and

$$\frac{\partial^2\tau}{\partial y^2}(x,y,z,r)|_{x=\frac{r}{2},y=0} = -\frac{2}{r}\frac{\partial\tau}{\partial x}(x,y,z,r)|_{x=\frac{r}{2},y=0}$$

$$\frac{\partial^2\tau}{\partial x\partial y}(x,y,z,r)|_{x=\frac{r}{2},y=0} = 0$$

This implies that

$$det\{\nabla_{xy}\nabla_{xy}[t-\tau(x,y,z,s)-\tau(x,y,z,r)]\}_{x=\frac{r}{2},y=0} =$$

$$[-\frac{\partial^2\tau}{\partial x^2}(x,y,z,s) - \frac{\partial^2\tau}{\partial x^2}(x,y,z,r)][-\frac{2}{r}\frac{\partial\tau}{\partial x}(x,y,z,s) + \frac{2}{r}\frac{\partial\tau}{\partial x}(x,y,z,r)]_{x=\frac{r}{2},y=0}$$

Invoking the translation invariance again in order make derivatives a function of s, we have

$$det\{\nabla_{xy}\nabla_{xy}[t-\tau(x,y,z,s)-\tau(x,y,z,r)]\}_{x=\frac{r}{2},y=0} =$$

$$[\frac{\partial^2\tau}{\partial x^2}(x,y,z,s) + \frac{\partial^2\tau}{\partial x^2}(x,y,z,s)][\frac{2}{r}\frac{\partial\tau}{\partial x}(x,y,z,s) + \frac{2}{r}\frac{\partial\tau}{\partial x}(x,y,z,s)]_{x=\frac{r}{2},y=0}$$

Note that since the travel time is always increasing away from the source, both of the terms in the square brackets are positive at $(x = \frac{r}{2}, y = 0)$. Thus

$$sgn \nabla_{xy} \nabla_{xy} [t - \tau(x,y,z,s) - \tau(x,y,z,r)]_{x=\frac{r}{2},y=0} = 2$$

Writing the determinant in a form more useful for the expression of $\delta p_0(r,t)$

$$det\{\nabla_{xy} \nabla_{xy} [t - \tau(x,y,z,s) - \tau(x,y,z,r)]\}_{x=\frac{r}{2},y=0,s=0} =$$
$$\frac{8}{r}\frac{\partial^2 \tau}{\partial x^2}(x,y,z,s)\frac{\partial \tau}{\partial x}(x,y,z,s)|_{x=\frac{r}{2},y=0,s=0}$$

Returning to (2.2) and substituting the previous results leaves us with

$$\delta p_0(r,t) = 2\frac{\partial^2}{\partial t^2}\int dz \frac{\delta c(z)}{c^3(z)}\int \frac{d\omega}{\omega} e^{i\omega[t-2\tau(x,z)]} a^2(x,z)$$
$$\frac{e^{\frac{i\pi}{2}}}{\sqrt{\frac{8}{r}\frac{\partial^2 \tau}{\partial x^2}(x,y,z,s)\frac{\partial \tau}{\partial x}(x,y,z,s)}}|_{x=\frac{r}{2},y=0,s=0}$$

or

$$\delta p_0(r,t) = e^{\frac{i\pi}{2}}\frac{\partial^2}{\partial t^2}\int dz \frac{\delta c(z)}{c^3(z)} \frac{a^2(x,z)}{\sqrt{\frac{2}{r}\frac{\partial^2 \tau}{\partial x^2}(x,y,z,s)\frac{\partial \tau}{\partial x}(x,y,z,s)}}\int \frac{d\omega}{\omega} e^{i\omega[t-2\tau(x,z)]}|_{x=\frac{r}{2},y=0,s=0}$$

Differentiation with respect to t then gives

$$\delta p_0(r,t) = ie^{\frac{i\pi}{2}}\frac{\partial}{\partial t}\int dz \frac{\delta c(z)}{c^3(z)} \frac{a^2(x,z)}{\sqrt{\frac{2}{r}\frac{\partial^2 \tau}{\partial x^2}(x,y,z,s)\frac{\partial \tau}{\partial x}(x,y,z,s)}}\int d\omega e^{i\omega[t-2\tau(x,z)]}|_{x=\frac{r}{2},y=0,s=0}$$

and by the definition of the delta function this is

$$\delta p_0(r,t) = 2\pi ie^{\frac{i\pi}{2}}\frac{\partial}{\partial t}\int dz \frac{\delta c(z)}{c^3(z)} \frac{a^2(x,z)}{\sqrt{\frac{2}{r}\frac{\partial^2 \tau}{\partial x^2}(x,y,z,s)\frac{\partial \tau}{\partial x}(x,y,z,s)}}\delta[t-2\tau(x,z)]|_{x=\frac{r}{2},y=0,s=0}$$

Using Euler's formula

$$e^{\frac{i\pi}{2}} = i$$

we have

$$\delta p_0(r,t) = -2\pi\frac{\partial}{\partial t}\int dz \frac{\delta c(z)}{c^3(z)} \frac{a^2(x,z)\delta[t-2\tau(x,z)]}{\sqrt{\frac{2}{r}\frac{\partial^2 \tau}{\partial x^2}(x,y,z,s)\frac{\partial \tau}{\partial x}(x,y,z,s)}}|_{x=\frac{r}{2},y=0,s=0} \qquad (2.3)$$

It is useful now to discuss the exact form of $a(x,z)|_{x=\frac{r}{2}}$. The form is as follows [13]

$$a(x,z)|_{x=\frac{r}{2}} = \frac{c(z)}{4\pi}\sqrt{\frac{\partial^2\tau}{\partial y^2}(x,y,z,s)|\nabla_{xyz}\tau(x,y,z,s)\times\nabla\theta(x,y,z,s)|}\,\Bigg|_{x=\frac{r}{2},y=0,s=0}$$

where $\theta(x,y,z,s)$ is the take off angle of the ray originating at s and arriving at (x,y,z).

Now, from the previous derivation of the Hessian we know

$$\frac{\partial\tau}{\partial y} = \frac{\partial\tau}{\partial n}\frac{\partial n}{\partial y} = \frac{y}{n}\frac{\partial\tau}{\partial n}$$

and

$$\frac{\partial^2\tau}{\partial y^2}(x,y,z,s)|_{x=\frac{r}{2},y=0} = \frac{2}{r}\frac{\partial\tau}{\partial x}(x,y,z,s)|_{x=\frac{r}{2},y=0}$$

which implies

$$a(x,z)|_{x=\frac{r}{2}} = \frac{c(z)}{4\pi}$$

$$\sqrt{\frac{2}{r}\frac{\partial\tau}{\partial x}(x,y,z,s)\left|\frac{\partial\tau}{\partial x}(x,y,z,s)\frac{\partial\theta}{\partial z}(x,y,z,s)-\frac{\partial\tau}{\partial z}(x,y,z,s)\frac{\partial\theta}{\partial x}(x,y,z,s)\right|}\,\Bigg|_{x=\frac{r}{2},y=0,s=0} \tag{2.4}$$

From Snell's Law we know

$$\frac{\sin\theta(x,y,z,s)}{c(0)} = \frac{\sin\psi(x,y,z,s)}{c(z)}$$

where $\psi(x,y,z,s)$ is the arrival angle at depth z for a ray originating at s. The Eikonal Equation tells us that

$$\frac{\sin\psi(x,y,z)}{c(z)} = \frac{\partial\tau}{\partial x}(x,y,z,s)$$

which implies that

$$\sin\theta(x,y,z,s) = c(0)\frac{\partial\tau}{\partial x}(x,y,z,s)$$

$$\cos\theta(x,y,z,s) = \sqrt{1-\sin^2\theta(x,y,z,s)} = \sqrt{1-c^2(0)\left(\frac{\partial\tau}{\partial x}(x,y,z,s)\right)^2}$$

and

$$c(0)\frac{\partial^2\tau}{\partial x^2}(x,y,z,s) = \frac{\partial \sin\theta}{\partial x}(x,y,z,s) = \cos\theta(x,y,z,s)\frac{d\theta}{dx}(x,y,z,s)$$

or

$$\frac{\partial\theta}{\partial x}(x,y,z,s) = \frac{c(0)}{\sqrt{1-c^2(0)\left(\frac{\partial\tau}{\partial x}(x,y,z,s)\right)^2}}\frac{\partial^2\tau}{\partial x^2}(x,y,z,s)$$

In addition,

$$c(0)\frac{\partial^2\tau}{\partial x\partial z}(x,y,z,s) = \frac{\partial \sin\theta}{\partial z}(x,y,z,s) = \cos\theta(x,y,z,s)\frac{\partial\theta}{\partial z}(x,y,z,s)$$

or

$$\frac{\partial\theta}{\partial z}(x,y,z,s) = \frac{c(0)}{\sqrt{1-c^2(0)\left(\frac{\partial\tau}{\partial x}(x,y,z,s)\right)^2}}\frac{\partial^2\tau}{\partial x\partial z}(x,y,z,s)$$

We can exchange the (x,y,z,s) arguments in place of (x,z) using the following definitions

$$\frac{\partial\theta}{\partial x}(x,z) = \frac{\partial\theta}{\partial x}(x,y,z,s)|_{y=0,s=0}$$

and

$$\frac{\partial\theta}{\partial z}(x,z) = \frac{\partial\theta}{\partial z}(x,y,z,s)|_{y=0,s=0}$$

Plugging these expressions into (2.4) we have

$$a(x,z)|_{x=\frac{r}{2}} =$$

$$\frac{c(z)}{4\pi}\sqrt{\frac{2}{r}\frac{\partial\tau}{\partial x}(x,z)\frac{c(0)}{\sqrt{1-c^2(0)\left(\frac{\partial\tau}{\partial x}(x,z)\right)^2}}\left|\frac{\partial\tau}{\partial x}(x,z)\frac{\partial^2\tau}{\partial x\partial z}(x,z)-\frac{\partial\tau}{\partial z}(x,z)\frac{\partial^2\tau}{\partial x^2}(x,z)\right|_{x=\frac{r}{2}}}$$

Because of our choice of the coordinate systems (x corresponds to r) we have

$$\frac{\partial}{\partial x}|_{s=0} = \frac{\partial}{\partial r}|_{s=0}$$

thus

$$a(x,z)|_{x=\frac{r}{2}} =$$

$$\frac{c(z)}{4\pi}\sqrt{\frac{2}{r}\frac{\partial\tau}{\partial r}(x,z)\frac{c(0)}{\sqrt{1-c^2(0)\left(\frac{\partial\tau}{\partial r}(x,z)\right)^2}}\left|\frac{\partial\tau}{\partial r}(x,z)\frac{\partial^2\tau}{\partial r\partial z}(x,z)-\frac{\partial\tau}{\partial r}(x,z)\frac{\partial^2\tau}{\partial r^2}(x,z)\right|}\Big|_{x=\frac{r}{2}}$$

and

$$a^2(x,z)|_{x=\frac{r}{2}} = a^2(\frac{r}{2},z) =$$

$$\frac{c^2(0)c^2(z)}{8\pi^2 r\sqrt{1-c^2(0)\left(\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\right)^2}}\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\left|\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\frac{\partial^2\tau}{\partial r\partial z}(\frac{r}{2},z)-\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\frac{\partial^2\tau}{\partial r^2}(\frac{r}{2},z)\right|$$

Substituting this into (2.3) and changing the dependence from (x, y, z, s) to (r, z) using previous definitions gives

$$\delta p_0(r,t) = \frac{-c^2(0)}{4\pi r}\frac{\partial}{\partial t}\int dz\frac{\delta c(z)}{c(z)}$$

$$\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\left|\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\frac{\partial^2\tau}{\partial r\partial z}(\frac{r}{2},z)-\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\frac{\partial^2\tau}{\partial r^2}(\frac{r}{2},z)\right|$$

$$\frac{\delta[t-2\tau(\frac{r}{2},z)]}{\sqrt{\left[1-c^2(0)\left(\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\right)^2\right]\frac{2}{r}\frac{\partial^2\tau}{\partial r^2}(\frac{r}{2},z)\frac{\partial\tau}{\partial r}(\frac{r}{2},z)}}$$

For the sake of sanity we define

$$a_1(r,z) = \frac{-c^2(0)\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\left|\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\frac{\partial^2\tau}{\partial r\partial z}(\frac{r}{2},z)-\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\frac{\partial^2\tau}{\partial r^2}(\frac{r}{2},z)\right|}{4\pi r\sqrt{\left[1-c^2(0)\left(\frac{\partial\tau}{\partial r}(\frac{r}{2},z)\right)^2\right]\frac{2}{r}\frac{\partial^2\tau}{\partial r^2}(\frac{r}{2},z)\frac{\partial\tau}{\partial r}(\frac{r}{2},z)}} \tag{2.5}$$

Thus we have

$$\delta p_0(r,t) = \frac{\partial}{\partial t}\int dz\frac{\delta c(z)}{c(z)}a_1(r,z)\delta[t-2\tau(\frac{r}{2},z)] \tag{2.6}$$

Now, we introduce the following change of variables:

$$\phi(r,\hat{t}) = z$$

$$T(r,z) = 2\tau(r,z)$$

such that

$$2\tau(r,\phi(r,\hat{t})) = T(r,\phi(r,\hat{t})) = \hat{t}$$

and

$$\phi(r, 2\tau(r,z)) = \phi(r, T(r,z)) = z$$

Then the following is true

$$\delta[t-2\tau(\frac{r}{2},z)]dz = \delta[t-2\tau(\frac{r}{2},\phi(\frac{r}{2},\hat{t}))]\frac{dz}{d\hat{t}}d\hat{t} = \delta[t-T(\frac{r}{2},\phi(\frac{r}{2},\hat{t}))]\frac{dz}{d\hat{t}}d\hat{t} = \delta(t-\hat{t})\frac{dz}{d\hat{t}}d\hat{t}$$

We then substitute into (2.6) to obtain

$$\delta p_0(r,t) = \frac{\partial}{\partial t}\int d\hat{t}\frac{dz}{d\hat{t}}\frac{\delta c(\phi(\frac{r}{2},\hat{t}))}{c(\phi(\frac{r}{2},\hat{t}))}a_1(r,\phi(\frac{r}{2},\hat{t}))\delta[t-\hat{t}]$$

The reproducing property of the delta function then gives

$$\delta p_0(r,t) = \frac{\partial}{\partial t}\left[\frac{dz}{dt}\frac{\delta c(\phi(\frac{r}{2},t))}{c(\phi(\frac{r}{2},t))}a_1(r,\phi(\frac{r}{2},t))\right]$$

This is nearly the final form of the perturbed Green's function. Note that this form contains explicit references to the travel time τ (in a_1) and no explicit references to θ. To be consistent with the original form of the Green's function, we change the variable r back to x. We now define an offset dependent amplitude function as follows

$$a(x,\phi(x,t)) = a_1(x,\phi(x,t))\frac{dz}{dt}$$

Then we can write the perturbation to the Green's function as

$$\delta p_0(x,t) = \frac{\partial}{\partial t}\left[\frac{\delta c(\phi(\frac{x}{2},t))}{c(\phi(\frac{x}{2},t))}a(x,\phi(x,t))\right]$$

Hence the perturbation to the seismogram is

$$\delta p(x,t) = f(t) * \delta p_0(\phi(x,t)) = f(t) * \frac{\partial}{\partial t}\left[\frac{\delta c(\phi(\frac{x}{2},t))}{c(\phi(\frac{x}{2},t))}a(x,\phi(x,t))\right]$$

Which is the convolutional model.

2.4 Forward map

With the convolution model rigorously derived, we now state it as follows

$$b(t,x) = f(t) * R(t,x) = f(t) * r(\phi(t,x))$$

where $b(t,x)$ is the model output (the response recorded at various receivers), $f(t)$ is the wave source, $R(t,x)$ is the offset dependent reflectivity, t is time, r is the depth dependent reflectivity, $\phi(t,x)$ is a change of variables function described below, and x is the horizontal spatial coordinate (or offset). Note that our forward map sets the offset dependent amplitude function $a(x, \phi(x,t))$ to 1. This approximation is justified for models with a small range of offsets. For a small range of offsets, the amplitude is nearly constant, hence it can pass through the convolution operator. Given the scale ambiguity inherent in this problem, the constant can then be considered to be 1.

The forward map assumes a horizontally layered medium, and hence a vertically dependent reflectivity $r(z)$ defined as

$$r(z) = \frac{d}{dz}\left(\frac{\delta c(z)}{c(z)}\right)$$

where $c(z)$ is the material velocity, $\delta c(z)$ represents small scale perturbations to the material velocity, and z is the vertical spatial coordinate in the model.

In order to perform the convolution in time, the vertically dependent reflectivity $r(z)$ must be transformed to an offset dependent reflectivity $R(t,x)$ via some change of variable $z = \phi(t,x)$. The forward map can handle a vertically variable velocity. It is, however, illustrative to describe this transformation for the case of a constant velocity medium ($c(z) = c$). Figure 2.2 describes the geometry for this case.

Figure 2.2 shows the ray path corresponding to a wave emitted from the source, which bounces off the reflector at depth z, travels up to the receiver at position x and is recorded at time $t(z,x)$. This time is referred to as the two-way travel time. The two-way travel time is dependent on the positions of the reflector and the receiver.

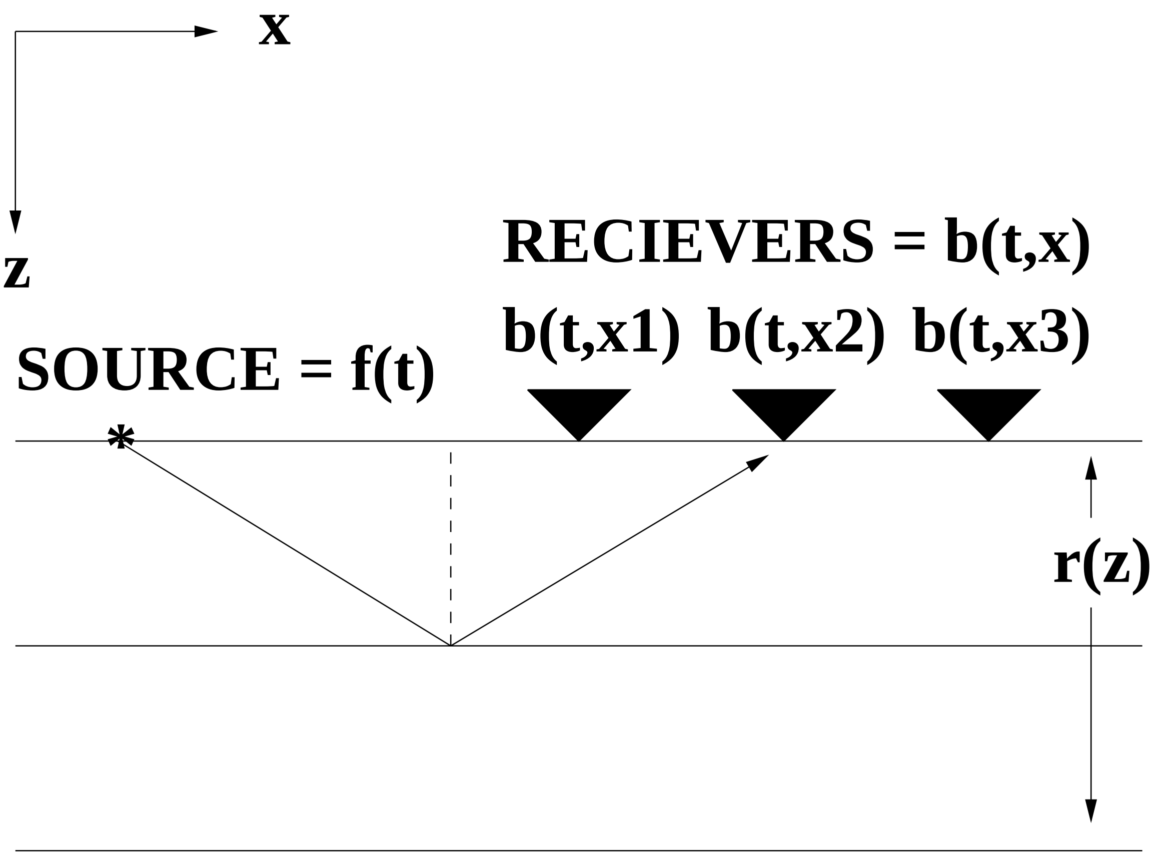

Figure 2.2 Geometry of the seismic experiment. A wave source (asterisk) emits waves (arrows) which travel to receivers (triangles). Waves reflecting off structure at depth z are recorded at receiver x at time t. For constant velocity the change of variables $z = \phi(t, x)$ can be computed using the Pythagorean Theorem. Layered velocities requires the change of variables to be computed using the travel time from the source to the receiver.

To compute $R(t,x)$ it is necessary to know at what time, for a given receiver, the reflection from a given depth point arrives at the receiver. For a layered constant velocity medium the Pythagorean theorem is used to compute the change of variables $z = \phi(x,t)$ such that

$$r(\phi(t,x)) = R(t,x)$$

The Pythagorean theorem states that the time it takes a wave to follow the aforementioned path is dictated by the following relationship

$$\frac{t^2}{4} = \frac{x^2}{4c^2} + \frac{z^2}{c^2}$$

this can be rewritten as

$$z = \phi(x,t) = \frac{1}{2}\sqrt{(ct)^2 - x^2}$$

Thus, for a constant velocity, layered medium

$$R(t,x) = r(\phi(t,x)) = r\left(\frac{1}{2}\sqrt{(ct)^2 - x^2}\right)$$

When the velocity of the medium is not constant, the two-way travel time is more complicated to compute. Methods of computing the two-way travel time are covered in subsection 3.2.2.

In summary, the forward map written in shorthand has the following form

$$A : (r(z), f(t)) \mapsto (r(\phi(t,x)), f(t)) \mapsto b(t,x)$$

where A denotes the forward map. Again, for the sake of brevity the above product vector is written in a more succinct form

$$m = (r(z), f(t))$$

This allows the above map to be described as

$$Am = b$$

2.5 Inversion

In the case of the convolutional model, the goal of inversion is to successfully map a set of known data $b(\tilde{t}, x)$ to a set of unknown parameters m. The most obvious means to achieve this goal is to find a mapping A^{-1} such that

$$A^{-1}\tilde{b} = m$$

If, however, A^{-1} does not exist, or it is too difficult to compute directly, then the process of finding m can be set up as an optimization problem

$$\min_{m \in M} \frac{1}{2} ||Am - \hat{b}||^2$$

where M is the set of all possible source and reflectivity pairs $(r(z) \in \Re^n, f(t) \in \Re^p)$. Various linear and non-linear optimization techniques can be used to find m. For a description of these methods and their implementation see chapter 3. For an discussion of the success of these various methods see chapter 4.

Chapter 3

Numerical Methods

3.1 Introduction

The notion of simultaneously inverting for the source and the reflectivity is not new. Previous work [10], however, was either done in the plane wave domain breaking the non-linear inverse problem into two linear sub-problems, or in the offset-time domain, but still breaking the non-linear problem into two linear sub-problems [7].

Breaking the non-linear problem into two linear problems is sometimes referred to as alternation, or coordinate search. In essence, the source is fixed and an inversion is done for the reflectivity. Inverting only for the reflectivity is a linear problem, so linear optimization techniques are used. Next, the reflectivity is fixed and an inversion is done for the source. Again inverting just for the source is a linear problem so linear optimization techniques are used. This alternating process is repeated until some convergence criteria is met.

The inversion in this thesis is done in the offset-time domain. More importantly, however, this is the first work to address the non-linear nature of the problem. In other words instead of just breaking the inversion into two linear sub-problems, inversion is also done on the full non-linear problem using a host of non-linear inversion techniques.

3.2 Implementation

The forward modeling and inversion of this problem is implemented within the C++ / HCL framework [5]. This section is intended as an explanation of the implementation from both a mathematical and a software perspective.

The generic elements or primary model components of this framework are the vector, the vector space and the operator. For the case of this model, the elements are specified as hfa vectors, hfa vector spaces, and hfa operators. HFA is an acronym for high frequency approximation, referring to the convolutional model as a high frequency approximation to the wave equation.

The simplest primary model component is denoted as a hfaModelVector. The parent class of hfaModelVector is HCLProductVector. Since it is a product vector, the hfaModelVector has two components, the source and the reflectivity. The construction of a hfaModelVector requires the following lines of code:

```
SGFVector source( 'sourceFile.sep' );
SGFVector reflectivity( 'reflectivityFile.sep' );
hfaModelVector srcRefProductVector( & source, & reflectivity );
```

Every hfaModelVector is associated with a hfaModelSpace. The construction of the hfaModelSpace may have the following form:

```
SGFSpace & SourceSpace = (SGFSpace &)source.Space();
SGFSpace & ReflectivitySpace = (SGFSpace &)reflectivity.Space();
hfaModelSpace SrcRefSpace(& SourceSpace, & ReflectivitySpace);
```

The third primary component is the hfa operator (hfaOp). A hfaModelVector is operated upon by a hfaOp. The parent class of hfaOp is HCLOpProductDomain. The hfaOp constructor has five components. Construction for a hfaOp may have the following form:

```
SGFSpace DataSpace( 'dataFormat.sep' );
SGFVector backgroundVelocity( 'velocityFile.sep' );
float stretch = 1.2;
float muteFactor = 2.1;
hfaOp ForwardMap( & SrcRefProductVectorSpace, & DataSpace, & back-
groundVelocity, stretch, muteFactor );
```

The SGFSpace DataSpace component dictates the output format of the hfaOp. In other words, it dictates the number and position of receivers, the time sampling of the receivers and any other specifications given by the SEP (Stanford Exploration Project) format. The SGFVector backgroundVelocity component specifies the discretized values of the background velocity of the model. The float stretch component specifies the extent to which NMO stretch (see section 3.2.3) may disfigure the data. The float mute component specifies the width of a linear smoothing filter that connects muted data to unmuted data.

Lastly, hfaOp is imaged using the Image method. The following code exemplifies one possible implementation:

```
SGFVector dataVector( & DataSpace, 'modelData.sep' );
ForwardMap.Image( SrcRefProductVector, dataVector );
```

In essence, the image method has three tasks:

- Convert the reflectivity as a function of space $r(z)$ to reflectivity as a function of space and time $R(t, x)$.
- Convolve the reflectivity $R(t, x)$ with the source $f(t)$.
- Mute the resulting traces.

A travel time table is used to convert the reflectivity as a function of space to a function of time and space. For details on how this is done see the subsection on depth to time conversions (3.2.2). The travel time table is a function of the data space and the background velocity. Thus, the construction of this table need not be repeated for each imaging of the hfa operator. To save computation the travel time table is built as part of the construction of the hfa operator.

The convolution of the reflectivity $R(t, x)$ and the source is done using the standard discrete convolution in the time domain as derived in section 2.1. The operator has the following form:

$$\sum_{i=-\infty}^{\infty} R(t_i, x) f(t - t_i)$$

Lastly, portions of the resulting traces are muted (zeroed-out) using a criteria (the stretch factor) specified by the user in the construction of the hfa operator. The characteristics of the mute operator are determined only by the data space, the stretch factor, the travel time table and the mute width. Thus, the construction of this operator need not be repeated for each imaging of the hfa operator. To save computation the mute operator is built as part of the construction of the hfa operator. For details on how this is done see the subsection on the Mute operator.

3.2.1 Travel Time

As stated above, the travel time operator is built in the constructor for the hfa operator. This is done to prevent an unnecessary repeat of computation and object construction. Before constructing the travel time operator, however, a travel time space is built. The space is simply an SGFSpace whose discretizations correspond to the spatial discretization of the data vector and reflectivity vector. An example construction using previous snippets of code follows:

```
float dz; backgroundVelocity.GetValue("d1",dz);
float z0; backgroundVelocity.GetValue("o1",z0);
```

```
float Nz; backgroundVelocity.GetValue("n1",Nz);
AbstractTable * ttSp = DataSpace.Parameters();
ttSp.PutValue("d1",dz);
ttSp.PutValue("o1",z0);
ttSp.PutValue("n1",Nz);
SGFSpace ttSpace = new SGFSpace( ttSp );
```

Now the travel time table can be constructed, and the travel time table can be retrieved. The travel time table is a private member of EvalTTOp, and the table is generated with the EvalTTOp constructor. The travel time table is accessed via the ReturnTT function.

```
evalTTOprator = new EvalTTOp( vel, ttspace, dt );
SGFVector & tt1 = evalTTOperator→ReturnTT();
```

Within the EvalTTOp constructor, the travel time table is constructed on a sampled grid. Each point on the grid has 1 value and two indices associated with it, an t, x, and z respectively. The x index represents the offset, or distance from the source to the receiver along the line $z = 0$. In this 2-D problem, it is assumed that the source and receivers are all located along the line $z = 0$, and that the receivers are equally spaced. The z index represents the depth, or the distance from the source to the depth point z along a vertical line. The t value is the time that it takes a ray to travel from the source down t a reflector at depth z and back up to a receiver at point x. This is commonly referred to as the *two-way travel time.* The values of t are computed using the following relation (Dix's formula) a second order approximation to the travel time for a layered medium. [14] [15].

$$t(t_0, x) = \sqrt{t_0^2 + \frac{x^2}{\frac{1}{t_0}\int_0^{t_0} c^2}}$$

where

$$t_0(z) = 2\int_0^z \frac{d\zeta}{c(\zeta)}$$

since

$$\frac{dt_0}{dz} = \frac{2}{c}$$

the expression for t can be rewritten as

$$t(z,x) = t_0(z)\sqrt{1 + \frac{x^2}{t_0(z)\int_0^z c(\zeta)d\zeta}}$$

Thus to compute the travel time, the t_0 integral and the integral over the velocity are computed using the midpoint rule with a discretization of the integral equal to the discretization of the z coordinate in the model. Once the integrals are computed, the results are plugged directly into the above relationship.

3.2.2 Depth to Time

The travel time table is a key tool for doing the change of variables or depth to time conversion that takes $r(z)$ to $r(\phi(t,x))$. This mapping is computed using a numerical scheme that is governed by the travel time table.

The mapping is done on a trace by trace basis, i.e. $r(z)$ gets mapped to $r(\phi(t,x)$ one x at a time. Thus, the first loop of the mapping algorithm is over the x index of the travel time table. The second loop of the algorithm is over the z index of the travel time table. The values within the travel time table dictate when on the time axis the reflectivity from a given depth will be placed.

Unfortunately, it is not likely that the time values from the travel time table will have a direct correspondence to the time discretizations of $r(\phi(t,x)$. In other words, the reflectivity at some depth point in the discretized form of $r(z)$ has a travel time associated with it via the travel time table. It is not likely that this travel time exactly matches one of the time points in the discretized form of $r(\phi(t,x)$. It is more likely that the travel time associated with a point in the discretized form of $r(z)$ fall in-between two time points in the discretized form of $r(\phi(t,x))$.

To overcome this problem a linear partition is computed based upon where the values from the travel time table land on the discretization of $r(\phi(t,x))$. For example, it the travel time is half way between two time points in the discretization, then the corresponding value of $r(z)$ is evenly split between the two points. Alternatively, if the travel time is two thirds of the way between the two points, then 1/3 of the corresponding value of $r(z)$ goes to one point and 2/3 goes to another. Lastly, to avoid aliasing, the travel time table is resampled and linearly interpolated on a finer scale within the mapping.

This whole process results in an algorithm with the following form in pseudo code where $TT(x,z)$ is the travel time table, and $\triangle t$ is the discretization spacing of time.

loop over x

loop over z

$\tau_1 = \frac{TT(x,z)}{\triangle t}$

$\tau_2 = \frac{TT(x,z+1)}{\triangle t}$

loop over resampling (im = 0:m)

$q = \frac{im}{m}$

$\tau = \tau_1 * (1.0 - q) + \tau_2 * q$

$k = int(floor(\tau))$

$\delta = \tau - k$

$R(t,x) = R(t,x) + (1.0 - \delta) * ((1.0 - q) * r(z) + q * r(z+1)$

$R(t+1,x) = R(t+1,x) + \delta * ((1.0 - q) * r(z) + q * r(z+1)$

end resampling loop

end z loop

end x loop

R(t,x)=R(t,x)/m

3.2.3 Stretch and Mute

When the reflectivity is mapped from a function of depth to a function of time, the support of the reflectivity is stretched as the offset increases. Stretching is a frequency transformation that shifts the reflectivity to lower frequencies. To understand the effects of stretch, recall the depth to time mapping for a constant velocity medium as derived in section 2.4

$$z = \phi(t, x) = \frac{1}{2}\sqrt{(ct)^2 - x^2}$$

Consider a fixed interval $\triangle z = z_2 - z_1$ on the depth axis. For $x = 0$ this depth interval will map onto a time interval $\triangle t = t_2 - t_1$, where for example $t_2 = \frac{2z_2}{c}$. If, however, x is increased, then the interval $\triangle t$ must also be increased in order to maintain a constant $\triangle z$. In other words, the $\triangle t$ interval is stretched as x increases.

In the forward map the source is convolved with $R(t, x)$. This results in data traces that get stretched as offset increases. The stretching is an artifact of the depth to time mapping and does not correspond to a real physical phenomenon. For small offsets, however, the effects of the stretching are negligible and ignored.

The user controls the amount of stretch allowed in the construction of the hfa-Operator. A stretch factor of 1.2 means that within a given trace (say the trace corresponding to $\tilde{x}$), once the length of the time interval becomes 1.2 times longer than the corresponding time interval for $x = 0$, then the value of the trace is set to zero. This process is referred to a muting.

If the muting occurs at a time in $R(t, x)$ has a large amplitude, an abrupt transformation in $R(t, x)$ will result. To avoid any numerical issues that may be generated by this abrupt transformation, the muting is extended over a series of time points with a linear interpolation. In other words, a portion of the trace (as determined by the stretch factor) is set to zero. Within the hfaOperator, the user specifies a time interval (muteWidth) over which the muting will transition linearly from 0 to the value of $R(t_{stretch} - muteWidth, x)$.

3.3 Optimization

As mentioned in chapter 2 the inverse problem of joint determination of the source and reflectivity is posed as an optimization problem.

If A represents the forward map, m represents a product vector containing the source and the reflectivity, and b is a set of data, then the inverse problem is defined as a least squares problem

$$\min_{m \in M} ||Am - b||_2^2$$

where M is the product vector space of all possible $(f(t), r(z))$ pairs. It is important to note here that A is a bi-linear operator. In other words, if the source is fixed and the reflectivity is free to vary, then the forward map is linear. Alternatively, if the reflectivity is fixed and the source is free to vary then the forward map is a linear operator. If, however, both the source and the reflectivity are free to vary, the forward map is non-linear.

It is important to make the bi-linear distinction because this characteristic allows different approaches to the inversion. The first approach (typically called alternation) is to use an algorithm like this:

1. Fix the source and invert for the reflectivity using a linear optimization technique
2. Fix the reflectivity and invert for the source using a linear optimization technique
3. If convergence criteria is not met, go to 1.

the second approach is to use a non-linear optimization technique on the full non-linear problem.

All previous simultaneous source, reflectivity inversion was done using the alternation approach [7] [11] [10] in either the offset-time domain, or the plane wave domain. This is the first work that the author is aware of that treats the full non-linear problem and does a comparison of various algorithms.

For alternation, the optimization algorithms included conjugate gradients and a Trust Region Steihaug-Toint truncated Newton method (here after referred to simply as trust region) [3].

For the non-linear problem the optimization algorithms used include limited memory BFGS, trust region, and a variation of the Kennet Sambridge algorithm [8] (referred to as the subspace algorithm) [3].

The subspace algorithm searches a 2 or 6 dimensional model of the objective function. The model is determined using either 2 vectors (the partial gradients with respect to the source and the reflectivity) or 6 vectors (the partial gradients with respect to the source and the reflectivity, plus the four images of the partial gradients under the partial Hessian). The subspace algorithm we used differs from the Kennet-Sambridge algorithm in that a trust region is imposed on the step. Specifically, the More'-Sorenson algorithm is used [12].

Lastly, Tikanhov regularization [6] was also implemented with the trust region and limited memory BFGS algorithms.

Chapter 4

Results

4.1 Introduction

In this chapter we use simulated data to present numerical evidence that supports the following statements

1. All optimization algorithms tested successfully reduce the relative error in the data residuals within 5%. In other words

$$\frac{||b_{sim} - b_{resim}||}{||b_{sim}||} < .05$$

   where b_{sim} is data simulated by running a predetermined source and reflectivity through the forward map, and b_{resim} is data obtained by running a source and reflectivity from the inversion through the forward map.

2. The trust region and limited memory BFGS algorithms are significantly faster at reducing the relative error in the data residual to below 5%

3. The shape of the source and reflectivity obtained from reducing the relative error in the data residuals below 5% can be markedly different from the shape of the source and reflectivity used to generate the simulated data.

4. When the source is oscillatory and the reflectivity has very compact support, the objective function can be reduced below 10^-6, but the source and reflectivity that is output from the inversion can have markedly different shapes compared to the shapes of the source and reflectivity used to generate the simulated data.

5. The above apparent non-uniqueness does not violate the theoretical results of Minkoff and Symes [11]

The numerical experiments to support the above hypotheses are broken into two categories: 1. experiments using reduction of the relative error in the data residuals to below 5% as the optimization convergence criteria, 2. experiments using reduction of the objective function to below 10^{-6} as the optimization convergence criteria. Each of these two categories is then broken into two subcategories: 1. experiments using a spike reflectivity to generate the simulated data, 2. experiments using a randomly generated reflectivity to generate the data. To reduce the error in the data residuals to below 5% the value of the spike reflectivity objective function must be reduced below 3.066 and the value of the random reflectivity objective function must be reduced below 36.23.

The following subsections detail the numerical experiments designed to support the above listed hypotheses. The first 3 hypotheses are supported in the subsection 4.2.1 which details experiments where the convergence criteria was that the relative error in the data residuals be reduced below 5%. Subsection 4.2.2 details the numerical evidence for the 4th hypothesis. Section 4.3 gives an analysis to support the 5th hypothesis.

4.2 Inversion Results

As stated in the previous chapter, inversion is performed via optimization of a least squares data residual objective function:

$$\min_{m \in M} \frac{1}{2} ||Am - b||_2^2$$

We used a wide range of optimization techniques to minimize $\frac{1}{2}||Am - b||^2$ including trust region, limited memory BFGS, Kennett subspace, and alternation.

The source used to generate the simulated data is shown in figure 4.1, the reflectivity used to generate the single spike simulated data is shown in figure 4.2 the reflectivity used to generate the random reflectivity is shown in figure 4.3. The simulated single spike data is shown in figure 4.4, and the simulated random data is shown in figure 4.5.

The inversions used a zero reflectivity and a source identical to the model source, but shifted away from zero time by 0.01 seconds and scaled by a factor of 0.5 as the starting guess.

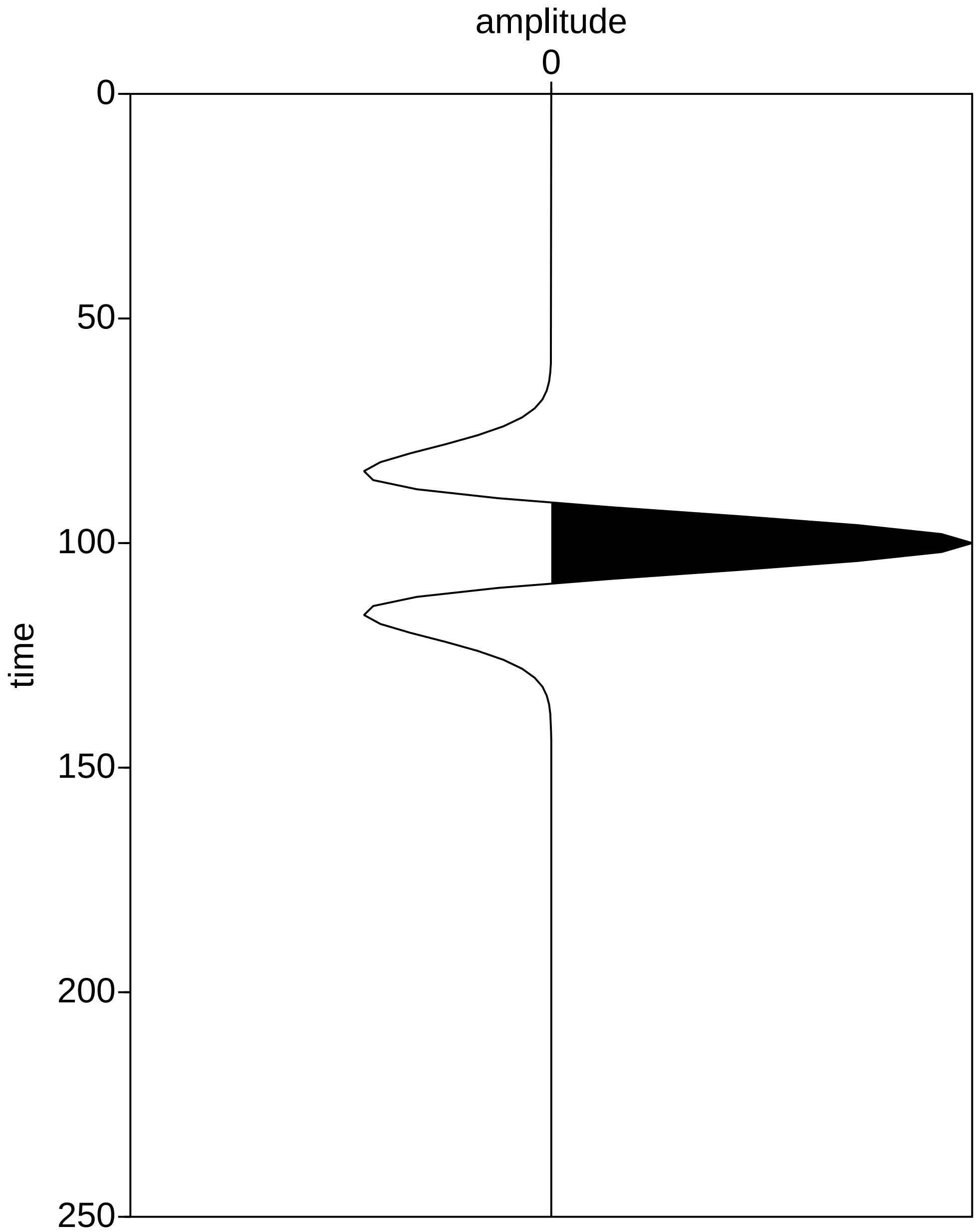

Figure 4.1 Model source used to generate the simulated data. Source is a Ricker Wavelet with 25 Hz center frequency centered at .10 seconds. SEP file parameters are as follows: $n1 = 126, d1 = 2, o1 = 0$

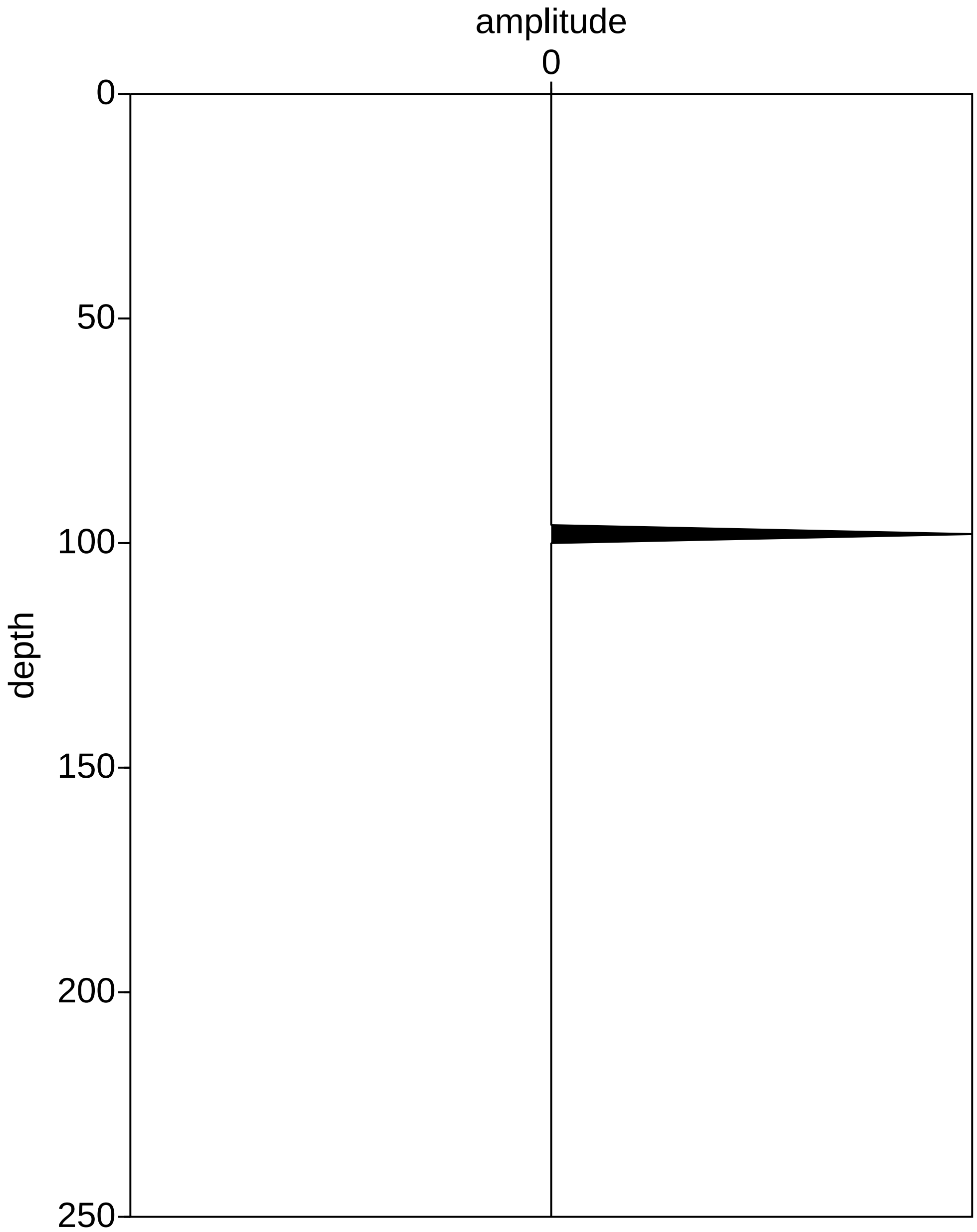

Figure 4.2 Single spike model reflectivity used to generate the simulated data. Values for the reflectivity are all zero except for the 50th element, which has an amplitude equal to one.

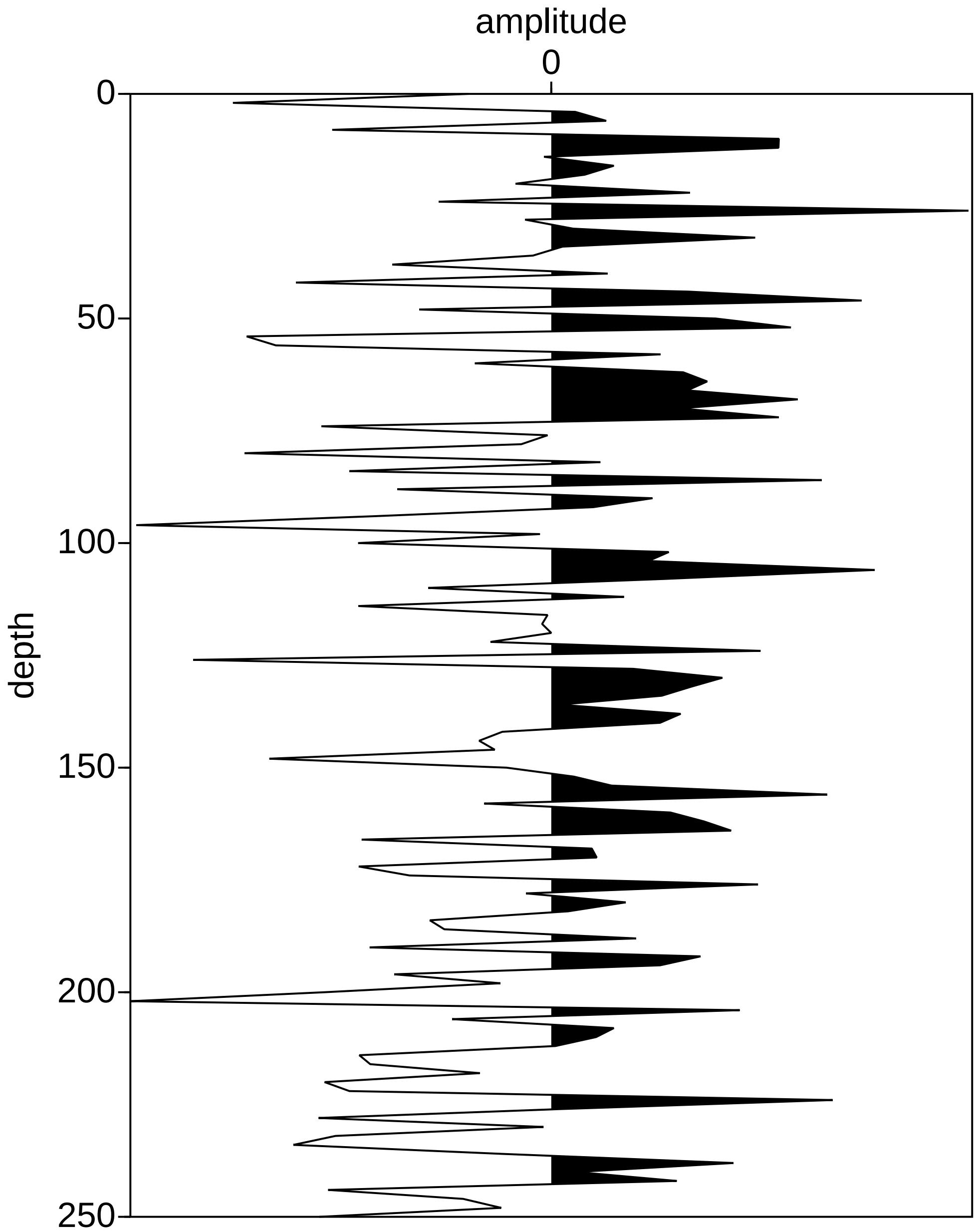

Figure 4.3 Random spike model reflectivity used to generate the simulated data. Values for the reflectivity were generate using the randn command in MATLAB. SEP file parameters are as follows: $n1 = 126, d1 = 2, o1 = 0$

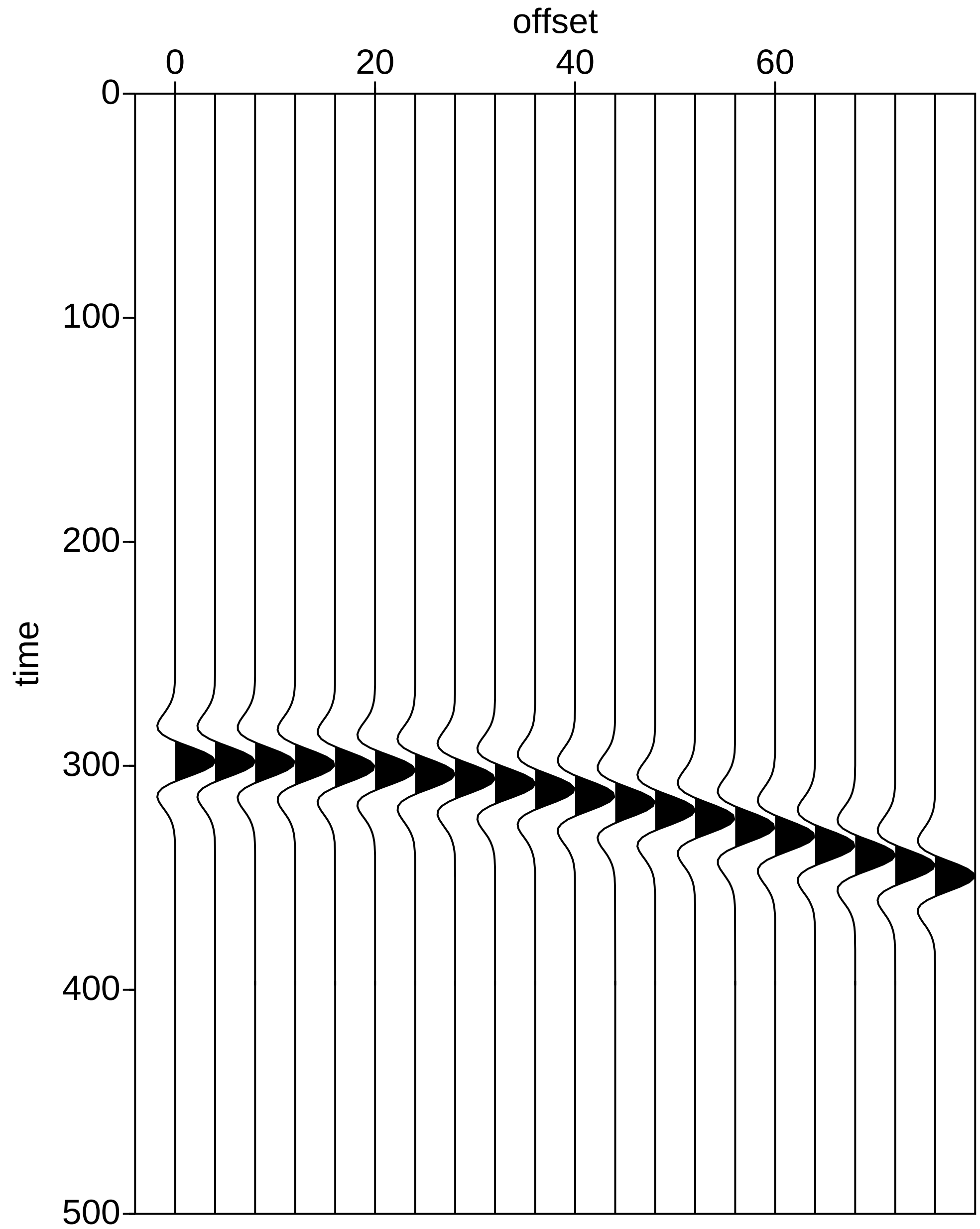

Figure 4.4 Simulated dated generated by the single spike reflectivity. Data was generated by running the single spike reflectivity and the model source through the forward map.

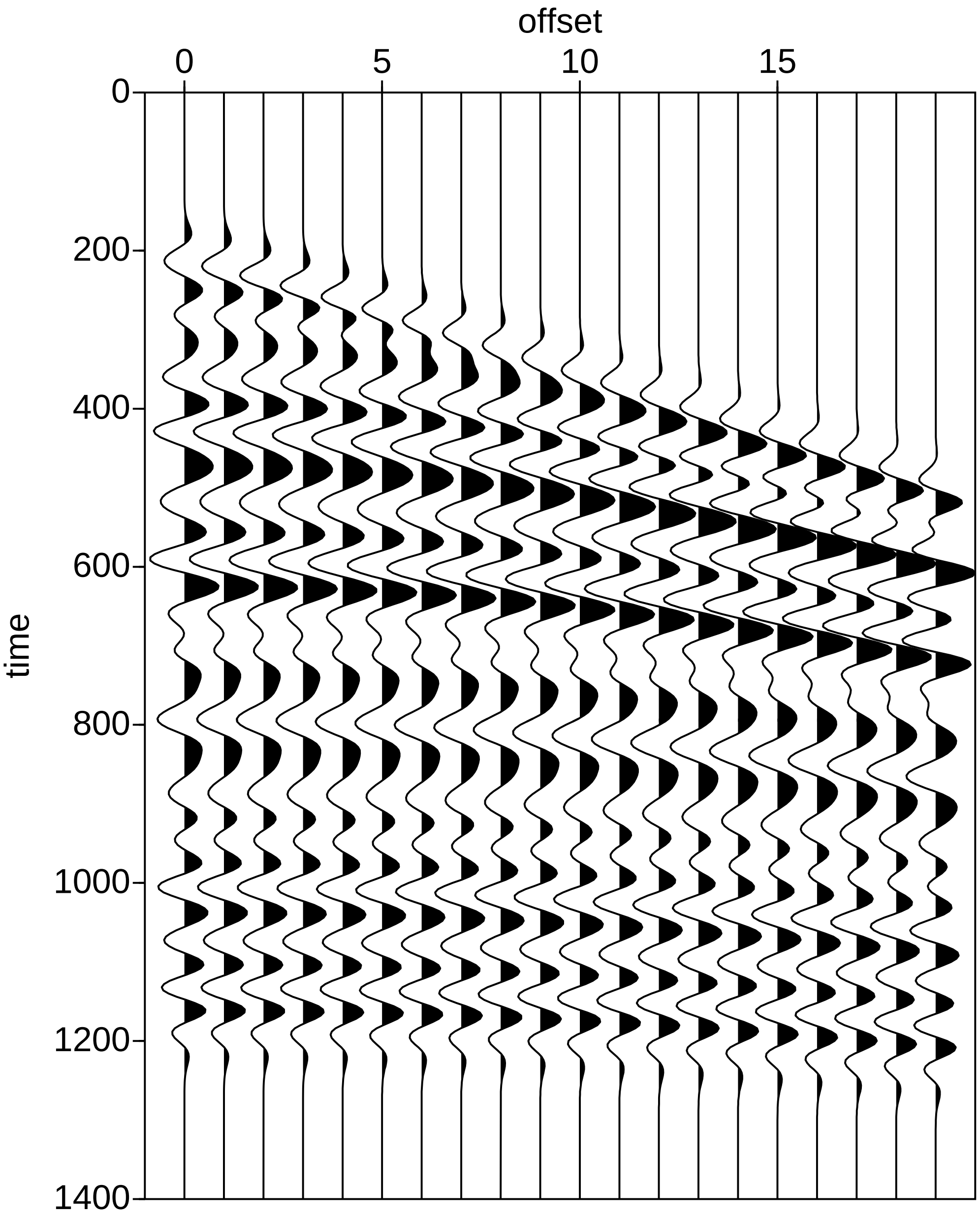

Figure 4.5 Simulated dated generated by the random spike reflectivity Data was generated by running the random spike reflectivity and the model source through the forward map.

4.2.1 Algorithm Ranking

This subsection will describe the evidence for the superiority of some algorithms over others, and the fact that the source and reflectivity obtained from reducing the relative error in the data residuals below 5% can be markedly different from the source and reflectivity used to generate the simulated data.

The section consists of 2 tables and a few example source and reflectivity plots. It should be noted that all the algorithms successfully reduced the objective function. Some algorithms, however, were faster than others at reducing the objective function.

Table 4.1 displays the results from the single spike reflectivity inversions. The table plots algorithms vs. iteration count and computation time on a dedicated Pentium II. The convergence criteria for these experiments was that the relative error in the data residual be less than 5%.

$$\frac{||b_{sim} - b_{guess}||}{||b_{sim}||} < .05$$

The optimization techniques can be broken into either the linear category, or the non-linear category. For a description of each optimization technique see section 3.3.

Alternation is the sole technique belonging to the linear category. Looking at the times for reducing the objective function, it is clear that Alternation is not the most successful technique of all those employed. Specifically, all of the non-linear techniques are faster at reducing the objective function. This is not surprising, since it has been shown by Kennett and Sambridge [8], that the subspace techniques can be superior to alternation for some non-linear problems in geophysics. Table 4.1 shows that for this particular inverse problem limited memory BFGS is the most successful algorithm in terms of reducing the objective function.

Table 4.1 indicates that the number of iterations taken by the alternation algorithm is sometimes less than the number taken by other algorithms. This number, however, is not indicative of the efficiency of the algorithm. Associated with each iteration are two sub-iterations of the trust region algorithm (one in each of the two alternation directions). For these experiments 20 iterations of the trust region algorithm were allowed in each of the 2 alternation directions. It was often the case that the source direction required less than 20 iterations to converge, while the reflectivity direction used all 20 iterations.

Table 4.2 displays the results from the random reflectivity inversions. The table plots algorithms vs. iteration count and computation time on a dedicated Pentium II. The convergence criteria for these experiments was that the relative error in the data residual be less than 5%.

As was the case with the single spike reflectivity, the alternation algorithm was the slowest to reduce the objective function. As the inverse problem became harder, however, the difference between the various algorithms became more apparent. Specifically, the trust region algorithm and limited memory BFGS algorithms clearly stand apart from the other algorithms in terms of rapidly reducing the objective function.

Note that the trust region and limited memory BFGS algorithms are approximately 100 times faster than alternation. This is remarkable given that previously published attempts at joint inversion for the source and reflectivity use only alternation.

Due to the nature of the inverse problem, however, reductions in the objective function rarely translate to an accurate inversion output in terms of the relative error of the model parameters. For example, figure 4.6 and figure 4.7 show the source and reflectivity output from the trust region experiment on the single reflector. Clearly this source and reflectivity are far from the model source and reflectivity. Figure 4.8 and 4.9 show the output from the trust region experiment on the random reflector. In this case the source retrieved from the inversion is accurate, but the reflectivity is lacking in high frequency information relative to the model reflectivity. The source and reflectivity output from other optimization algorithms is similar.

ALGORITHM	ITERATIONS	TIME
Trust Region	6	0 minutes 8 seconds
Limited Memory BFGS	29	0 minutes 3 seconds
6 D Subspace	38	0 minutes 37 seconds
2 D Subspace	132	0 minutes 43 seconds
Alternation	22	3 minutes 7 seconds

Table 4.1 Speed of various algorithms using the single spike reflectivity and a convergence criteria that the relative error in the data residual be less than 5%. The initial guess reflectivity consisted of the zero vector. The initial guess source consisted of the model source shifted away from zero time by 0.01 seconds and scaled by a factor of 0.5. Inversion parameters are identical to the forward map parameters

ALGORITHM	ITERATIONS	TIME
Trust Region	25	0 minutes 56 seconds
Limited Memory BFGS	2093	1 minutes 0 seconds
6 D Subspace	971	22 minutes 8 seconds
2 D Subspace	4341	32 minutes 5 seconds
Alternation	426	127 minutes 41 seconds

Table 4.2 Speed of various algorithms using the random reflectivity and a convergence criteria that the relative error in the data residual be less than 5%. The initial guess reflectivity consisted of the zero vector. The initial guess source consisted of the model source shifted away from zero time by 0.01 seconds and scaled by a factor of 0.5. Inversion parameters are identical to the forward map parameters

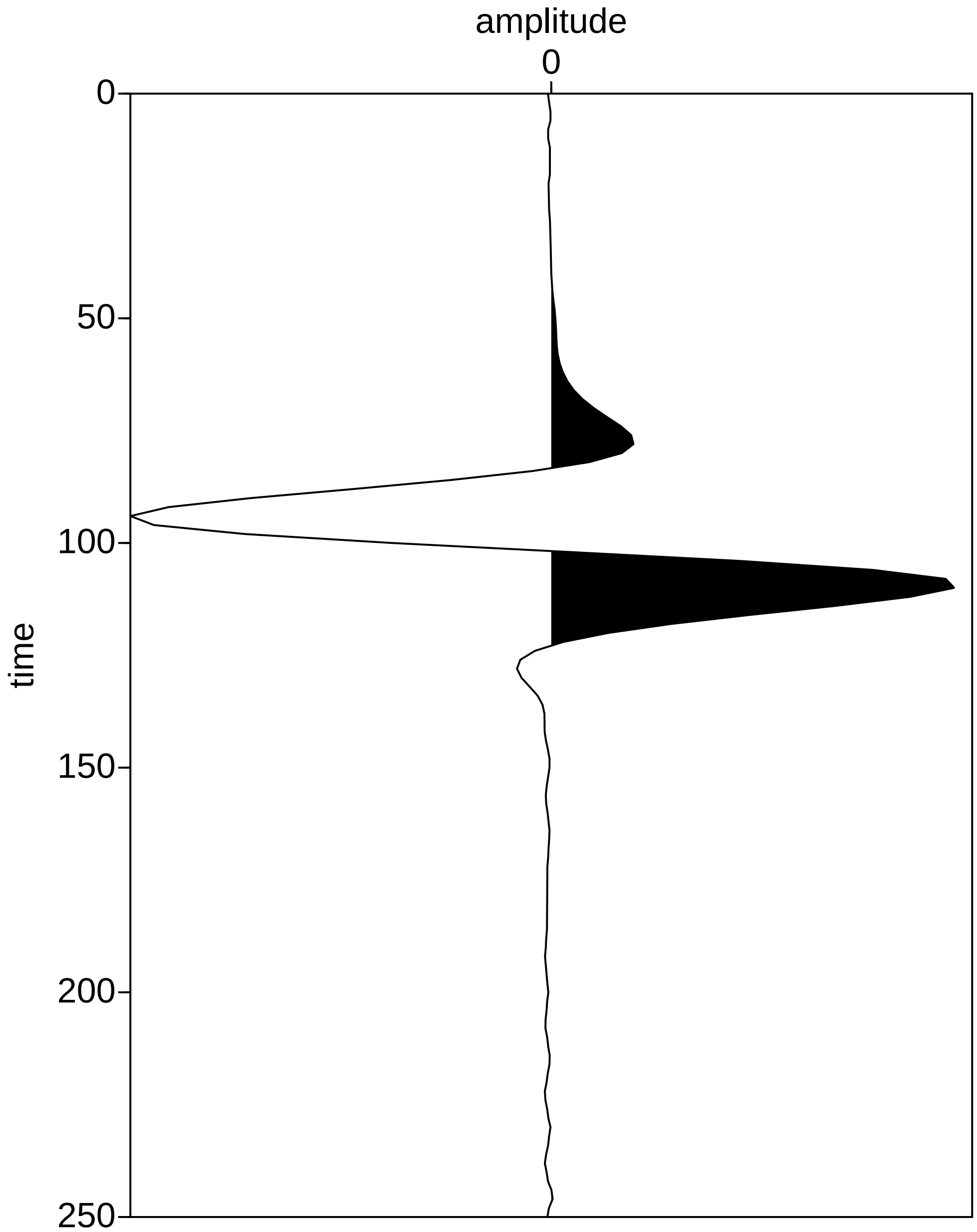

Figure 4.6 Source inversion result for spike reflectivity using the Alternation algorithm

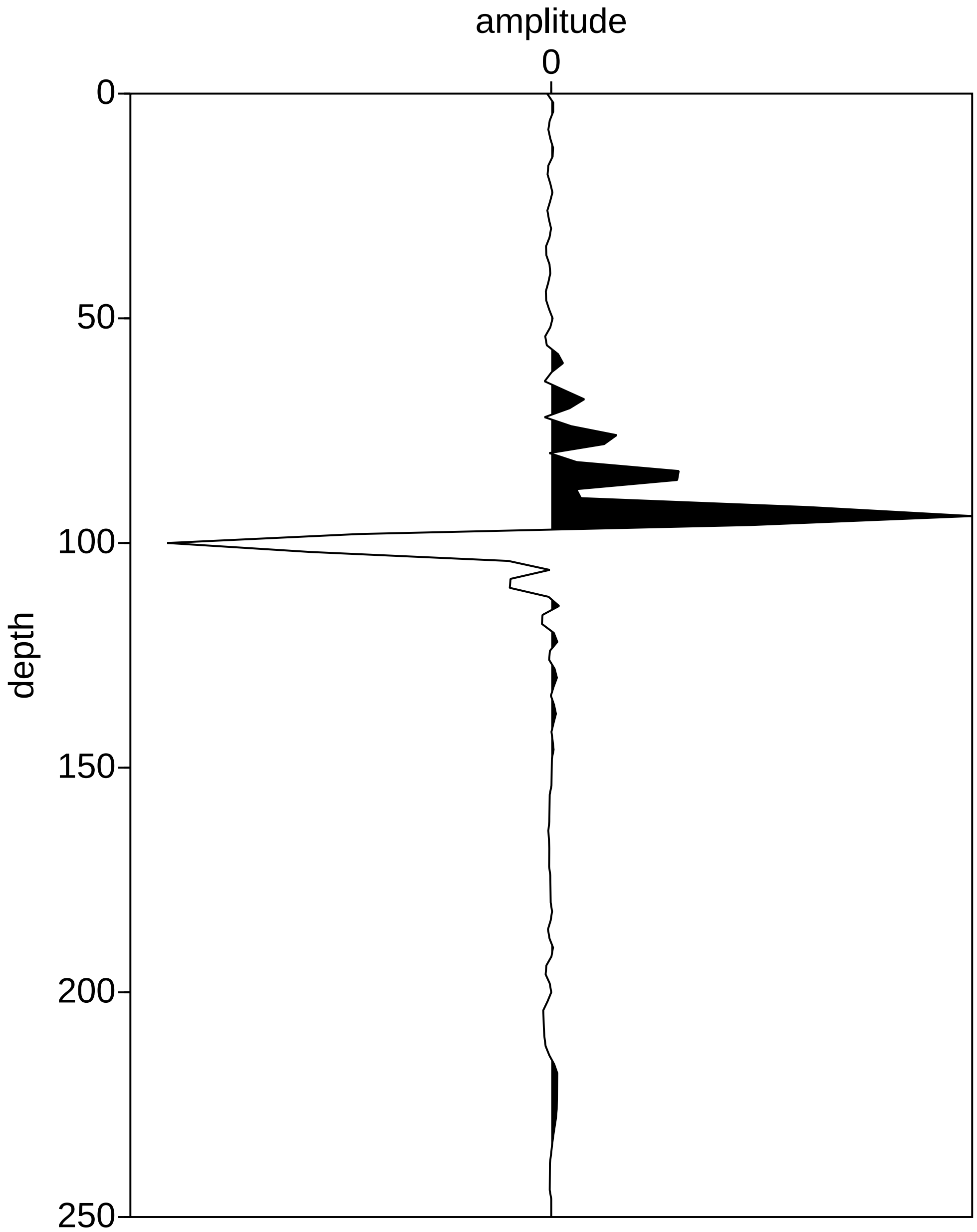

Figure 4.7 Reflectivity inversion result for spike reflectivity using the Alternation algorithm

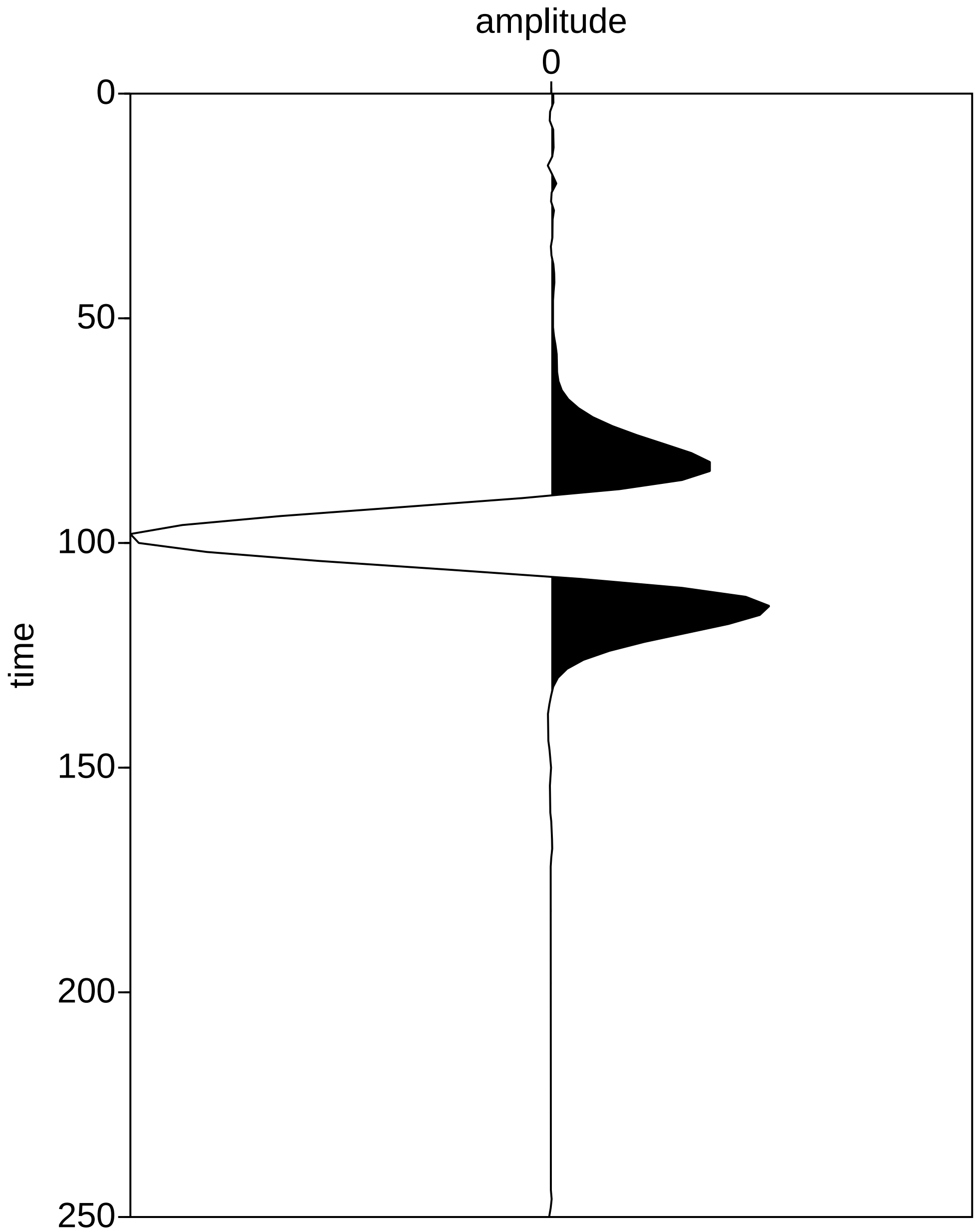

Figure 4.8 Source inversion result for random reflectivity using the Alternation algorithm

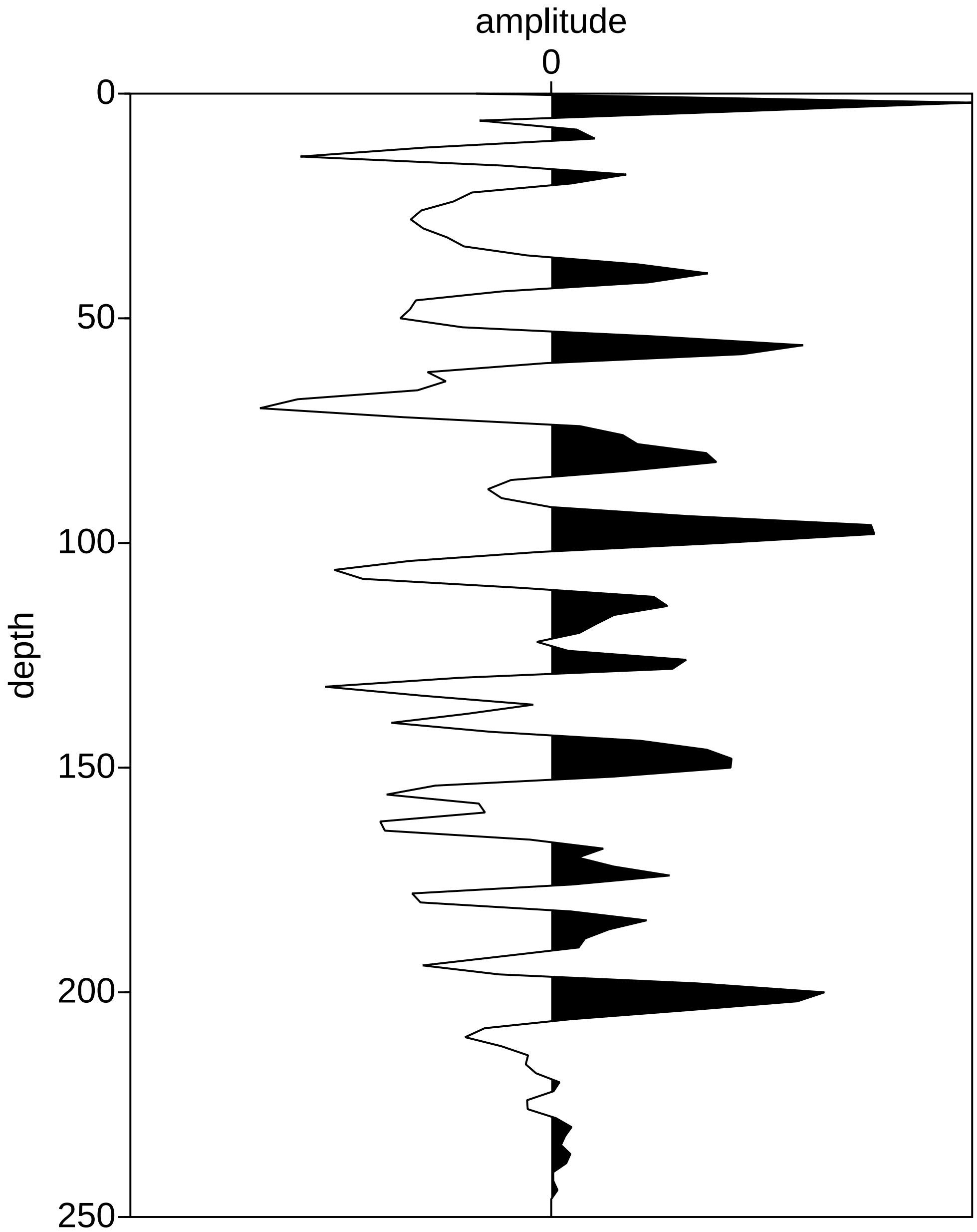

Figure 4.9 Reflectivity inversion result for random reflectivity using the Alternation algorithm

4.2.2 Apparent Non-Uniqueness

This subsection details the numerical evidence showing that when the source is oscillatory and the reflectivity has very compact support, the objective function can be reduced below 10^-6, but the source and reflectivity that is output from the inversion can be markedly different from the source and reflectivity used to generate the simulated data. In other word, this subsection gives numerical evidence for a non-uniqueness of the inverse problem.

Table 4.3 displays the results from the single spike reflectivity inversions. The table plots algorithms vs. iteration count, computation time on a dedicated Pentium II at the final iteration, relative error of the final reflectivity and relative error of the final source. The convergence criteria for these experiments was that the value of the objective function be reduced below 10^{-6}.

Looking at the relative error columns in table 4.3 it is clear that Alternation is not the most accurate technique of all those employed. In fact, based strictly on the relative errors, it can be argued that alternation is the least successful. Relative error is computed in a standard way

$$E_R = \frac{||f_{model} - f_{result}||_2}{||f_{model}||_2}$$

for both the source and reflectivity. Because of the scale ambiguity inherent in this problem, however, the model and the result are first scaled to have a norm of one. Hence, relative error is computed as follows:

$$E_R = \left\| \frac{f_{model}}{||f_{model}||_2} - \frac{f_{result}}{||f_{result}||_2} \right\|_2$$

Figure 4.10 and figure 4.11 show the source and reflectivity resulting from the alternation algorithm. These results do not bear a strong resemblance to the model source and reflectivity. The results from the trust region algorithm, however, bear

ALGORITHM	ITERATIONS	TIME	ERROR SOURCE	ERROR REFLECTIVITY
Trust Region	3997	1 hour 44 minutes	0.1410	0.2453
Limited Memory BFGS	3238	0 hours 4 minutes	0.3374	0.4466
6 D Subspace	64751	18 hours 9 minutes	0.6178	0.8236
2 D Subspace	23297	2 hours 7 minutes	1.1776	1.2349
Alternation	321	1 hour 3 minutes	1.3451	1.3451

Table 4.3 Speed and accuracy of various algorithms using the single spike reflectivity. The initial guess reflectivity consisted of the zero vector. The initial guess source consisted of the model source shifted away from zero time by 0.01 seconds and scaled by a factor of 0.5. Inversion parameters are identical to the forward map parameters

a noticeably closer resemblance as evidenced by figure 4.12 and figure 4.13 as well as the relative errors from table 4.3. In terms of the relative error the trust region algorithm performs the best in retrieving the model source and reflectivity. These results suggest that even though the value of the objective function is extremely small, the source and reflectivity output from the inversion can be markedly different.

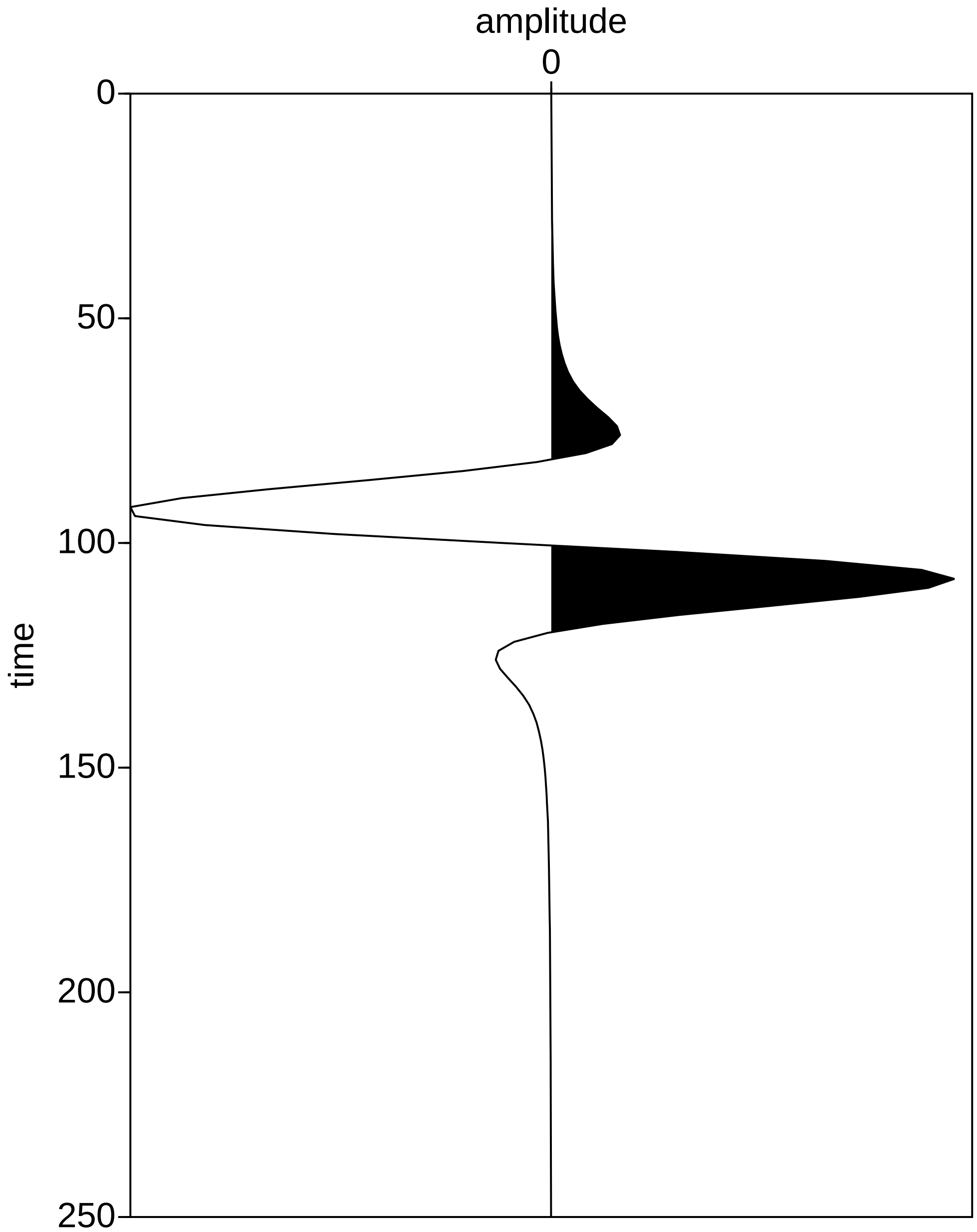

Figure 4.10 Source inversion result for spike reflectivity using the Alternation algorithm

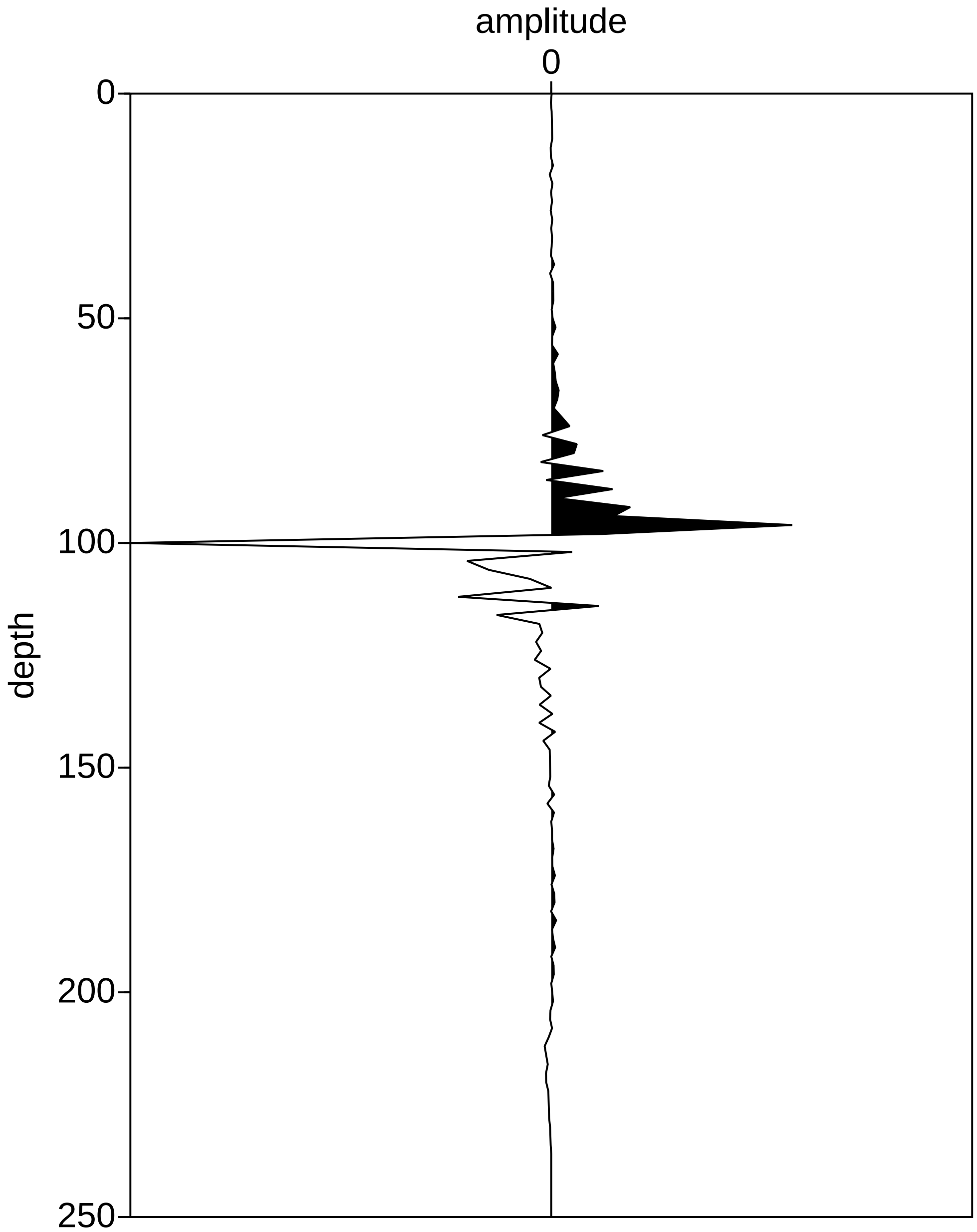

Figure 4.11 Reflectivity inversion result for spike reflectivity using the Alternation algorithm

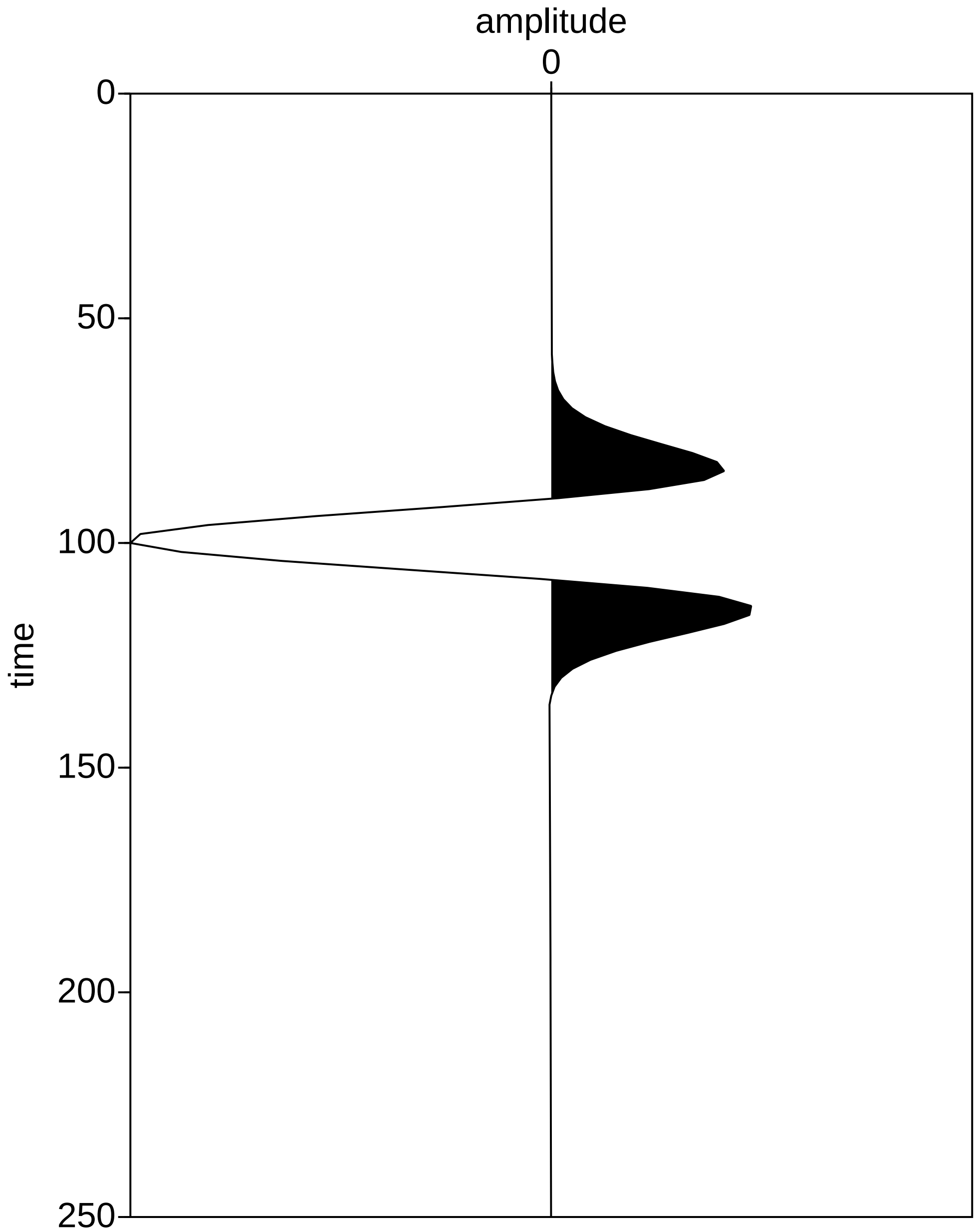

Figure 4.12 Source inversion result for spike reflectivity using the Trust Region algorithm

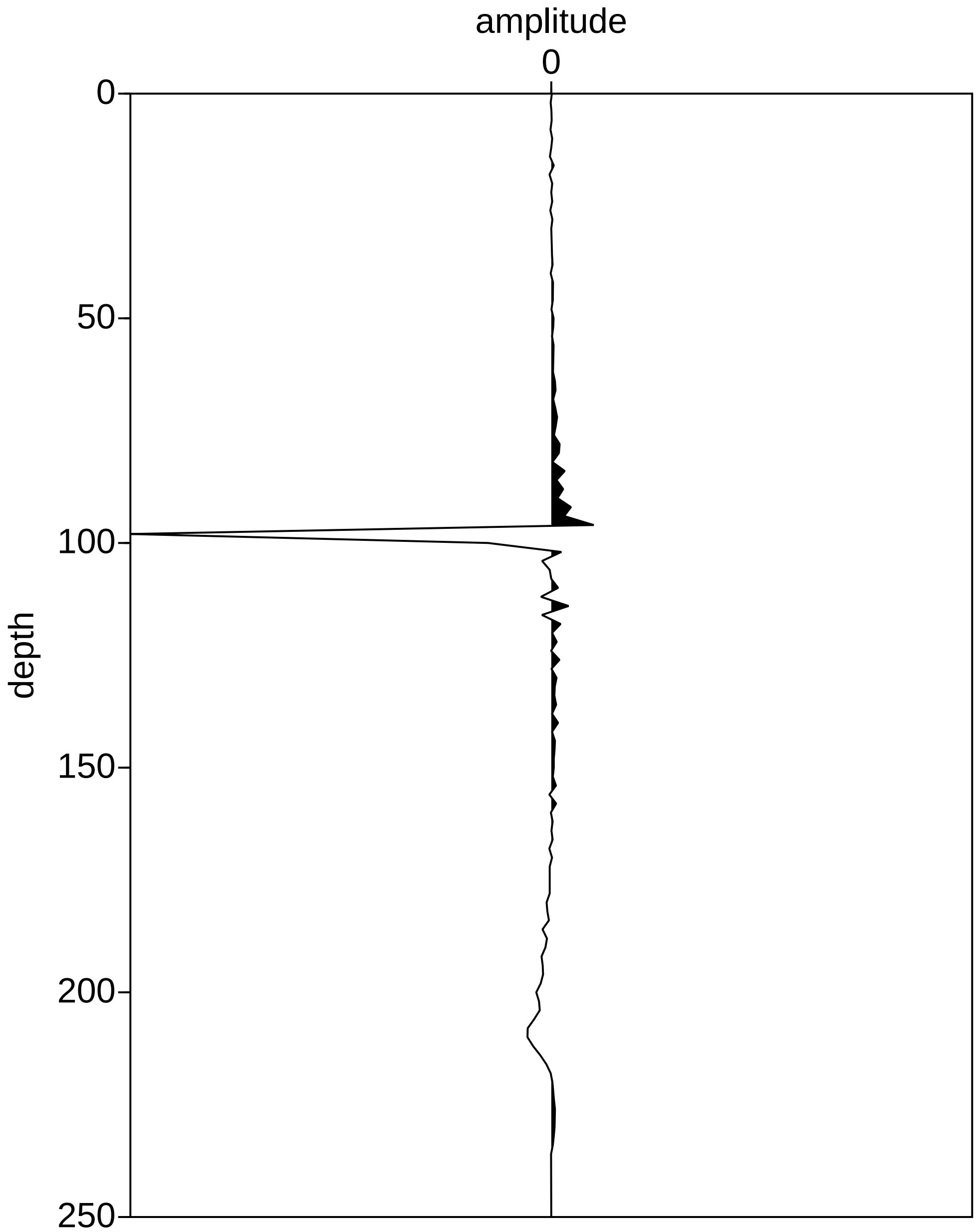

Figure 4.13 Reflectivity inversion result for spike reflectivity using the Trust Region algorithm

Table 4.4 displays the results from random reflectivity inversions. The table plots algorithms vs. iteration count, computation time on a dedicated Pentium II at the final iteration, relative error of the final reflectivity and relative error of the final source. The plots consist of the final source and reflectivity for each algorithm. The convergence criteria for these experiments was that the value of the objective function be reduced below 10^{-6}.

ALGORITHM	**ITERATIONS**	**TIME**	**ERROR SOURCE**	**ERROR REFLECTIVITY**
Trust Region	14459	8 hours 41 minutes	2.4e-5	0.1615
Limited Memory BFGS	15811	0 hours 24 minutes	0.0092	0.7060
6 D Subspace	108314	42 hours 8 minutes	0.0003	0.3066
2 D Subspace	141353	17 hours 53 minutes	0.0005	0.3983
Alternation	9096	56 hours 5 minutes	0.0004	0.2603

Table 4.4 Success of various algorithms using the random spike reflectivity. The initial guess reflectivity consisted of the zero vector. The initial guess source consisted of the model source shifted away from zero time by 0.01 seconds and scaled by a factor of 0.5. Inversion parameters are identical to the forward map parameters

As was the case with the single spike reflectivity, the trust region algorithm was the most effective at obtaining an accurate source and reflectivity. Comparing the results of the trust region algorithm visually in figure 4.14 and figure 4.15 and via the relative error in table 4.4, it is clear that this algorithm accurate reproduced the source and reflectivity. As was the case with the single spike experiment.

The alternation algorithm also accurately reproduced the source and reflectivity. This is evidenced by the relative error from table 4.4, figure 4.16 and figure 4.17. Note, however that the difference between the alternation results and the trust region results is dramatically reduced in comparison to the single spike reflectivity experiments. In fact, in all of our experiments, only the single spike reflectivity experiments generated sources that varied so greatly in shape across algorithms.

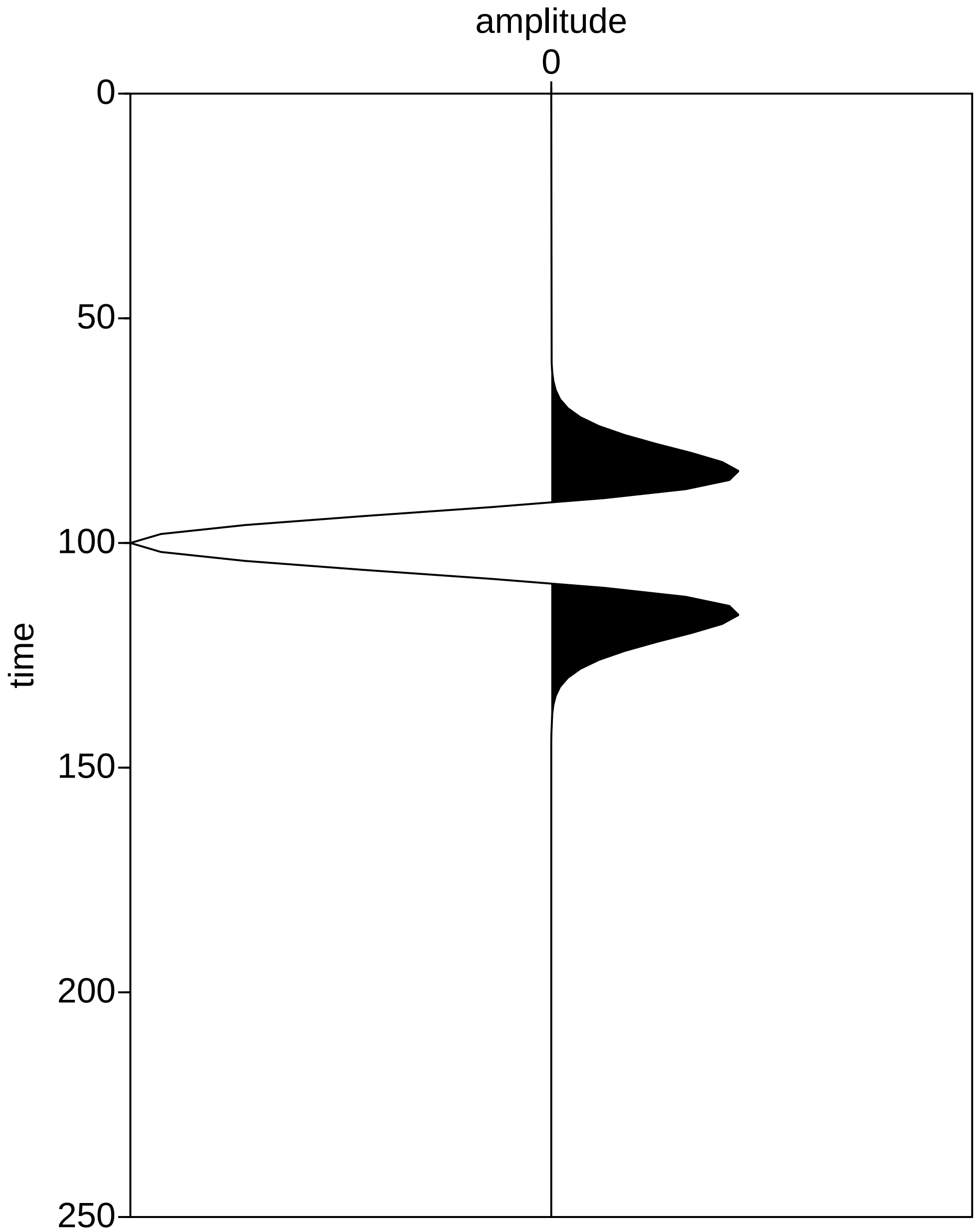

Figure 4.14 Source inversion result for random reflectivity using the Trust Region algorithm

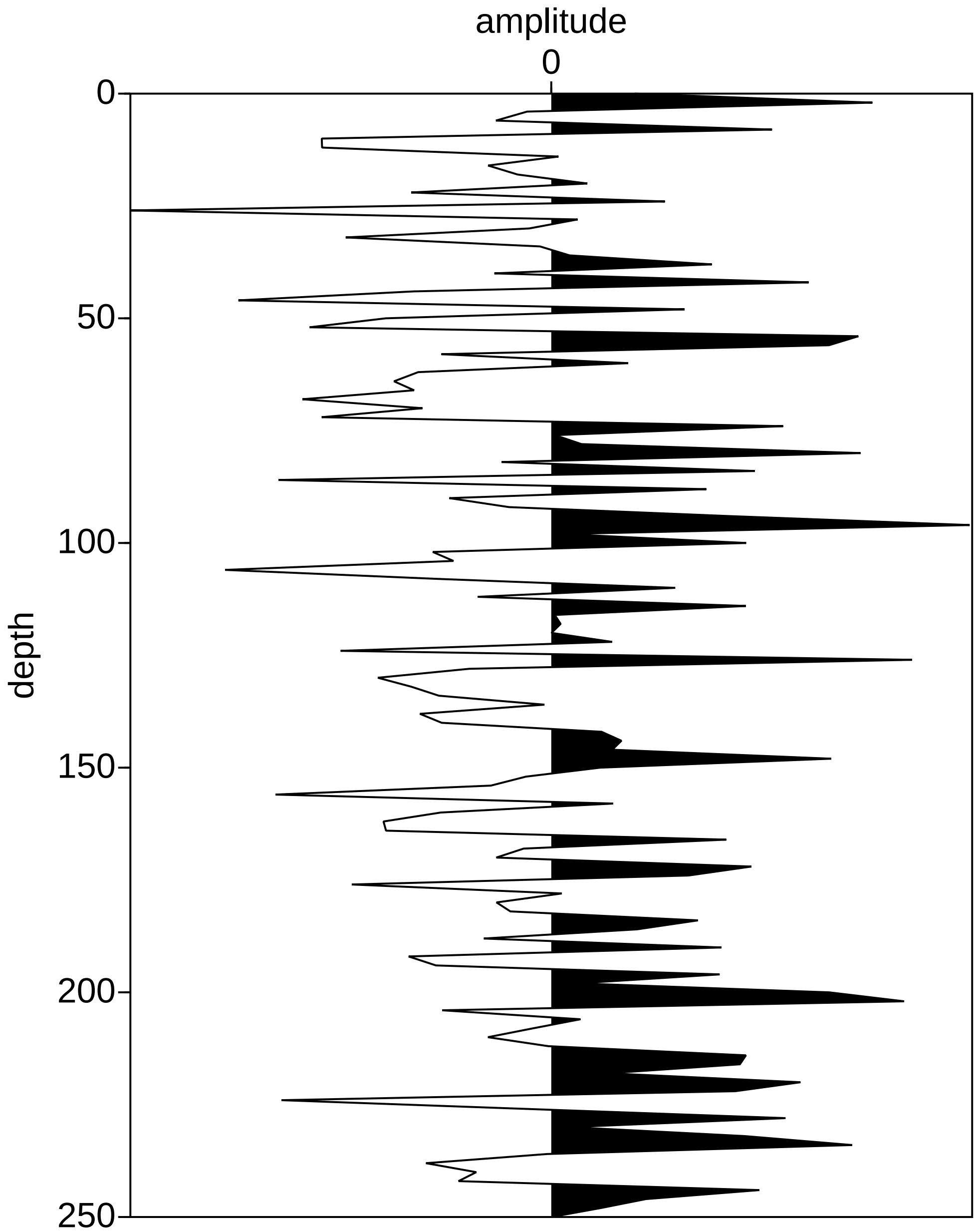

Figure 4.15 Reflectivity inversion result for random reflectivity using the Trust Region algorithm

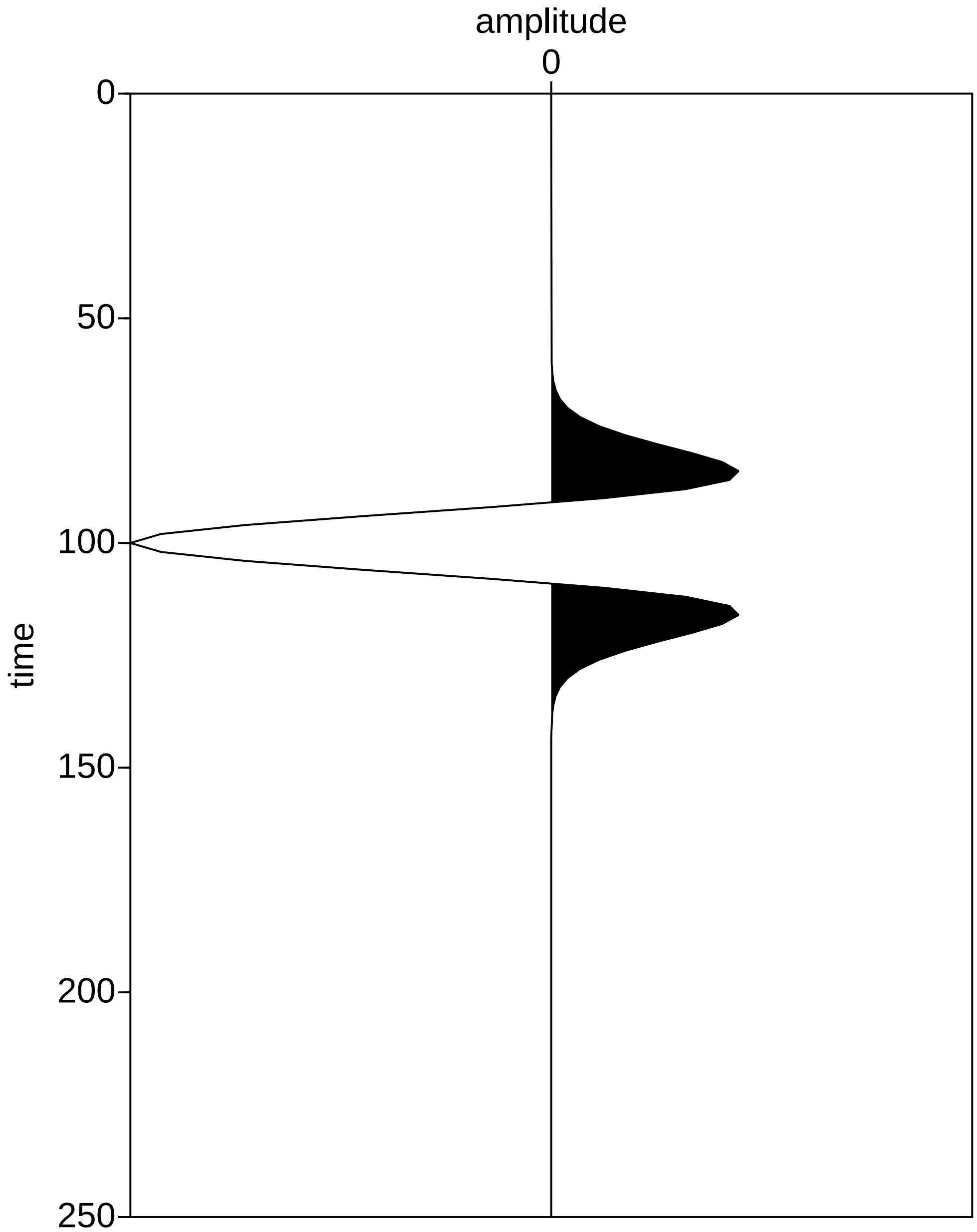

Figure 4.16 Source inversion result for random reflectivity using the Alternation algorithm

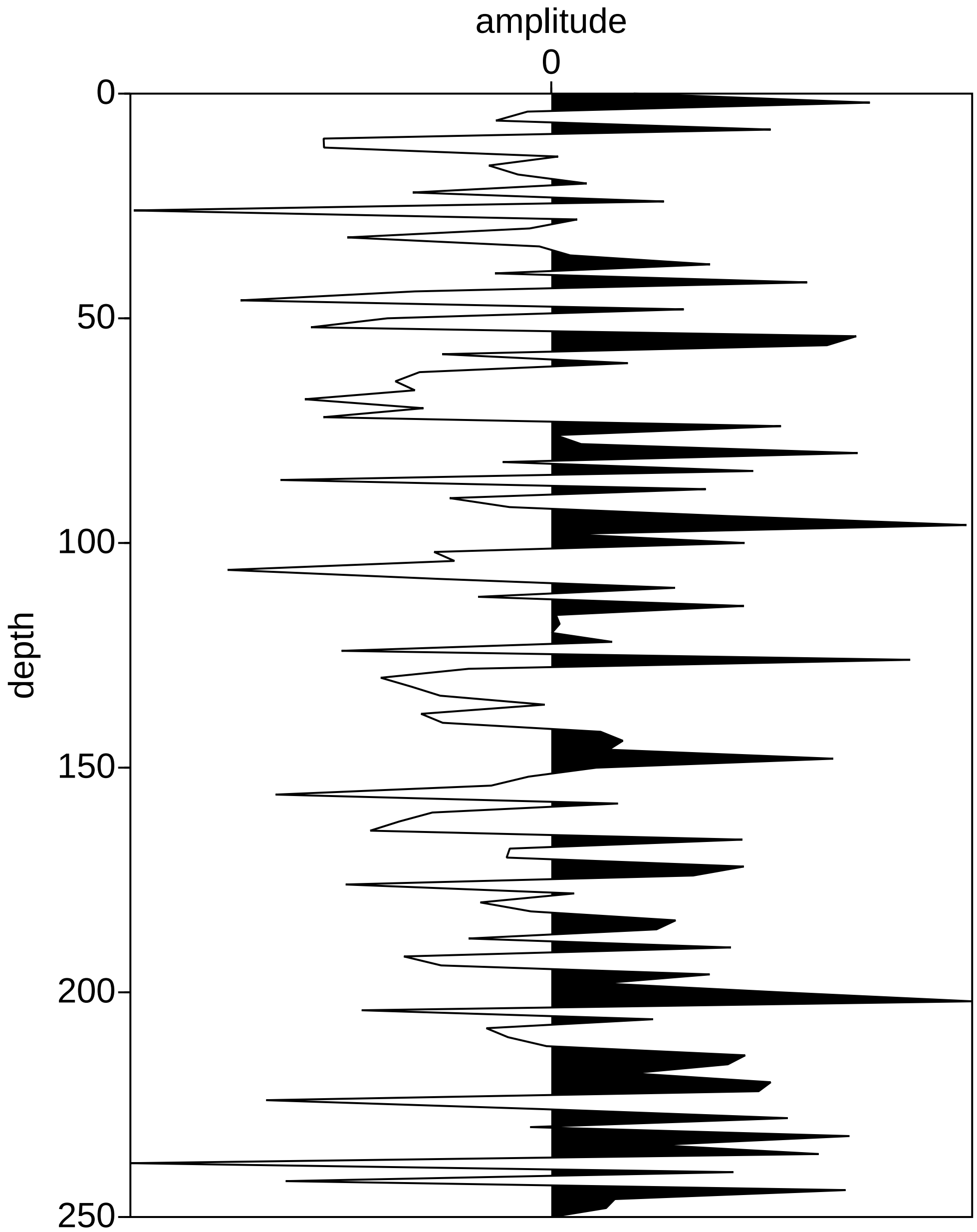

Figure 4.17 Reflectivity inversion result for random reflectivity using the Alternation algorithm

To investigate the possible non-uniqueness in the single spike reflectivity inverse problem, experiments were performed where each algorithm was allowed a progressively increasing number of maximum iterations. For example, 5 experiments were done where the trust region algorithm was allowed 50, 100, 1000, 1100, 1200, 1300, 1400 and 1500 iterations of the 3997 iterations required to converge.

For each algorithm a similar behavior results. First, the algorithm quickly (in less than 50 iterations) moves to a place on the objective function where the source and reflectivity have shapes markedly different from the model source and reflectivity (see figure 4.18 and figure 4.19). This move greatly reduces the value of the objective function (from 10^3 to 10^{-1}). Notice that the reflectivity at the 50th iteration has a form similar to the Hilbert Transform of a delta function (figure 4.20 shows the Hilbert Transform of a delta function).

The algorithm then slowly (over hundreds or thousands of iterations) moves toward a place on the objective function where the source and reflectivity closely resemble the model source and reflectivity.

During the course of this slow movement, the reflectivity varies smoothly. In other words, the reflectivity at the 50th iteration looks very similar to the reflectivity at the 300th iteration (see figure 4.19 and figure 4.22 respectively). The difference between the two amounts to a slight decrease in the amplitude of the secondary spikes, and a slight shift of the primary spike.

The source varies smoothly as well. The difference between the source at the 50th iteration and the 300th iteration amounts to a small phase shift and a small decrease in the amplitude of the primary lobe (see figure 4.18 and figure 4.21).

Similar behavior is observed for each optimization algorithm. A characteristic worth noting is that the algorithms with the lower relative errors in table 4.3 moved away from the Hilbert Transform of the model reflectivity faster. The fact that this behavior is ubiquitous across algorithms coupled with the fact that all of the results reduced the objective function below 10^{-6} implies that some sort or family of computationally non-unique solutions exist for the spike reflectivity inverse problem. By computationally non-unique, we mean that the objective function has values less than 10^{-6}

These results raise a number of questions

- Is this spike reflectivity non-uniqueness a contradiction of the uniqueness results of Minkoff and Symes [11]?
- Why does the non-uniqueness only occur in the spike reflectivity experiments?
- Does the Hilbert transform play a pivotal role in the generation of the family of solutions?

We will address these questions in the following section.

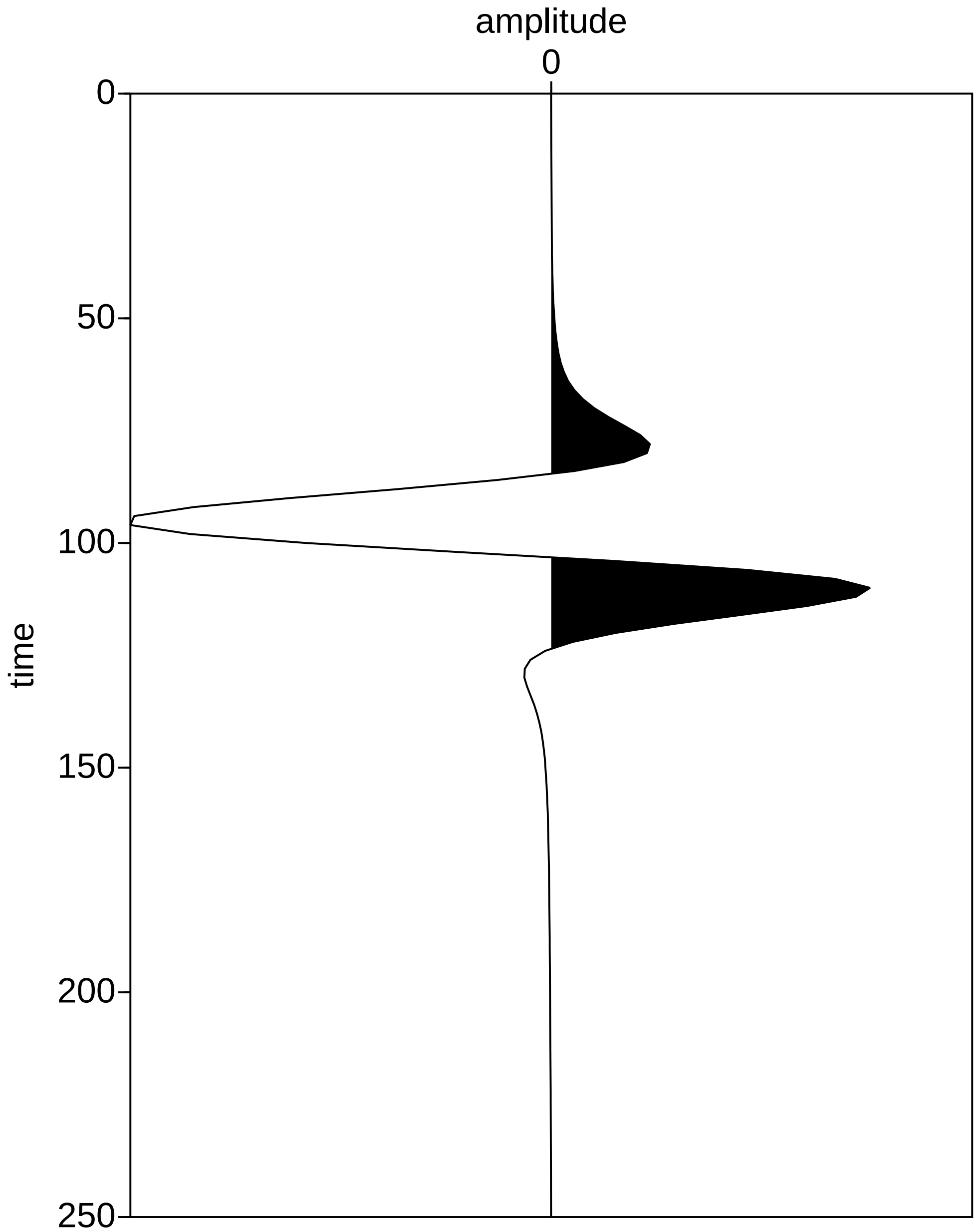

Figure 4.18 Source inversion result for spike reflectivity using the trust region algorithm halted at the 50th iteration

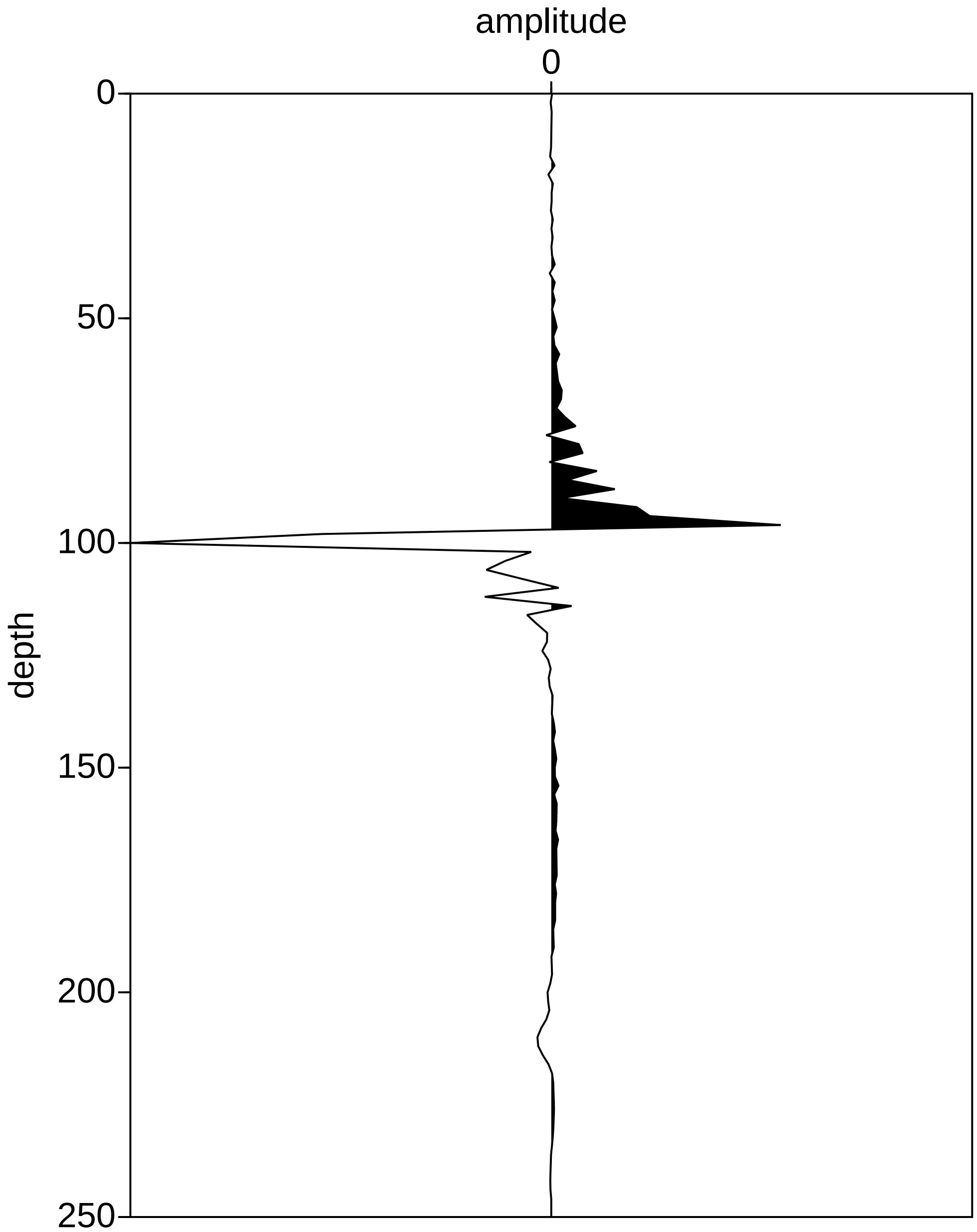

Figure 4.19 Reflectivity inversion result for spike reflectivity using the trust region algorithm halted at the 50th iteration

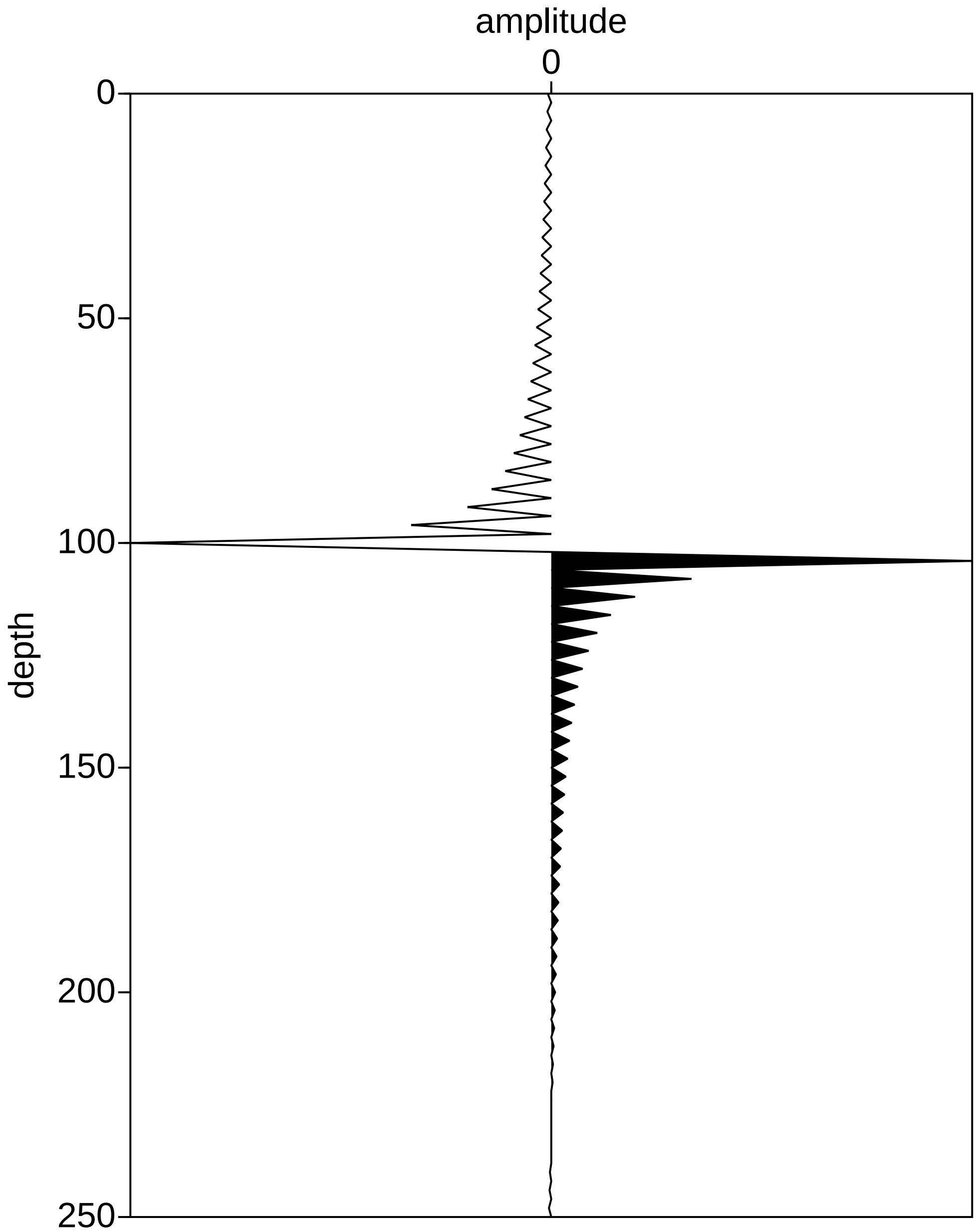

Figure 4.20 Hilbert Transform of a spike reflectivity located slightly deeper than the model reflectivity.

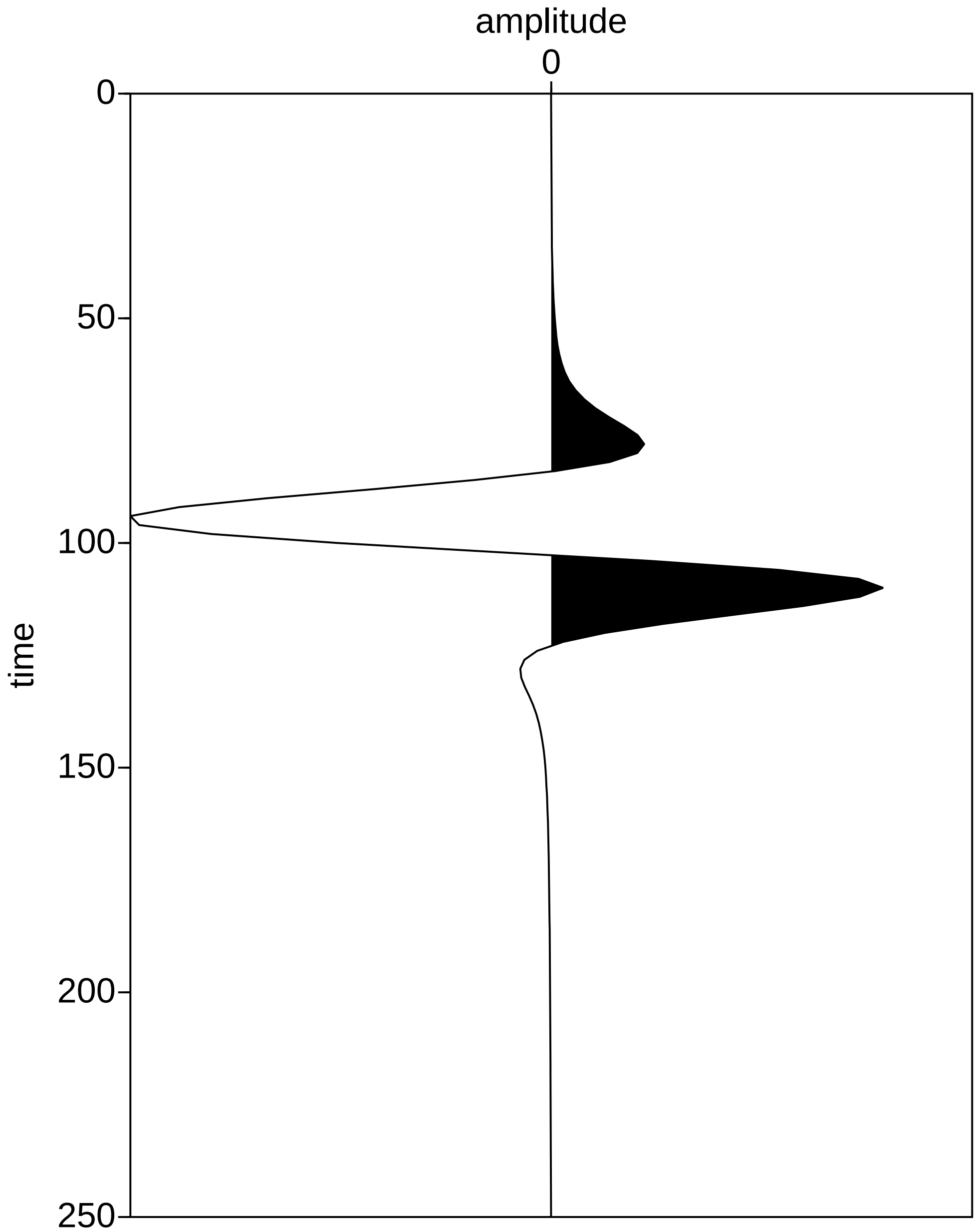

Figure 4.21 Source inversion result for spike reflectivity using the trust region algorithm halted at the 300th iteration

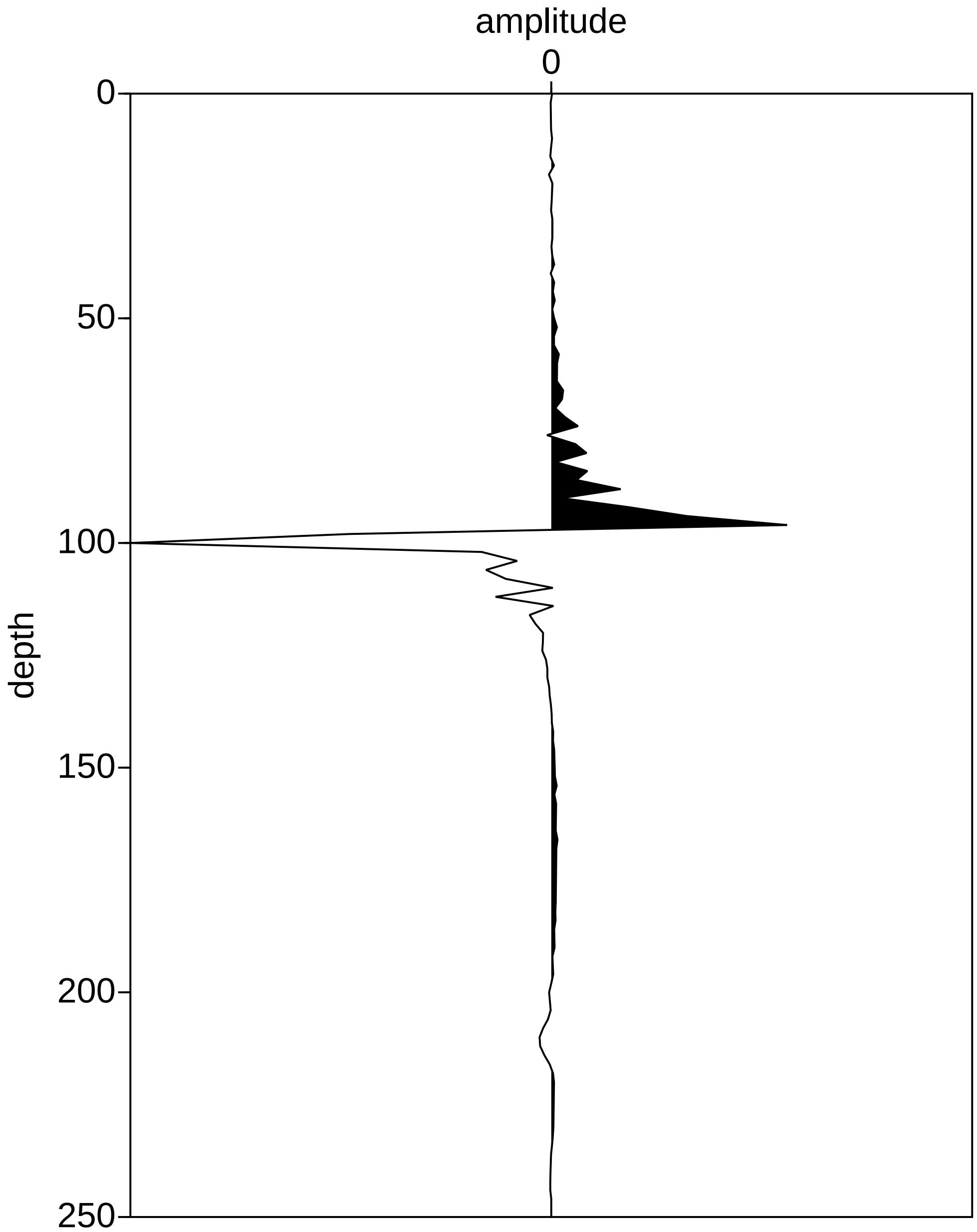

Figure 4.22 Reflectivity inversion result for spike reflectivity using the trust region algorithm halted at the 300th iteration

4.3 Uniqueness

The numerical experiments in subsection 4.2.2 give compelling evidence for non-unique solutions to the spike reflectivity inverse problem. At first glance this appears to fly in the face of the theoretical results of Minkoff and Symes [11]. In this section it is shown for the first time that the output of the convolutional model does not uniquely determine a source and reflectivity, at least in terms of the realities of computation. In addition, it is explained why this result does not contradict the results of Minkoff and Symes. We also explain the role that the Hilbert transform plays in the non-uniqueness.

We begin with an analysis showing why the spike reflectivity inversion experiment can generate effectively non-unique solutions. The non-uniqueness of the solution to the inverse problem plagues the convolutional model. For example, if the data set $b(t,x)$ consists only of a single trace, it is ridicules to even mention a unique solution to the inverse problem.

To understand why, recall that the convolution in time is equivalent to multiplication in the Fourier domain. Thus, we can write

$$b\widehat{(x,\omega)} = R\widehat{(\omega,x)}\widehat{f(\omega)}$$

where the hat indicates a Fourier transform, and ω is frequency. With only a single trace, inversion for $f(t)$ and $r(z)$ involves the determination of two frequency (w) factors from one.

In addition, a scale ambiguity exists in the convolutional model. The scale ambiguity can be summarized with the following equation

$$b(t,x) = f(t) * r(\phi(t,x)) = (af(t)) * \left(\frac{r(\phi(t,x))}{a}\right)$$

The scale ambiguity, however, is easily dealt with by constraining the size of either the source or the reflectivity. For example

$$\int dt f^2(t) = 1$$

With the size of either the source or reflectivity constrained, the exact extent to which non-uniqueness plagues the convolutional model is still an unanswered question. It is known that for a constant velocity medium the perturbational plane wave seismogram [1] [9] uniquely determines perturbations to the source and the velocity profile. In addition, for a slowly varying background velocity, perturbations in the source and reflectivity are uniquely determined by a perturbed seismogram within some band pass [11]. These results, however, do not speak directly to the non-uniqueness of inversion results from the convolutional model in the offset-time domain (the domain in which most geophysical applications take place) with non-zero fit error.

Evidence from numerical experiments (see section 4.2) suggests that if the support of $r(z)$ is small (a delta function), the inversion results can become horrifyingly non-unique in terms of the shape of the source and reflectivity. It is shown in the following derivation that if $r(z)$ is a delta function, the results of the inversion will be computationally non-unique.

Recall that in the previous subsection the reflectivity output from the inversion had a form similar to the Hilbert Transform of the delta function. From the definition of the Hilbert transform

$$\widehat{Hg}(\omega) = \begin{cases} i\hat{g}(\omega) & \omega > 0 \\ -i\hat{g}(\omega) & \omega < 0 \end{cases}$$

where $\hat{g}$ denotes the Fourier transform of g, it can be concluded that

$$Ha_1(t) * Ha_2(t) = -a_1(t) * a_2(t)$$

or in terms more familiar

$$[-Hf(t)] * H[r \circ \phi(t,x)] = f(t) * r \circ \phi(x,t)$$

or

$$[-Hf(t)] * (Hr) \circ \phi(x,t) - Hf(t) * \{H[r \circ \phi(x,t)] - (Hr) \circ \phi(x,t)\} = f(t) * r \circ \phi(x,t)$$

One of the properties of the convolution operator is that convolving something regular with something oscillatory results in something very small. In the experiments performed to test the code the source was always a Ricker wavelet (something oscillatory). If we can show that the term in the curly brackets is something regular, then convolving that term with the source will produce something small and

$$[-Hf(t)] * (Hr) \circ \phi(x,t) \doteq f(t) * r \circ \phi(x,t)$$

which in terms of the scale ambiguity inherent in the inverse problem means that the inversion result may be non-unique. Note that this non-uniqueness is not of the form typically experienced in inversion of the convolutional model. In other words it is not the case that the source and reflectivity from output have the same shape but differ by a scale factor, or the source and reflectivity are determined only within a given pass band. In other words for the spike reflectivity inverse problem, the source and reflectivity derived via inversion may have a markedly different shape and scale within a given pass band.

To determine if an operation results in an output which is regular, an inner product is computed with the operation and a test function $u \in C^{\infty}(\Re)$. If the inner product is equivalent to an inner product with the test function u and something smooth, then the operation is regular. So we wish to know if

$$(\{H[\delta \circ \phi(x,t)] - (H\delta) \circ \phi(x,t)\}, u(t)) = (g(t), u(t))$$

where $g(t)$ is smooth. We can write this as left side inner product as

$$(\delta \circ \phi(x,t), Hu(t)) - (H\delta, \psi'(z)u \circ \psi(z))$$

since H is self adjoint, and $\psi(z)$ is the inverse of $\phi(x,t)$. Recombining the inner product gives

$$(\delta(z), \psi' Hu \circ \psi(z) - H(\psi'(z)u \circ \psi(z)))$$

and by the reproducing property of the delta function, this is

$$\psi'(0)Hu(\psi(0)) - H(\psi' u \circ \psi)(0)$$

Hence we wish to know is the above form is smooth.

Without a loss of generality, we can assume that $\phi(0) = 0$. This implies that $\psi(0) = 0$. Thus we must show that

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) \tag{4.1}$$

is smooth. Let us begin by examining part of the first term in 4.1, $Hu(0)$. Using the integral form of the Hilbert Transform we have

$$Hu(0) = \lim_{\epsilon \to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \frac{u(s)}{-s}$$

Now define a function $m(s)$ as follows

$$m(s) = 1 \text{ for } |s| < 1$$

$$m(s) = 0 \text{ for } |s| << 0$$

Returning to the integral form of the Hilbert Transform we have

$$Hu(0) = \lim_{\epsilon \to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \frac{1 - m(s)}{-s} u(s) + \lim_{\epsilon \to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \frac{m(s)}{-s} u(s)$$

By the definition of $m(s)$, the first term on the right hand side is defined at the limit. In addition, this integrand is smooth, so the entire first term is smooth. In other words we can write the above expression as

$$\psi'(0)Hu(0) = (p_1(t), u(t)) + \psi'(0) \lim_{\epsilon\to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \frac{m(s)}{-s} u(s)$$

where p_1 is a smooth function

We now examine the integral form of the second term of equation 4.1.

$$-H(\psi' u \circ \psi)(0) = -\lim_{\epsilon\to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \frac{\psi'(s)u(\psi(s))}{-s}$$

We can again make use of the function $m(s)$ to show that

$$-H(\psi' u \circ \psi)(0) = (p_2(t), u(t)) - \lim_{\epsilon\to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \frac{\psi'(s)u(\psi(s))}{-s} m(s)$$

where $(p_2(t), u(t))$ is the value of the integral as $\epsilon \to 0$, and p_2 is a smooth function. Hence (4.1) becomes

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) =$$
$$(p(t), u(t)) + \psi'(0) \lim_{\epsilon\to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \frac{m(s)}{-s} u(s) - \lim_{\epsilon\to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \frac{\psi'(s)u(\psi(s))}{-s} m(s)$$

where $p = p_1 + p_2$. If we perform a change of variables ($t = \psi(s)$, $s = \phi(t)$) on the second integral pair we have

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) =$$
$$(p(t), u(t)) + \psi'(0) \lim_{\epsilon\to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \frac{m(s)}{-s} u(s) - \lim_{\epsilon\to 0} \int_{-\infty}^{\psi(-\epsilon)} + \int_{\psi(\epsilon)}^{\infty} dt \frac{u(t)}{-\phi(t)} m(\phi(t))$$

or

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) =$$
$$(p(t), u(t)) + \lim_{\epsilon\to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \psi'(0) \frac{m(s) - m(\phi(s))}{-s} u(s)$$
$$+ \lim_{\epsilon\to 0} \int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} ds \left[\psi'(0) - \frac{s}{\phi(s)} \right] m(\phi(s)) \frac{u(s)}{-s}$$

$$+\lim_{\epsilon\to 0}\left[\left(\int_{-\infty}^{-\epsilon}+\int_{\epsilon}^{\infty}\right)-\left(\int_{-\infty}^{\psi(-\epsilon)}+\int_{\psi(\epsilon)}^{\infty}\right)\right] ds\frac{u(s)}{-\phi(s)}m(\phi(s))$$

In the limit as $\epsilon \to 0$ the integrand of the first integral pair goes to zero. In addition, away from zero the part of the integrand not involving u is smooth. Therefore, we can again modify $p(t)$ to absorb the first part of the integrand and write

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t))$$
$$+\lim_{\epsilon\to 0}\int_{-\infty}^{-\epsilon}+\int_{\epsilon}^{\infty} ds\left[\psi'(0)-\frac{s}{\phi(s)}\right] m(\phi(s))\frac{u(s)}{-s}$$
$$+\lim_{\epsilon\to 0}\left[\left(\int_{-\infty}^{-\epsilon}+\int_{\epsilon}^{\infty}\right)-\left(\int_{-\infty}^{\psi(-\epsilon)}+\int_{\psi(\epsilon)}^{\infty}\right)\right] ds\frac{u(s)}{-\phi(s)}m(\phi(s))$$

where $p(t)$ is a smooth function. Now if we do a Taylor expansion of $\phi(s)$ we can say

$$\left[\psi'(0)-\frac{s}{\phi(s)}\right]=\frac{1}{\phi'(0)}-\frac{s}{\phi(0)+s\phi'(0)+\frac{s^2}{2}\phi''(0)+O(s^3)}$$

or

$$\left[\psi'(0)-\frac{s}{\phi(s)}\right]=\frac{1}{\phi'(0)}\left[1-\frac{1}{1+\frac{s}{2}\frac{\phi''(0)}{\phi'(0)}+O(s^2)}\right]$$

recalling that $\phi(0) = 0$. Now the denominator of the last term on the right hand side can be thought of as $1 + sv(s)$, where $v(s)$ is a smooth function. Then using the expansion of $\frac{1}{1+x} = 1 - x + \ldots$ and canceling terms we have

$$\left[\psi'(0)-\frac{s}{\phi(s)}\right] \doteq \frac{sv(s)}{\phi'(0)}$$

Substituting back into the integral expression gives

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t))$$
$$+\lim_{\epsilon\to 0}\int_{-\infty}^{-\epsilon}+\int_{\epsilon}^{\infty} ds\left[\frac{sv(s)}{\phi'(0)}\right] m(\phi(s))\frac{u(s)}{-s}$$
$$+\lim_{\epsilon\to 0}\left[\left(\int_{-\infty}^{-\epsilon}+\int_{\epsilon}^{\infty}\right)-\left(\int_{-\infty}^{\psi(-\epsilon)}+\int_{\psi(\epsilon)}^{\infty}\right)\right] ds\frac{u(s)}{-\phi(s)}m(\phi(s))$$

Note that the s cancels in the first integral pair, so we can take the limit as $\epsilon \to 0$. The integrand without the $u(s)$ factor in the first integral pair is then smooth, so we can again modify our smooth function $p(t)$ and write

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t))$$
$$+ \lim_{\epsilon \to 0} \left[\left(\int_{-\infty}^{-\epsilon} + \int_{\epsilon}^{\infty} \right) - \left(\int_{-\infty}^{\psi(-\epsilon)} + \int_{\psi(\epsilon)}^{\infty} \right) \right] ds \frac{u(s)}{-\phi(s)} m(\phi(s))$$

Which can be rewritten as

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t)) + \lim_{\epsilon \to 0} \int_{\psi(-\epsilon)}^{-\epsilon} + \int_{\epsilon}^{\psi(\epsilon)} ds \frac{u(s)}{-\phi(s)} m(\phi(s))$$

For sufficiently small ϵ the domain of integration is a small interval near zero. Near zero we know from the definition of m that $m(\phi(s)) = 1$. Hence

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t)) + \lim_{\epsilon \to 0} \int_{\psi(-\epsilon)}^{-\epsilon} + \int_{\epsilon}^{\psi(\epsilon)} ds \frac{u(s)}{-\phi(s)}$$

If we perform another change of variables $t = \phi(s)$

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t)) + \lim_{\epsilon \to 0} \int_{-\epsilon}^{\phi(-\epsilon)} + \int_{\phi(\epsilon)}^{\epsilon} dt \psi'(t) \frac{u(\phi(t))}{-t}$$

We can again do a Taylor expansion of $u(\phi(t))$ to obtain

$$u(\phi(t)) = u(0) + t u_1(t)$$

where $u_1(t)$ contains the higher order terms of the expansion. This gives

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t))$$
$$+ \lim_{\epsilon \to 0} u(0) \int_{-\epsilon}^{\phi(-\epsilon)} + \int_{\phi(\epsilon)}^{\epsilon} dt \frac{\psi'(t)}{-t} + \lim_{\epsilon \to 0} \int_{-\epsilon}^{\phi(-\epsilon)} + \int_{\phi(\epsilon)}^{\epsilon} dt \psi'(t) \frac{t u_1(t)}{-t}$$

In the second integral pair, the t terms cancel and we are left with an integral which vanishes as $\epsilon \to 0$, so

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t)) + \lim_{\epsilon \to 0} u(0) \int_{-\epsilon}^{\phi(-\epsilon)} + \int_{\phi(\epsilon)}^{\epsilon} dt \frac{\psi'(t)}{-t}$$

Similarly, we can do a Taylor expansion of $\psi'(t)$ to obtain

$$\psi'(t) = \psi'(0) + t\psi_1(t)$$

where $\psi_1(t)$ contains the higher order terms of the expansion. As above, this gives

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t)) + \lim_{\epsilon\to 0} u(0)\psi'(0) \int_{-\epsilon}^{\phi(-\epsilon)} + \int_{\phi(\epsilon)}^{\epsilon} \frac{dt}{-t}$$

or

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t)) - u(0)\psi'(0) \lim_{\epsilon\to 0} \left[-\int_{-\phi(-\epsilon)}^{\epsilon} + \int_{\phi(\epsilon)}^{\epsilon} \right] \frac{dt}{t}$$

Evaluation of the remaining explicit integrals gives

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t)) - u(0)\psi'(0) \lim_{\epsilon\to 0} \{\ln[-\phi(-\epsilon)] - \ln[\phi(\epsilon)]\}$$

Recalling that $\phi(0) = 0$, a Taylor expansion of $\phi(\epsilon)$ gives us

$$\phi(\epsilon) = \epsilon(\phi'(0) + \frac{\epsilon}{2}\phi''(0) + \ldots)$$

and an expansion of $-\phi(-\epsilon)$ gives us

$$-\phi(-\epsilon) = \epsilon(\phi'(0) - \frac{\epsilon}{2}\phi''(0) + \ldots)$$

Plugging these expressions into the natural log terms we have

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t))$$

$$-u(0)\psi'(0) \lim_{\epsilon\to 0} \{\ln[\epsilon(\phi'(0) - \frac{\epsilon}{2}\phi''(0) + \ldots)] - \ln[\epsilon(\phi'(0) + \frac{\epsilon}{2}\phi''(0) + \ldots)]\}$$

or

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t))$$

$$-u(0)\psi'(0) \lim_{\epsilon\to 0} \{\ln[\phi'(0) - \frac{\epsilon}{2}\phi''(0) + \ldots] - \ln[\phi'(0) + \frac{\epsilon}{2}\phi''(0) + \ldots]\}$$

Which in the limit as $\epsilon \to 0$ is

$$\psi'(0)Hu(0) - H(\psi' u \circ \psi)(0) = (p(t), u(t))$$

Thus the left hand side is equivalent to a smooth function, and

$$(\{H[\delta \circ \phi(x,t)] - (H\delta) \circ \phi(x,t)\}, u(t)) = (g(t), u(t))$$

where $g(t)$ is smooth. Which implies that the operation in the left inner product is regular and

$$[-Hf(t)] * (H\delta) \circ \phi(x,t) \doteq f(t) * \delta \circ \phi(x,t)$$

when $r(z)$ approximates a delta function.

This analysis can be boiled down to saying that when the support of the reflectivity is small and the source is oscillatory, an alternative source and reflectivity can be found which are very different from the model source and reflectivity, but reduce the objective function to near machine precision. In other words, the spike reflectivity inverse problem has computationally non-unique solutions.

This result does not, however, contradict the uniqueness results of Minkoff and Symes [11]. Their results applied to the plane wave domain, and these results are for the offset time domain. Yet, it is reasonable to expect that their results could be extended to the offset time domain. Their uniqueness result is not contradicted for a more important reason: the delta function does not satisfy the conditions placed on the reflectivity.

The determination of perturbations in the reflectivity by perturbations in the data is constrained by a "whiteness" property of the reflectivity. In other words, in order for the reflectivity to be well determined by the data, the reflectivity must have a large whiteness. For an interval $[\zeta_{min}, \zeta_{max}] \in \Re$ and $0 < \triangle\zeta < \zeta_{max} - \zeta_{min}$, the whiteness is defined for $r \in L^2(\Re)$, $r \neq 0$ as

$$W(\zeta_{min}, \zeta_{max}, \triangle\zeta; r)$$
$$= inf \left\{ \frac{\frac{1}{\zeta_+ - \zeta_-} \int_{\zeta_-}^{\zeta_+} |\hat{r}|^2}{\|r\|^2_{L^2(\Re)}} : \zeta_{min} \leq \zeta_- < \zeta_+ \leq \zeta_{max}, \frac{1}{2}\triangle\zeta \leq \zeta_+ - \zeta_- \leq \triangle\zeta \right\}$$

The whiteness is a measure of the uniformity of the distribution of Fourier components averaged over frequency interval of $\triangle\zeta$.

Consider a reflectivity of the following form

$$r_\epsilon(z) = \begin{cases} \epsilon & \text{if } -\frac{\epsilon}{2} \leq z \leq \frac{\epsilon}{2} \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\frac{1}{\zeta_+ - \zeta_-} \int_{\zeta_-}^{\zeta_+} dk |\hat{r_\epsilon}|^2 = \frac{4}{\zeta_+ - \zeta_-} \int_{\zeta_-}^{\zeta_+} dk \frac{\sin^2 \frac{k\epsilon}{2}}{\epsilon^2 k^2}$$

and

$$\|r_\epsilon\|^2_{L^2(\Re)} = \frac{1}{\epsilon}$$

So for any fixed ζ_+ and ζ_-

$$\lim_{\epsilon \to 0} \frac{\frac{1}{\zeta_+ - \zeta_-} \int_{\zeta_-}^{\zeta_+} dk |\hat{r_\epsilon}|^2}{\|r_\epsilon\|^2_{L^2(\Re)}} = \lim_{\epsilon \to 0} \frac{\frac{4}{\zeta_+ - \zeta_-} \int_{\zeta_-}^{\zeta_+} dk \frac{\sin^2 \frac{k\epsilon}{2}}{\epsilon^2 k^2}}{\frac{1}{\epsilon}} = \lim_{\epsilon \to 0} \frac{4 \int_{\zeta_-}^{\zeta_+} dk \frac{\sin^2 \frac{k\epsilon}{2}}{k^2}}{(\zeta_+ - \zeta_-)\epsilon} = 0$$

So as the support of the reflectivity decreases, the whiteness also decreases. In the limit, the whiteness goes to zero, and perturbations to the reflectivity can no longer be determined by the data.

Lastly, the experiments in subsection 4.2.2 gave numerical evidence that there is a family of solutions to the spike reflectivity inverse problem. The above analysis suggests the existence of only two near solutions (the model source/reflectivity and the Hilbert transform of the model source/reflectivity).

In answer to the above fact we offer that the preceding analysis suggests a family of solutions of the form

$$f_\theta = \cos(\theta) f + \sin(\theta) H f$$

and

$$\delta_\theta = -\sin(\theta)H\delta + \cos(\theta)\delta$$

so that

$$f_\theta * \delta_\theta = -(\sin\theta\cos\theta)f * H\delta + (\cos^2\theta)f * \delta - (\sin^2\theta)Hf * H\delta + (\sin\theta\cos\theta)Hf * \delta$$

using the preceding analysis we can say

$$f_\theta * \delta_\theta \doteq -(\sin\theta\cos\theta)f * H\delta + (\cos^2\theta)f * \delta + (\sin^2\theta)f * \delta + (\sin\theta\cos\theta)Hf * \delta$$

The Hilbert transform is a convolution operator in the sense that it is multiplication in the time domain, hence it commute and we can write

$$f_\theta * \delta_\theta \doteq -(\sin\theta\cos\theta)Hf * \delta + (\cos^2\theta)f * \delta + (\sin^2\theta)f * \delta + (\sin\theta\cos\theta)Hf * \delta$$

or

$$f_\theta * \delta_\theta \doteq f * \delta$$

Which implies that there is a circle of near solutions to the spike reflectivity inverse problem.

Chapter 5

Conclusions

For large seismic data sets finite difference modeling of the wave equation may be computationally prohibitive. The convolutional model of the seismogram provides a computationally cheaper means of modeling seismic data. The convolutional model is, however, only an approximation to the acoustic wave equation. In fact, we derive the convolutional model as a high frequency approximation to the linearized wave equation.

With a fixed source, convolution is a bijective operator within a given pass band. Therefore, it is possible to use the convolutional model to invert for the reflectivity and obtain a unique result when the source is known. If, however, the source is not known, or it is inaccurately known the reflectivity obtained via inversion may not be correct.

The source is rarely known to the level of detail necessary to obtain a reflectivity to desired accuracy. To circumvent this problem the convolutional model can be used for a joint inversion of the source and reflectivity. The uniqueness of the source and reflectivity is theoretically guaranteed under some conditions. Specifically, a quasi-impulsive, isotropic point source can be uniquely recovered from reflection seismograms using linear perturbation approximations. In an acoustic constant density, constant background velocity medium, perturbations in the source uniquely determine perturbations in plane wave seismograms. In addition, if the material velocity does not change too rapidly the source can be uniquely determined within a given pass band. Minkoff and Symes also showed that as the slowness aperture is widened the extent to which the reflectivity and source can be separately determined improves.

Further, if the slowness aperture is closed (there exists only one seismogram) there is no chance of recovering the source and reflectivity.

Computationally, however, up until now it has not been clear which methods are most desirable for joint inversion of the source and reflectivity. We find that when the inversion is posed as an optimization problem, where one is trying to minimize a data residual objective function is least squares sense, treating the joint inversion as a full non-linear optimization problem (simultaneously inverting for the source and reflectivity) is more effective than taking advantage of the bi-linear nature of the convolutional model and using an alternation algorithm or Kennett subspace algorithm. Of the non-linear optimization algorithms we tested, we found that the trust region Steihaug-Toint algorithm and the limited memory BFGS were the fastest at reducing the objective function such that the relative error in the data residuals was less than 5%.

In some instances, when the theoretical constraints given by Minkoff and Symes are not met, the inversion of the convolutional model will generate a source and reflectivity that are effectively not unique. The non-uniqueness encountered by these experiments is not the non-uniqueness typically encountered in the inversion of the convolutional model. Specifically, the shape of the source and reflectivity output from an inversion can be markedly different from the shape of the source and reflectivity used to generate the simulated data. In other words this is not just a scale effect, and the non-uniqueness occurs within any pass band.

Bibliography

[1] K. Bube, P. Lailly, P. Sacks, F. Santosa, and W.W. Symes. Simultaneous determination of source wavelet and velocity profile using impulsive point-source data from a layered fluid. *Geophys. J.*, 95:449–462, 1988.

[2] L. Debnath and P. Mikusinski. *Introduction to Hilbert Spaces with Applications*. Academic Press, San Diego, 1999.

[3] J.E. Dennis, Jr. and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, 1983.

[4] J. Duistermaat. Fourier integral operators. Lecture notes, Courant Institute, New York, 1973.

[5] Mark S. Gockenbach and William W. Symes. The Hilbert Class Library: a library of abstract C++ classes for optimization and inversion. *Computers and Mathematics with Applications*, 32, 1996.

[6] P. C. Hansen. *Rank Deficient and Discrete Ill-posed Problems*. SIAM, Philadelphia, 1998.

[7] W.S. Harlan. Simultaneous velocity filtering of hyperbolic reflections and balancing of offset-dependent wavelets. *Geophysics*, 11:1455–1465, 1989.

[8] B. Kennett, Sambridge M., and Williamson P. Subspace methods for large inverse problems with multiple parameter classes. *Geophysical Journal*, 94:237–247, 1988.

[9] R.M. Lewis. *Source-Velocity Identification for a Layered Model of Reflection Seismology.* PhD thesis, Department of Mathematical Sciences, Rice University, Houston, Texas, U.S.A, 1989.

[10] S. E. Minkoff and W. W. Symes. Full waveform inversion of marine reflection data in the plane-wave domain. *Geophysics*, 62(2):540–553, 1997.

[11] S.E. Minkoff and W.W. Symes. Estimating the energy source and reflectivity by seismic inversion. *Inverse Problems*, 11:383–395, 1995. Also in Proc. SPIE, 1994.

[12] J.J. More and D. C. Sorenson. On the use of directions of negative curvature in a modified newton method. *Math. Prog.*, 16:1–20, 1979.

[13] J.L. Qian. *Geometric Optics for quasi-P Waves: Theories and Numerical Methods.* PhD thesis, Department of Computational and Appled Mathematics, Rice University, Houston, Texas, U.S.A, 2000.

[14] William Symes. All stationary points of differential semblance are asymptotic global minizers: Layered acoustics. Technical report, The Rice Inversion Project, Rice University, Houston, Texas, USA, 1999.

[15] O. Yilmaz. *Seismic Data Processing: Investigations in Geophysics No. 2.* Society of Exploration Geophysicists, Tulsa, 1987.

[16] N. Young. *An Introduction to Hilbert Space.* Cambridge University Press, New York, 1988.

[17] A. Ziolkowski. *Deconvolution.* International Human Resources Development Corporation, Boston, 1984.

[18] A. Ziolkowski. Why don't we measure seismic signatures? *Geophysics*, 56(2):190–201, 1991.