Seismic Moment Rate Function Inversions from Very Long Period Signals Associated With Strombolian Eruptions at Mount Erebus, Antarctica

by

Christian Levi Lucero

Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Science in Mathematics

with Operations Research and Statistics Option

New Mexico Institute of Mining and Technology

Socorro, New Mexico

February 2nd, 2007

ABSTRACT

The inverse problem for the recovery of the seismic source moment tensor is a fundamental problem in seismology. A particular seismic source phenomenon seen at Mount Erebus, Antarctica is that of explosive decompression of a gas slug formed beneath the volcano's lava lake and the subsequent forces generated as the lava lake recovers from this disruption. The resulting time series obtained from a number of broadband seismometers placed around the volcano can be described by the summation of convolutions between the moment tensor components and their corresponding Green's functions. In this thesis, I present a new solution method for this problem that uses the properties of Toeplitz matrices. In particular, the Toeplitz matrix structure allows for fast matrix-vector multiplication by means of the Fast Fourier Transform. By combining the use of Toeplitz matrices, the implicit storage of the Green's functions, and an iterative method, it becomes possible to work on large data sets. Iterative inverse techniques, such as Conjugate Gradient Least Squares (CGLS) method, are easily adapted to use Fast Toeplitz Multiplication, and explicit regularization is introduced for extra stability. However, the choice of a regularization parameter can require the calculation and evaluation of numerous solutions.

ACKNOWLEDGMENT

I would like to thank my loving and supporting wife Danielle. She has always been there by my side while we worked through the endless nights.

I would like to thank all my friends. Many of you have loaned me your thoughts and your assistance when I needed it. I will never forget, and I hope to return the favors for years to come. Most importantly, thanks for sharing the great times that we've had together. Time hasn't always permitted us to enjoy as many moments together as we would have liked, but thanks for making the effort to enjoy what we could.

To my parents, I thank you for your love and support and your understanding while I spent these long tedious years in school.

Finally, to my advisory committee, thank you for your guidance and support and most of all your patience and understanding as everything that can go wrong, did go wrong. In particular, to Brian Borchers and Rick Aster, thank you for helping me to reach a new milestone in my journey thru life, while helping to open up a few new exciting paths.

This dissertation was typeset with LaTeX[1] by the author.

[1] LaTeX document preparation system was developed by Leslie Lamport as a special version of Donald Knuth's TeX program for computer typesetting. TeX is a trademark of the American Mathematical Society. The LaTeX macro package for the New Mexico Institute of Mining and Technology dissertation format was adapted from Gerald Arnold's modification of the LaTeX macro package for The University of Texas at Austin by Khe-Sing The.

TABLE OF CONTENTS

LIST OF FIGURES

This dissertation is accepted on behalf of the faculty of the Institute by the following committee:

Brian Borchers, Advisor

Christian Levi Lucero Date

CHAPTER 1

INTRODUCTION

1.1 The Geophysical problem

Mount Erebus, located on Ross Island, Antarctica, is a large strato-volcano which experiences Strombolian (small-to-medium blasts with relatively low-viscosity lava) eruptions due to the explosive decompression of large gas slugs which form beneath its exposed magma (lava) lake [1]. These eruptions are highly cyclic in nature with each cycle involving the formation of a the gas slug, explosive decompression, and magmatic recharge of the lava lake. This persistent volcanic activity has been observed on Mount Erebus for at least 8 years allowing for the fruitful deployment of broadband seismometers on and around the volcano as seen in figure 1.2. Short-period seismic explosion-related seismograms superimposed on oscillatory very long period signals related to the response of the magmatic conduit have been observed by means of near field broadband seismometers 0.7 to 2.5 km from the lava lake [1].

Broadband seismometers on Erebus have been permanently placed since 2004 at 6 sites. Each seismometer records data in three components, (x,y,z) directions which are then rotated into a source radial-coordinate system. The transformation for a given displacement vector **u** emanating from

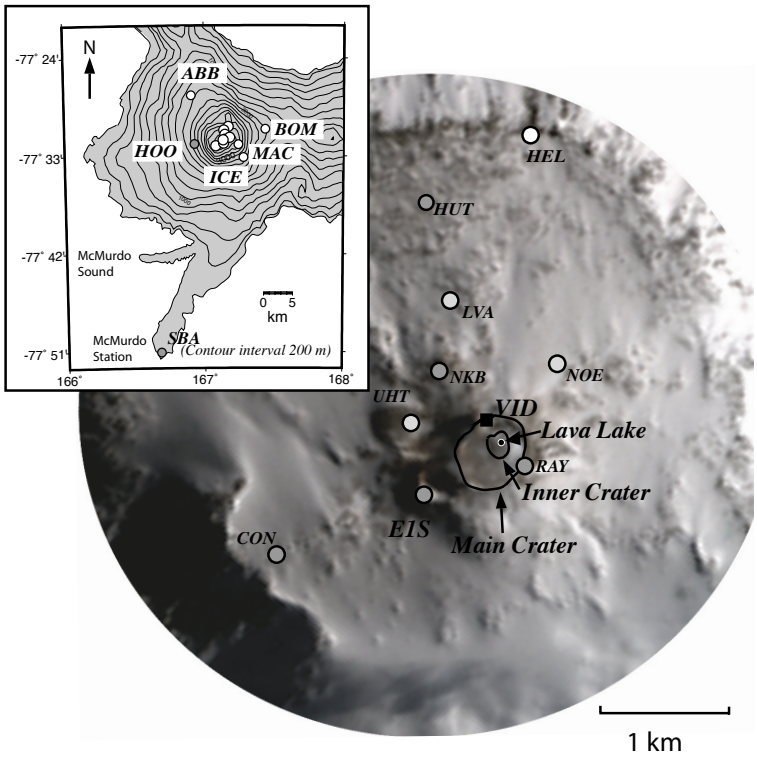

Figure 1.1: Erebus Station Map. The stations used in the stacked data inversion are CON, E1S, HOO, LEH, NKB, RAY

the source location center is computed by

$$\begin{bmatrix} \mathbf{u}_R \\ \mathbf{u}_T \\ \mathbf{u}_Z \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \\ \mathbf{u}_z \end{bmatrix}. \tag{1.1}$$

This rotation results in the radial, transverse (the direction corresponding to the angle θ perpendicular to the radial direction) and vertical components for each station.

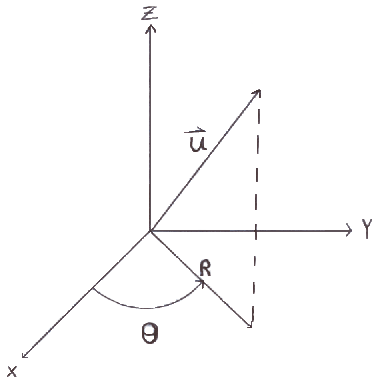

Figure 1.2: Rotation of coordinates into a source radial-coordinate system.

The stacked data used in this study was generated in 2005 from these 6 stations by means of summing the seismogram amplitudes aligned along travel times [1]. Using stacked data significantly helps to increase the signal-to-noise ratio by effectively reducing the noise observed from the oceanic microseism, that is, helping to remove the faint tremors measured by seismic equipment that are completely unrelated to the observed event and corresponding primarily to ocean waves in the nearby Ross Sea. The Green's functions which describe the impulse response are calculated by means of a homogeneous half

space with flat topography [7]. Unfortunately this half space assumption is not entirely accurate in regards to building Green's functions which incorporate the topography of a volcano but instead are more applicable for work in models where the earth is flat. The source location was determined by McNamara [9] and adjusted later by Aster [3] with an inferred point-source depth of 400 meters. The seismic moment tensor utilized here is a generalization to the formulation used to estimate the source parameters of a seismic model where faulting caused by displacement discontinuity can be instead represented by equivalent body forces to describe a seismic source [11]. These general source components correspond to forces acting in particular directions called single forces and force-couples. A force-couple consists of two forces acting together along the same axis or acting upon a lever arm as to cause a torque.

The force-couples are typically denoted by subscripts which define their direction of influence. For example, $\mathbf{M}_{yx}$ and $\mathbf{M}_{xy}$ describe single-couples of forces of magnitude f separated by distance d which act in the $(\pm x)$ and a clockwise manner in the xy-plane, respectively. Double couples are simply the combination of the two opposing single-couples in a plane such as $\mathbf{M}_{xy} + \mathbf{M}_{yx}$ and are especially useful in seismology because they are equivalent force couples for faulting. The tensor components $\mathbf{M}_{ij}$ are therefore considered to be symmetric in most sources, this arises because internal processes cannot generate a net torque, and thus eliminates the need for their independent recovery. The seismic moment tensor therefore represents the equivalent body forces for seismic sources by the moment tensor $\mathbf{M}$ whose components are the nine force

couples [11]

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{xx} & \mathbf{M}_{xy} & \mathbf{M}_{xz} \\ \mathbf{M}_{xy} & \mathbf{M}_{yy} & \mathbf{M}_{yz} \\ \mathbf{M}_{xz} & \mathbf{M}_{yz} & \mathbf{M}_{zz} \end{pmatrix}. \tag{1.2}$$

The seismic source can then be represented explicitly as an ordered vector containing the six independent double couples and three single directional forces

$$\mathbf{m} = \begin{pmatrix} \mathbf{M}_{xx}(t), & \mathbf{M}_{xy}(t), & \cdots, & \mathbf{M}_{zz}(t), & \mathbf{F}_x(t), & \mathbf{F}_y(t), & \mathbf{F}_z(t) \end{pmatrix}^T \tag{1.3}$$

where $\mathbf{F}_x$, $\mathbf{F}_y$, $\mathbf{F}_z$ are single forces acting in the (x,y,z) directions respectively. In the context of a volcanic eruption, such sources arise from accelerated magma or gas. A designated ordering of the model shall be used by labeling $m_1 = M_{xx}$, $m_2 = M_{xy}$, ..., $m_9 = F_z$ for a total of nine tensor components. All nine moment tensor components are not necessary for each assumed model, and instead attempted recovery for any combination of the tensor components my be desirable, depending on the source.

The recorded seismogram at the jth seismometer is defined by

$$d_j(t) = \sum_{i=1}^{k} g_{ji}(t) * m_i(t), \qquad j = 1, 2, \ldots, n \tag{1.4}$$

where k is the number of tensor components and g_{ji} are the corresponding Green's functions for each tensor component, and $m_i(t)$ are the moment tensor components. Use of the seismic moment tensor simplifies the inversion process to the solution of the system (1.4) for the moment tensor component vectors and can be used to describe non-slip seismic sources which generate observable seismic waves, such as volcanic eruptions [11].

The seismic moment tensor is used to directly identify the nature of the seismic source and shall be denoted in the following manner. First a Mogi source [10] is a symmetric spherical source defined by the moment tensor

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{xx} & 0 & 0 \\ 0 & \mathbf{M}_{xx} & 0 \\ 0 & 0 & \mathbf{M}_{xx} \end{bmatrix} \tag{1.5}$$

and is generally used to describe a source which radiates outward/inward in an explosive/implosive manner. A dilitational source is defined by the three tensor components

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{xx} & 0 & 0 \\ 0 & \mathbf{M}_{yy} & 0 \\ 0 & 0 & \mathbf{M}_{zz} \end{bmatrix} \tag{1.6}$$

and may generally describe a source that radiates in a disk-like manner along a general spatial orientation. The dilitational plus single vertical force model builds upon the dilitational model by allowing inertial forces in the source. Specifically, the dilitational plus three single forces model adds the directional single forces $\mathbf{F}_x$, $\mathbf{F}_y$, $\mathbf{F}_z$ to the general dilitational model. A full model consists dilitational components, the three directional single forces and the additional double-couples including $\mathbf{M}_{xy}$. In general, the addition of the model components increases the computational complexity due to the increasing ill-posedness of the problem, while at the same time should allow for better fits to the data. It is of primary importance to see the relative contributions of each tensor component and understand its role in the overall fit to the data.

1.2 The Mathematical problem

Before addressing the full moment tensor inversion problem, I will demonstrate the recovery of a single component model. The general convolution

equation to describe a seismogram is given by Lay and Wallace [8]

$$d(t) = m(t) * e(t) * i(t) \tag{1.7}$$

where $*$ denotes convolution, $d(t)$ is the seismogram, $m(t)$ is a single component moment rate function, $e(t)$ is the Green's function that represents the Earth responses to a single-force or force-couple impulses, and $i(t)$ is the instrument response. We gain the total response $g(t)$ by convolving the instrument and earth responses (i.e. $g(t) = e(t) * i(t)$) and then simplify our basic convolution equation to

$$d(t) = m(t) * g(t). \tag{1.8}$$

The inverse problem of deconvolution requires minimal computational effort so long as you work entirely in the frequency domain. It is sufficient in the simple deconvolution problem between two time-series to carry out the deconvolution using the convolution theorem and the Fourier transform. Taking the Fourier transform of both sides of (1.8) results in the equation $D(f) = G(f)M(f)$, where we are now concerned with each quantity as a function of frequency, f. A simple division in the frequency domain (spectral division) recovers the Fourier transform of the model and then by taking the inverse Fourier transform the original time domain model is recovered

$$m(t) = \mathcal{F}^{-1}\left(\frac{D(f)}{G(f)}\right) \tag{1.9}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform. In practice the method of spectral division seldom works well as instability arises from the division by $G(f)$ which may have components near zero. In particular, the occurrence of noise in a signal may generally occur over a broad range of frequencies which produce small non-zero components in $D(f)$ that can become amplified in spectral division.

Prefiltering of data is a regularization technique which can greatly improve the stability by eliminating noise in the data before attempting to recover the model. If knowledge about the data frequency content is known in advance, then an appropriate high/low/band pass filter. For example, if the model is assumed to have no high frequency components, then the a low pass filter can be used on the data to eliminate all frequencies greater than a theoretical cutoff frequency.

In Tikhonov regularization, we use

$$m_\lambda(t) = \mathcal{F}^{-1}\left(\frac{G(f)^* D(f)}{G(f)^* G(f) + \lambda}\right) \tag{1.10}$$

where λ is a small positive constant, and * denotes the complex-conjugate. The product $G(f)^*G(f) \geq 0$ ensures that the denominator will always be positive, that is, even if $G(f) = 0$ the addition of $\lambda > 0$ prevents division by zero. A further result is when $G(f)$ is big then the affect of λ is negligible.

To perform any numerical calculations, discretization must be done by time-sampling the data and Green's function. Inversion can then be done on the discrete convolution

$$\mathbf{d} = \mathbf{g} * \mathbf{m} \tag{1.11}$$

where $\mathbf{d}$, $\mathbf{g}$, $\mathbf{m}$ are vectors of length n describing the data, Green's function, and model, respectively. The nth component of the truncated linear convolution is computed by

$$(m * g)_n = \sum_{k=0}^{N-1} m_k g_{n-k}, \qquad n = 0, \ldots, N-1 \tag{1.12}$$

where the negative indices on $g_{n-k} = 0$.

In this thesis we will only need the first n components of the convolutions as the data and the model vectors are to have the same length. Otherwise convolution between two vectors of length n results in a vector of length $2n-1$. So then the actual data vector obtained by the linear convolution is a truncated convolution representing only the first n elements of the computation. Alternatively, (1.11) can be expressed as the matrix-vector product $\mathbf{d} = \mathbf{Gm}$ where $\mathbf{G}$ is a Toeplitz matrix formed from the Green's function vector as

$$\mathbf{G} = \begin{bmatrix} g_1 & 0 & 0 & \cdots & 0 & 0 \\ g_2 & g_1 & 0 & \cdots & 0 & 0 \\ g_3 & g_2 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 & 0 \\ g_{n-1} & \ddots & \ddots & \ddots & g_1 & 0 \\ g_n & g_{n-1} & \cdots & g_3 & g_2 & g_1 \end{bmatrix}. \tag{1.13}$$

The result of $\mathbf{Gm}$ is the first n elements obtained by $\sum_{k=0}^{N-1} m_k g_{n-k}$, i.e. the truncated linear convolution.

As in the continuous case, computation of the discrete convolution can also be expressed by the convolution theorem, however the time series vectors will need to be zero padded, that is the addition of zeros to pad the time-series to be length $2n$ to avoid wrap around effects when taking the Fourier transform. Thus (1.11) becomes

$$\mathbf{D} = \mathbf{GM} \tag{1.14}$$

where $\mathbf{D}$ and $\mathbf{M}$ are vectors containing the discrete Fourier transforms of the zero-padded time-series d_n, and m_n respectively, and $\mathbf{G}$ is a diagonal matrix whose diagonal is the discrete Fourier transform of the discrete zero-padded Green's function. Instability arises from $\mathbf{G}$ being nearly singular. The forward

problem in (1.14) can be simply viewed as the element-wise product of the FFT of $\mathbf{G}$ and $\mathbf{M}$, thus becoming

$$D_k = G_k M_k, \qquad k = 0, 1, \ldots, 2N \tag{1.15}$$

for $k = 0, 1, \ldots, 2N$, thus $\mathbf{M}$ could be recovered quickly by the element-wise division

$$M_k = \frac{D_k}{G_k}, \qquad k = 0, 1, \ldots, 2N \tag{1.16}$$

where clearly if any element G_k is small, and D_k is nonzero, then the resulting M_k becomes large. Therefore division by zero is an obvious problem and the main pitfall of this approach.

What is needed is a scheme which filters out near zero elements of G_k while retaining the larger influential components. One possible fix is to simply perturb all the complex G_k by a small positive real number λ so that (1.16) now becomes

$$M_k = \frac{D_k}{G_k + \lambda}. \tag{1.17}$$

This addition serves to eliminate the problem of division by very small numbers by selecting an appropriate λ. Clearly when G_k is much larger than λ, then λ will have little effect on M_k, therefore selection of an appropriately small sized λ is desired. This method also has a pitfall if $G_k = -\lambda$, which still results in division by zero.

Another issue is that we want M to be Hermitian, that is, $M_k = M^*_{N-k}$ where * denotes complex conjugate. It is a fact that $\mathbf{m}$ is real valued if and only if M is Hermitian [4]. Therefore, M must be Hermitian in order to recover a real-valued model $\mathbf{m}$.

Tikhonov regularization addresses these problems by solution of

$$\mathbf{D} = \mathbf{GM}$$

for $\mathbf{M}$ by means of the minimization problem

$$\min \left(\|\mathbf{GM} - \mathbf{D}\|_2^2 + \lambda \|\mathbf{M}\|_2^2 \right). \tag{1.18}$$

Here $\| \cdot \|_2$ denotes the Euclidean norm, where for a complex vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, $\|\mathbf{x}\|_2 = \sqrt{x_1^* x_1 + x_2^* x_2 + \cdots + x_n^* x_n}$. The function to be minimized can be expanded by means of matrix algebra

$$f(\mathbf{M}) = \|\mathbf{GM} - \mathbf{D}\|_2^2 + \lambda \|\mathbf{M}\|_2^2 = (\mathbf{GM} - \mathbf{D})^* (\mathbf{GM} - \mathbf{D}) + \lambda \mathbf{M}^* \mathbf{M} \tag{1.19}$$

where $(*)$ represents the conjugate transpose. Further simplification yields

$$f(\mathbf{M}) = \mathbf{M}^* \mathbf{G}^* \mathbf{GM} - 2\mathbf{G}^* \mathbf{D} + \mathbf{D}^* \mathbf{D} + \lambda \mathbf{M}^* \mathbf{M}. \tag{1.20}$$

We compute the gradient

$$\nabla f(\mathbf{M}) = 2\mathbf{G}^* \mathbf{GM} - 2\mathbf{G}^* \mathbf{D} + 2\lambda \mathbf{M} \tag{1.21}$$

and setting $\nabla f(\mathbf{M}) = \mathbf{0}$ results in the normal equations

$$(\mathbf{G}^* \mathbf{G} + \lambda \mathbf{I})\mathbf{M} = \mathbf{G}^* \mathbf{D}. \tag{1.22}$$

Solution of this system is also makes use of the element-wise division in the form

$$M_k = \frac{G_k^* D_k}{G_k^* G_k + \lambda} \tag{1.23}$$

where now the denominator is never zero for positive λ.

This final result is the discrete form of the Tikhonov regularization method seen in equation (1.10).

A more complicated problem is to look at a summation of convolutions as seen in the recovery of the seismic moment tensor model components associated with Strombolian eruptions observed at Mount Erebus in Antarctica. The general inverse problem is to solve for the moment rate function which itself consists of moment tensor force couples. Since each Green's function corresponds to a different moment tensor and the wave equation describing the seismic wave is linear, by Lay and Wallace [8] our equation is now described by the system,

$$d_n(t) = \sum_{i=1}^{k} m_i(t) * g_{ni}(t) \tag{1.24}$$

where n specifics the station and the radial, transverse, or vertical component, and g_{in} is the Green's function corresponding to the i-th moment tensor for each component.

The signals are discretized and truncated to ensure full coverage of the event (and part of its coda). Regularization by discretization is imposed whenever numerical calculations are performed. The sampling rate or level of discritization has a direct effect on the stability of the inversion. Increasing the sampling rate improves the residual fit, but decreases stability due to the inclusion of small singular values of $\mathbf{G}$ [2].

Since we have multiple seismic stations, n more generally represents the radial, transverse, and vertical components for each station (i.e. $n = 3\times$

the number of stations).

$$\mathbf{d}_j = \sum_{i=1}^{k} \mathbf{g}_{ji} * \mathbf{m}_i, \qquad j = 1, 2, \ldots, n \tag{1.25}$$

or more explicitly

$$\begin{aligned}
\mathbf{d}_1 &= \mathbf{g}_{11} * \mathbf{m}_1 + \mathbf{g}_{12} * \mathbf{m}_2 + \cdots + \mathbf{g}_{1k} * \mathbf{m}_k, \\
\mathbf{d}_2 &= \mathbf{g}_{21} * \mathbf{m}_1 + \mathbf{g}_{22} * \mathbf{m}_2 + \cdots + \mathbf{g}_{2k} * \mathbf{m}_k, \\
&\vdots \\
\mathbf{d}_n &= \mathbf{g}_{n1} * \mathbf{m}_1 + \mathbf{g}_{n2} * \mathbf{m}_2 + \cdots + \mathbf{g}_{nk} * \mathbf{m}_k,
\end{aligned}$$

We want to solve for the moment tensor components $\mathbf{m}_1$, $\mathbf{m}_2$, etc. Alternatively, we can represent the convolution between two vectors using a matrix-vector product using the Toeplitz matrix formulation seen in (1.13). So then (1.25) now can be represented by

$$\begin{aligned}
\mathbf{d}_1 &= \mathbf{G}_{11}\mathbf{m}_1 + \mathbf{G}_{12}\mathbf{m}_2 + \cdots + \mathbf{G}_{1k}\mathbf{m}_k, \\
\mathbf{d}_2 &= \mathbf{G}_{22}\mathbf{m}_1 + \mathbf{G}_{22}\mathbf{m}_2 + \cdots + \mathbf{G}_{2k}\mathbf{m}_k, \\
&\vdots \\
\mathbf{d}_n &= \mathbf{G}_{n1}\mathbf{m}_1 + \mathbf{G}_{n2}\mathbf{m}_2 + \cdots + \mathbf{G}_{nk}\mathbf{m}_k,
\end{aligned}$$

or equivalently by the blocked matrix system

$$\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \mathbf{d}_n \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & \cdots & \mathbf{G}_{1k} \\ \mathbf{G}_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{G}_{n1} & \cdots & \cdots & \mathbf{G}_{nk} \end{bmatrix} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_k \end{bmatrix} \tag{1.26}$$

The signals obtained are 200 seconds in duration with a sampling rate of 40 samples per second. A data vector is therefore 8000 samples long while the concatenation of data recorded from 18 channels results in 144000 element vector requiring approximately 1.1 megabytes of storage. Each model component is also 8000 elements long and a full model vector consists of up to 9

model components resulting in a vector of 72000 elements and 0.55 megabytes of storage. The full G is therefore 144000 by 72000 in size and requires 77 gigabytes for full storage. The resulting data sets are therefore large enough to require the need for large dense matrices which describe the convolutions and subsequently impacts inverse methods which require explicit storage of large matrices. This limitation requires large amounts of computer memory as seen in the method used by Sara McNamara [9], and often prompts the need to work on down-sampled data sets. To simplify this problem, we would like to do as many frequency-domain convolutions as possible in order to use the speed of the FFT as well as avoid explicitly storing the G matrices. Because of the storage requirements, methods such as singular value decomposition (SVD) are impossible on current computers, but iterative methods which use only matrix vector multiplications are much better suited to low storage computational algorithms.

In this instance, recovery of a series of model vectors would be more tedious and not readily available (but still attainable) by means of the convolution theorem. To do the inversion on this system, we must consider the larger system $\mathbb{D} = \mathbb{G}\mathbb{M}$. Here $\mathbb{D}$ is a concatenated vector of the Fourier transformed data vectors $\mathbb{D} = (\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_n)^T$, $\mathbb{M}$ is a concatenated vector of the Fourier transformed model vectors $\mathbb{M} = (\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k)^T$, and $\mathbb{G}$ is an blocked matrix of $n \times k$ diagonalized Fourier transformed Green's functions each appropriately matching the data channel and the model vector. Instability arises once again from $\mathbb{G}$ being nearly singular while naively attempting spectral division by inversion of the $\mathbb{G}$ matrix. Instead, we seek to use an alternative method which uses the speed of the Fast Fourier Transform to tackle

this inverse problem by first representing this system of equations in a more concise form.

Circulant matrices, which make possible "Fast Toeplitz Multiplication", seem perfect for use in converting the problem to use only matrix-vector multiplications which are ideal for low-memory iterative techniques such as conjugate gradient least squares (CGLS). Additionally, Tikhonov regularization can be explicitly applied, thus allowing for better control over regularization. In Chapter 2, I will introduce the important properties of circulant matrices along with the CGLS method which will be used for carrying out the moment tensor inversion.

1.3 What has been done before

In her 2004 thesis, Sara McNamara worked on the Mount Erebus data inverse problem. For her inversion, McNamara used an iterative inverse method presented by Lay and Wallace [8]. Starting again with the basic continuous forward problem,

$$d_n(t) = \sum_{i=1}^{k} m_i(t) * g_{in}(t) \tag{1.27}$$

Lay and Wallace proposed a parameterization method consisting of a series of overlapping triangles (i.e. triangle basis functions) as seen in figure 1.3. The parameterization represents the model as

$$m_i(t) = \sum_{j=1}^{M} B_{ij} b(t - \tau_j), \qquad i = 1, \ldots, k. \tag{1.28}$$

The forward problem with the inclusion of triangle basis functions becomes:

$$d_n(t) = \sum_{j=1}^{M} \sum_{i=1}^{k} B_{ij} [b(t - \tau_j) * g_{in}(t)] \tag{1.29}$$

where M is the total number of triangle functions, B_{ij} the height of the jth triangle, and $b(t-\tau_j)$ is triangle centered at time τ_j. In this system we must

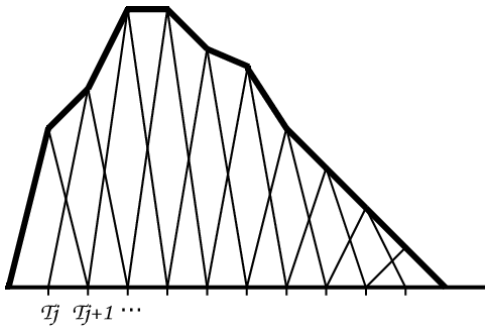

Figure 1.3: Parameterization of an arbitrarily shaped source function using triangle basis functions.

solve for, B_{ij} the triangle heights which will be used to estimate the moment tensor m_i [8].

Computationally, we consider the discrete approximation and begin the actual inversion computation from a generic initial triangle heights $\mathbf{B}_{ij} = \mathbf{0}$. Iterations which revise these heights are carried out by minimizing the residual from the difference of the observed and synthetic seismograms, i.e. $\Delta\mathbf{d} = \mathbf{d}_{obs} - \mathbf{d}_{syn}$. The height $\mathbf{B}_{ij}$ is then updated each iteration by solving the equation $\Delta\mathbf{d} = \mathbf{A}\Delta\mathbf{P}$, Where $\mathbf{A}$ is a Jacobian matrix (where $A_{ij} = \partial d_i/\partial P_j$). By adding the resulting $\Delta\mathbf{P}$ to $\mathbf{B}_{ij}$, the triangle heights are adjusted and then used generate a new $\mathbf{d}_{syn}$ to repeat the process until convergence. The final model is then obtained by using equation (1.28).

For solving the actual inverse problem, McNamara used a singular value decomposition which was slow and suffered from needing large amounts of storage. In 2006, Aster [3] continued work on the Mount Erebus problem

by using the Lay and Wallace method but allowing for the inverse problem to solved by means of the CGLS algorithm. This adjustment helped to improve the speed of the algorithm dramatically and eliminate the need for storing the SVD matrices. For comparison, the algorithm using the SVD typically took a couple of hours and was reduced to about 20 minutes by using CGLS. Aster began by prefiltering the data to remove high frequency components which are not expected to be included in the recovered model. This also helps to reduce computational effort by reducing the computational effort to attempt to fit high frequency noise in the data..

One major problem with this regularization scheme is that it is not possible to obtain a measure of the regularization effect, nor is it possible to accurately control its level of influence during the inversion. It is therefore not clear whether regularization by means of discretization from triangle basis functions has any real benefits over other methods of regularization and it unclear what negative affects might result.

The main storage requirement for the method of parameterization by discretization using triangle basis functions is the dense Jacobian matrix $\mathbf{A}$. Given the number of seismic station n, the data sample length s, the number of model components c, and the number of triangles used in the discretization p, then the storage is as follows, the # of rows of $\mathbf{A} = (3 \times n \times s)$, and the # of columns of $\mathbf{A} = (c \times p)$. Thus the final storage of $\mathbf{A}$ is $3nscp$.

CHAPTER 2

THEORETICAL FRAMEWORK

Numerical computations of convolutions of the form $\mathbf{a} * \mathbf{b}$ are expensive except when using the well established Convolution Theorem (CT) which invokes the Fast Fourier Transform (FFT). Discrete convolutions can be carried out in $O(n^2)$ time on n length vectors by computing $(\mathbf{a} * \mathbf{b})_m = \sum_{k=0}^{n} a_k b_{m-k}$. Using the convolution theorem and the FFT reduces the convolution of two vectors, $\mathbf{a}$,$\mathbf{b}$ to an element-wise product between the FFT of $\mathbf{a}$ and the complex vector from the FFT of $\mathbf{b}$, i.e. $\mathbf{a} * \mathbf{b} = \mathrm{FFT}(\mathbf{a}) \cdot \mathrm{FFT}(\mathbf{b})$ in only $O(n \ln(n))$ time.

When dealing with the fast Fourier transform it must be noted that there are unexpected effects when applying the convolution theorem directly. The discrete convolution of two sequences, one of length m and the other n, results in a $n + m - 1$ length sequence. When using the FFT (and its inverse) it is assumed that these sequences are periodic in the time domain. The period assumed by the FFT, say size l, must then be longer than $n + m - 1$ to allow room for the entire $n + m - 1$ length convolution to fit in an l-periodic result. Otherwise if $l < n+m-1$, undesirable wraparound effects give a result different than the expected linear convolution computation. These wraparound effects correspond to cyclic, or circular convolution and can be avoided by padding the sequences with zeros (zero padding) so that the number of significant terms is

less than the period of the FFT.

For the convolution between two n length sequences, zero padding each sequence by n zeros is sufficient to contain the $2n-1$-element convolution. It should be noted that this product is still computed in $O(n\ln(n))$ time. When convolving two n length vectors to produce the first n elements of the convolution, the FFT with zero padding is still used. The final result is simply truncated to include the first n elements from the convolution theorem.

2.1 Fast Toeplitz Multiplication

The general convolution operation can be greatly simplified to a simple matrix-vector multiplication by means of a convolution matrix. For example, the convolution of vectors $\mathbf{a}$, and $\mathbf{b}$, results in the following system:

$$\mathbf{a} * \mathbf{b} = \begin{bmatrix} a_1 & 0 & \cdots & 0 & 0 \\ a_2 & a_1 & 0 & \cdots & 0 \\ a_3 & a_2 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \\ a_n & a_{n-1} & \cdots & a_2 & a_1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} \tag{2.1}$$

This method of computing the convolution only returns the first n elements of the convolution. This convolution matrix is a structured matrix known as a Toeplitz matrix. In general a Toeplitz matrix is a matrix whose entries are constant along each diagonal, which allows Toeplitz matrices to be formed implicitly by only knowing the first column and first row. Therefore a general

Toeplitz matrix is well structured and easily defined by its unique structure

$$\mathbf{T} = \begin{bmatrix} T_1 & T_0 & T_{-1} & \cdots & T_{-n+1} & T_{-n+2} \\ T_2 & T_1 & T_0 & \ddots & \ddots & T_{-n+1} \\ T_3 & T_2 & T_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & T_0 & T_{-1} \\ \vdots & & \ddots & T_2 & T_1 & T_0 \\ T_n & T_{n-1} & \cdots & T_3 & T_2 & T_1 \end{bmatrix}. \tag{2.2}$$

However when using a convolution matrix, where the first row has only one nonzero entry, all that is ultimately needed is the first column to precisely describe the entire matrix.

This matrix-vector multiplication itself can be carried out more quickly by using another structured matrix called a circulant matrix. A circulant matrix shown in (2.3) is a Toeplitz matrix where each column is a "circular-shifted" version of its predecessor [5].

$$\mathbf{C} = \begin{bmatrix} c_1 & c_n & \cdots & c_3 & c_2 \\ c_2 & c_1 & \cdots & c_4 & c_3 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ c_n & c_{n-1} & \cdots & c_2 & c_1 \end{bmatrix} \tag{2.3}$$

Circulant matrices have a very useful property [12]:

$$\mathbf{Cx} = \mathbf{F}^{-1}(\mathbf{Fc} \cdot \mathbf{Fx}) \tag{2.4}$$

where $\mathbf{c}$ is the first column of the circulant matrix $\mathbf{C}$. Here $\mathbf{F}$ and $\mathbf{F}^{-1}$ refer to the Fourier and the inverse Fourier transform matrix operators, respectively. The notation "$\cdot$" in this instance describes an element-wise product of the vectors not a dot product. A proof of (2.4) is as follows:

The circulant matrix $\mathbf{C}$ can be obtained from the use of permutation matrices acting upon the descriptive vector $\mathbf{c} = (c_1, c_2, \ldots, c_n)^T$. Specifically, $\mathbf{C}$ can be formed from the polynomial equation

$$\mathbf{C} = c_1\mathbf{I} + c_2\mathbf{P} + c_3\mathbf{P}^2 + \cdots + c_n\mathbf{P}^{n-1} \tag{2.5}$$

where $\mathbf{P}$ is a permutation matrix of the form

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \vdots & \ddots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{2.6}$$

The fast Fourier computation and its inverse are equivalent to the product of the Fourier matrix and a vector, i.e. $\text{fft}(x) = \mathbf{F}\mathbf{x}$ and $\text{ifft}(y) = \mathbf{F}^{-1}\mathbf{y}$. Where the Fourier matrix $\mathbf{F}$ of order n is

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \cdots & \omega^{(n-1)^2} \\ & & & \end{bmatrix}$$

and $\omega = e^{2\pi i/n}$ is the primitive nth root of unity. Using the Fourier matrix notation, we can rearrange the operators from equation (2.4) as

$$\mathbf{F}\mathbf{C}\mathbf{F}^{-1} = \text{diag}(\mathbf{F}\mathbf{c}) \tag{2.7}$$

and work to show the equivalence of both sides. Starting from the left hand side

$$\mathbf{F}\left(\sum_{k=1}^{n} c_k\mathbf{P}^{k-1}\right)\mathbf{F}^{-1} = \sum_{k=1}^{n} c_k(\mathbf{F}\mathbf{P}^{k-1}\mathbf{F}^{-1}) = \sum_{k=1}^{n} c_k(\mathbf{F}\mathbf{P}\mathbf{F}^{-1})^{k-1}. \tag{2.8}$$

This result is the same as $(\mathbf{FPF}^{-1})^{k-1}$ since clearly $(\mathbf{FPF}^{-1})^2 = \mathbf{FPF}^{-1}\mathbf{FPF}^{-1} = \mathbf{FP}^2\mathbf{F}^{-1}$, and thus inductively $(\mathbf{FPF}^{-1})^{k-1} = \mathbf{FP}^{k-1}\mathbf{F}^{-1}$.

Note the effect of the general product $\mathbf{FP}^{k-1}\mathbf{F}^{-1}$. Multiplication by the permutation matrix $\mathbf{P}$ cycles through the columns of $\mathbf{F}$ therefore the result of $\mathbf{FP}^{k-1}\mathbf{F}^{-1} = \text{diag}[\text{col}_{(k) \mod (n)}(\mathbf{F})]$. Therefore, the product of $\mathbf{FPF}^{-1} = \text{diag}(1, w, \ldots, w^{n-1})$ and thus (2.8) becomes

$$\sum_{k=1}^{n} c_k(\mathbf{FPF}^{-1})^{k-1} = \sum_{k=1}^{n} c_k \left(\text{diag}(1, w, \ldots, \omega^{n-1})\right)^{k-1} \tag{2.9}$$

Thus the left hand side of (2.7) is

$$\mathbf{FCF}^{-1} = \text{diag}\begin{pmatrix} c_1 + \cdots + c_n \\ c_1 + c_2 w + \cdots + c_n w^{n-1} \\ c_1 + c_2 w^2 + \cdots + c_n w^{2(n-1)} \\ \cdots \\ c_1 + c_2 w^{n-1} + \cdots + c_n w^{(n-1)^2} \end{pmatrix} \tag{2.10}$$

.

The right hand side of (2.8) is a far quicker calculation as the matrix-vector product $\mathbf{Fc}$ can be expressed as

$$\mathbf{Fc} = \begin{bmatrix} 1 & 1 & \vdots & 1 \\ 1 & w & \vdots & w^{n-1} \\ \cdots & \cdots & \vdots & \cdots \\ 1 & w^{n-1} & \cdots & w^{(n-1)^2} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = c_1 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \cdots + c_n \begin{bmatrix} 1 \\ w^{n-1} \\ \vdots \\ w^{(n-1)^2} \end{bmatrix} \tag{2.11}$$

Summing up the vectors gives

$$\mathbf{Fc} = \begin{bmatrix} c_1 + \cdots + c_n \\ c_1 + c_2 w + \cdots + c_n w^{n-1} \\ c_1 + c_2 w^2 + \cdots + c_n w^{2(n-1)} \\ \vdots \\ c_1 + c_2 w^{n-1} + \cdots + c_n w^{(n-1)^2} \end{bmatrix}. \tag{2.12}$$

We finally see that (2.7) holds.

The Toeplitz matrix described in the convolution in (2.1) is not a circulant matrix, instead we must reform the product by imbedding the Toeplitz matrix in a larger proper circulant matrix. Each row and column in a circulant matrix must be cyclic, so each column and row must be continued so that each possess the same information as the next just shifted. To extend each column we simply add n zeros the same procedure done before taking the FFT which helped us to avoid wrap-around problems from circular convolution. So from the original Toeplitz matrix,

$$\mathbf{A} = \begin{bmatrix} a_1 & a_0 & a_{-1} & \cdots & a_{-n+2} & a_{-n+1} \\ a_2 & a_1 & a_0 & \ddots & \ddots & a_{-n+2} \\ a_3 & a_2 & a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_0 & a_{-1} \\ \vdots & & \ddots & a_2 & a_1 & a_0 \\ a_n & a_{n-1} & \cdots & a_3 & a_2 & a_1 \end{bmatrix}. \tag{2.13}$$

a circulant Toeplitz matrix is easily determined

$$\mathbf{C} = \left[\begin{array}{ccccc|ccccc} a_1 & a_0 & \cdots & a_{-n+2} & a_{-n+1} & 0 & a_n & \cdots & a_3 & a_2 \\ a_2 & a_1 & \ddots & \ddots & a_{-n+2} & a_{-n+1} & 0 & \ddots & a_4 & a_3 \\ a_3 & a_2 & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_1 & a_0 & a_{-1} & & \ddots & 0 & a_n \\ a_n & a_{n-1} & \cdots & a_2 & a_1 & a_0 & a_{-1} & \cdots & a_{-n+1} & 0 \\ \hline 0 & a_n & \cdots & a_3 & a_2 & a_1 & a_0 & \cdots & a_{-n+2} & a_{-n+1} \\ a_{-n+1} & 0 & \ddots & a_4 & a_3 & a_2 & a_1 & \ddots & \ddots & a_{-n+2} \\ \vdots & \vdots & \ddots & \ddots & \vdots & a_3 & a_2 & \ddots & \ddots & \vdots \\ a_{-1} & & \ddots & 0 & a_n & \vdots & & \ddots & a_1 & a_0 \\ a_0 & a_{-1} & \cdots & a_{-n+1} & 0 & a_n & a_{n-1} & \cdots & a_2 & a_1 \end{array}\right] \tag{2.14}$$

To compute the product $\mathbf{Ax}$ we imbed $\mathbf{A}$ within a circulant matrix $\mathbf{C}$, and pad $\mathbf{x}$ with n zeros[5].

$$\mathbf{C}\begin{bmatrix}\mathbf{x}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B} & \mathbf{A}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{Tx}\\ \mathbf{Bx}\end{bmatrix} \tag{2.15}$$

The blocked matrix $\mathbf{B}$ is an $n \times n$ matrix encompassing the column elements of $\mathbf{A}$ that are needed to perform a proper circular-shift of the first column and first row of $\mathbf{A}$. More specifically,

$$\mathbf{B} = \begin{bmatrix} 0 & a_n & \cdots & a_3 & a_2 \\ a_{-n+1} & 0 & \ddots & a_4 & a_3 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{-1} & & \ddots & 0 & a_n \\ a_0 & a_{-1} & \cdots & a_{-n+1} & 0 \end{bmatrix} \tag{2.16}$$

The $\mathbf{B}$ matrix never needs to be formed explicitly as $\mathbf{C}$ is simply a Toeplitz matrix with columns described by the circular-shift of vector

$$\mathbf{c} = (a_1, a_2, \ldots, a_n, 0, a_{-n+1}, \ldots, a_{-1}, a_0)^T.$$

Once in this form, we can use the FFT property to calculate

$$\mathbf{y} = \mathbf{C}\begin{bmatrix}\mathbf{x}\\ \mathbf{0}\end{bmatrix} = \mathbf{F}^{-1}\left(\mathbf{Fc}\cdot\mathbf{F}\begin{bmatrix}\mathbf{x}\\ \mathbf{0}\end{bmatrix}\right) \tag{2.17}$$

in $2n\ln(2n)$ or $O(n\ln(n))$ time, i.e. "FFT time". The actual solution is $\mathbf{y}(1:n) = \mathbf{Ax}$ corresponding to the original $n \times n$ Toeplitz matrix product. The circulant matrix $\mathbf{C}$ does not contain any additional information that is not already present in the original Toeplitz matrix $\mathbf{A}$, thus there is no need for any additional storage beyond that of the first column of $\mathbf{A}$. To do the actual computations, routines have been written that never actually form the

circulant matrices explicitly, but instead carry forth the fast Toeplitz multiplication on the length n vectors $\mathbf{a}$, and $\mathbf{x}$ by zero padding, then carrying out the computation in (2.17) and finally storing the first n elements of the result. Therefore we can achieve fast Toeplitz multiplication of a convolution matrix with FFT speeds and through implicit storage of the matrix.

Recalling the seismic moment tensor inversion problem

$$\mathbf{d}_j = \sum_{i=1}^{k} \mathbf{g}_{ji} * \mathbf{m}_i, \qquad j = 1, 2, \ldots, n \tag{2.18}$$

we can represent our problem instead by using Toeplitz matrices:

$$\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \mathbf{d}_n \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & \cdots & \mathbf{G}_{1k} \\ \mathbf{G}_{21} & \ddots & & \vdots \\ \vdots & & & \\ \mathbf{G}_{n1} & \cdots & & \mathbf{G}_{nk} \end{bmatrix} \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_k \end{bmatrix} \tag{2.19}$$

Here $\mathbf{G}_{i,j}$ are Toeplitz matrices which encompass the Green's functions describing the total response for each model component i at each channel j. Now we have the linear system $\mathbf{d} = \mathbf{G}\mathbf{m}$ which can be solved using a number of techniques.

Obviously for this particular system, we want to exploit the efficiency of the method of fast Toeplitz matrix-vector multiplication. Implicit storage and calculation with this method allows for minimal computational cost and can be be utilized through use of iterative techniques. However, if we explicitly tried to store this $\mathbf{G}$ matrix, we would quickly run into storage issues as $\mathbf{G}$ has tens of thousands of rows and columns also making SVD-based pseudo-inverse and Tikhonov regularization solutions impractical [2]. In general, to store a double precision matrix, we need 8 bytes of memory per matrix element. Thus

a 12000 element vector requires approximately 0.09 Megabytes of storage. By contrast, a 12000 by 12000 element matrix needs more than 1.07 Gigabytes to store explicitly. Therefore implicit storage of Toeplitz matrices offers an amazing storage advantage for each $\mathbf{G}_{ij}$ matrix. In the next two chapters, I will be considering systems with up to 9 model components and 18 data channels, which would require over 173 Gigabytes if explicit storage was used, but all that is really needed is less than 20 Megabytes for the entire $\mathbf{G}$ matrix using the implicit storage scheme!

2.2 Inversion with Conjugate Gradient Least Squares (CGLS)

Some inverse methods such as singular value decomposition (SVD) become impractical as the number of parameters increases. In particular to store the SVD result for an $n \times n$ matrix, three $n \times n$ matrices are needed. Aside from the one diagonal matrix, the SVD does not produce sparse matrices, nor are they well structured. There is also no effective scheme for computing the SVD from the implicitly stored matrices that are required to store all of $\mathbf{G}$ matrix, nor any way to store the resulting matrices sparsely.

Many iterative methods are well suited to use the fast Toeplitz multiplication since they may only require the resulting vector from a matrix-vector multiplication. The method of conjugate gradient least squares (CGLS) is an iterative method that is particularly effective in the solution of linear inverse problems. As the name implies, CGLS is the method of conjugate gradients applied to the solution of the general least squares problem $\min\|\mathbf{Gm} - \mathbf{d}\|_2^2$.

By expanding the function to be minimized

$$\|\mathbf{Gm}-\mathbf{d}\|_2^2 = (\mathbf{Gm}-\mathbf{d})^T(\mathbf{Gm}-\mathbf{d}) = \mathbf{m}^T\mathbf{G}^T\mathbf{Gm} - 2\mathbf{G}^T\mathbf{d} + \mathbf{d}^T\mathbf{d}$$

and then setting the gradient with respect to $\mathbf{m}$ to zero results in the normal equations $\mathbf{G}^T\mathbf{Gm} = \mathbf{G}^T\mathbf{d}$. We see that $\mathbf{G}^T\mathbf{G}$ is positive semidefinite, that is $\mathbf{m}^T\mathbf{G}^T\mathbf{Gm} \geq 0$ for all $\mathbf{m}$ since $\mathbf{m}^T\mathbf{G}^T\mathbf{Gm} = \|\mathbf{Gm}\|^2$. $\mathbf{G}^T\mathbf{G}$ is singular in the case where $\mathbf{G}$ is rank deficient.

Within the CGLS algorithm, multiplications involving the $\mathbf{G}$ matrix are needed as well as multiplications involving the transpose of $\mathbf{G}$. The $\mathbf{G}^T$ matrix consists of the transpose of the blocked $\mathbf{G}_{i,j}$ matrices as well as the elements within each of these sub-matrices.

$$\mathbf{G}^T = \begin{bmatrix} \mathbf{G}_{11}^T & \mathbf{G}_{12}^T & \cdots & \mathbf{G}_{1k}^T \\ \mathbf{G}_{21}^T & \ddots & & \vdots \\ \vdots & & & \\ \mathbf{G}_{n1}^T & \cdots & & \mathbf{G}_{nk}^T \end{bmatrix} \tag{2.20}$$

The resulting $\mathbf{G}^T$ matrix is also a Toeplitz matrix since the transpose of a Toeplitz matrix is also a Toeplitz matrix. Therefore all multiplications involving $\mathbf{G}^T$ can also be done using the fast multiplication technique.

The main feature of CGLS (and in fact CG) is the construction of mutually conjugate basis vectors $\mathbf{p}_i$ in which we iteratively approach the solution $\mathbf{x}^*$ as a summation of these basis vectors $\mathbf{x} = \sum_{i=0}^{n-1} \alpha_i \mathbf{p}_i$. During each iteration, each basis vector is a correction in the form of a search direction and step of appropriate length toward the final solution $\mathbf{x}^*$.

The CGLS algorithm outlined by [2] is as follows: Given a system of equations $\mathbf{Gm} = \mathbf{d}$, let $k = 0$, $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{p}_{-1} = \mathbf{0}$, $\beta_0 = 0$, $\mathbf{s}_0 = \mathbf{d}$, and $\mathbf{r}_0 = \mathbf{G}^T\mathbf{s}_0$. Repeat the following steps until convergence,

1. If $k > 0$, let $\beta_k = \dfrac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{r}_{k-1}^T \mathbf{r}_{k-1}}$.
2. Let $\mathbf{p}_k = \mathbf{r}_k + \beta_k \mathbf{p}_{k-1}$.
3. Let $\mathbf{w}_k = \mathbf{G}\mathbf{p}_k$.
4. Let $\alpha_k = \dfrac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{w}_k^T \mathbf{w}_k}$.
5. Let $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$.
6. Let $\mathbf{s}_{k+1} = \mathbf{s}_k - \alpha_k \mathbf{w}_k$.
7. Let $\mathbf{r}_{k+1} = \mathbf{G}^T \mathbf{s}_{k+1}$.
8. Let $k = k + 1$.

CGLS starts with a beginning model $\mathbf{x} = \mathbf{0}$ then, in each iteration step, the model increases from the addition of basis vectors $\mathbf{x} = \sum_{i=0}^{n-1} \alpha_i \mathbf{p}_i$. Stopping the CGLS algorithm early results in a short solution which has a regularization effect referred to as implicit regularization [6].

Initially for the Mount Erebus data, CGLS with early termination was used. This works by observing the iteration history plot of $\|\mathbf{G}\mathbf{m}_k - \mathbf{d}\|$ versus $\|\mathbf{m}_k\|$. The resulting plot resembles an "L" shaped curve. We then choose to terminate the algorithm when a distinct corner on the L appears. This corner represents a point of appropriate trade-off between the misfit $\|\mathbf{G}\mathbf{m}_k - \mathbf{d}\|$ and $\|\mathbf{m}_k\|$. This approach however does not impose enough of a regularization effect by itself as the noise levels still have significant influence on the recovered model. Therefore extra regularization must be used with CGLS to adequately recover the model.

2.3 CGLS with Explicit Regularization

Tikhonov regularization attempts to obtain stability by solving the damped least squares problem

$$\min \|\mathbf{Gm} - \mathbf{d}\|_2^2 + \alpha^2 \|\mathbf{m}\|_2^2 \tag{2.21}$$

which uses an adjustable regularization parameter $\alpha > 0$. The solution to the damped least squares problem ensures minimization of the model $\mathbf{m}$ which is the primary regularization effect. There may be many valid models which satisfy the least squares solution, but by introducing this new criterion the solution the model will be of minimal length. This is known as Zeroth-order Tikhonov regularization.

In general, Tikhonov regularization imposes a filter factor on the term $\alpha^2\|\mathbf{Lm}\|$ where $\mathbf{L}$ is a roughening matrix which incorporates a particular filter on the model $\mathbf{m}$, in Zeroth-order problem (2.21) $\mathbf{L} = \mathbf{I}$. An alternative view of the damped least squares problem is viewing the problem as an ordinary least squares problem obtained by augmenting the least squares problem

$$\min \|\mathbf{Gm} - \mathbf{d}\|^2 + \alpha^2 \|\mathbf{Lm}\|^2$$

as

$$\min \left\| \begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \tag{2.22}$$

The function to be minimized with respect to $\mathbf{m}$ can be simplified using matrix algebra

$$f(\mathbf{m}) = \left\| \begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right\|_2^2 = \left(\begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right)^T \left(\begin{bmatrix} \mathbf{G} \\ \alpha\mathbf{L} \end{bmatrix} \mathbf{m} - \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \right) \tag{2.23}$$

or equivalently

$$f(\mathbf{m}) = \mathbf{m}^T \begin{bmatrix} \mathbf{G}^T & \alpha \mathbf{L}^T \end{bmatrix} \begin{bmatrix} \mathbf{G} \\ \alpha \mathbf{L} \end{bmatrix} \mathbf{m} - 2\mathbf{m}^T \begin{bmatrix} \mathbf{G}^T & \alpha \mathbf{L}^T \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{d}^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix}. \tag{2.24}$$

The gradient with respect to $\mathbf{m}$ is needed for minimization, which by expansion becomes

$$\nabla f(\mathbf{m}) = 2\left(\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L}\right)\mathbf{m} - 2\mathbf{G}^T\mathbf{d}. \tag{2.25}$$

and thereby setting the gradient to zero and solving for $\mathbf{m}$ gives the new normal equations $(\mathbf{G}^T\mathbf{G} + \alpha^2\mathbf{L}^T\mathbf{L})\mathbf{m} = \mathbf{G}^T\mathbf{d}$ [2]. Therefore, if $\mathbf{L}^T\mathbf{L}$ is nonsingular, then the addition of $\alpha^2\mathbf{L}^T\mathbf{L}$ to $\mathbf{G}^T\mathbf{G}$ is also nonsingular, resulting in the desired stability. Filter factors of all types can be used generally based upon preference of the desired regularization effects. The level of influence can be directly controlled by adjusting the regularization parameter α.

The choice for the values of α to use is not arbitrary, instead, CGLS is run with a variety of α values. A log-log plot of the of the final misfit $\|\mathbf{Gm}-\mathbf{d}\|$ versus the semi-norm $\|\mathbf{Lm}\|$ for each value of α used in the CGLS algorithm produces a L-shaped curved called an "L-curve" plot [6]. The L-curve is produced because the misfit and semi-norm are functions of the regularization parameter α, reducing α increases the semi-norm $\|\mathbf{Lm}\|$ and decreases the misfit. Explicit regularization is a means of choosing the value of α which helps best stabilize the solution while not over influencing the recovered model. Selection of α is based upon the **L-curve criterion** (although there are alternative selection criteria that can be used), which selects that α which gives the solution closest to the corner of the L-curve [2].

Explicit regularization is needed for the moment tensor inversion associated with the Mount Erebus data. The nature of the noise present in the seismogram, which contains background noise from the oceanic microseism, requires the use of an additional filter. Thus the $\mathbf{K}$ filter is a Toeplitz matrix which houses a high-pass FIR filter which filters out signals with periods less than or equal to 5 seconds. The $\mathbf{K}$-filter is identical to the dimensions of $\mathbf{G}$ and is therefore able to use fast Toeplitz multiplication.

2.4 Storage Requirements

CGLS with fast Multiplication requires the storage of the large $\mathbf{G}$ matrix. The fast Toeplitz multiplication method allows for implicit storage of each of the sub-matrices as vectors each the length of a data channel vector. In the most efficient form of the algorithm, the precomputed FFT of $\mathbf{G}$ and $\mathbf{G}^T$ are stored instead of $\mathbf{G}$. Given a the number of seismic stations n, the length of a data channel vector s, and the number of model components c. The storage of $\mathbf{G}$ is thus $(3\times n\times s\times c) = 3\times(\#\text{ of stations})\times(\text{length of data channel vector})\times(\#\text{ of model components})$. The storage requirement for the FFT of $\mathbf{G}$ is then four times that of storing $\mathbf{G}$ alone since the FFT is on the zero padded vectors $\mathbf{g}_{ij}$ and the result is complex. Thus the storage for both the FFT of $\mathbf{G}$ and $\mathbf{G}^T$ is $24nsc$. Using Tikhonov regularization requires the additional storage of the FFT of both $\mathbf{L}$ and $\mathbf{L}^T$ bringing the storage total to $48nsc$. This is an improvement over the parameterization method which had a storage need of $3nscp$. The two algorithms would be identical in the case where $p = 12$, but this small number of triangles is probably insufficient in most moment tensor inversions. Therefore the CGLS algorithm with fast Toeplitz multiplication

and Tikhonov regularization has an advantage when it comes to storage.

2.5 Model Fit

To evaluate the quality of the recovered model obtained from any inverse method, we must evaluate the goodness of the fit of the synthetic data generated by computing the forward problem and compare this result to observed data. A simple quantitative measure of the goodness of fit is computed by $\|\mathbf{Gm} - \mathbf{d}\|$. Another measure of fit used by seismologists is the variance reduction (VR), computed by

$$\text{Variance Reduction} = \left(1 - \frac{\|\mathbf{Gm} - \mathbf{d})\|^2}{\|\mathbf{d}\|^2}\right) \times 100\%. \tag{2.26}$$

CHAPTER 3

SYNTHETIC MODELS

In this chapter I will present synthetic models for 3 seismic moment tensor configurations. These include the following models: Mogi, Dilitational, and the Dilitational plus three Single Forces. The nature of these synthetics are to be very general time-series which may not necessarily represent seismic moment tensor time functions, but rather any mathematical tensor function of interest.

3.1 Synthetic Mogi Model

The Mogi model is the most basic system represented by a single moment tensor component. A generic synthetic model was created and used in the forward problem to produce data. A clean data channel seen in figure 3.2 needs a level of noise added which may serve to represent what it seen by an actual instrument. The nature of the noise is that of white noise mixed with an oscillatory signal which represents what may be seen from the oceanic microseism.

The CGLS algorithm with Tikhonov regularization does a nice job of recovering the model from the noisy data. The fit to the data is quite good with a variance reduction of 99.85 %.

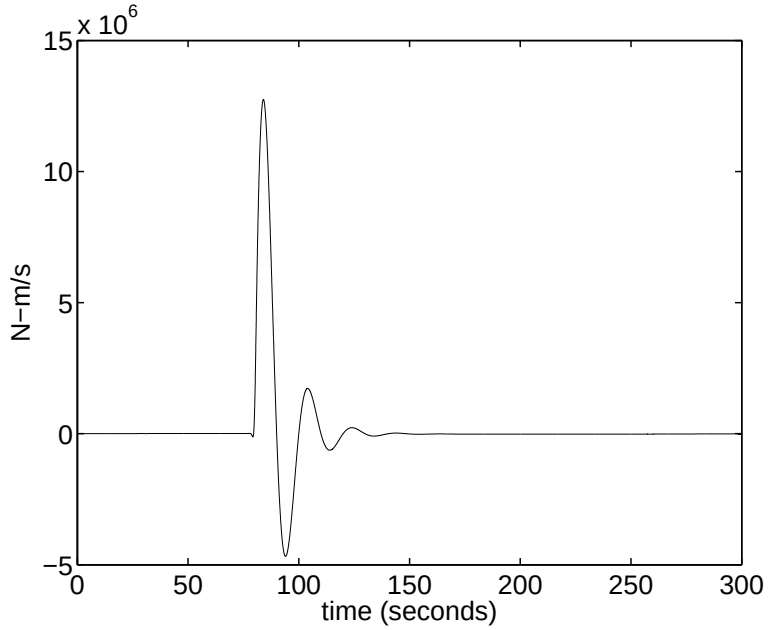

Figure 3.1: Original Synthetic Mogi Model

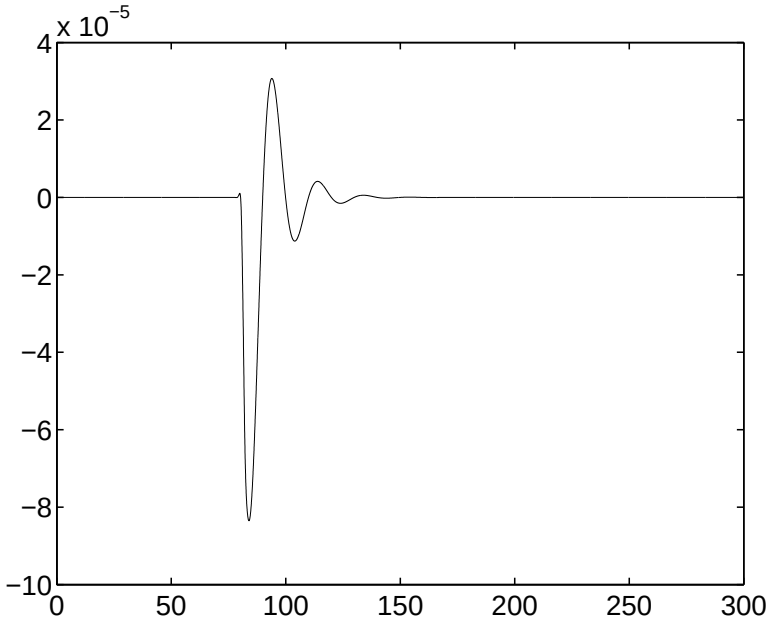

Figure 3.2: Original Synthetic Data CON R component

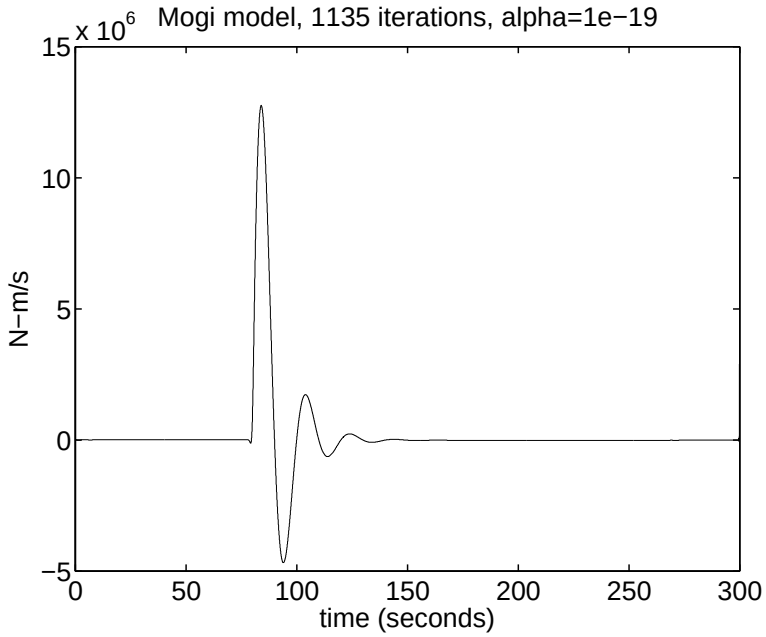

Figure 3.3: Recovered Synthetic Mogi Model

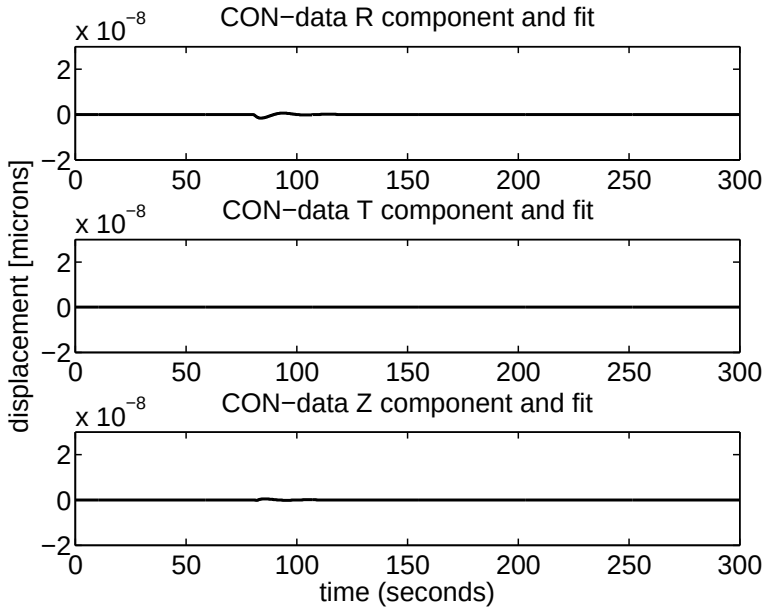

Figure 3.4: Synthetic Mogi Model CON Data Fit

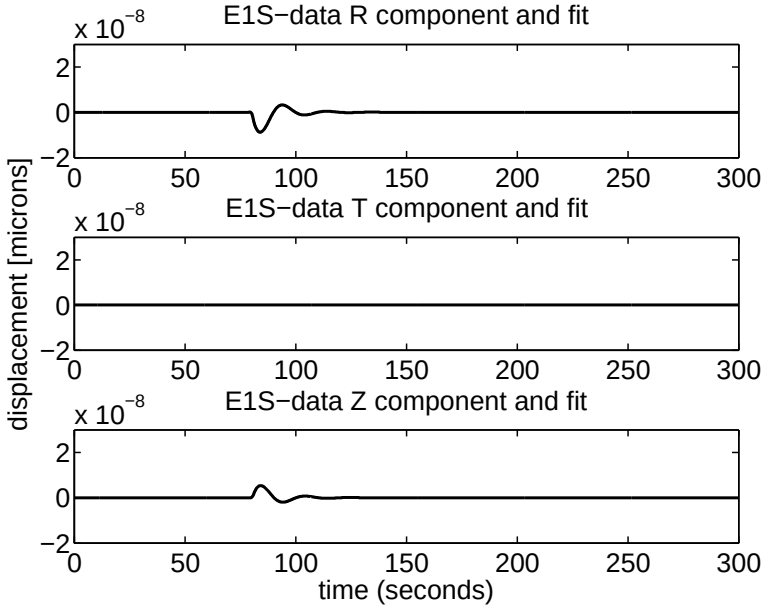

Figure 3.5: Synthetic Mogi Model E1S Data Fit

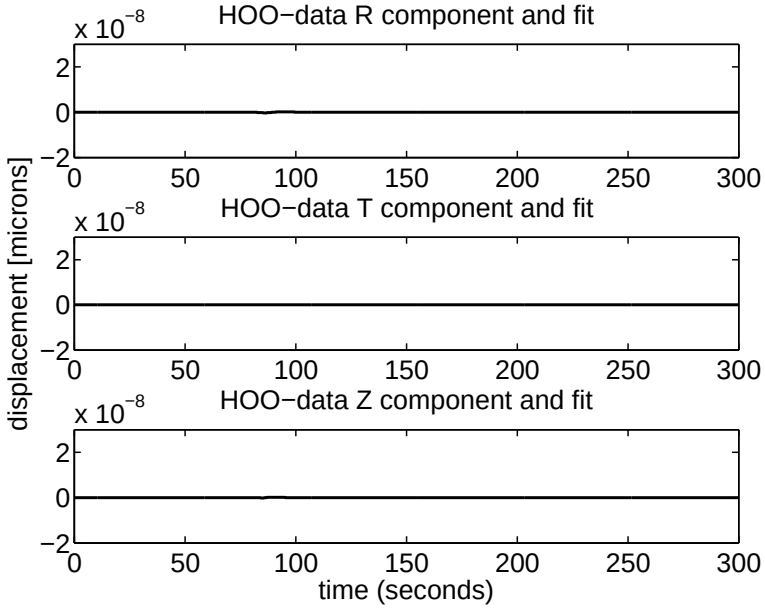

Figure 3.6: Synthetic Mogi Model HOO Data Fit

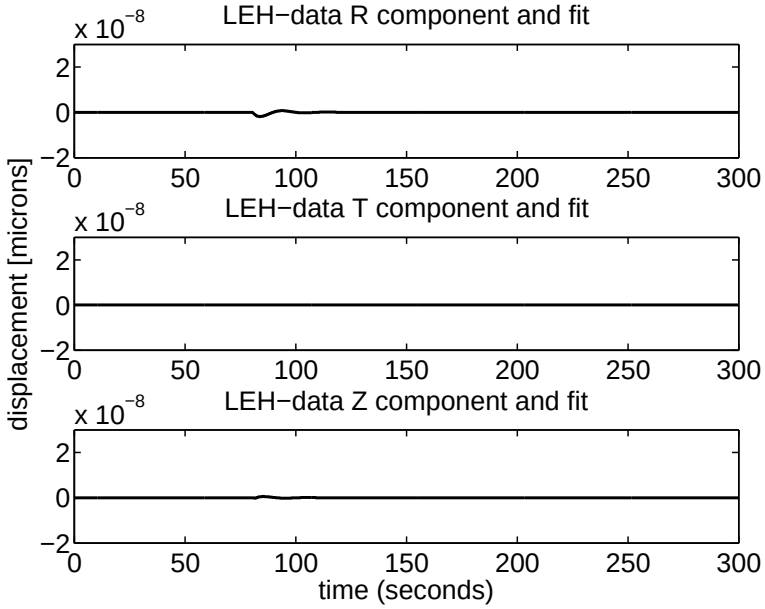

Figure 3.7: Synthetic Mogi Model LEH Data Fit

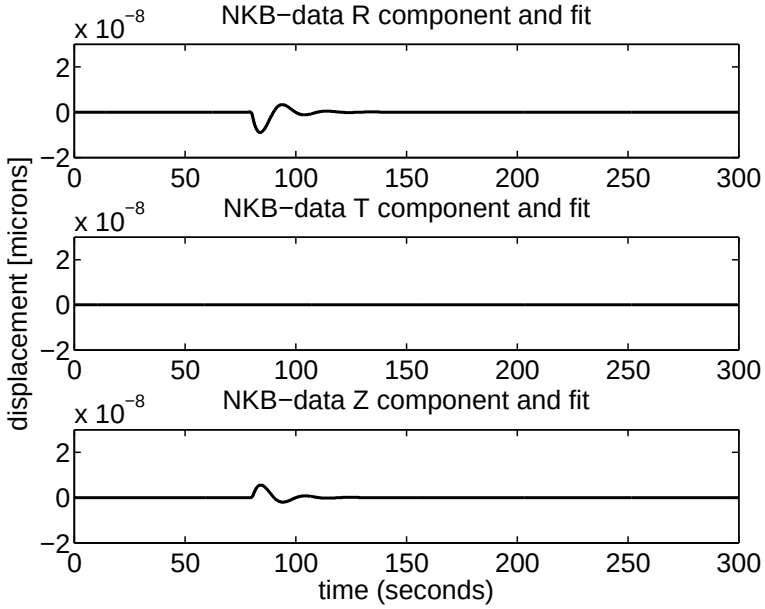

Figure 3.8: Synthetic Mogi Model NKB Data Fit

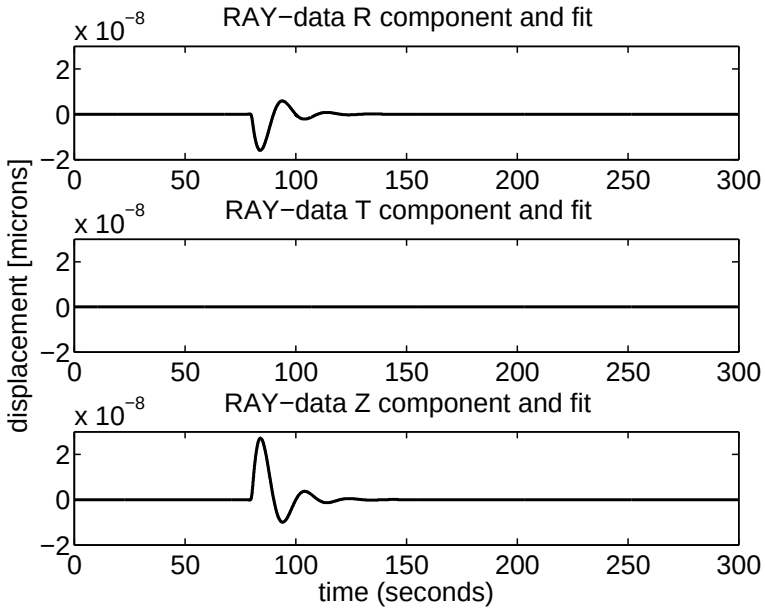

Figure 3.9: Synthetic Mogi Model RAY Data Fit

3.2 Synthetic Dilitational Model

The Dilitational model is the next step up in complexity as the moment tensor has three components to recover and additional Green's functions. In figure 3.10 there are three unique synthetic models used to generate the synthetic data. The recovered models are smooth and free of noise. A variance reduction of 99.94% further illustrates an excellent fit to the data.

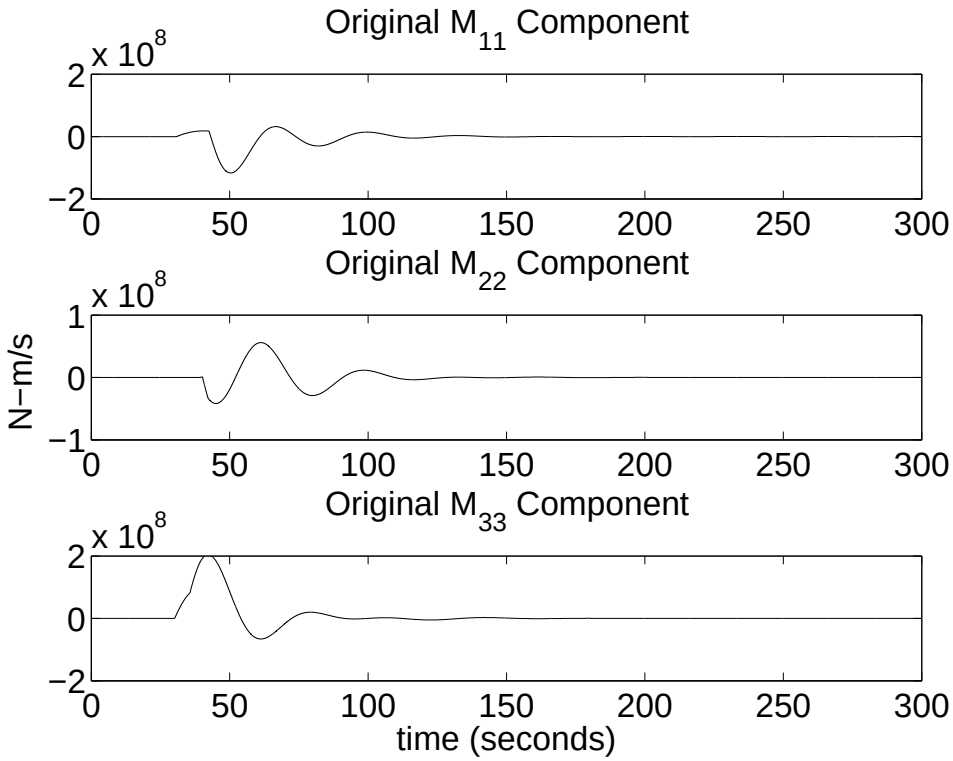

Figure 3.10: Synthetic Dilitational Original M Components

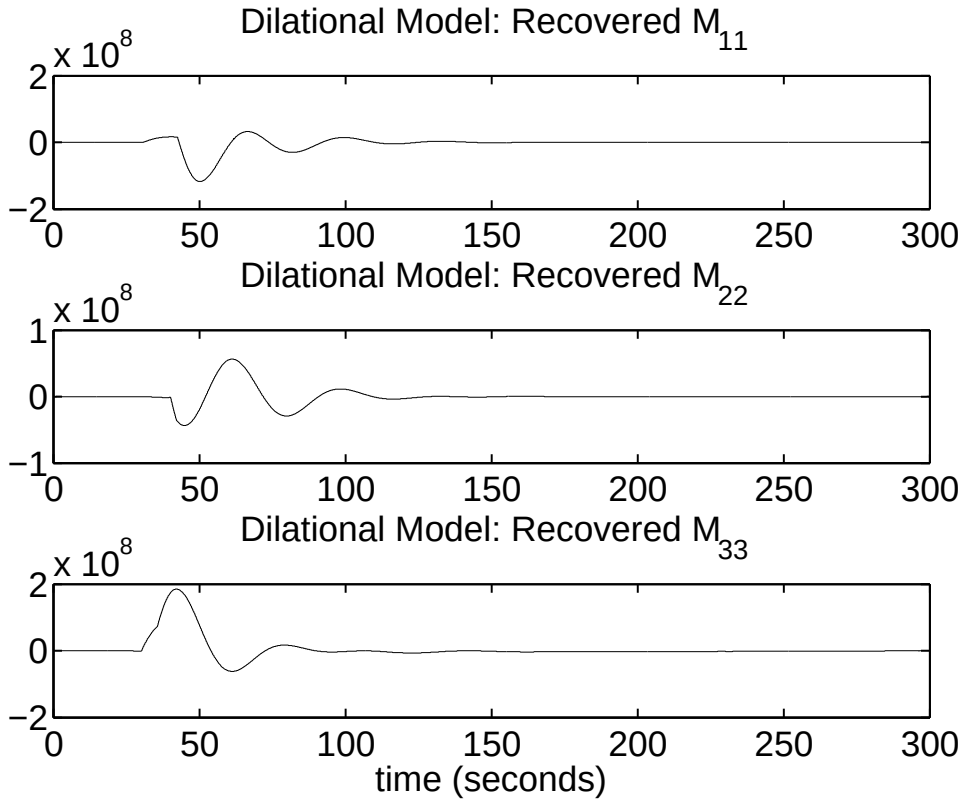

Figure 3.11: Synthetic Dilitational Recovered M Components

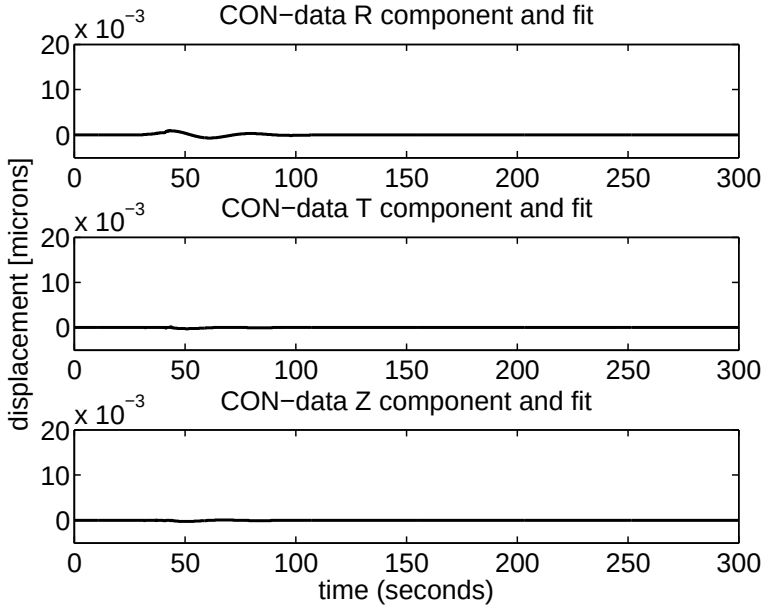

Figure 3.12: Synthetic Dilitational Model CON Data Fit

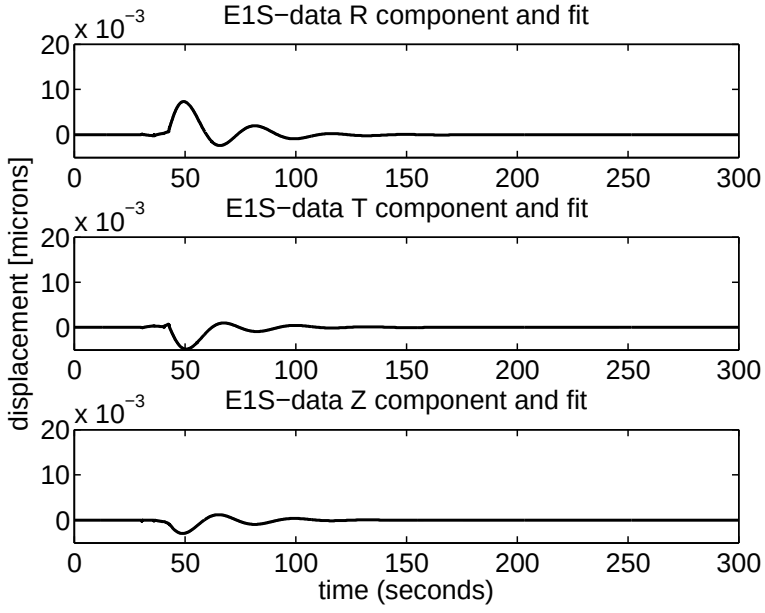

Figure 3.13: Synthetic Dilitational Model E1S Data Fit

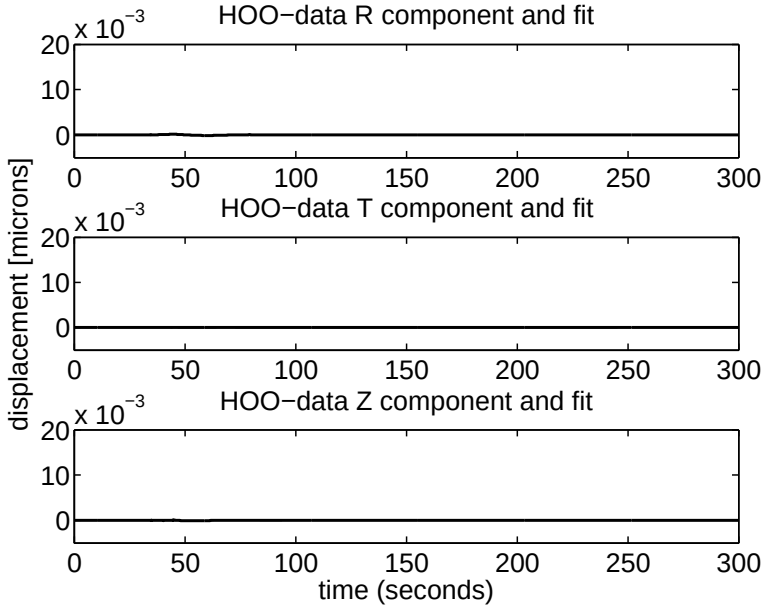

Figure 3.14: Synthetic Dilitational Model HOO Data Fit

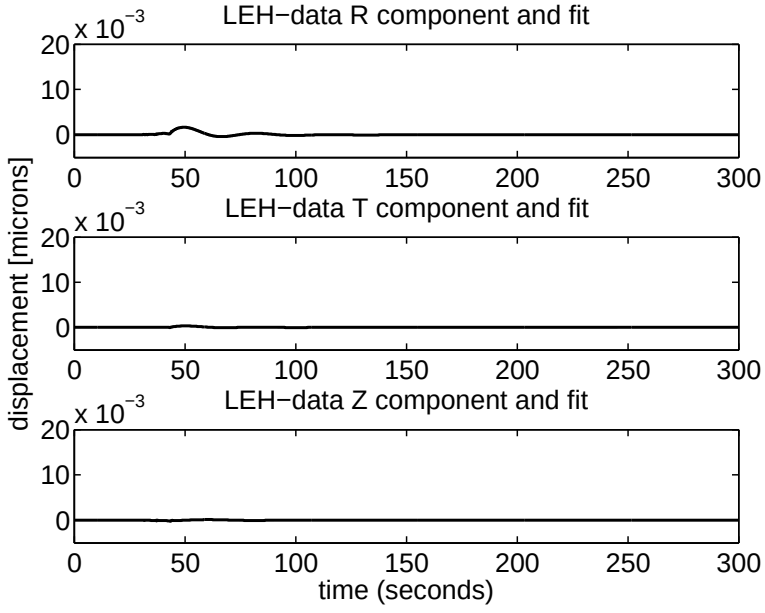

Figure 3.15: Synthetic Dilitational Model LEH Data Fit

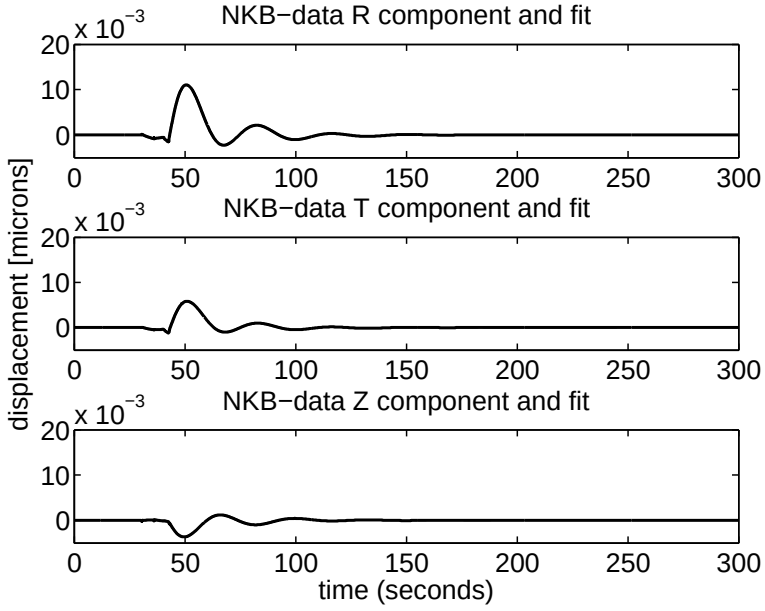

Figure 3.16: Synthetic Dilitational Model NKB Data Fit

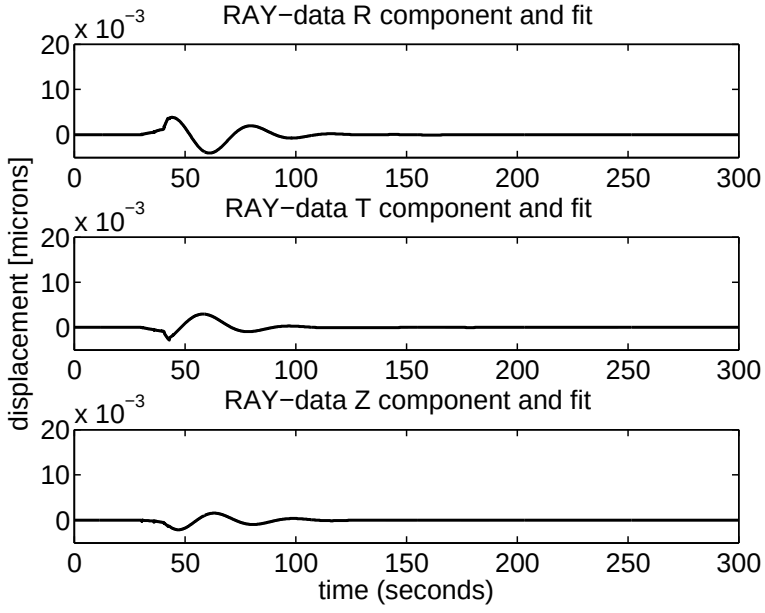

Figure 3.17: Synthetic Dilitational Model RAY Data Fit

3.3 Synthetic Dilitational Plus Three Directional Single Forces Model

The synthetic dilitational plus three directional single forces model has six model components. In the previous two synthetic examples the recovered models were smooth and free of high frequency noise. This example illustrates a problem that will be seen in working with real data. The three components describing the single forces are pretty well recovered but the dilitational components have several problems. First the dilitational components seem to contain lots of high frequency noise. Secondly, the models seem to contain the periodical noise from the simulated oceanic microseism. Finally, while models do have some resemblance to the original dilitational components, the magnitudes are not equivalent.

This model has a variance reduction of 98.72% which would suggest a very good fit to the data, but also suggests over-fitting of the data. The presence of high frequency noise and mismatched magnitudes suggests problems with the level of regularization.

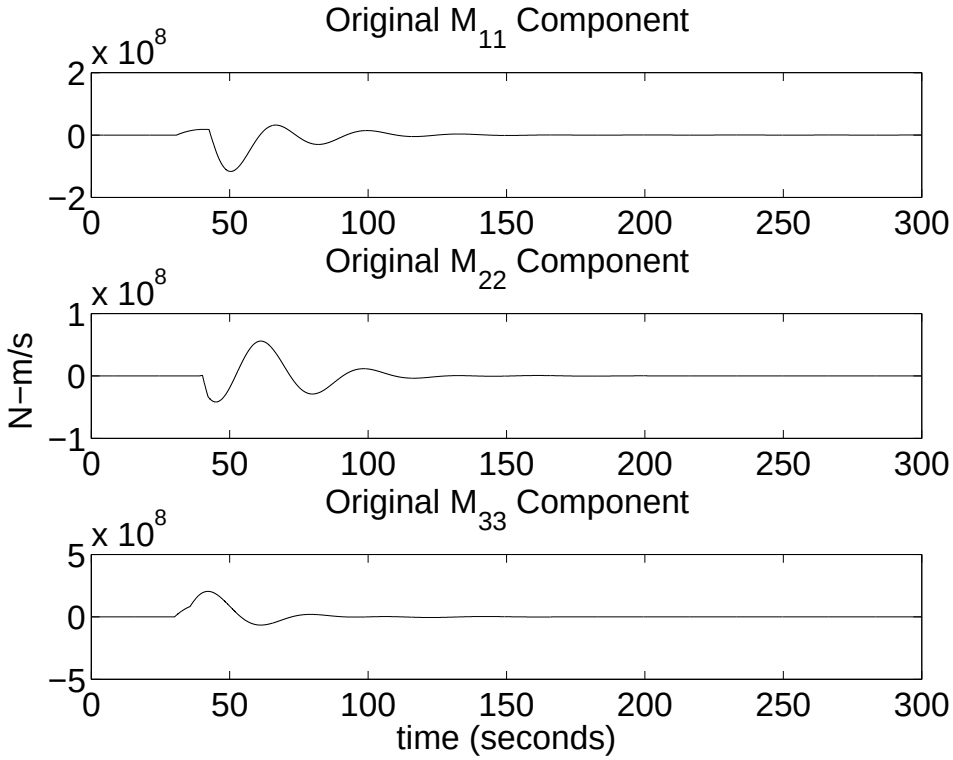

Figure 3.18: Synthetic Dilitational Plus Three Directional Single Forces Original M Components

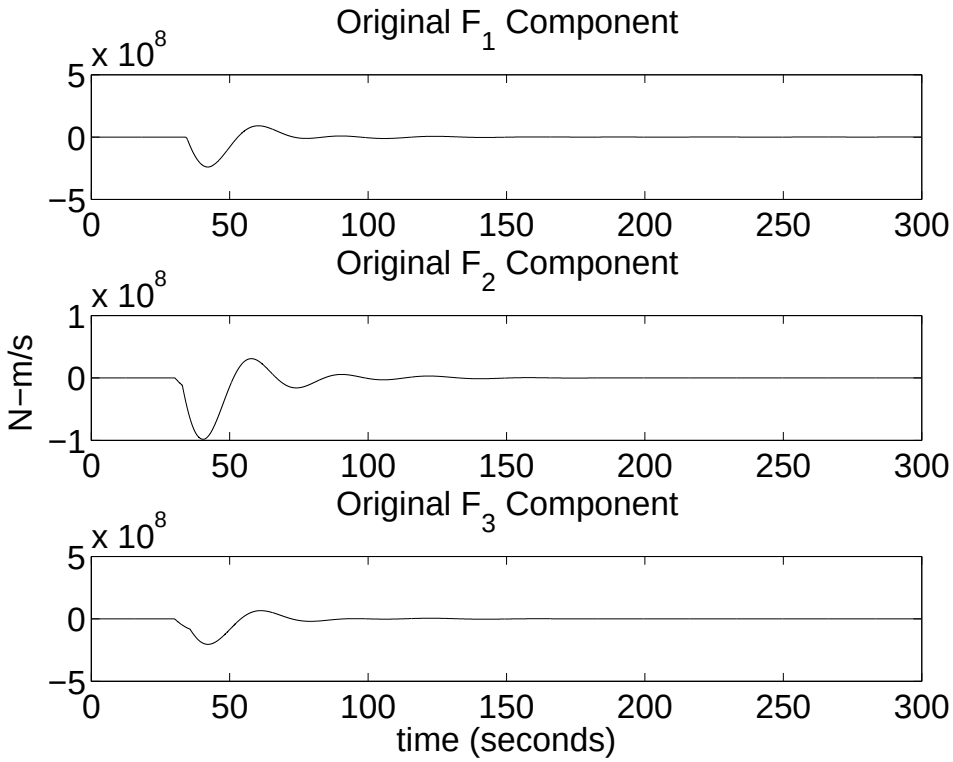

Figure 3.19: Synthetic Dilitational Plus Three Directional Single Forces Original F Components

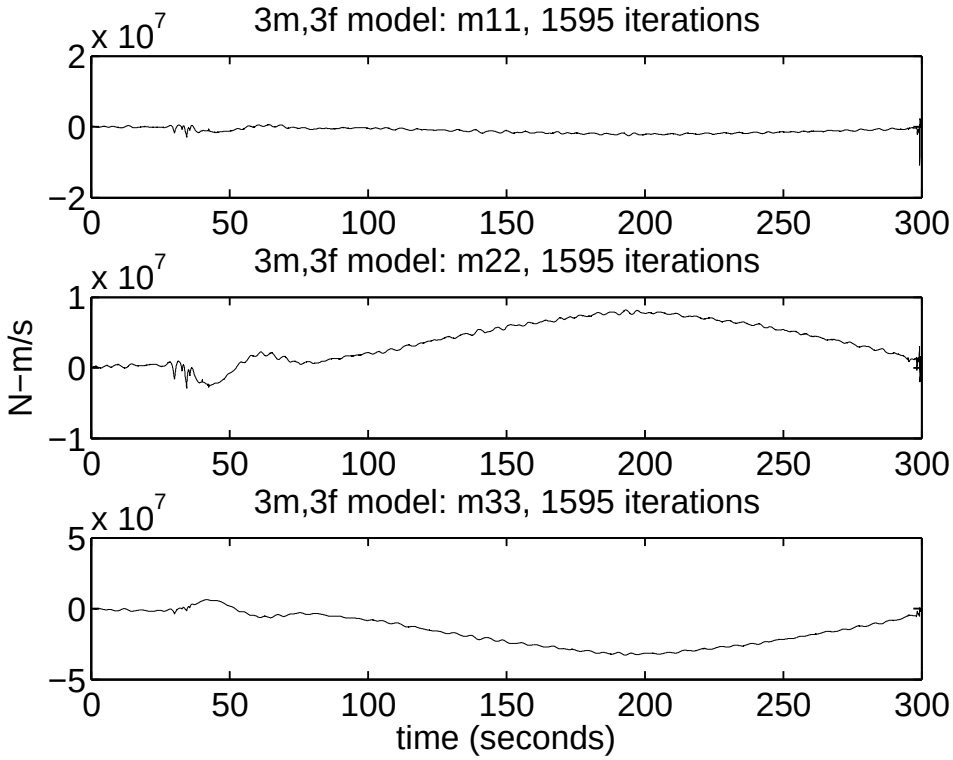

Figure 3.20: Synthetic Dilitational Plus Three Directional Single Forces Recovered F Components

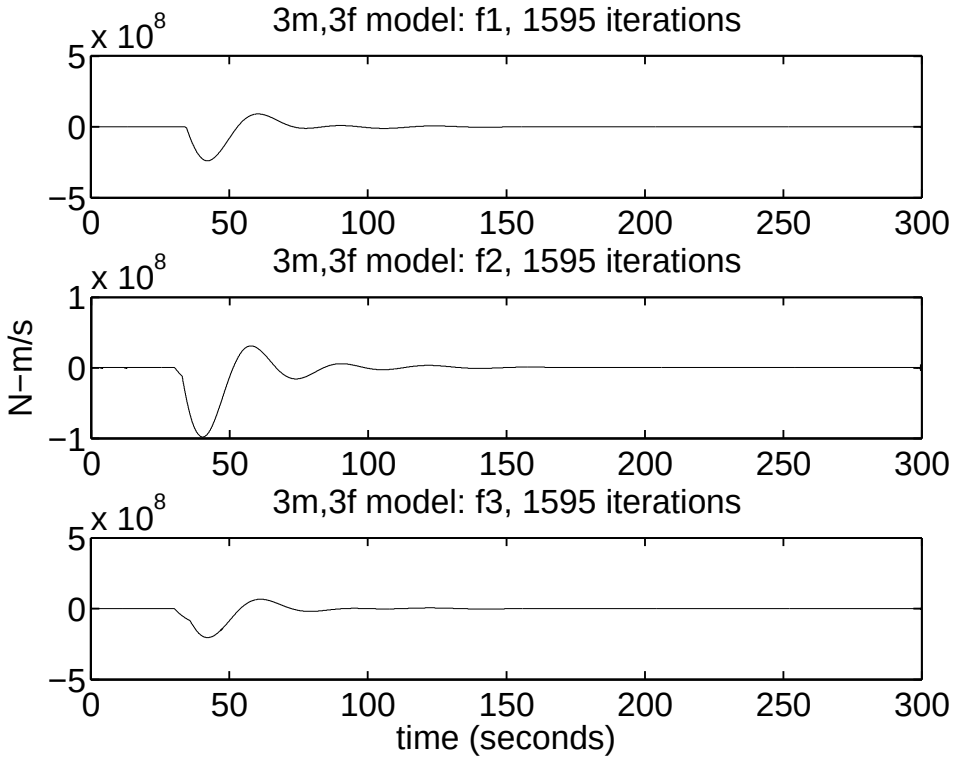

Figure 3.21: Synthetic Dilitational Plus Three Directional Single Forces Recovered F Components

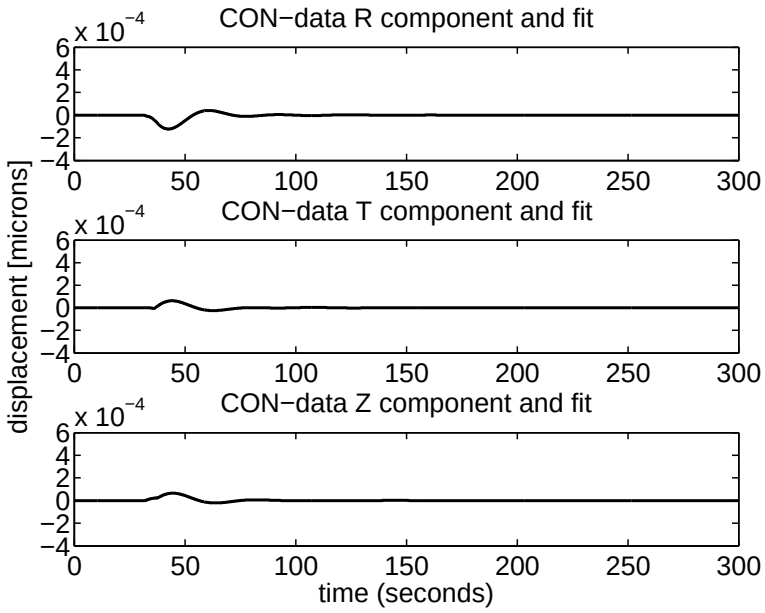

Figure 3.22: Synthetic Dilitational Plus Three Directional Single Forces Model CON Data Fit

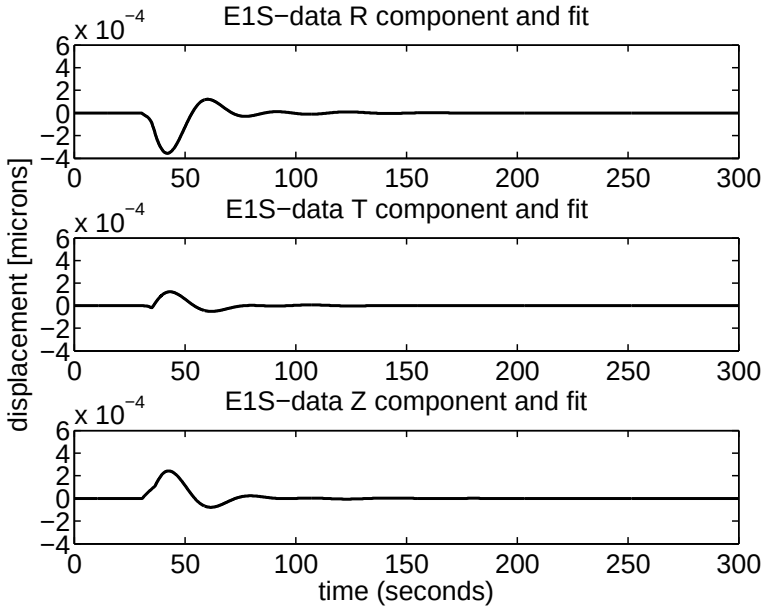

Figure 3.23: Synthetic Dilitational Plus Three Directional Single Forces Model E1S Data Fit

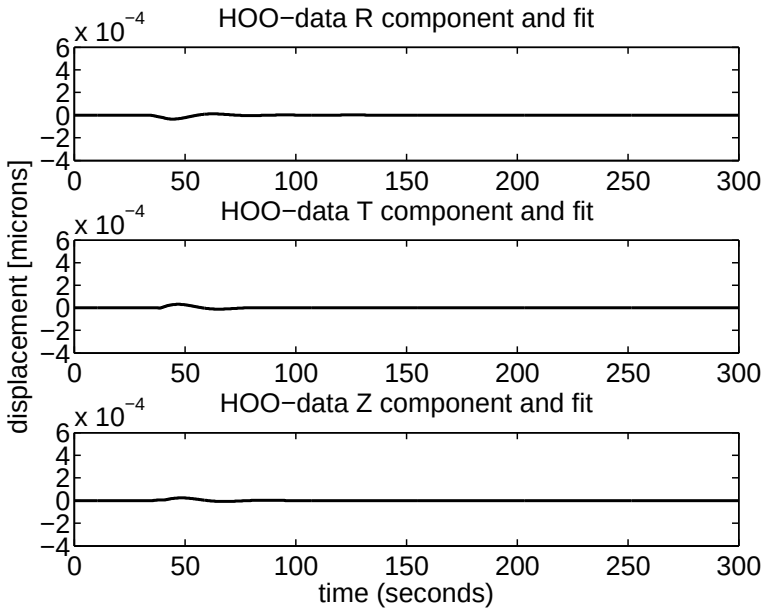

Figure 3.24: Synthetic Dilitational Plus Three Directional Single Forces Model HOO Data Fit

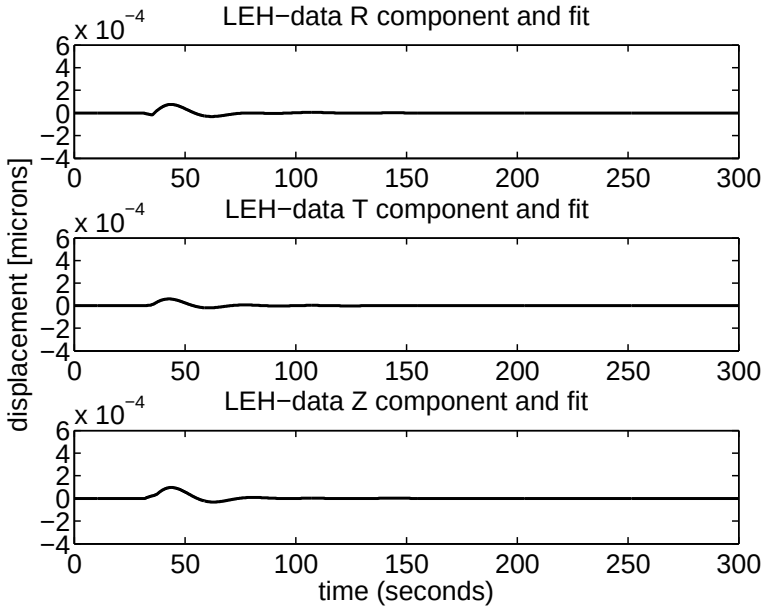

Figure 3.25: Synthetic Dilitational Plus Three Directional Single Forces Model LEH Data Fit

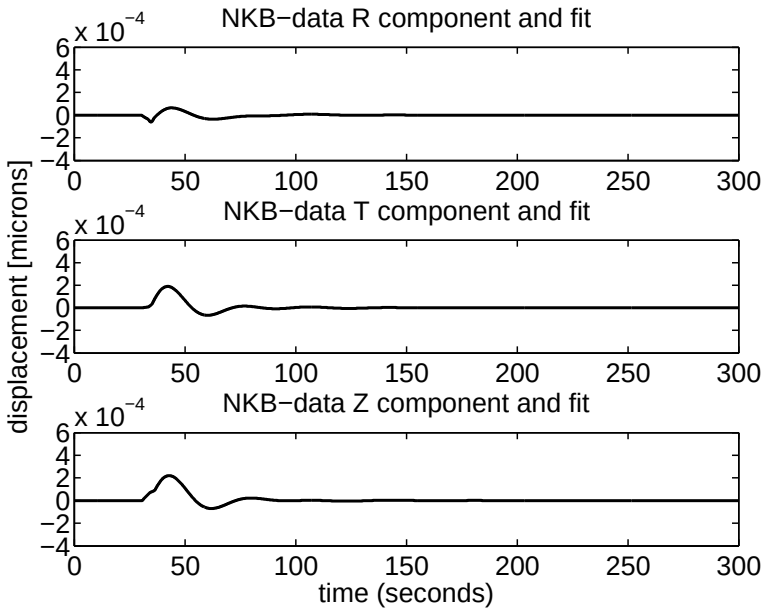

Figure 3.26: Synthetic Dilitational Plus Three Directional Single Forces Model NKB Data Fit

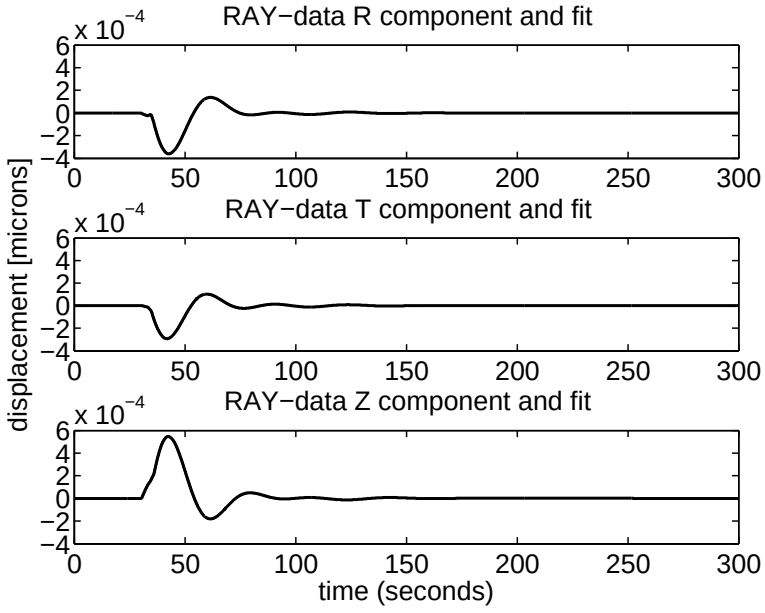

Figure 3.27: Synthetic Dilitational Plus Three Directional Single Forces Model RAY Data Fit

CHAPTER 4

STACKED DATA

The stacked data was run using CGLS with explicit regularization describing the least squares problem in (**??**). Six model configurations were considered for inversion each consisting of a Green's function matrix describing the according Green's function for each double couple (or single force) at the desired depth.

The Green's functions are computed by Fortran code based upon the methods outlined by Johnson [7] for the source location determined by Aster [3] at depths of 200, 400, 800, and 1200 meters. The following inversions are all done using the 400 meter depth Green's functions.

4.1 The Mogi Model

The inversion process requires running the CGLS algorithm with regularization for a series of trials to find the appropriate regularization parameter by using the L-curve criterion. The associated L-curve in figure 4.1 shows a distinct corner solution with regularization parameter $\alpha = 5 \times 10^{-}19$. The associated recovered Mogi model appears well behaved and free of high frequency noise.

The data fits illustrate a few interesting results. Some stations are very well fit, while others exhibit a slight anomaly. This anomaly is a slight

misalignment of signal amplitudes and is mostly explained by the CGLS algorithm itself. The CGLS algorithm favors the signals with larger amplitudes and will generally do its best to fit those associated signals. These signals generally correspond to stations closer to the volcano source. The stations further away, with their smaller amplitudes, and therefore have much less influence on the data fits resulting in a recovered model which best fits the closer stations. The stations E1S, NKB, and RAY show good fits to the data in both amplitude and frequency. However for the overall data fit the variance reduction is only 61.69%.

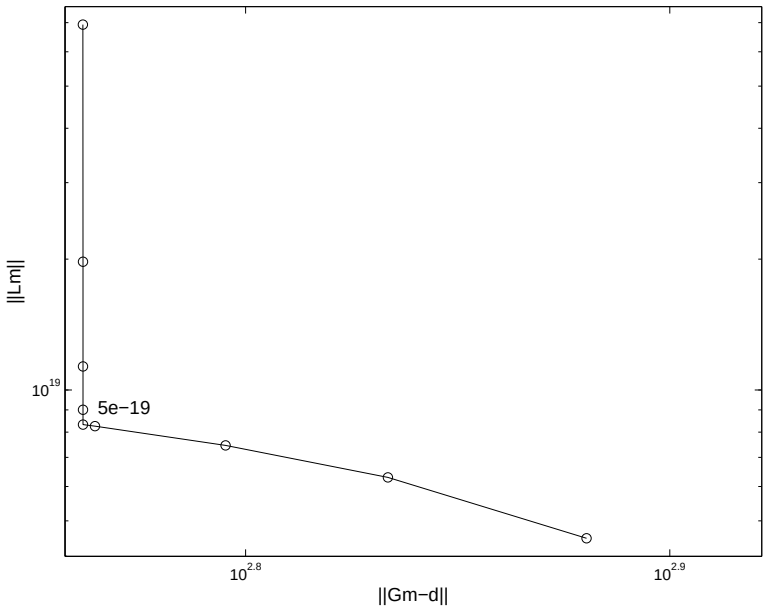

Figure 4.1: Mogi Model Stacked 2005 Data L-curve

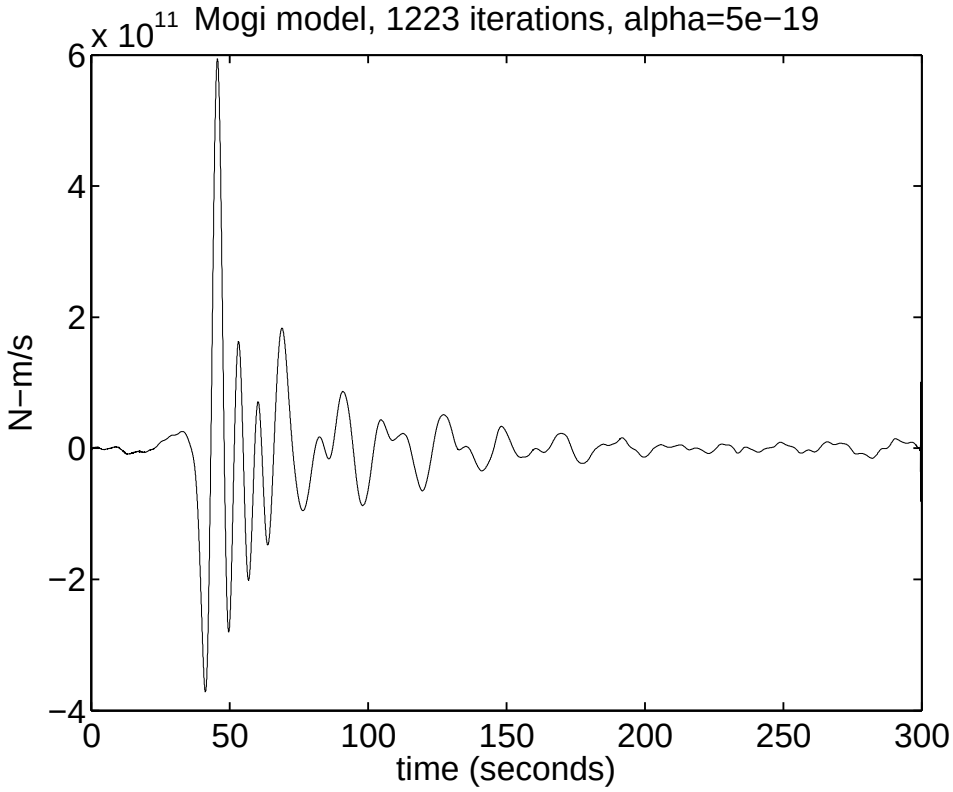

Figure 4.2: The Recovered Mogi Model on Stacked 2005 Data

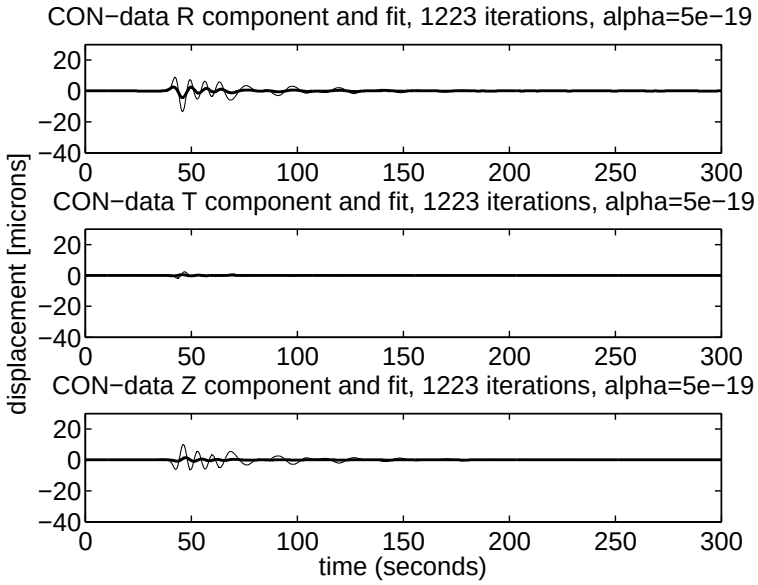

Figure 4.3: Stacked 2005 Data Mogi Model CON Data Fit

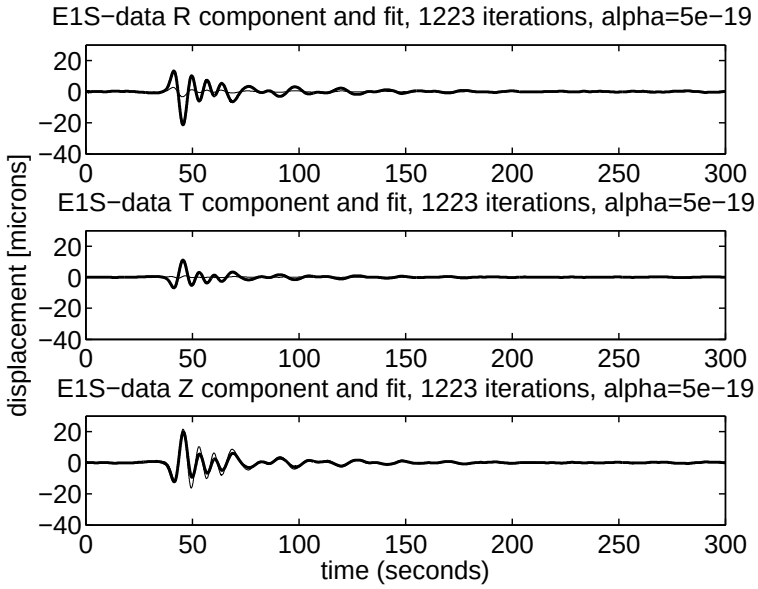

Figure 4.4: Stacked 2005 Data Mogi Model E1S Data Fit

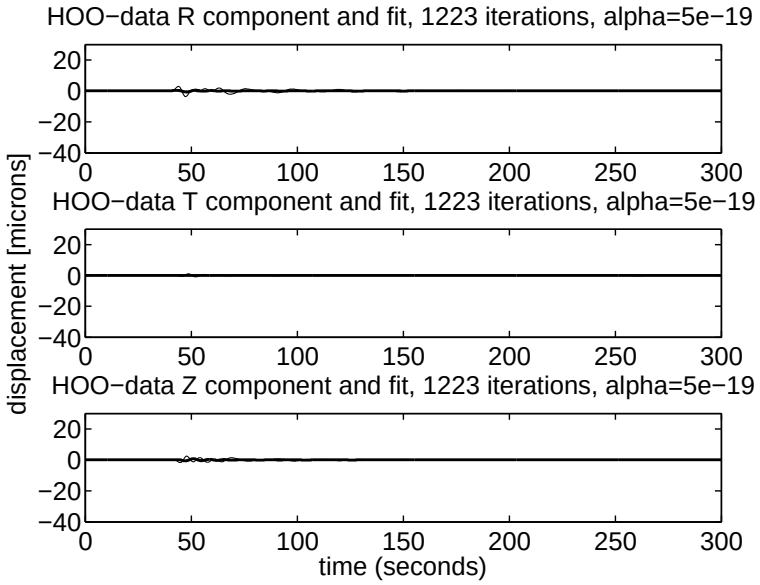

Figure 4.5: Stacked 2005 Data Mogi Model HOO Data Fit

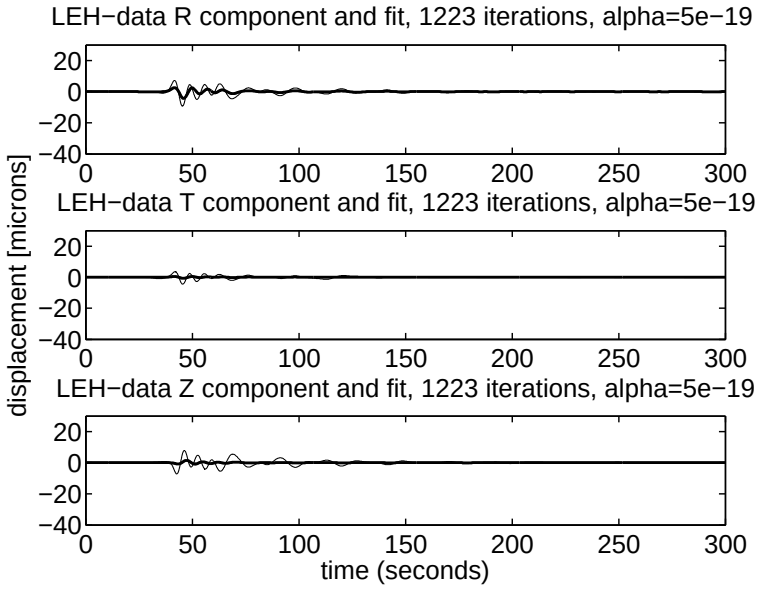

Figure 4.6: Stacked 2005 Data Mogi Model LEH Data Fit

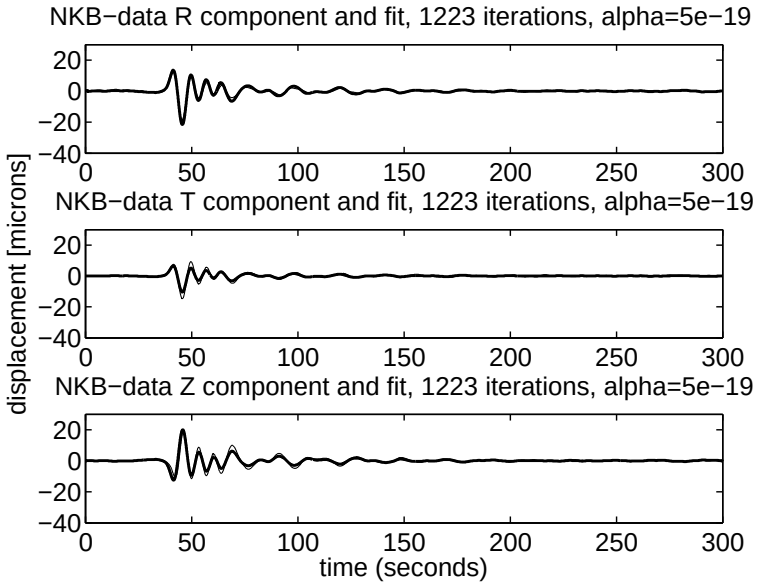

Figure 4.7: Stacked 2005 Data Mogi Model NKB Data Fit

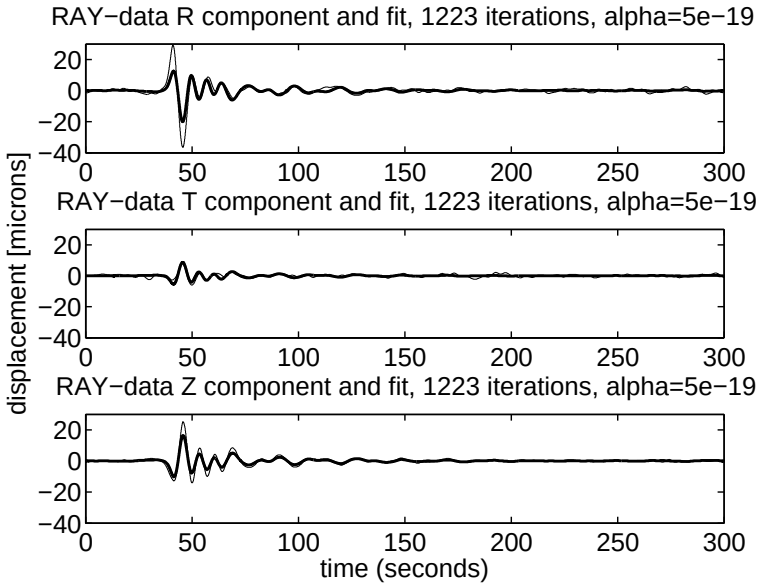

Figure 4.8: Stacked 2005 Data Mogi Model RAY Data Fit

4.2 Dilitational Plus Three Directional Single Forces Model

The dilitational plus three directional single forces model is presented for a regularization parameter $\alpha = 3 \times 10^{-}19$ and a filter matrix with a 5 second cutoff period. Note that in this instance, the CGLS algorithm has not converged, but was instead stopped at 24,000 iterations. The recovered dilitational components for the dilitational plus three directional single forces seem sufficiently regularized and of appropriate amplitude. The single force components do contain a level of high frequency noise. This is most likely due to the CGLS algorithm not fully converging. The variance reduction for this model is 90.24% which suggests a very good fit to the data.

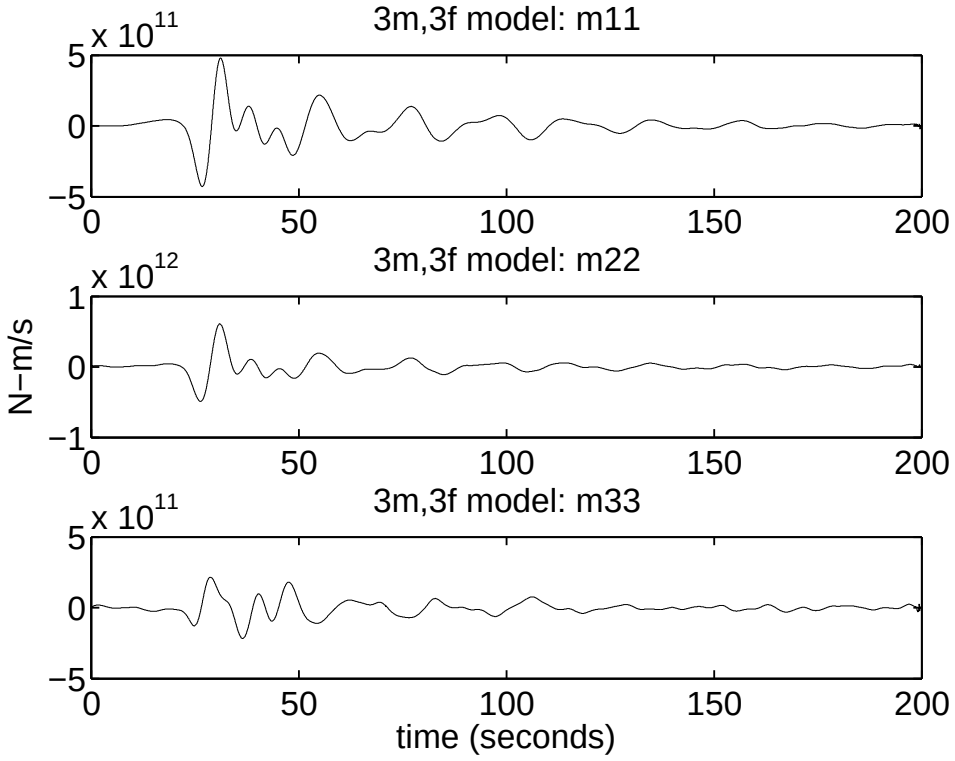

Figure 4.9: The Recovered 3m3f Model m_1, m_2, m_3 components on Stacked 2005 Data

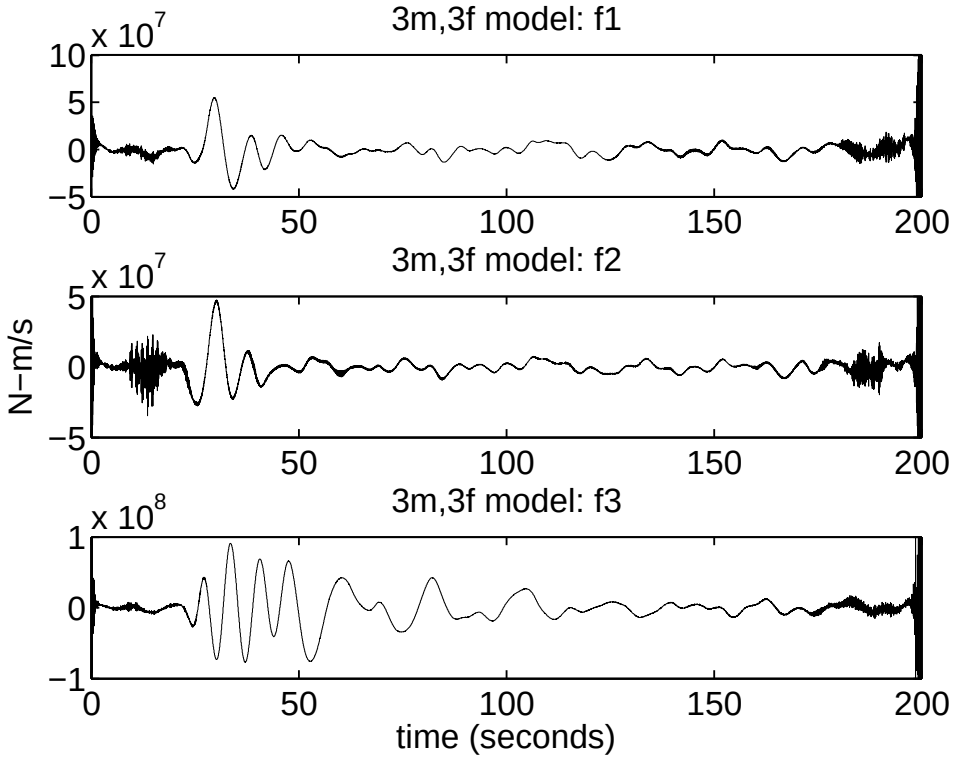

Figure 4.10: The Recovered 3m3f Model f_1, f_2, f_3 components on Stacked 2005 Data

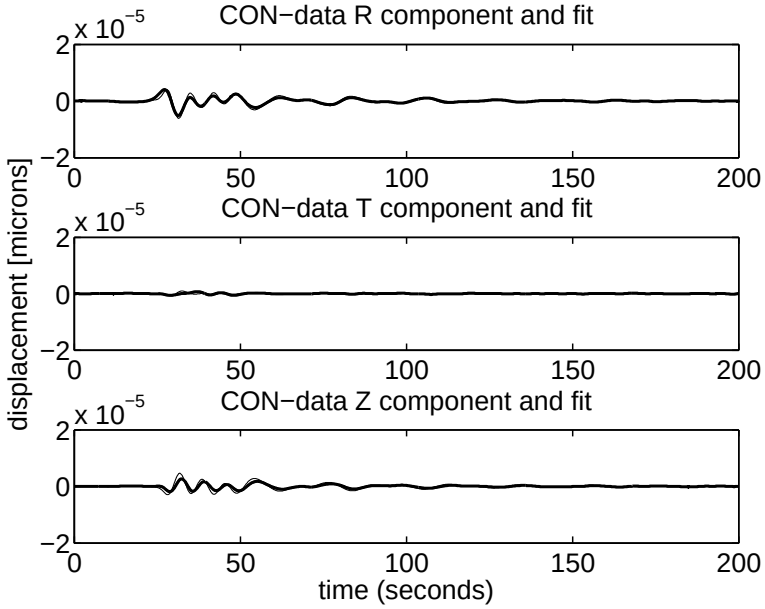

Figure 4.11: Stacked 2005 Data 3m3f Model CON Data Fit

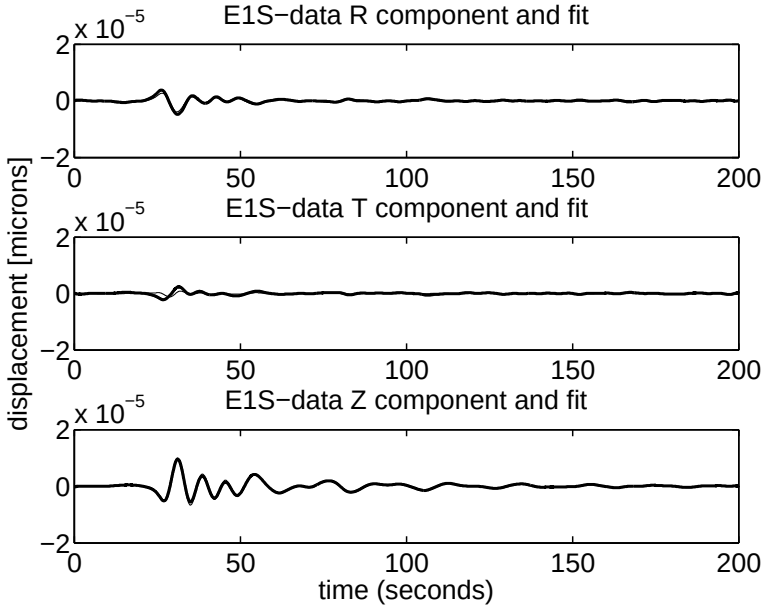

Figure 4.12: Stacked 2005 Data 3m3f Model E1S Data Fit

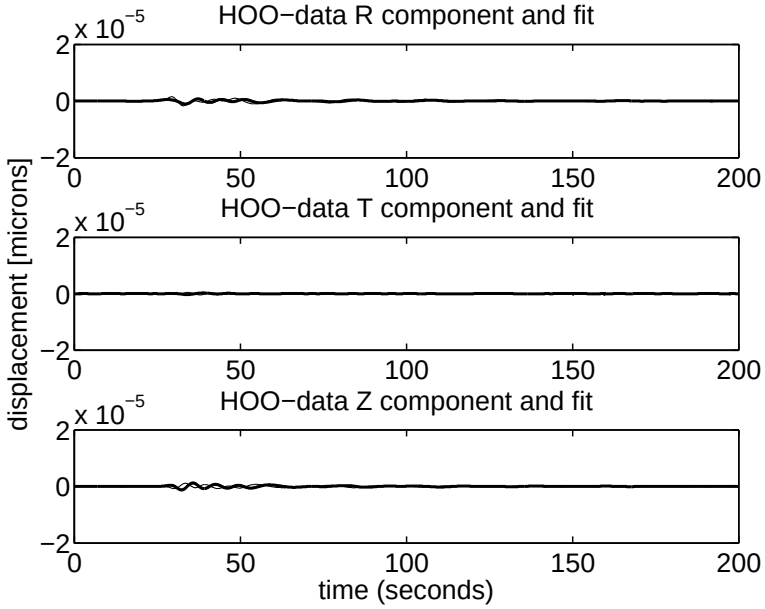

Figure 4.13: Stacked 2005 Data 3m3f Model HOO Data Fit

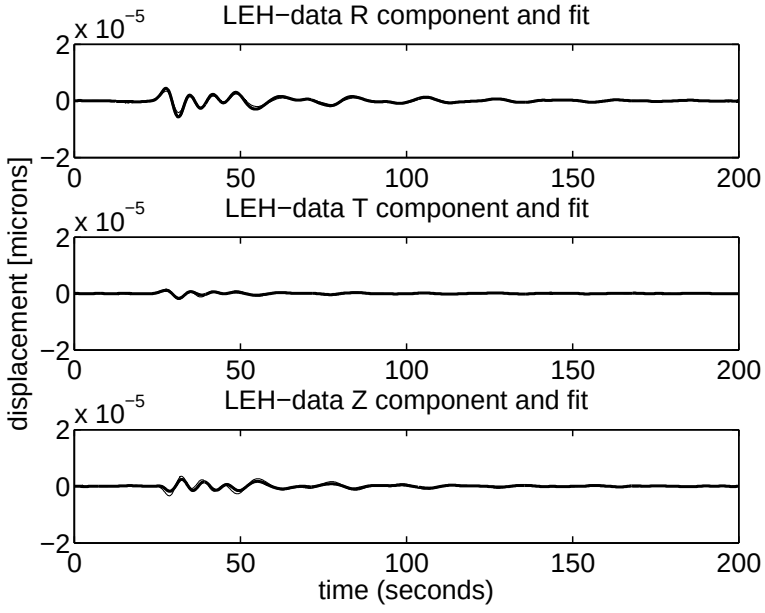

Figure 4.14: Stacked 2005 Data 3m3f Model LEH Data Fit

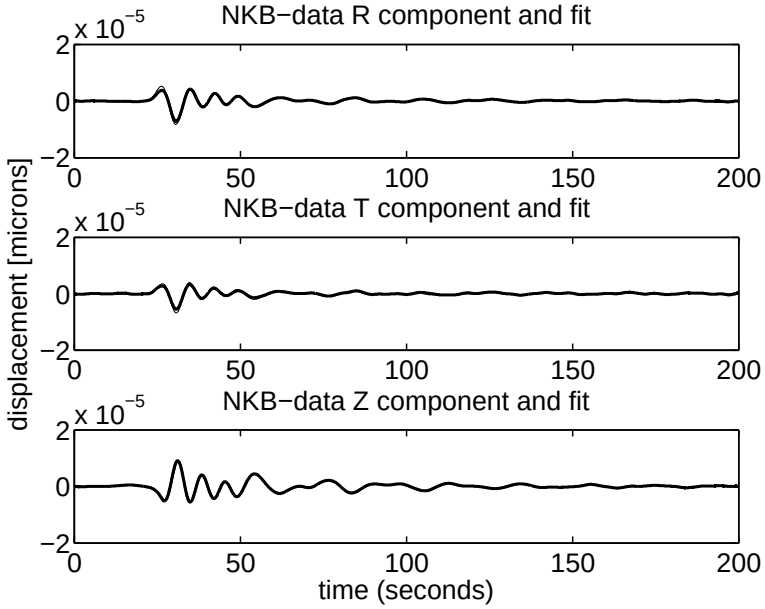

Figure 4.15: Stacked 2005 Data 3m3f Model NKB Data Fit

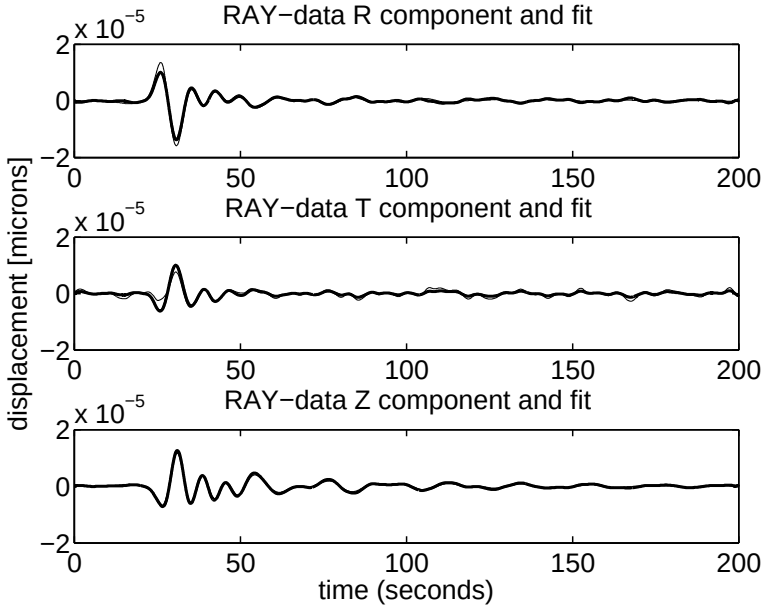

Figure 4.16: Stacked 2005 Data 3m3f Model RAY Data Fit

4.3 Comparison of Methods

The method of parameterization by discretization using triangle basis functions was implemented by Aster [3]. The best fit to the data was with a dilitational model with three directional single forces which obtained a variance reduction of 90.23%.

Below are plots of the recovered model for the method parameterization by discretization using triangle basis functions.

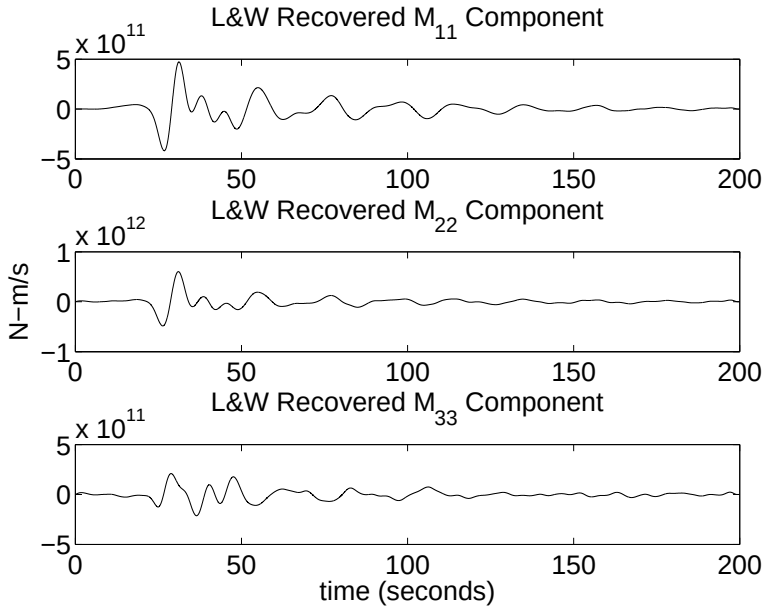

Figure 4.17: Recovered Dilitational Components For Parameterization Method

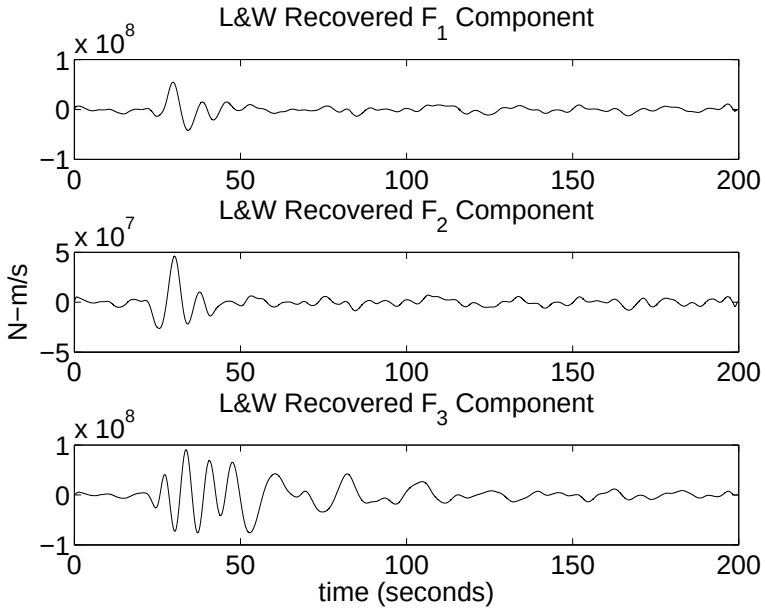

Figure 4.18: Recovered Single Force Components For Parameterization Method

The results show that both methods do obtain the same general models, but CGLS with Tikhonov regularization still contains some high frequency components. This shows that both methods give agreeable results. For a direct comparison, the recovered model from both methods are overlayed below:

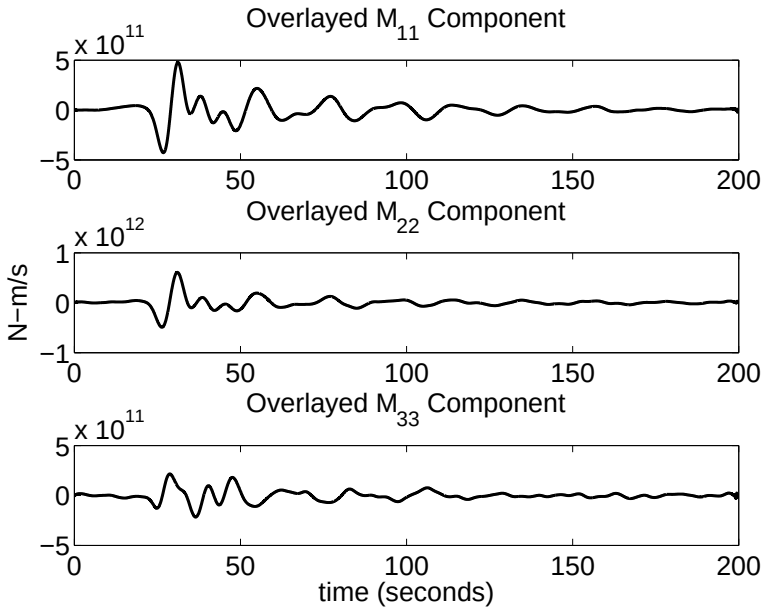

Figure 4.19: Comparison Recovered Dilitational Components For Parameterization Method

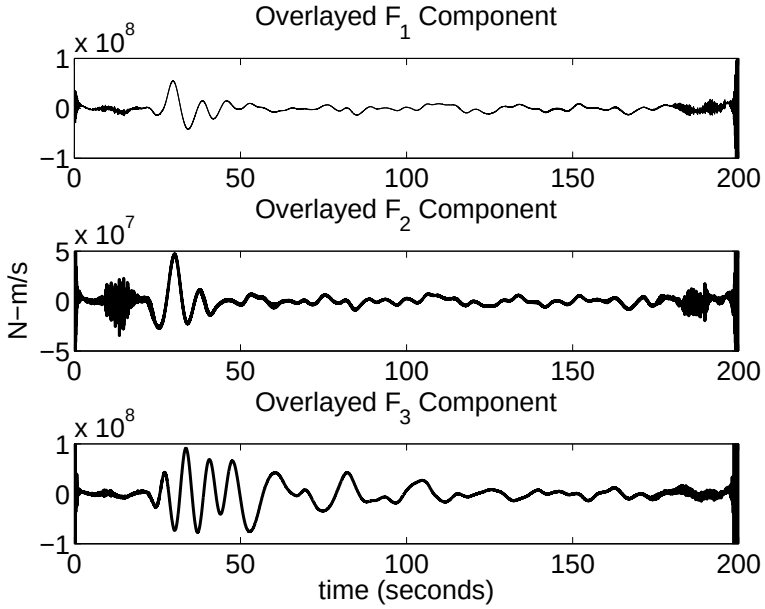

Figure 4.20: Comparison of Recovered Single Force Components For Parameterization Method

CHAPTER 5

CONCLUSIONS

This thesis presents the idea of using fast Toeplitz multiplication as a technique for inverse problems involving sums of convolutions. This technique is then applied to the study of the strombolian eruption events observed at Mount Erebus, Antarctica. The main conclusions can be summarized as follows:

Fast Toeplitz multiplication can be adapted for use in the conjugate gradient least squares algorithm and other iterative inverse techniques. The ability to store large matrices implicitly as vectors and use low memory matrix-vector multiplication techniques makes the handling of large data sets possible whereas explicit storage would be impossible without the use of supercomputers. Fast Toeplitz multiplication also reduces computational effort by carrying out multiplication by means of the Fast Fourier Transform in $O(n \ln(n))$ time versus the traditional $O(n^2)$ time for typical matrix vector multiplications. This means that work can be done on full length data sets without the need to down sample.

The CGLS method which works to find the least squares solution is easily adapted to incorporate increased levels of regularization. By itself, CGLS implicitly regularizes the solution by means of early termination of the algorithm when an appropriate balance of the model misfit $\|\mathbf{Gm} - \mathbf{d}\|$ and the model norm $\|\mathbf{m}\|$ has been achieved. This implicit regularization was not

sufficient enough in practice on the Mount Erebus problem sets as it failed to eliminate enough of the noise within the data and produce a sufficiently smooth recovered model. Implicit regularization also requires an unpredictable amount of iterations to meet our conditions which causes too much wasted computational effort.

Tikhonov regularization is well suited for use within the CGLS algorithm and has the ability to use different filters which together have combined effects for regularization. Identification of the nature of the noise helps in the construction of these filters and can greatly help to speed up the CGLS algorithm. The filter chosen was a roughening matrix designed from a FIR filter to help remove noise from the oceanic microseism. CGLS with Tikhonov regularization requires finding an optimal regularization parameter α which we selected by means of the L-curve criterion. This meant multiple runs of CGLS for various α values were needed. This lead to considerable computational effort, including wasted effort, since there is no prior knowledge about the choice of values of α to use. This process is easily parallelized by running the algorithm upon multiple processors for different values of α.

To test the effectiveness of CGLS with Tikhonov regularization, synthetic examples similar to the system which describes the Mount Erebus problem are needed. Synthetic moment tensor models were used to then compute data by using the Green's functions from the Mount Erebus problem. Noise was then added by combining a waveform describing the oceanic microseism with Gaussian white noise. CGLS with Tikhonov regularization was used to recover the model. The resulting fit to the noisy data is determined by the vari-

ance reduction to be 99.85% for a Mogi model. It was noted that the channels which describe seismic stations farther from the source are most susceptible to interference from noise due to small displacement amplitudes.

The CGLS algorithm with Tikhonov regularization was run on the Mount Erebus data for four different possible model specifications which describe the nature of the source. The model which assumes a dilitational source with three single force directional forces had the best fit to the data with a variance reduction of 90.24%. This result compares well with the parameterization by discretization with triangle basis functions implemented by Aster [3], who obtained a variance reduction of 90.23% for the same model configuration.

The CGLS algorithm was not allowed to run until convergence, which may mean that more of the high frequency components will be removed over time. Also, slight modifications to the filtering matrix may help to filter these components more easily and thus speed up convergence.

The two methods give very similar results. The parameterization method is quicker and should be thought of to be generally superior for quick calculations on smaller problems that have either shorter data sets or fewer triangle basis functions. The parameterization method is not ideal to use on larger data sets or when using a greater number of triangles as the storage requirements can be tremendous. The largest model for comparison in this thesis used about 600 Megabytes to implement this method. Another drawback is that when using this parameterization it is not possible to invert for any high frequency components which may be of interest. It works well in this example, as seismologists are generally interested in the lower frequency elements of this

particular seismic source.

The CGLS with Tikhonov regularization is a more all-purpose moment tensor inversion scheme which has the potential to obtain any frequency of interest within a signal. The problem is the computational effort when finding an appropriate regularization parameter. The memory requirements are feasible on current hardware and for the largest model in this thesis only used 200 Megabytes. Implicit storage of the matrices therefore makes a significant difference in the ability to work with much larger data sets.

In comparison, Aster's implementation of the Lay and Wallace method uses much more storage. If we consider scaling the problem to 100 seismic stations, then this requires approximately 5 times the current storage or 2.93 Gigabytes of storage for the parameterization method and 0.98 Gigabytes for CGLS. On current notebook machines, 1 Gigabyte of memory is typical, and up to 4 Gigabytes is feasible. Every parameter scales accordingly in both methods except the number of triangles. Increasing the number of triangles will help to improve recovery of higher frequency components in the model at the additional cost of storage, however the CGLS method will recover any range of the frequency at no additional storage cost. If attempting to recover higher model frequencies for largest model, the parameterization method would run out memory, while the CGLS method would require the only original 200 Megabytes.

Bibliography

[1] R. Aster, S.Y. Mah, P. Kyle, W. McIntosh, N. Dunbar, J. Johnson, M. Ruiz, and S. McNamara. Very long period oscillations of Mount Erebus volcano. *Journal of Geophysical Research*, 108 (B11,2522), 2003.

[2] Richard Aster, Brian Borchers, and Clifford Thurber. *Parameter Estimation and Inverse Problems*. Elsevier Academic Press, New York, 2005.

[3] Richard Aster, S.Y. Mah, and S. McNamara. Variable and stable broadband seismic characteristics of mount Erebus strombolian eruptions. In Preparation, 2006.

[4] Ronald N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, New York, third edition, 2000.

[5] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, third edition, 1996.

[6] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, Philadelpha, 1998.

[7] Lane R. Johnson. Green's function for Lamb's problem. *Geophysical Journal of the Royal Astronomical Society*, 37 (99-131), 1974.

[8] Thorne Lay and Terry C. Wallace. *Modern Global Seismology*. Academic Press, New York, 1995.

[9] Sara McNamara. Source Mechanism Inversion of Very Long Period Signals Associated with Strombolian Eruptions at Mount Erebus, Antartica. Master's thesis, New Mexico Institute of Mining and Technology, Socorro, New Mexico, August 2004.

[10] K. Mogi. Relations of the eruptions of various volcanoes and the deformations of the ground surfaces around them. *Bulletin of the Earthquake Research Institute, University of Tokyo*, 36 (99-134), 1958.

[11] Seth Stein and Michael Wysession. *An Introduction To Seismology, Earthquakes, and Earth Structure*. Blackwell Publishing, Malden, Massachusetts, 2003.

[12] Gilbert Strang. *Linear Algebra And Its Applications*. Harcourt Brace Jovanovich, Inc., New York, third edition, 1988.