IMAGING FROM SPARSE MEASUREMENTS

By

Yi Fang

A Thesis Submitted to the Graduate Faculty of Rensselaer Polytechnic Institute in Partial Fulfillment of the Requirements for the Degree of DOCTOR OF PHILOSOPHY

Major Subject: Mathematics

Approved by the Examining Committee:

Margaret Cheney, Thesis Adviser

Steven Roecker, Member

Joyce McLaughlin, Member

Randolph Franklin, Member

Rensselaer Polytechnic Institute Troy, New York

December 2008 (For Graduation December 2008)
IMAGING FROM SPARSE MEASUREMENTS

By

Yi Fang

An Abstract of a Thesis Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute
in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY
Major Subject: Mathematics

The original of the complete thesis is on file
in the Rensselaer Polytechnic Institute Library

Examining Committee:

Margaret Cheney, Thesis Adviser
Steven Roecker, Member
Joyce McLaughlin, Member
Randolph Franklin, Member

Rensselaer Polytechnic Institute
Troy, New York

December 2008
(For Graduation December 2008)
© Copyright 2008
by
Yi Fang
All Rights Reserved

ii
CONTENTS

LIST OF TABLES ......................................................... iv
LIST OF FIGURES ....................................................... v
ACKNOWLEDGMENT ....................................................... xiii
ABSTRACT ................................................................. xiv
1. Introduction .......................................................... 1
2. Forward Model ......................................................... 4
   2.1 Helmholtz Equation and Perturbation Model ................. 4
   2.2 Inverse Problem .................................................. 5
   2.3 The Born Approximation ....................................... 6
   2.4 Forward Data Model ............................................. 6
3. Backprojection Reconstruction and Weighting Problem .......... 8
   3.1 Inversion Formula ............................................... 8
   3.2 Analysis of the Image .......................................... 8
      3.2.1 The Point-Spread Function ............................... 10
   3.3 The Weighting Problem ........................................ 13
   3.4 Constant Weight Function Versus Non-constant Weight Function ... 15
   3.5 Discussion of Some Existing Methods ........................ 17
4. Determination of Optimization Weights ............................ 19
   4.1 Least Squares Minimization ................................... 19
   4.2 Solution by Singular Value Decomposition .................. 22
   4.3 Settings and Assumptions of Numerical Simulation ........ 26
   4.4 Reconstruction Experiments .................................. 30
      4.4.1 Geometry and Simulation Data ............................ 31
      4.4.2 Reconstruction with Constant Weights ................. 36
      4.4.3 Reconstruction with $\lambda_0 = 1$ ....................... 41
      4.4.4 Reconstruction with $\lambda_0 = 500$ .................... 43
      4.4.5 Reconstruction with $\lambda_0 = 10,000$ .................. 49
      4.4.6 Reconstruction with $\lambda_0 = 120,000$ ............... 53
LIST OF TABLES

4.1 Locations of sources and receivers as shown in Fig. 4.2. The units are km. .................................................. 31

4.2 Optimization weights at $y_0$ with $\lambda_0 = 1$. .................................................. 40

4.3 Optimization weights at $y_0$ with $\lambda_0 = 500$. .................................................. 45

4.4 Optimization weights at $y_0$ with $\lambda_0 = 10,000$. ........................................ 50

4.5 Optimization weights at $y_0$ with $\lambda_0 = 120,000$. ........................................ 54

4.6 Optimization weights at $y_0$ with $\lambda_0 = 200,000$. ........................................ 56

4.7 List of parameters. All reconstructions were done with 16 sources and 16 receivers, and the wave speed $c$ was 5km/s. .................................................. 61

C.1 List of the singular values $s_1$ to $s_{116}$ of matrix $A$ at $y_0$. .................. 91

C.2 List of the singular values $s_{117}$ to $s_{256}$ of matrix $A$ at $y_0$. .................. 92
LIST OF FIGURES

2.1 Examples of single scattering and multiple scattering. .......................... 6

3.1 The bisector of one source-receiver pair in a homogeneous medium at
imaging point $x$. .................................................................................. 11

3.2 An example of plot of all the bisectors at one imaging point in 2-D
case. The two axes correspond to the two entries of vector $\nabla_x \tau(x_s, x) + \nabla_x \tau(x, x_r)$. The units are $1/m$. ................................. 12

3.3 Bisecting line segments for an imaging point in 2-D case. The horizontal
axis is $k_1$ and the vertical axis is $k_2$. .................................................. 14

3.4 Point-spread function with constant weight function. The horizontal
axis is $z_1$ and the vertical axis is $z_2$. $z = x - y$ in (3.17). The units of
point-spread function values are $1/m^2$. .................................................. 15

3.5 Point-spread function with non-constant weight function. The horizon-
tal axis is $z_1$ and the vertical axis is $z_2$. $z = x - y$ in (3.17). The units
of point-spread function values are $1/m^2$. .................................................. 16

4.1 An example of mesh in a 2-D $z$-computational domain. The number in
each pixel represents the index of each pixel. The first pixel (the black
pixel labeled as 1) is the center of the $z$-computational domain with
$\bar{z}_1 = 0$. ...................................................................................... 21

4.2 Geometry of sources (red stars) and receivers (blue circles). The hori-
zontal axis is $y_1$ and the vertical axis is $y_3$. The yellow square is the
computational domain for reconstruction experiments. ............................. 30

4.3 Actual image with a 10 meter by 10 meter point-like scatterer at the
center. The horizontal axis is $y_1$ and the vertical axis is $y_3$. The units
on each of the two axes is $km$. ............................................................... 32

4.4 Time-domain scattered wave calculated at receiver $x_{r_1}$ from source $x_{s_5}$. 34

4.5 The bisecting line segments for the scatterer $y_0$. Each line segment is
colored according to its weight, which is given by the grey scale to the
right. In this case, all the weights are equal to 1. The horizontal axis is
$k_1$ and the vertical axis is $k_2$. The units on each axis are $rad/km$. .... 36

4.6 Point-spread function at scatterer $y_0$ using constant weights. The hori-
zontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are
$km$. The units of the point-spread function values are $1/km^2$. .......... 37
4.7 Reconstruction without noise using constant weights. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km. 38

4.8 Reconstruction with 10% noise in scattered waves using constant weights. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km. 38

4.9 Reconstruction with up-to-50-meter positioning error in the locations of sources and receivers using constant weights. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km. 39

4.10 The bisecting line segments at the scatterer $y_0$, colored according to the optimization weights with $\lambda_0 = 1$. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are rad/km. Note that many of the bisectors have small weights and cannot be seen with this color scale. Moreover, some weights have large positive values and some have large negative values. 41

4.11 Point-spread function at the scatterer $y_0$ with optimization weights for $\lambda_0 = 1$. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are km. The units of the point-spread function values are $1/km^2$. 42

4.12 Reconstruction without noise using optimization weights for $\lambda_0 = 1$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km. The gray scale has been chopped to show the behavior near the origin; the large values in regions farther from the origin are shown as 1 or $-1$. 43

4.13 Reconstruction with 10% noise in scattered waves using optimization weights for $\lambda_0 = 1$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km. 44

4.14 Reconstruction with up-to-50-meter positioning error in the locations of sources and receivers using optimization weights for $\lambda_0 = 1$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km. 44

4.15 The bisecting line segments at the scatterer $y_0$, colored according to the optimization weights with $\lambda_0 = 500$. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are rad/km. Gray-scale corresponds to the weight values on bisecting line segments. 46

4.16 Point spread function at the scatterer $y_0$ using optimization weights for $\lambda_0 = 500$. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are km. The units of the point-spread function values are $1/km^2$. 47
4.17 Reconstruction without noise using optimization weights for $\lambda_0 = 500$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are $km$.

4.18 Reconstruction with 10% noise in scattered waves using optimization weights for $\lambda_0 = 500$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are $km$.

4.19 Reconstruction with up-to-50-meter positioning error in the locations of sources and receivers using optimization weights for $\lambda_0 = 500$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are $km$.

4.20 The bisecting line segments at the scatterer $y_0$, colored according to the optimization weights with $\lambda_0 = 10,000$. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$. Note that most bisecting line segments have positive weights.

4.21 Point spread function at the scatterer $y_0$ using optimization weights for $\lambda_0 = 10,000$. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The units of the point-spread function values are $1/km^2$.

4.22 Reconstruction without noise using optimization weights for $\lambda_0 = 10,000$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are $km$.

4.23 Reconstruction with 10% noise in scattered waves using optimization weights for $\lambda_0 = 10,000$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are $km$.

4.24 Reconstruction with up-to-50-meter positioning error in the locations of sources and receivers using optimization weights for $\lambda_0 = 10,000$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are $km$.

4.25 The bisecting line segments at the scatterer $y_0$, colored according to the optimization weights with $\lambda_0 = 120,000$. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$. Note weights are almost constant.

4.26 Point spread function at the scatterer $y_0$ using optimization weights for $\lambda_0 = 120,000$. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The units of the point-spread function values are $1/km^2$. 
4.27 Reconstruction without noise using optimization weights for \( \lambda_0 = 120,000 \). The horizontal axis is \( x_1 \) and the vertical axis is \( x_3 \). The units on each axis are \( km \).

4.28 Reconstruction with 10% noise in scattered waves using optimization weights for \( \lambda_0 = 120,000 \). The horizontal axis is \( x_1 \) and the vertical axis is \( x_3 \). The units on each axis are \( km \).

4.29 Reconstruction with up-to-50-meter positioning error in the locations of sources and receivers using optimization weights for \( \lambda_0 = 120,000 \). The horizontal axis is \( x_1 \) and the vertical axis is \( x_3 \). The units on each axis are \( km \).

4.30 The bisecting line segments at the scatterer \( y_0 \), colored according to the optimization weights with \( \lambda_0 = 200,000 \). The horizontal axis is \( k_1 \) and the vertical axis is \( k_3 \). The units on each axis are \( rad/km \). Note that weights are almost constant.

4.31 Point spread function at the scatterer \( y_0 \) using optimization weights for \( \lambda_0 = 200,000 \). The horizontal axis is \( z_1 \) and the vertical axis is \( z_3 \). The units on each axis are \( km \). The units of the point-spread function values are \( 1/km^2 \).

4.32 Singular values of matrix \( A \) at \( y_0 \). The five lines correspond to the five regularization parameters \( \lambda_0 = 1, \lambda_0 = 500, \lambda_0 = 10,000, \lambda_0 = 120,000 \) and \( \lambda_0 = 200,000 \). The \( \lambda_0 = 1 \) line and the \( \lambda_0 = 500 \) line are very close to the horizontal axis.

4.33 Logarithm of the singular values of matrix \( A \) at \( y_0 \). The five lines correspond to the logarithm of the five regularization parameters \( \lambda_0 = 1, \lambda_0 = 500, \lambda_0 = 10,000, \lambda_0 = 120,000 \) and \( \lambda_0 = 200,000 \).

4.34 Sources (red stars) and receivers (blue circles) locations of geometry 1. The horizontal axis is \( y_1 \) and the vertical axis is \( y_3 \). Green pentagrams and squares represent the chosen imaging points, where optimization weights and point-spread functions are calculated. Bisecting line segments plots are shown only at green pentagram locations.

4.35 Plots of the point-spread functions of geometry 1 using constant weights. The horizontal axis is \( z_1 \) and the vertical axis is \( z_3 \). The units on each axis are \( km \). The units of the point-spread function values are \( 1/km^2 \). The locations correspond to the \( 7 \times 5 \) different imaging points designated by green pentagrams and squares in Fig. 4.34.
4.36 Plots of point-spread functions of geometry 1 using optimization weights. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The units of the point-spread function values are $1/km^2$. The locations are the same as those in Fig. 4.35.

4.37 Half-maximum-value contours of the point-spread functions calculated with optimization weights (the contours filled with black) and constant weights (the outer contours) of geometry 1. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The $7 \times 5$ different imaging points correspond to both green pentagrams and squares in Fig. 4.34. The number shown outside the parenthesis is half-maximum-value area with optimization weights. The number shown inside the parenthesis is half-maximum-value area with constant weights.

4.38 Bisecting line segments plots of $4 \times 3$ different imaging points corresponding to the green pentagrams in Fig. 4.34 of geometry 1. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$.

4.39 Bisecting line segments plots of $4 \times 3$ different imaging points corresponding to the green pentagrams in Fig. 4.34 of geometry 1. Each bisecting line segment is colored according to the optimization weights. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$.

4.40 Sources and receivers locations of geometry 2. Sources and receivers have exact same locations at blue circles. The horizontal axis is $y_1$ and the vertical axis is $y_3$. Green pentagrams and squares represent the chosen imaging points, where optimization weights and point-spread functions are calculated. Bisecting line segments plots are shown only at green pentagram locations.

4.41 Plots of point-spread functions of geometry 2 using constant weights. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The units of the point-spread function values are $1/km^2$. Again the locations correspond to the $7 \times 5$ points marked by green pentagrams and squares in Fig. 4.40.

4.42 Plots of point-spread functions of geometry 2 using optimization weights. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The units of the point-spread function values are $1/km^2$. The locations are the same as those in Fig. 4.41.
4.43 Half-maximum-value contours of point-spread functions calculated with optimization weights (the contours filled with black) and constant weights (the outer contours) of geometry 2. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The $7 \times 5$ different imaging points correspond to both green pentagrams and squares in Fig. 4.40. The number shown outside the parenthesis is half-maximum-value area with optimization weights. The number shown inside the parenthesis is half-maximum-value area with constant weights.

4.44 Bisecting line segments plots of $4 \times 3$ different imaging points corresponding to the green pentagrams in Fig. 4.40 of geometry 2. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$.

4.45 Bisecting line segments plots of $4 \times 3$ different imaging points corresponding to the green pentagrams in Fig. 4.40 of geometry 2. Each bisecting line segment is colored according to the optimization weights. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$.

4.46 Sources (red stars) and receivers (blue circles) locations of geometry 3. The horizontal axis is $y_1$ and the vertical axis is $y_3$. Green pentagrams and squares represent the chosen imaging points, where optimization weights and point-spread functions are calculated. Bisecting line segments plots are shown only at green pentagram locations.

4.47 Plots of point-spread functions of geometry 3 using constant weights. The units on each axis are $km$. The units of the point-spread function values are $1/km^2$. Again the locations correspond to the $7 \times 5$ imaging points designated by both green pentagrams and squares in Fig. 4.46.

4.48 Plots of point-spread functions of geometry 3 using optimization weights. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The units of the point-spread function values are $1/km^2$. The locations are the same as those in Fig. 4.47.

4.49 Half-maximum-value contours of point-spread functions calculated with optimization weights (the contours filled with black) and constant weights (the outer contours) of geometry 3. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The $7 \times 5$ different imaging points correspond to both green pentagrams and squares in Fig. 4.46. The number shown outside the parenthesis is the half-maximum-value area with optimization weights. The number shown inside the parenthesis is half-maximum-value area with constant weights.
4.50 Bisecting line segments plots of $4 \times 3$ different imaging points corresponding to the green pentagrams in Fig. 4.46 of geometry 3. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$.

4.51 Bisecting line segments plots of $4 \times 3$ different imaging points corresponding to the green pentagrams in Fig. 4.46 of geometry 3. Each bisecting line segment is colored according to the optimization weights. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$. 

77
ACKNOWLEDGMENT

This work was supported by the NSF under contract number DMS-0327634 and by the Air Force Office of Scientific Research\(^1\) under agreement number FA9550-06-1-0017. Some computing resources were provided by the DARPA/Geo* project at RPI. Other computing resources were provided by IPRPI center.

I am very grateful to my adviser, Margaret Cheney, for her guidance, encouragement and support throughout my study in RPI over the past six years. Thanks to Steven Roecker for all of his helpful advice and comments that have contributed to my research accomplishments. I would like to acknowledge Joyce Mclaughlin, Daniel Renzi, David Isaacson, Randolph Franklin, Kui Lin, Jing Hu, Ashley Thomas, Fengyan Li, Leslie Greengard, Il-Young Son, Trond Varslot, Ling Wang, Ning Zhang, Polina Zheglova, Hector Morales, Matthew Ferrara, Sava Dediu, and Jeong-Rock Yoon for their useful discussions and suggestions.

\(^1\)Consequently the U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.
ABSTRACT

In this dissertation we consider the inverse problem for the scalar wave equation with sparse and non-equally spaced sources and receivers. We develop a method to weight different parts of the data differently to compensate for nonuniform sampling.

We use the single-scattering (Born) approximation and an inversion formula based on a filtered version of the adjoint operator of the forward model. We study the point-spread function to determine the resolution of the reconstruction. For sparsely positioned sources and receivers, the point-spread function can be approximated by a weighted sum of oscillatory functions. A regularized least-squares method can be formulated to determine weights that make the point-spread function as close as possible to the Dirac delta function. Once the weights are determined, the same set of weights can be applied to form an image from measured data.

We test our minimization scheme with different regularization parameters. The sensitivity of the reconstruction with respect to noise and positioning error is tested. We can choose regularization parameters properly to improve resolution and gain stability at the same time.

This method also applies to the regular grid problem. We show examples of point-spread functions constructed with weights corresponding to three different types of source-receiver geometry with different frequency bands. These results not only show that using the right weights improves the resolution relative to reconstructions with constant weights, but also illustrate the relation between resolution and the source-receiver geometry and bandwidth.
CHAPTER 1

Introduction

Wave propagation arises in many fields, such as electromagnetics, acoustics and seismology, and has been studied since 17th century. For example, Daniel Bernoulli investigated the problem of a vibrating string. With certain assumptions and approximations, the wave propagation is governed by the \textit{scalar wave equation} or \textit{acoustic wave equation}. The wave equation has been thoroughly studied, both analytically and numerically. Numerous methods have been developed to calculate the wave field in space and time, given the wave source and wave speed of the medium. The problem of finding the wave field from knowledge of the medium and the wave source is called the \textit{forward problem}. We note that although the dependence of the wave field on the source is linear, the dependence on the medium is nonlinear.

In medical ultrasound imaging, radar imaging, and seismology, we are interested instead in the \textit{inverse problem}, in which we want to recover the wave speed of the medium from the recorded scattered waves, assuming the source configuration is known. For example, people are interested in imaging the geological fault in the Parkfield area using microquake data. They hope to form high resolution images using seismic data recorded with geophones which are installed on the surface of earth. In this inverse problem, the source locations and waveforms need to be approximated first.

The inverse problem is nonlinear. But the relationship between the medium and the scattered field can be approximated by a linear mapping called the \textit{Born approximation} or \textit{single-scattering} approximation. It was shown by Beylkin \cite{Beylkin1984, Beylkin1993} that this linear operator is a \textit{Fourier integral operator} (FIO) \cite{Hormander1983, Hormander1988}. He shows that the transform defined by the forward operator is actually a \textit{generalized Radon transform} \cite{Beylkin1993}, which is a type of FIO. An asymptotic inversion of the generalized Radon transform can be derived using FIO theory \cite{Hormander1983, Hormander1988}. Just as for the classical \textit{Radon transform} \cite{Kak1988}, the inverse operator that maps scattered wave to wave speed is a filtered adjoint operator of the forward operator. This method has the physical
interpretation of \textit{backprojection}.

Many inverse problems ultimately reduce to a problem in inverting the Fourier transform from data known at a certain set of points. In most such problems, the Fourier data is not uniformly sampled on a cartesian grid. For example, in radar imaging [14], data is usually collected on a polar coordinate grid. The data recorded in \textit{magnetic resonance imaging} (MRI) [22] has a spiral sampling pattern. To handle data on a non-cartesian grid, one can either interpolate to a cartesian grid, or attempt to handle the Fourier inversion directly. In the latter approach, ultimately a weight function needs to be chosen to compensate for the nonuniform sampling. In the literature of MRI, this weight function is also referred to as a \textit{sampling density compensation function}.

To handle Fourier transforms on irregular grids, in the last few decades there has been a lot of research focusing on the \textit{Nonuniform fast fourier transform} (NUFFT). The NUFFT arises not only in many inverse problems, but also in signal processing and image processing. For example, [5, 6, 7, 8, 9] have presented methods using \textit{interpolation} and \textit{gridding} techniques that can achieve fast calculation of NUFFT with high accuracy. Other researchers [21, 11, 13] have studied the sampling density weight function in gridding method in MRI. Bydder et al [3] have evaluated different density weight functions for the gridding method in MRI. Samsonov et al [2] have presented a different method to calculate the sampling density weight function by optimizing the \textit{point-spread function} based on the gridding method.

Interpolation and gridding techniques have been proven to work well in MRI when the data is sampled on a dense set. However, in many seismic imaging and radar imaging problems, sources and receivers are very sparsely located on an irregular grid. For example, in the Parkfield area, people use sparsely located geophones to measure the microquake data because the locations of the geophones are restricted by geographical complexities such as hard rocks, lakes and swamps, etc. In the case when the data is sparsely sampled, interpolation and gridding method may break down.

For radar and seismic imaging problems, we cannot apply NUFFT directly because the phase function of the oscillatory term in the inverse operator is not a
linear function of the integral variables. In these cases, there is an additional step of parameterizing the source-receiver geometry. We can avoid this additional step by seeking a weight function that not only compensates for the nonuniform sampling but also accounts for the irregularity of the source-receiver geometry. We don’t use the interpolation or gridding method, however, because it requires parameterizing the source-receiver geometry which we want to avoid.

In this paper, we address situations in which sources and receivers are sparsely located on an irregular grid. Our problem is to find a weight function that provides a suitable trade-off between resolution and stability in the reconstruction process. In Chapter 2, we derive the mathematical model for the forward problem. In chapter 3, an inverse formula is derived using the results from the theory of FIO. In chapter 4, we present an optimization method based on the property of generalized functions [19]. Here we address the trade-off between resolution and stability. We show the results of simulations for different weight functions and different geometries. In Chapter 5, we discuss remaining problems and possible future work.
CHAPTER 2
Forward Model

Our strategy for formulating the problem is to start with the Helmholtz equation in which we model scatterers by a perturbation of a smooth background wave speed. We use the Born(single-scattering) approximation.

2.1 Helmholtz Equation and Perturbation Model

The propagation of waves that emanate from a point source at time $t = 0$ in a homogeneous medium is governed by the 3-D scalar wave equation

$$\nabla^2_x \hat{u}(\mathbf{x}, \mathbf{x}_s, t) - \frac{1}{v^2(\mathbf{x})} \frac{\partial^2}{\partial t^2} \hat{u}(\mathbf{x}, \mathbf{x}_s, t) = -\delta(\mathbf{x} - \mathbf{x}_s)\delta(t),$$

(2.1)

where $\hat{u}$ represents the displacement in time domain, $\mathbf{x} \in \mathbb{R}^3$ is the position in space, $\mathbf{x}_s \in \mathbb{R}^3$ is the source position, $v(\mathbf{x})$ is the wave speed which depends on position, and $\delta(\mathbf{x})$ and $\delta(t)$ are 3-D and 1-D Dirac delta function respectively. We take the inverse Fourier transform on both sides of wave equation to obtain the Helmholtz equation

$$\nabla^2_x u(\mathbf{x}, \mathbf{x}_s, \omega) + \frac{\omega^2}{v^2(\mathbf{x})} u(\mathbf{x}, \mathbf{x}_s, \omega) = -\delta(\mathbf{x} - \mathbf{x}_s),$$

(2.2)

where $u$ is the displacement in frequency domain, $\omega$ is the angular frequency.

We consider the case in which the wave speed is the sum of two terms, a smooth background speed and a more singular term that models reflecting surfaces between different media. For the seismic case, the different media correspond to different rock types; for the radar case, the different media correspond to air and other objects such as vehicles and buildings.

The smooth background wave speed $c(\mathbf{x})$ can be simply chosen to be the speed of light in radar problems. For seismic problems, the background wave speed can be approximated by several approaches [28, 29, 30]. We write

$$\frac{1}{v^2(\mathbf{x})} = \frac{1}{c^2(\mathbf{x})}(1 + \alpha(\mathbf{x})).$$

(2.3)
and we refer to $\alpha(x)$ as the perturbation.

The field $u(x, x_s, \omega)$ from a point-like source can be written as the sum of an incident wave field $G_0(x, x_s, \omega)$ and a scattered wave field $u^s(x, x_s, \omega)$:

$$u(x, x_s, \omega) = G_0(x, x_s, \omega) + u^s(x, x_s, \omega),$$  \hspace{1cm} (2.4)

where $G_0(x, x_s, \omega)$ represents Green’s function and satisfies

$$\nabla^2_x G_0(x, x_s, \omega) + \frac{\omega^2}{c^2(x)} G_0(x, x_s, \omega) = -\delta(x - x_s).$$  \hspace{1cm} (2.5)

We substitute (2.3) and (2.4) into (2.2), and subtract from (2.5) to obtain

$$\nabla^2_x u^s(x, x_s, \omega) + \frac{\omega^2}{c^2(x)} u^s(x, x_s, \omega) = -\frac{\omega^2}{c^2(x)} \alpha(x) \left( G_0(x, x_s, \omega) + u^s(x, x_s, \omega) \right)$$  \hspace{1cm} (2.6)

The scattered wave $u^s(x, x_s, \omega)$ satisfies the Helmholtz equation with the source term $\frac{\omega^2}{c^2(x)} \alpha(x) \left( G_0(x, x_s, \omega) + u^s(x, x_s, \omega) \right)$. We use Green’s method [21] to solve for $u^s(x, x_s, \omega)$. This results in the integral equation for the scattered field $u^s(x, x_s, \omega)$:

$$u^s(x, x_s, \omega) = \omega^2 \int_{\mathbb{R}^3} \frac{\alpha(y)}{c^2(x)} G_0(x, y, \omega) \left( G_0(y, x_s, \omega) + u^s(y, x_s, \omega) \right) dy.$$  \hspace{1cm} (2.7)

### 2.2 Inverse Problem

Definition (Forward Problem): Given the perturbation $\alpha(x)$, find the scattered wave $u^s(x, x_s, \omega)$.

Definition (Inverse Problem): Recover the medium parameter $\alpha(x)$ from the scattered wave $u^s(x_r, x_s, \omega)$ at given frequency $\omega$ and given locations $x_s$’s and $x_r$’s.

In most applications, the scattered wave is recorded in the time domain. To get the scattered wave in the frequency domain, we simply take the inverse fourier transform of the time-domain signal.

Note that the exact locations of the sources may not be known for some ap-
Figure 2.1: Examples of single scattering and multiple scattering.

2.3 The Born Approximation

Equation (2.7) is an integral equation that must be solved, rather than simply a formula for $u^s(x, x_s, \omega)$. For the inverse problem, where we know $c(y)$, $G_0(x, y, \omega)$ and the values of $u^s(x, x_s, \omega)$ at certain receivers, the right side of (2.7) contains a product of two unknown quantities: $u^s(x, x_s, \omega)$ in the scattering region and $\alpha(x)$, which is what we would like to find. To linearize the problem, we use the Born approximation to neglect the $u^s(x, x_s, \omega)$ term on the right side of (2.7). The Born approximation is valid when either the perturbation $\alpha(x)$ is small, or when most of the received energy is due to single scattering. The Born approximation is thus

$$u_B^s(x, x_s, \omega) = \omega^2 \int_{\mathbb{R}^3} \frac{\alpha(y)}{c^2(y)} G_0(x, y, \omega) G_0(y, x_s, \omega) dy. \quad (2.8)$$

2.4 Forward Data Model

In equation (2.8), we are still missing an explicit expression for $G_0(x, y, \omega)$. In high frequency applications, $G_0(x, y, \omega)$ can be represented by the WKBJ series [31]. We approximate this series by retaining only the first order term, which produces
simple expressions

\[ G_0(x, y, \omega) = A(y, x) e^{i\omega \tau(y, x)}, \] (2.9)

where \( \tau(x_s, x) \) is the travel time for the wave propagating from \( x_s \) to \( x \) and can be computed by solving the Eikonal equation (see Appendix A). \( A(x_s, x) \) is the amplitude that satisfies the first order transport equation.

We replace \( G_0(x, x_s, \omega) \) in (2.8) by the above expression and replace \( x \) by \( x_r \) since data are only collected at locations of receivers \( x_r \)’s. We thus obtain a model for the received data:

\[ u_s^\omega(x_r, x_s, \omega) = \omega^2 \int_{\mathbb{R}^3} \frac{a(y)}{c^2(y)} a(x_s, y, x_r) e^{i\omega \phi(x_s, y, x_r)} dy, \] (2.10)

where \( a(x_s, y, x) \) and \( \phi(x_s, y, x) \) are defined by

\[ a(x_s, y, x_r) = A(x_s, y) A(y, x_r), \] (2.11)

\[ \phi(x_s, y, x_r) = \tau(x_s, y) + \tau(y, x_r). \] (2.12)

The recording locations \((\omega, x_s, x_r)\) need to cover a three dimensional domain in order to reconstruct three dimensional objects. We assume that the source-receiver geometry can be parameterized by the two dimensional variable \( \sigma = (\sigma_1, \sigma_2) \). We will use \( \sigma \) to represent \((x_s, x_r)\) in the paper.

In the next chapter, we will give a brief derivation of a formula for inverting of (2.10).
CHAPTER 3
Backprojection Reconstruction and Weighting Problem

In this chapter, we follow the approach of [4, 1] in deriving the inversion formula by considering the filtered adjoint operator of the forward formula. Then we determine the right filter function by analyzing the relationship between the image \( I(x) \) and the actual perturbation \( \alpha(x) \).

3.1 Inversion Formula

We look for an inverse to (2.10) in terms of a filtered adjoint:

\[
I(x) = \int \int b(x, \sigma)u_{D}(\sigma, \omega)e^{-i\omega \phi(x, \sigma)}d\omega d\sigma.
\]

(3.1)

Here \( I(x) \) is called the image of \( \alpha(x) \) and the expression for \( b(x, \sigma) \) is determined below.

This reconstruction algorithm can be interpreted in terms of backprojection, which arises in the inverting of the classical Radon transform. In fact, we can carry out the \( \omega \) integration in the equation above to obtain

\[
I(x) = \int b(x, \sigma)\hat{u}^{s}_{D}(\sigma, \phi(x, \sigma))d\sigma.
\]

(3.2)

This equation says the data from the source-receiver pair \( \sigma \) at time \( t \) is first filtered by \( b(x, \sigma) \), then backprojected to (i.e., spread out over) the surface \( \{x|\phi(x, \sigma) = t\} \). The image \( I(x) \) is then the superposition of all the back projected data from all source-receiver pairs. Because of this physical interpretation, we refer to this method as the backprojection method.

3.2 Analysis of the Image

To determine \( b(x, \sigma) \), we analyze the relationship between the image \( I(x) \) and the actual perturbation \( \alpha(x) \). To do this, we plug (2.10) into (3.1) and change the
order of integration. This results in

\[ I(x) = \int K(x, y) \alpha(y) dy, \tag{3.3} \]

where \( K(x, y) \) is the point-spread function

\[ K(x, y) = \int_{\mathbb{R}^3} \frac{\omega^2}{c^2(y)} a(y, \sigma) b(x, \sigma) e^{-i\omega(\phi(x, \sigma) - \phi(y, \sigma))} d\omega d\sigma. \tag{3.4} \]

To obtain a perfect image, we would like the right hand side of the equation (3.3) to be a convolution of \( \alpha(x) \) with \( \delta(x) \), which we write as

\[ \delta(x - y) = \frac{1}{(2\pi)^3} \int_{\mathbb{R}^3} e^{-ik(x-y)} dk. \tag{3.5} \]

We analyze (3.4) by the method of stationary phase [23, 24]. Differentiating the phase of (3.4) with respect to \( \omega \) and \( \sigma \), we find that the critical conditions are

\[ \phi(x, \sigma) = \phi(y, \sigma) \]
\[ \omega \nabla_\sigma \phi(x, \sigma) = \omega \nabla_\sigma \phi(y, \sigma) \tag{3.6} \]

We assume here that there are solutions to (3.6) only when \( x = y \) (see [1, 4]). In the neighborhood of \( x = y \), we can use the following approximations in (3.4).

\[ b(x, \sigma) \approx b(y, \sigma) \tag{3.7} \]
\[ \omega(\phi(x, \sigma) - \phi(y, \sigma)) \approx \omega \nabla_x \phi(x, \sigma) \cdot (x - y). \tag{3.8} \]

In (3.4) we replace \( b(x, \sigma) \) by \( b(y, \sigma) \) and make the change of variables \( (\omega, \sigma) \mapsto k = \omega \nabla_x \phi(x, \sigma) \) to get

\[ K(x, y) \approx \int_{\mathbb{R}^3} \frac{\omega^2(k)}{c^2(y)} a(y, \sigma(k)) b(y, \sigma(k)) |J(k)| e^{-ik(x-y)} dk, \tag{3.9} \]
where $J(k) = \det(\frac{\partial(\omega, \sigma)}{\partial k})$. Note that

$$\frac{1}{J(k)} = \det\left(\frac{\partial k}{\partial(\omega, \sigma)}\right),$$

$$= \omega^2 \det\left(\begin{bmatrix} \nabla_x \phi(x, \sigma) \\ \frac{\partial}{\partial \sigma_1} \nabla_x \phi(x, \sigma) \\ \frac{\partial}{\partial \sigma_2} \nabla_x \phi(x, \sigma) \end{bmatrix}\right),$$

$$= \omega^2 h(x, \sigma).$$

Comparing (3.9) with (3.5), we see that we should choose $b(y, \sigma(k))$ such that

$$\omega^2(k) \frac{a(y, \sigma(k)) b(y, \sigma(k)) |J(k)|}{c^2(y)} = \frac{1}{(2\pi)^3}. $$

Thus we have an explicit form for $b(y, \sigma)$, namely

$$b(y, \sigma) = \frac{c^2(y) |h(y, \sigma)|}{(2\pi)^3 a(y, \sigma)}. \quad (3.13)$$

Using the above expression for $b(y, \sigma)$ in equation (3.1), the inverse formula to reconstruct $\alpha(x)$ can be derived, which is exactly the formula given by Bleistein et al. [1].

$$I(x) = \int \int \frac{c^2(x) |h(x, \sigma)|}{(2\pi)^3 a(x, \sigma)} u_D^s(\sigma, \omega) e^{-i\omega \nabla_x \phi(x, \sigma)} d\omega d\sigma. \quad (3.14)$$

### 3.2.1 The Point-Spread Function

Because our data are band-limited, and the source-receiver geometry is limited, the integration region of (3.9) is only a bounded domain, rather than all of $\mathbb{R}^3$ space. As a result, $\alpha(x)$ cannot be perfectly recovered. However, the point-spread function $K(x, y)$ quantifies the degree to which our image faithfully represents the true $\alpha(x)$. Using (3.13) in equation (3.9), we get

$$K(x, y) = \frac{1}{(2\pi)^3} \int |h(x, \sigma)| \int \omega^2 e^{-i\omega \nabla_x \phi(x, \sigma)(x-y)} d\omega d\sigma.$$  \quad (3.15)$$

The degree to which $K(x, y)$ approximates a delta function is determined by the source-receiver geometry and the frequency band of the data. In cases that sources and receivers are sparsely located on an irregular grid, we can approximate
the $\sigma$ integral of (3.15) by the following weighted sum:

$$K_W(x, y) \approx \frac{1}{(2\pi\bar{c})^3} \sum_{\sigma} W(x, \sigma) \int \omega^2 e^{-i\omega \nabla_x \phi(x, \sigma) \cdot (x-y)} d\omega$$  \hspace{1cm} (3.16)$$

$$= \frac{1}{(2\pi\bar{c})^3} \sum_{x_s, x_r} W(x, x_s, x_r)$$
$$\times \int \omega^2 e^{-i\omega (\nabla_x \tau(x_s, x) + \nabla_x \tau(x, x_r)) \cdot (x-y)} d\omega,$$  \hspace{1cm} (3.17)$$

where $W(x, \sigma)$ or $W(x, x_s, x_r)$ is the unknown weight function, and $\bar{c}$ is the average of the background velocity. Here we scale the problem by $\frac{1}{c}$ (see more details in Appendix B) to avoid small results.

Note that both the travel times $\tau(x_s, x)$ and $\tau(x, x_r)$ satisfy the Eikonal equation. The vectors $\nabla_x \tau(x_s, x)$ and $\nabla_x \tau(x, x_r)$ have the same length, namely $\frac{1}{c(x)}$, and are tangential to the ray path at the point $x$. The sum of the two vectors is a vector in the direction of the bisector of the angle between the vectors, and its

Figure 3.1: The bisector of one source-receiver pair in a homogeneous medium at imaging point $x$. 
length in general is between 0 and $\frac{2}{c(x)}$. Fig. 3.1 shows the bisector at one imaging point from one source-receiver pair in a homogeneous medium. The plot of all the bisectors at an imaging point characterizes the resolution can be achieved at that point. An example of all the bisectors at one imaging point is shown in (Fig. 3.2). In general, the direction that most bisectors point has the best resolution. The direction that fewer bisectors point has worse resolution. In the band-limited case
where $\omega$ is bounded, we refer to the line segment
\begin{equation}
\{\omega(\nabla_x \tau(x_s, x) + \nabla_x \tau(x, x_r)) : \omega \text{ ranges over the measured angular frequencies}\}
\end{equation}

as the bisecting line segment. The bisecting line segment is actually the bisector scaled by frequency; hence, it points in the same direction as the bisector. We denote by $\Xi_x$ the set of points in the union of all the bisecting line segments. This set $\Xi_x$ determines the resolution of backprojection.

Although resolution is different in different directions, we can characterize the overall resolution at a point $x_0$ by the volume (or area in 2-D) of the enclosed region determined by the half-maximum-value contour of the point-spread function. A point $y$ on the half-maximum-value contour satisfies
\begin{equation}
K(x_0, y) = \frac{1}{2} K(x_0, x_0).
\end{equation}

We refer it as half-maximum-value volume in 3-D and half-maximum-value area in 2-D.

### 3.3 The Weighting Problem

Given any imaging point $x$, we would like to determine $W(x, \sigma)$ or $W(x, x_s, x_r)$ such that $K_W(x, y)$ is as close as possible to the Dirac delta function. Once $W(x, \sigma)$ is known, the weight function can be used in the inversion formula (3.14) to approximate $I(x)$, as we see from the following calculation:

\begin{align}
I_W(x) & \approx \frac{1}{(2\pi c)^3} \sum_\sigma W(x, \sigma) \int \frac{c^2(x)}{a(x, \sigma)} u_D(\sigma, \omega) e^{-i\omega \phi(x, \sigma)} d\omega \\
& = \frac{1}{(2\pi c)^3} \sum_\sigma W(x, \sigma) \int \frac{c^2(x)}{a(x, \sigma)} \omega^2 \\
& \times \int R^3 \frac{\alpha(y)}{c^2(y)} a(y, \sigma) e^{-i\omega (\phi(x, \sigma) - \phi(y, \sigma))} dy d\omega \\
& \approx \int R^3 \frac{1}{(2\pi c)^3} \sum_\sigma W(x, \sigma) \int \omega^2 e^{-i\omega \nabla_x \phi(x, \sigma) \cdot (x - y)} d\omega \times \alpha(y) dy \\
& \approx \int R^3 K_W(x, y) \alpha(y) dy,
\end{align}
Figure 3.3: Bisecting line segments for an imaging point in 2-D case. The horizontal axis is $k_1$ and the vertical axis is $k_2$.

where in the second line we have used (2.10) and in the third we have used (3.7) and (3.8).
Figure 3.4: Point-spread function with constant weight function. The horizontal axis is $z_1$ and the vertical axis is $z_2$. $z = x - y$ in (3.17). The units of point-spread function values are $1/m^2$.

3.4 Constant Weight Function Versus Non-constant Weight Function

The trivial weight function is constant weight function by setting $W(x, x_s, x_r) = 1$. This is equivalent to calculating the average of the backprojected data from all the source-receiver pairs in reconstruction. The disadvantage of constant weight function lies in its weighting each bisector the same. Consider the case which has most bisectors pointing in the horizontal direction at one imaging point, but only one bisector in the vertical direction as shown in Fig. 3.3. In this figure, we show both the positive and negative parts of the frequency band. Fig. 3.4 shows the point-spread function calculated with the constant weight function at that imaging point. The point-spread function consists of vertical ridges, which provide good resolution in the horizontal direction and almost no resolution in the vertical direction. We can imagine that if the constant weight function is used in a backprojection reconstruction, the backprojected data from the source-receiver pair that gives the
vertical bisector is going to be overwhelmed by the backprojected data from the many source-receiver pairs that give horizontal bisectors. However, if we assign small weight values to the horizontal bisectors and a large weight value to the vertical bisector, then we get good resolution in both directions as shown in Fig. 3.5.

There is another way to see the disadvantage of constant weight function in this example. Suppose initially we have only one source-receiver pair that gives a vertical bisector. We begin with good resolution in the vertical direction, but no resolution in the horizontal direction. Then we add in more source-receiver pairs that give horizontal bisectors using constant weight function. We gain resolution in the horizontal direction but completely lose resolution in the vertical direction. In other words, using more data gives worse results! On the other hand, if we add in these source-receiver pairs using a well-chosen non-constant weight function, we can gain resolution in the horizontal direction, while not losing resolution in the vertical direction. Thus choosing a good weight function allows us to use additional
data to improve the reconstruction. This extreme example motivates us to seek non-constant weight functions.

### 3.5 Discussion of Some Existing Methods

Here we discuss some approaches that we considered for finding appropriate weight functions.

- **Backus-Gilbert method** [27]: The Backus-Gilbert method is an inversion approach that based on minimizing the variance of the point-spread function. This method works well in one dimensional problems but fails in higher dimensions, because we are constructing the point-spread function by adding together ridge-like terms, each of which has infinite variance.

- **Interpolation and gridding-based methods**: Interpolation and gridding works well for applications such as MRI in which the sampling is nonuniform but dense. But for very sparse data, these methods are not expected to work well. At the same time, in order to use interpolation or gridding to evaluate the integral such as (3.14) and (3.15), the first step is to parameterize the source-receiver geometry, which is difficult for our geophysical problem.

- **The Voronoi diagram** [12]: For a problem in a bounded domain, the weight function can be derived from a Voronoi diagram. However, the Voronoi diagram weight function can only compensate for the nonuniform sampling and not for the irregularity of the source-receiver geometry, which is characterized by the determinant $h(x, \sigma)$ in (3.14) and (3.15). Moreover, using weights derived from a Voronoi diagram also requires parameterizing the source-receiver geometry first. Finally, our problem involves an unbounded domain, and it is unclear what weights should be assigned to the Voronoi cells on the boundary of the computational domain.

- **Optimization using data as a constraint**: One approach is to express the desired perturbation $\alpha$ in terms of some basis, and minimization of the $L^2$ norm of the $\alpha$ using the data as a constraint. This method can be applied to the sparse
measurement situation, but this approach involves formulating the problem in a different way. One disadvantage of this method lies in its dependence on the data set, which could be very large. Moreover, any change in the data requires the entire minimization procedure be carried out again. Another disadvantage is that the size of the problem is determined by the number of voxels in the mesh of the computational domain. It could be computationally expensive because a global solver is usually needed to solve the forward problem. The third disadvantage is that this approach provides no information about the resolution of the image.

In the next chapter, we will present an optimization method to determine a weight function $W(x, x_s, x_r)$ that achieves better resolution than the constant one. The advantage of this method is that it does not require the parametrization of the source-receiver geometry. Moreover, it does not depend on the measured data itself, but only on the frequency band and the locations of sources and receivers. Once the weight function is determined with this method, the same weight function can be applied to the inversion formula with any data set recorded with the same frequency band using the same source-receiver geometry.
CHAPTER 4
Determination of Optimization Weights

In this chapter, we develop a least-squares minimization method to determine the weight function. Then we test our method by numerical simulation.

4.1 Least Squares Minimization

We propose an optimization approach to find the weight function that is optimal in the sense that our point-spread function best approximates the Dirac delta function. To determine what “best” means, we recall that the delta function has the following property

\[ f(x) = \int \delta(x - y) f(y) \, dy, \quad (4.1) \]

for every smooth test function \( f(x) \).

Thus we convolve a sequence of test functions with our point-spread function. If we can determine the weight function so that each test function can be recovered, the weight function should give a point-spread function close to a delta function. In particular, given any fixed imaging point \( x \), we choose non-overlapping real test functions \( \chi_{jkl}(z) \) to be a smooth approximation of the characteristic function supported on the \((j, k, l)\)th voxel.

\[
\int \frac{1}{(2\pi \bar{c})^3} \sum_{x_s, x_r} W(x_s, x_r) \int \omega^2 e^{-i\omega(\nabla_{x, x_s, x_r} + \nabla_{x, x_s, x_r}) \cdot z} d\omega \chi_{jkl}(z) \, dz = \chi_{jkl}(0), \quad (4.2)
\]

where we have replaced \( x - y \) by \( z \) in the point-spread function \( K_W \) and dropped the \( x \) dependence in \( W \) and \( K_W \) because each calculation is done with \( x \) fixed.

Note that because the sources and receivers are discrete, for each \( x \), the weight function \( W(x, \cdot) \) is a vector whose dimension equals the number of source-receiver pairs. Here we use the term \textit{weights} to refer to the elements of this vector. Note
that the weights are different for different imaging points \( x \).

In (4.2), we can use a 3-D non-uniform fast Fourier transform (NUFFT) [8] to carry out the \( z \) integration, and we sample the result at the non-uniform locations \( \omega (\nabla_x \tau (x_s, x) + \nabla_x \tau (x, x_r)) \), then carry out the \( \omega \) integration. However, if the voxel size \( \Delta z \) is small, in particular if it satisfies \( \Delta z < \min \{ \frac{\pi}{4} c(x) \omega \} \), then we can approximate the integrand in the following way and thus significantly decrease the computational cost:

\[
\int \int \omega^2 e^{-i\omega \nabla_x (\tau(x_s, x) + \tau(x, x_r)) \cdot z} d\omega \chi_{jkl}(z) dz \approx V \int \omega^2 e^{-i\omega \nabla_x (\tau(x_s, x) + \tau(x, x_r)) \cdot \bar{z} \text{ijkl}} d\omega, \tag{4.3}
\]

Here \( \bar{z} \text{ijkl} \) is the center point of \((j, k, l)\)th voxel and \( V \) represents the volume of each voxel. Without losing generality, we assume \( \chi_{jkl}(\bar{z} \text{ijkl}) = 1 \). With (4.3), (4.2) becomes

\[
\frac{1}{(2\pi \bar{c})^3} \sum_{x_s, x_r} W(x_s, x_r) \int \omega^2 e^{-i\omega \nabla_x (\tau(x_s, x) + \tau(x, x_r)) \cdot \bar{z} \text{ijkl}} d\omega = \begin{cases} 1/V & \text{if } \bar{z} \text{ijkl} = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}
\]

We can write equation (4.4) in matrix form as \( Aw = r \). We enumerate the source-receiver pairs as \((x_s^{(n)}, x_r^{(n)})\) for \( n = 1, \ldots, N \), and we enumerate the voxels using the index \( m = 1, \ldots, M \). Here, \( M \geq N \), and \( m \) is an enumeration of the indices \( j, k, l \). Then there are \( M \) test functions \( \chi_m(z) \) for \( m = 1, \ldots, M \). The voxel centers are denoted \( \bar{z}_m \). We assume the first voxel (pixel in 2-D) contains the origin, i.e., \( \bar{z}_1 = 0 \). We use the term “computational domain” to refer to the physical region of interest (denoted by the variable \( y \)) which is the same as the image region (denoted by the variable \( y \)). The point-spread function, however, depends on \( z = x - y \); the computational region in \( z \) we call the “\( z \)-computational domain”. An example of a mesh in a 2-D \( z \)-computational domain is shown in Fig. 4.1. Let \( w \) be the \( N \) dimensional weight vector \( W(x_s^{(n)}, x_r^{(n)}) \), and let \( A \) be the \( M \)-by-\( N \) matrix with \((m, n)\)th entry \( A_{mn} \) defined to be

\[
A_{mn} = \frac{\lambda_m}{(2\pi \bar{c})^3} \int \omega^2 e^{-i\omega \nabla_x (\tau(x_s^{(n)}, x) + \tau(x_s^{(n)}, x_r^{(n)})) \cdot \bar{z} \text{m}} d\omega, \tag{4.5}
\]

where \( \{\lambda_m\}_{m=1}^M \) are introduced as regularization factors. We multiply both sides
Figure 4.1: An example of mesh in a 2-D $z$-computational domain. The number in each pixel represents the index of each pixel. The first pixel (the black pixel labeled as 1) is the center of the $z$-computational domain with $\bar{z}_1 = 0$.

of $m$th equation by a positive constant $\lambda_m$. These regularization parameters don’t change the equations at all. However, they will play a role in the optimization scheme that we present later. In the numerical simulations, we will choose $\lambda_1 \geq 1$ and $\lambda_m = 1$ for $m \geq 2$ to emphasize the importance of the center peak of the point-spread function. Now, $r$ is the $M$-dimensional vector with components

$$r_m = \lambda_m \chi_m(\bar{z}_m) = \begin{cases} \lambda_m / V & \text{if } \bar{z}_m = 0, \\ 0 & \text{otherwise}. \end{cases}$$

(4.6)
Theoretically, we can solve $Aw = r$ to find the weight function $w$. However, matrix $A$ is usually large and ill-conditioned. We need to regularize our problem to get stable solutions. Tikhonov regularization is adopted here to achieve the goal. Using $\ast$ for adjoint, we define the functional $F$

\[
F(w) = \lambda_0^2 ||w||_2^2 + ||Aw - r||_2^2 - ||r||_2^2
\]

\[
= \lambda_0^2 w^* w + (Aw - r)^* (Aw - r) - r^* r,
\]

\[
= \lambda_0^2 w^* w + w^* A^* A w - r^* A w - w^* A^* r,
\]

\[
= w^* (\lambda_0^2 I + A^* A) w - r^* A w - w^* A^* r,
\]

where $|| \cdot ||_2$ represents $L^2$ norm, and $\lambda_0$ is called Tikhonov regularization parameter. Note that $F(w)$ is always real. The optimization scheme we use to compute weight function is to minimize $F(w)$, i.e.,

\[
w = \text{argmin} F(w).
\]

### 4.2 Solution by Singular Value Decomposition

Matrix $\lambda_0^2 I + A^* A$ is self-adjoint and positive semi-definite. It is positive definite when $\lambda_0 \neq 0$, therefore, $F(w)$ has one and only one critical point, which must be the solution to (4.11). This can be seen later in the explanation of equation (4.21). The critical point of $F(w)$ is the point where the first order derivative of $F(w)$ is 0, i.e., to find the solution, we need to solve $\frac{\partial F}{\partial g} = 0$ and $\frac{\partial F}{\partial h} = 0$, where $w = g + i h$, and $g$ and $h$ are real vectors.

First, we expand $F$ with respect to $g$ and $h$, using $\ast$ for adjoint and $'$ for
transpose:

\[
\mathcal{F}(w) = \lambda_0^2 (g + ih)^*(g + ih) + (g + ih)^*A^*A(g + ih) - r^*A(g + ih) \\
-(g + ih)^*A^*r
\]

(4.12)

\[
= \lambda_0^2 (g' - ih')(g + ih) + (g' - ih')A^*A(g + ih) - r^*A(g + ih) \\
-(g' - ih')A^*r
\]

(4.13)

\[
= \lambda_0^2 (g'g + h'h) + g'A^*Ag + h'A^*Ah - ih'A^*Ag + ig'A^*Ah \\
-r^*Ag - ir^*Ah - g'A^*r + ih'A^*r
\]

(4.14)

Then we take the derivative of \( \mathcal{F} \) with respect to \( h \) to get

\[
\frac{\partial \mathcal{F}}{\partial h} = 2\lambda_0^2 h + A^*Ah + (A^*A)'h - iA^*Ag + i(A^*A)'g - iA'r \\
+iA^*r.
\]

(4.15)

Next we multiply \( \frac{\partial \mathcal{F}}{\partial h} \) by \( i \) and calculate \( \frac{\partial \mathcal{F}}{\partial g} \).

\[
i \frac{\partial \mathcal{F}}{\partial h} = 2i\lambda_0^2 h + iA^*Ah - (A^*A)'g + i(A^*A)'h + A^*Ag + A'r \\
-A^*r.
\]

(4.16)

\[
\frac{\partial \mathcal{F}}{\partial g} = 2\lambda_0^2 g + A^*Ag + (A^*A)'g - i(A^*A)'h + iA^*Ah - A'r \\
-A^*r.
\]

(4.17)

Using the two equations above, we obtain

\[
\frac{\partial \mathcal{F}}{\partial g} + i \frac{\partial \mathcal{F}}{\partial h} = 2\lambda_0^2 (g + ih) + 2A^*A(g + ih) - 2A^*r, \quad (4.18)
\]

\[
= 2\lambda_0^2 w + 2A^*Aw - 2A^*r. \quad (4.19)
\]

Since \( \frac{\partial \mathcal{F}}{\partial g} = 0 \) and \( \frac{\partial \mathcal{F}}{\partial h} = 0 \), we have

\[
\frac{1}{2} \left( \frac{\partial \mathcal{F}}{\partial g} + i \frac{\partial \mathcal{F}}{\partial h} \right) = \lambda_0^2 w + A^*Aw - A^*r = 0. \quad (4.20)
\]
Thus, we get the linear equations

$$\lambda_0^2 w + A^* Aw = A^* r. \quad (4.21)$$

When $\lambda_0 \neq 0$, the matrix $\lambda_0^2 I + A^* A$ is positive definite, and in particular it must be full rank [18]. Thus, the linear system (4.21) has a unique solution, implying that $F(w)$ has one and only one critical point which minimizes our objective function. In order to get a stable solution, we solve this linear system by the singular value decomposition (SVD) method [17], from which an explicit formula for the solution can be derived. Let the SVD of $A$ be

$$A = U S V^*, \quad (4.22)$$

where $U$ and $V$ are $M$-by-$M$ and $N$-by-$N$ unitary matrices, $S$ is a $M$-by-$N$ rectangular diagonal matrix. Let $s_n$ on the main diagonal of $S$ denote singular values of $A$. With (4.22), (4.21) becomes

$$\lambda_0^2 w + (USV^*)^* USV^* w = (USV^*)^* r. \quad (4.23)$$

We use the fact that the conjugate transpose of a unitary matrix is its inverse to write $I = V V^* = V I V^*$, and with this we convert (4.23) into

$$\lambda_0^2 V I V^* w + VS^* SV^* w = VS^* U^* r, \quad (4.24)$$

which is equivalent to

$$V(\lambda_0^2 I + S^* S)V^* w = VS^* U^* r. \quad (4.25)$$

Note that when $\lambda_0 > 0$, matrix $\lambda_0^2 I + S^* S$ is non-singular and invertible. Thus the
solution to (4.25) is

\[ \mathbf{w} = V (\lambda_0^2 \mathbf{I} + S^* S)^{-1} V^* V S^* U^* r, \] (4.26)

\[ = V (\lambda_0^2 \mathbf{I} + S^* S)^{-1} S^* U^* r, \] (4.27)

\[ = V Q U^* r, \] (4.28)

where we define

\[ Q = (\lambda_0^2 \mathbf{I} + S^* S)^{-1} S^* . \] (4.29)

\( Q \) is a \( N \)-by-\( M \) rectangular diagonal matrix with main diagonal entries

\[ q_n = \frac{s_n}{s_n^2 + \lambda_0^2} \] (4.30)

where \( n = 1, \ldots, N \). We call the weights calculated from this method the optimization weights. The optimization weights represent all the components of the vector of the optimization weight function \( W \) at a certain imaging point.

Solving (4.11) does not guarantee that the optimization weight function \( \mathbf{w} \) is non-negative. By \( \mathbf{w} \) being non-negative, we mean both real and imaginary part of \( \mathbf{w} \) are non-negative. For the corresponding MRI problem, Samsonov [2] and Bydder et al [3] suspect that negative weights come from instability of the problem and give worse reconstruction than non-negative weights. In our particular problem, the Tikhonov regularization parameter \( \lambda_0 \) controls how close to zero the optimization weights are. We can choose a small \( \lambda_0 \) to get wild behaved weights, or a relatively bigger \( \lambda_0 \) to make most weights positive. To see the effects of different weights on point-spread functions and reconstructions, we have done numerical simulations to compare the results from different weights by choosing different \( \lambda_0 \). Ideally, we want to choose \( \lambda_0 \) so that both point-spread functions and backprojection reconstructions are improved. We will show numerical results in the next section to find a good choice of Tikhonov regularization parameter.

We conclude this section by summarizing the process to calculate the optimization weights by the following steps:
1. Write source-receiver pairs as a list \((x_s^{(n)}, x_r^{(n)})\), where \(n\) is no greater than the number of source-receiver pairs.

2. Create a \(J\)-by-\(K\)-by-\(L\) mesh in the \(z\)-computational domain. Write all the voxels in mesh as an ordered list with the first voxel centered at the origin. The center point of \(m\)th voxel is \(\bar{z}_m\), where \(m \leq J \times K \times L\).

3. Choose an imaging point \(\bar{x}\).

4. Set \(\lambda_m = 1\) for \(m \geq 2\), and choose \(\lambda_0\) and \(\lambda_1\). Both parameters should be positive.

5. Construct the matrix \([A_{mn}]\) by (4.5) from \(m\)th test function and \(n\)th source-receiver pair.

6. Construct the vector \(r\) by (4.6).

7. Find the singular value decomposition of the matrix \(A\).

8. Construct the rectangular diagonal matrix \(Q\) by (4.30) from the singular values of \(A\) and \(\lambda_0\).

9. Calculate the optimization weights \(w\) at \(\bar{x}\) by (4.28).

10. Pick the next imaging point, go back to step 4 and follow the same procedure until the optimization weights have been calculated for all the imaging points.

### 4.3 Settings and Assumptions of Numerical Simulation

In this section, we make some assumptions to simplify the problem for the purpose of numerical simulation. In order to avoid the computational cost of solving full 3-D problems, we assume the medium parameters \(v(\bar{x}), c(\bar{x})\) and \(\alpha(\bar{x})\) vary only in the \(x\) and \(z\) directions. We also assume all the sources and receivers are located in the plane \(y = 0\). In this scenario, wave propagation is still governed by 3-D Helmholtz equation. But for any source-receiver pair, the wave needs to be computed only on the plane \(y = 0\). Numerically, we can ignore any computation that involves nonzero values of the variable \(y\). In this case, the second element of
our 3-D variables is always zero. Thus we can drop the second entry and represent them as 2-D variables. For instance, $\mathbf{x} = (x_1, x_3)$, $\mathbf{y} = (y_1, y_3)$, $\mathbf{z} = (z_1, z_3)$ and $\mathbf{k} = (k_1, k_3)$. For further simplicity, we assume the background velocity is constant everywhere, i.e., $c(\mathbf{x}) = c = \bar{c}$. In this case, ray paths become straight lines and the calculation of the travel time, Green’s function and $\nabla_\mathbf{x} \tau(\mathbf{x}, \mathbf{y})$ becomes the simple evaluation of the following formulae:

$$
\tau(\mathbf{x}, \mathbf{y}) = \frac{\|\mathbf{x} - \mathbf{y}\|}{c},
$$

(4.31)

$$
G_0(\mathbf{x}, \mathbf{y}, \omega) = \frac{e^{i\omega c \|\mathbf{x} - \mathbf{y}\|}}{4\pi \|\mathbf{x} - \mathbf{y}\|},
$$

(4.32)

$$
\nabla_\mathbf{x} \tau(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} - \mathbf{y}}{c \|\mathbf{x} - \mathbf{y}\|}.
$$

(4.33)

A wave emanates from each source and is recorded by every receiver. The number of source-receiver pairs is equal to the number of sources times the number of receivers. We also assume no two sources emit waves at the same time. In other words, they are well separated in time so that the waves from different sources do not interfere with each other.

For applications that use real signals, for example, seismic imaging, both positive and negative frequency content can be obtained by simply applying the inverse Fourier transform to the measurements. In this case, $\mathbf{A}$ is a real matrix, because the imaginary part of the integrated function in equation (4.5) is an odd function. However, $\mathbf{A}$, $\mathbf{w}$ and $\mathbf{r}$ might be complex for applications that use analytic signals, for example, radar imaging, where recorded data is preprocessed by removing the negative frequency content. For simplicity here, we assume we start with real data and the frequency $\omega$ satisfies $\omega \in \Omega = \{\omega|\omega_l \leq |\omega| \leq \omega_h\}$, where $\omega_h$ and $\omega_l$ are highest positive frequency and lowest positive frequency. Thus, $\mathbf{A}$, $\mathbf{w}$ and $\mathbf{r}$ are all real quantities. And because of the symmetry in the frequency band, the equation in (4.3) doesn’t change if the sign of $z_m$ is reversed. Although we plan to choose the $z$-computational domain to be a square centered at origin, in our 2-D simulations, we only use the half of the region with a non-negative horizontal coordinate to decrease the dimension of $\mathbf{A}$.

To form the matrix $\mathbf{A}$, we need to calculate the $\omega$ integration in (4.5), which
involves the evaluation of the indefinite integral

\[ g(\omega) = \int \omega^2 e^{-i\omega p} d\omega \]

\[ = \left( -\frac{\omega^2}{ip} + \frac{2\omega}{p^2} + \frac{2}{ip^3} \right) e^{-i\omega p}. \]  

(4.34)

(4.35)

We use the equation above to evaluate \( g(\omega) \) when \( \omega_h |p| \) isn’t too small. But for \( |p| \) small such that \( \omega_h |p| \) is also small, we use the Taylor expansion:

\[ e^{-i\omega p} \approx 1 - i\omega p + \frac{(-i\omega p)^2}{2}. \]  

(4.36)

Thus, when \( \omega_h |p| \) is small, \( g(\omega) \) can be approximated by

\[ g(\omega) \approx \int \omega^2 (1 - i\omega p - \frac{(\omega p)^2}{2}) d\omega, \]

\[ = \frac{1}{3} \omega^3 - \frac{ip}{4} \omega^4 - \frac{p^2}{10} \omega^5. \]  

(4.37)

(4.38)

Now, we can carry out the \( \omega \) integration in (4.5) using formulae (4.35) and (4.38). Let \( p = \nabla_x (\tau(x^{(n)}_s, x) + \tau(x, x^{(n)}_r)) \cdot \bar{z}_m \). For \( \omega_h |p| \) is not small, we carry out the integral in \( \Omega \) to obtain

\[ \int_{\Omega} \omega^2 e^{-i\omega p} d\omega = g(\omega_h) - g(-\omega_h) - g(\omega_l) + g(-\omega_l), \]

\[ = 2 \left( \frac{2\omega_h}{p^2} \cos(\omega_h p) - \frac{\omega_h^2}{p} - \frac{2}{p^3} \right) \sin(\omega_h p) \]

\[ - 2\omega_l \left( \frac{\omega_l^2}{p} - \frac{2}{p^3} \right) \sin(\omega_l p). \]  

(4.39)

(4.40)

For \( \omega_h |p| \) is small,

\[ \int_{\Omega} \omega^2 e^{-i\omega p} d\omega \approx 2 \left( \frac{1}{3} \omega_h^3 - \frac{p^2}{10} \omega_h^5 - \frac{1}{3} \omega_l^3 + \frac{p^2}{10} \omega_l^5 \right). \]  

(4.41)

Use these results in (4.5), we obtain the expression for \( A_{mn} \). When \( \omega_h |p| \) is not
small,

$$A_{mn} = \frac{2\lambda_m}{(2\pi\bar{c})^3} \left( \frac{2\omega_h}{p^2} \cos(\omega_h p) - \left( \frac{\omega_h^2}{p} - \frac{2}{p^3} \right) \sin(\omega_h p) ight) - \frac{2\omega_l}{p^2} \cos(\omega_l p) + \left( \frac{\omega_l^2}{p} - \frac{2}{p^3} \right) \sin(\omega_l p) \right). \quad (4.42)$$

And, when $\omega_h |p|$ is small,

$$A_{mn} \approx \frac{2\lambda_m}{(2\pi\bar{c})^3} \left( \frac{1}{3} \omega_h^3 - \frac{p^2}{10} \omega_h^5 - \frac{1}{3} \omega_l^3 + \frac{p^2}{10} \omega_l^5 \right). \quad (4.43)$$

We need to decide what regularization parameters we want to choose. We have assumed $\lambda_1$ corresponds to the test function supported on the pixel that contains origin which is actually the center of the point-spread function at any fixed imaging point. We choose $\lambda_1 \geq 1$, and $\lambda_m = 1$ for $m \geq 2$ to emphasize the importance of the center peak of the point-spread function. Usually, $\lambda_1$ should be large if the number of test functions is large. Otherwise, the solution to our least squares problem tends to zero as the number of equations in the problem becomes larger, since only the first entry of $r$ is nonzero. This phenomenon can be seen from the QR factorization solution of the problem. We will choose $\lambda_1$ so that point-spread functions at different locations have almost same center height. Once $\lambda_1$ is determined, $\lambda_0$ should be chosen accordingly. However, determining $\lambda_0$ not only depends on how different weights affect the point-spread functions but also depends on the stability of reconstructions. We hope to choose $\lambda_0$ to calculate optimization weights to improve point-spread functions and get stable backprojection reconstructions. Reconstruction tests must be done to see what value of $\lambda_0$ is good.

Before we show numerical results we need to point out the units we use in our computation. In our simulations, the units on the horizontal and the vertical axes in the point-spread function plot, in the reconstructed image, and in the half-maximum-value contour plot are $km$. The units of point-spread function values are $1/km^2$. The image $I(x)$ doesn’t have a unit in our formulation. The units on the horizontal and the vertical axes in the plots of bisecting line segments are $rad/km$. Because of our choice of scaling, the weight function $w$ doesn’t have a
Figure 4.2: Geometry of sources (red stars) and receivers (blue circles). The horizontal axis is $y_1$ and the vertical axis is $y_3$. The yellow square is the computational domain for reconstruction experiments.

Throughout this chapter, we consider these units as default units, unless explicitly stated.

4.4 Reconstruction Experiments

In this section, we test our optimization scheme by reconstructing a point scatterer. Simulation results will be shown to illustrate how the regularization parameter $\lambda_0$ affects reconstruction.
Table 4.1: Locations of sources and receivers as shown in Fig. 4.2. The units are km.

<table>
<thead>
<tr>
<th>Source</th>
<th>Coordinates</th>
<th>Receiver</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{s1}$</td>
<td>(3.00, 2.00)</td>
<td>$x_{r1}$</td>
<td>(0.80, 0.20)</td>
</tr>
<tr>
<td>$x_{s2}$</td>
<td>(3.50, 1.75)</td>
<td>$x_{r2}$</td>
<td>(1.00, 0.20)</td>
</tr>
<tr>
<td>$x_{s3}$</td>
<td>(3.20, 3.00)</td>
<td>$x_{r3}$</td>
<td>(1.20, 0.20)</td>
</tr>
<tr>
<td>$x_{s4}$</td>
<td>(3.10, 4.50)</td>
<td>$x_{r4}$</td>
<td>(2.40, 0.20)</td>
</tr>
<tr>
<td>$x_{s5}$</td>
<td>(3.30, 5.00)</td>
<td>$x_{r5}$</td>
<td>(2.50, 0.20)</td>
</tr>
<tr>
<td>$x_{s6}$</td>
<td>(3.15, 6.00)</td>
<td>$x_{r6}$</td>
<td>(2.60, 0.20)</td>
</tr>
<tr>
<td>$x_{s7}$</td>
<td>(3.25, 4.50)</td>
<td>$x_{r7}$</td>
<td>(3.00, 0.20)</td>
</tr>
<tr>
<td>$x_{s8}$</td>
<td>(3.30, 4.40)</td>
<td>$x_{r8}$</td>
<td>(3.20, 0.20)</td>
</tr>
<tr>
<td>$x_{s9}$</td>
<td>(3.75, 3.75)</td>
<td>$x_{r9}$</td>
<td>(3.25, 0.20)</td>
</tr>
<tr>
<td>$x_{s10}$</td>
<td>(3.00, 3.50)</td>
<td>$x_{r10}$</td>
<td>(3.70, 0.20)</td>
</tr>
<tr>
<td>$x_{s11}$</td>
<td>(2.90, 5.15)</td>
<td>$x_{r11}$</td>
<td>(3.75, 0.20)</td>
</tr>
<tr>
<td>$x_{s12}$</td>
<td>(5.00, 4.00)</td>
<td>$x_{r12}$</td>
<td>(3.90, 0.20)</td>
</tr>
<tr>
<td>$x_{s13}$</td>
<td>(2.75, 5.50)</td>
<td>$x_{r13}$</td>
<td>(4.50, 0.20)</td>
</tr>
<tr>
<td>$x_{s14}$</td>
<td>(2.00, 2.50)</td>
<td>$x_{r14}$</td>
<td>(4.80, 0.20)</td>
</tr>
<tr>
<td>$x_{s15}$</td>
<td>(3.25, 4.00)</td>
<td>$x_{r15}$</td>
<td>(5.30, 0.20)</td>
</tr>
<tr>
<td>$x_{s16}$</td>
<td>(3.00, 4.10)</td>
<td>$x_{r16}$</td>
<td>(6.50, 0.20)</td>
</tr>
</tbody>
</table>

4.4.1 Geometry and Simulation Data

We intend to use equation (2.10) to construct simulation data. The source-receiver geometry is shown in Fig. 4.2. The yellow square corresponds to our computational domain. In this geometry, sources and receivers are located outside our computational domain to avoid the situation that an imaging point coincides with a source or a receiver. The exact coordinates of sources and receivers can be found in Table 4.1. The actual image shown in Fig. 4.3 is a homogeneous medium with a point-like scatterer at $y_0 = (4.3, 2.1)$. The point-like scatterer is a small square of side length 10 meters. Mathematically, it is defined as

$$\alpha(y) = \begin{cases} 
1 & \text{if } \|y - y_0\|_\infty \leq 0.005 \text{km}, \\
0 & \text{otherwise},
\end{cases}$$

(4.44)

where $\| \cdot \|_\infty$ represents infinity norm and is defined to be the maximum absolute value of all elements of a vector.
Figure 4.3: Actual image with a 10 meter by 10 meter point-like scatterer at the center. The horizontal axis is $y_1$ and the vertical axis is $y_3$. The units on each of the two axes is $km$.

The simulation data is then calculated by equation (2.10) with $\alpha(x)$ defined by (4.44). Note that when the background speed is constant,

$$\alpha(x_s, y, x_r) = \frac{1}{16\pi^2 \|x_s - y\| \|y - x_r\|},$$  \hfill (4.45)  

$$\phi(x_s, y, x_r) = \frac{\|x_s - y\| + \|y - x_r\|}{c}.$$  \hfill (4.46)

Since the size of the point-like scatterer is relatively small, we use the following
approximation of the scattered wave:

\[
\begin{align*}
u_s^D(x_r, x_s, \omega) &= \frac{1}{16\pi^2 c^2} \frac{\omega^2}{\|x_s - y\|\|y - x_r\|} e^{i\frac{\omega}{c}(|x_s - y| + |y - x_r|)} dy, \quad (4.47) \\
&\approx \frac{10^{-4} \omega^2}{16\pi^2 c^2 \|x_s - y_0\|\|y_0 - x_r\|} e^{i\frac{\omega}{c}(|x_s - y_0| + |y_0 - x_r|)}, \quad (4.48)
\end{align*}
\]

where \(10^{-4}\) comes from the area of the point-like scatterer. Note that once we make the above approximation, it is equivalent to setting

\[
\alpha(y) = 10^{-4} \delta(y - y_0), \quad (4.49)
\]

implying that this point-like scatterer is actually a point scatterer at \(y_0\).

In order to simulate band-limited data, we consider \(u_s^D(x_r, x_s, \omega)\) above to be zero outside \(\Omega = \{\omega: \omega_l \leq |\omega| \leq \omega_h\}\). Then we take the Fourier transform of \(u_s^D(x_r, x_s, \omega)\) to get the time-domain scattered wave \(\hat{u}_s^D(x_r, x_s, t)\).

\[
\begin{align*}
\hat{u}_s^D(x_r, x_s, t) &= \int_{\Omega} e^{-i\omega t} u_s^D(x_r, x_s, \omega) d\omega, \quad (4.50) \\
&= \frac{10^{-4}}{16\pi^2 c^2 \|x_s - y_0\|\|y_0 - x_r\|} \int_{\Omega} \omega^2 e^{-i\omega(t-\phi)} d\omega, \quad (4.51)
\end{align*}
\]

where \(\phi = \frac{\|x_s - y\| + \|y - x_r\|}{c}\). The \(\omega\) integration in the equation above can be evaluated using (4.35) and (4.38). Thus, when \(\omega_h |t - \Phi|\) is not small,

\[
\begin{align*}
\hat{u}_s^D(x_r, x_s, t) &= \frac{10^{-4}}{8\pi^2 c^2 \|x_s - y_0\|\|y_0 - x_r\|} \left( \frac{2\omega_h}{(t-\phi)^2} \cos(\omega_h(t-\phi)) \\
&\quad - \left( \frac{\omega_h^2}{(t-\phi)} - \frac{2}{(t-\phi)^2} \right) \sin(\omega_h(t-\phi)) - \frac{2\omega_l}{(t-\phi)^2} \cos(\omega_l(t-\phi)) \\
&\quad + \left( \frac{\omega_l^2}{(t-\phi)} - \frac{2}{(t-\phi)^2} \right) \sin(\omega_l(t-\phi)) \right) \quad (4.52)
\end{align*}
\]

When \(\omega_h |t - \Phi|\) is small,

\[
\begin{align*}
\hat{u}_s^D(x_r, x_s, t) &\approx \frac{10^{-4}}{8\pi^2 c^2 \|x_s - y_0\|\|y_0 - x_r\|} \left( \frac{1}{3} \omega_h^3 - \frac{(t-\Phi)^2}{10 \omega_h^5} \right) \\
&\quad - \frac{1}{3} \omega_l^3 + \frac{(t-\Phi)^2}{10 \omega_l^5}. \quad (4.53)
\end{align*}
\]
Fig. 4.4 shows an example of the time-domain scattered wave calculated at receiver $x_{r1}$ from source $x_{s5}$. We can approximate the arrival time of the reflected wave by finding the peak, which is at around $t = 1.45s$ in this example. This means that the scattered wave takes 1.45 seconds in total to propagate from source to scatterer, then from scatterer to receiver.

Here we need to point out that because we don’t have zero frequency data, backprojection by (3.14) cannot recover the actual image even when the sources and receivers are equally spaced. But the correct locations of discontinuities can be recovered (see [1]). However, more information about a point scatterer can be recovered if the image is scaled properly. Let $\alpha(y) = R\delta(y - y_0)$ and substitute the
expression into (3.21). By computing the image value at the scatterer \( y_0 \), we obtain

\[
I_{W}(y_0) = \frac{1}{(2\pi \bar{c})^3} \sum_{\sigma} W(y_0, \sigma) \int_{\Omega} \frac{c^2(y_0)}{a(y_0, \sigma)} \omega^2 \\
\times \int_{R^3} \frac{R\delta(y - y_0)}{c^2(y)} a(y, \sigma) e^{-i\omega(\phi(y_0, \sigma) - \phi(y, \sigma))} dy d\omega
\]

\[
= \frac{R}{(2\pi \bar{c})^3} \sum_{\sigma} W(y_0, \sigma) \int_{\Omega} \omega^2 d\omega.
\] \hspace{1cm} (4.54)

This suggests that the image value is \( R \) multiplied by the k-space “volume” \( V_K(y_0) \), which is given by

\[
V_K(x) = \frac{1}{(2\pi)^3} \int dk \\
= \frac{1}{(2\pi)^3} \int |h(x, \sigma)| \int_{\Omega} \omega^2 d\omega d\sigma \\
\approx \frac{1}{(2\pi \bar{c})^3} \sum_{\sigma} W(x, \sigma) \int_{\Omega} \omega^2 d\omega.
\] \hspace{1cm} (4.56)

If we scale the image by dividing it by the k-space “volume” (4.56), then we can recover the amplitude \( R \) at the scatterer. The quantity \( V_K(x) \) is exactly the center peak of the point-spread function at \( x \). Since we choose \( \lambda_1 \) big enough so that the point-spread function at different locations in our computational domain has almost the same peak value, the scaling factor \( V_K(x) \) is almost a constant everywhere. However, scaling the point-spread function changes neither the resolution nor the stability of backprojection. Because our purpose here is to show resolution and stability, and the scaling factor is almost invariant in our computational domain, we ignore the scaling in our simulations.

Next, we calculate the optimization weights by choosing \( \lambda_0 = 1, \lambda_0 = 500, \lambda_0 = 10,000, \lambda_0 = 120,000 \) and \( \lambda_0 = 200,000 \) using the same frequency band from 5Hz to 50Hz. Throughout the calculations, the background velocity is \( c(x) = 5 \text{km/s} \), the \( z \)-computational domain is the square \( ||z||_\infty \leq 0.6 \text{km} \), the mesh size \( \Delta z \) is 0.01 km, and the regularization parameter \( \lambda_1 \) is 20. We use these optimization weights and constant weights to calculate point-spread functions and reconstruct the point scatterer.
Figure 4.5: The bisecting line segments for the scatterer $y_0$. Each line segment is colored according to its weight, which is given by the grey scale to the right. In this case, all the weights are equal to 1. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $\text{rad/km}$.

4.4.2 Reconstruction with Constant Weights

The trivial choice of weights is constant weights. We simply set $w = 1$ to see the point-spread function at scatterer location, and to see the reconstructed image. In simulations, we scale the constant weights so that the point-spread function has the same height as the optimization weights case.

Fig. 4.5 shows the bisecting line segments, colored according to the corresponding weight. The bisecting line segment is the set of points defined by (3.18). We plot only the positive frequency part, since the bisecting line segment of negative frequency is simply reversing the sign. In the constant weights case, since each bisecting line segment has the same weight value, Fig. 4.5 is simply a plot of all the bisecting line segments with the same grey scale value.

Fig. 4.6 shows the point-spread function at the scatterer $y_0$ calculated with constant weights. This point-spread function has a high peak at the center, but blurs out in the vertical direction. This means that the reconstruction at this point
Figure 4.6: Point-spread function at scatterer $y_0$ using constant weights. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are km. The units of the point-spread function values are $1/\text{km}^2$.

has better resolution in the horizontal direction but worse resolution in the vertical direction. Note that although there are bisecting line segments pointing in the vertical direction, there are many more bisecting line segments are pointing in the horizontal direction. Weighting each direction the same ruins the resolution in the vertical direction. We hope to choose optimization weights to compensate for this effect.

Fig. 4.7 shows the reconstruction using constant weights and simulation data without noise. The reconstruction is similar to the point-spread function shown in Fig. 4.6. The image and the point-spread function are related by equation (3.23) and should be almost the same if $\alpha(x)$ is a point-like scatterer. However, there is difference between them that lies in the difference in the amplitude function and phase function between (3.17) and (3.21), where we have approximated $a(y, \sigma)$ by $a(x, \sigma)$, and $\phi(x, \sigma) - \phi(y, \sigma)$ by $\nabla_x \phi(x, \sigma) \cdot (x - y)$. The reconstruction simulation result suggests these approximations work well with constant weights.

Fig. 4.8 shows the reconstruction with 10% Gaussian noise in the simulation
Figure 4.7: Reconstruction without noise using constant weights. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km.

Figure 4.8: Reconstruction with 10% noise in scattered waves using constant weights. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km.
data using constant weights. Although it is noisier than the reconstruction without noise, the point scatterer is still localized. This shows that the reconstruction with constant weights is not sensitive to noise in data.

Fig. 4.9 shows the reconstruction with up-to-50-meter positioning error in the locations of sources and receivers using constant weights. Here 50 meters is the half minimum wavelength. Although 50-meter positioning error seems unlikely for geophysical experiments, for other applications such as radar, positioning errors of half a wavelength are difficult to avoid. For positioning errors larger than a wavelength, other methods such as autofocus techniques [25] would need to be applied to compensate for the error.

In Fig. 4.9, the center of reconstructed image has lower value than in Fig. 4.7 because it is not well focused. However, a peak is located at the correct scatterer position. This shows that constant weights are not sensitive to positioning error.
Table 4.2: Optimization weights at $y_0$ with $\lambda_0 = 1$.

<table>
<thead>
<tr>
<th>$x_{s_1}$</th>
<th>$x_{r_1}$</th>
<th>$x_{r_2}$</th>
<th>$x_{r_3}$</th>
<th>$x_{r_4}$</th>
<th>$x_{r_5}$</th>
<th>$x_{r_6}$</th>
<th>$x_{r_7}$</th>
<th>$x_{r_8}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.9769</td>
<td>2.9901</td>
<td>1.1756</td>
<td>-0.9660</td>
<td>0.2031</td>
<td>0.6100</td>
<td>0.3736</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.0686</td>
<td>103.4645</td>
<td>14.8646</td>
<td>39.9549</td>
<td>-85.0221</td>
<td>53.3586</td>
<td>-34.4318</td>
<td>12.8120</td>
<td></td>
</tr>
<tr>
<td>149.9825</td>
<td>-1.3665</td>
<td>-134.9614</td>
<td>-23.2294</td>
<td>-13.9866</td>
<td>103.3017</td>
<td>-22.4166</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-90.0686</td>
<td>103.4645</td>
<td>14.8646</td>
<td>39.9549</td>
<td>-85.0221</td>
<td>53.3586</td>
<td>-34.4318</td>
<td>12.8120</td>
<td></td>
</tr>
<tr>
<td>-196.5213</td>
<td>-44.7740</td>
<td>-2.4722</td>
<td>-0.7842</td>
<td>12.3127</td>
<td>-87.3816</td>
<td>-1.9411</td>
<td>27.6090</td>
<td></td>
</tr>
<tr>
<td>-0.0758</td>
<td>-0.5784</td>
<td>0.7712</td>
<td>-0.1708</td>
<td>0.0981</td>
<td>0.6151</td>
<td>0.1930</td>
<td>0.3386</td>
<td></td>
</tr>
<tr>
<td>1.7810</td>
<td>1.4181</td>
<td>-0.7404</td>
<td>1.0439</td>
<td>1.0033</td>
<td>0.9728</td>
<td>0.7504</td>
<td>0.6748</td>
<td></td>
</tr>
<tr>
<td>0.3891</td>
<td>0.3080</td>
<td>-0.3126</td>
<td>0.2523</td>
<td>0.0913</td>
<td>0.4172</td>
<td>1.0444</td>
<td>15.3340</td>
<td></td>
</tr>
<tr>
<td>-32.6102</td>
<td>55.3625</td>
<td>-50.8512</td>
<td>-7.0214</td>
<td>11.5186</td>
<td>-52.5777</td>
<td>5.2745</td>
<td>-17.7751</td>
<td></td>
</tr>
<tr>
<td>31.6791</td>
<td>143.3376</td>
<td>-60.5188</td>
<td>-29.2162</td>
<td>-125.0759</td>
<td>-67.9785</td>
<td>102.0557</td>
<td>5.8228</td>
<td></td>
</tr>
<tr>
<td>6.4268</td>
<td>31.0328</td>
<td>-27.5962</td>
<td>-2.9662</td>
<td>-4.6776</td>
<td>106.6140</td>
<td>94.4317</td>
<td>-0.7593</td>
<td></td>
</tr>
<tr>
<td>21.7685</td>
<td>27.8876</td>
<td>-51.5884</td>
<td>-0.8350</td>
<td>30.9815</td>
<td>-21.2939</td>
<td>165.5673</td>
<td>-46.1496</td>
<td></td>
</tr>
<tr>
<td>28.9207</td>
<td>49.3142</td>
<td>-74.3624</td>
<td>2.0149</td>
<td>26.9453</td>
<td>-25.7871</td>
<td>221.3428</td>
<td>25.2829</td>
<td></td>
</tr>
<tr>
<td>-41.6813</td>
<td>-170.4430</td>
<td>96.5955</td>
<td>30.1132</td>
<td>125.1369</td>
<td>-38.8116</td>
<td>-137.0560</td>
<td>3.7277</td>
<td></td>
</tr>
<tr>
<td>0.2786</td>
<td>0.4525</td>
<td>-0.3297</td>
<td>0.0914</td>
<td>0.1206</td>
<td>-0.1066</td>
<td>-1.5953</td>
<td>6.6445</td>
<td></td>
</tr>
<tr>
<td>-21.6514</td>
<td>-66.2378</td>
<td>87.1666</td>
<td>5.1572</td>
<td>-34.1777</td>
<td>20.8870</td>
<td>-144.9117</td>
<td>49.9501</td>
<td></td>
</tr>
<tr>
<td>-3.0645</td>
<td>-21.5640</td>
<td>20.9442</td>
<td>-3.2871</td>
<td>0.3446</td>
<td>-0.0289</td>
<td>-0.5677</td>
<td>-0.0363</td>
<td></td>
</tr>
<tr>
<td>-17.1316</td>
<td>-60.1311</td>
<td>74.8232</td>
<td>1.3504</td>
<td>-18.8956</td>
<td>14.9819</td>
<td>-117.7476</td>
<td>-18.5222</td>
<td></td>
</tr>
<tr>
<td>0.5282</td>
<td>0.7276</td>
<td>-0.7053</td>
<td>0.3968</td>
<td>0.3704</td>
<td>0.3216</td>
<td>0.2050</td>
<td>0.1621</td>
<td></td>
</tr>
<tr>
<td>-6.4244</td>
<td>-0.4740</td>
<td>0.8792</td>
<td>0.0327</td>
<td>0.3763</td>
<td>2.2332</td>
<td>-105.9191</td>
<td>-0.3436</td>
<td></td>
</tr>
</tbody>
</table>
Figure 4.10: The bisecting line segments at the scatterer $y_0$, colored according to the optimization weights with $\lambda_0 = 1$. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $\text{rad/km}$. Note that many of the bisectors have small weights and cannot be seen with this color scale. Moreover, some weights have large positive values and some have large negative values.

4.4.3 Reconstruction with $\lambda_0 = 1$

Now, we choose $\lambda_0 = 1$ and calculate the optimization weights. Fig. 4.10 shows the bisecting line segments colored by the optimization weights calculated with $\lambda_0 = 1$. The bisecting line segments should be the same as Fig. 4.5. But many of the bisecting line segments have small weights and cannot be seen with this color scale. Moreover, some weights have large positive values and some have large negative values. Table 4.2 shows the exact values of the optimization weights for $\lambda_0 = 1$. The weight values are listed by each combination of source and receiver. In our least squares problem, the parameter $\lambda_0$ is the scaling factor for the norm of the weight vector. It controls how large the optimization weight values are. We find that when $\lambda_0 = 1$, the weight values lie in a very wide range, roughly from -200 to 300.

Fig. 4.11 shows the point-spread function at the scatterer. The point-spread
Figure 4.11: Point-spread function at the scatterer $y_0$ with optimization weights for $\lambda_0 = 1$. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The units of the point-spread function values are $1/km^2$.

The point-spread function is calculated using the optimization weights with $\lambda_0 = 1$. The point-spread function at the scatterer is improved by using optimization weights. In the vertical direction, it is much narrower than the one calculated from constant weights.

Fig. 4.12 shows the reconstruction without noise in data using optimization weights for $\lambda_0 = 1$. The gray scale has been chopped to show the behavior near the origin with the large values in regions farther from the origin are shown as 1 or $-1$. The original reconstructed image falls into the range of $(-35, 25)$ roughly. This figure shows a localized scatterer at the center, but the reconstruction is corrupted away from the scatterer. We have mentioned in the reconstruction simulation with constant weights that approximations have been made in the amplitude function and phase function in (3.21) to get (3.23). With this weight function calculated for $\lambda_0 = 1$, these approximations are good when $x$ is close to $y$, but not when they are far apart, where the error terms between these approximations are significantly amplified by large magnitude weights.

Fig. 4.13 shows the reconstruction with 10% Gaussian noise in the simulation
Figure 4.12: Reconstruction without noise using optimization weights for $\lambda_0 = 1$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km. The gray scale has been chopped to show the behavior near the origin; the large values in regions farther from the origin are shown as 1 or −1.

data. Fig. 4.14 shows the reconstruction with up-to-50-meter positioning error in the locations of sources and receivers.

4.4.4 Reconstruction with $\lambda_0 = 500$

Fig. 4.15 shows the bisecting line segments colored by the optimization weights calculated with $\lambda_0 = 500$. Again, the bisecting line segments should be the same as Fig. 4.5 and Fig. 4.10. Table 4.3 shows the exact values of optimization weights. When $\lambda_0 = 500$, the weight values lie in a much smaller range compared with the optimization weights calculated with $\lambda_0 = 1$. Note there are still a lot of weight values that are negative. Next, we calculate the point-spread function with this optimization weight function.

Fig. 4.16 shows the point-spread function at scatterer using the optimization weights calculated with $\lambda_0 = 1$. This point-spread function is much better than the
Figure 4.13: Reconstruction with 10% noise in scattered waves using optimization weights for $\lambda_0 = 1$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km.

Figure 4.14: Reconstruction with up-to-50-meter positioning error in the locations of sources and receivers using optimization weights for $\lambda_0 = 1$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km.
Table 4.3: Optimization weights at $y_0$ with $\lambda_0 = 500$.  

<table>
<thead>
<tr>
<th></th>
<th>$x_{r_1}$</th>
<th>$x_{r_2}$</th>
<th>$x_{r_3}$</th>
<th>$x_{r_4}$</th>
<th>$x_{r_5}$</th>
<th>$x_{r_6}$</th>
<th>$x_{r_7}$</th>
<th>$x_{r_8}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{s_1}$</td>
<td>2.7827</td>
<td>-3.0172</td>
<td>1.9224</td>
<td>1.3154</td>
<td>-1.1963</td>
<td>0.4362</td>
<td>0.5553</td>
<td>0.7596</td>
</tr>
<tr>
<td>$x_{s_2}$</td>
<td>1.2213</td>
<td>-1.1936</td>
<td>1.3818</td>
<td>1.6000</td>
<td>-1.9725</td>
<td>2.0713</td>
<td>1.1627</td>
<td>-0.8502</td>
</tr>
<tr>
<td>$x_{s_3}$</td>
<td>1.2309</td>
<td>-1.4509</td>
<td>0.8560</td>
<td>0.0059</td>
<td>-0.809</td>
<td>0.0419</td>
<td>0.0444</td>
<td>-0.433</td>
</tr>
<tr>
<td>$x_{s_4}$</td>
<td>0.3794</td>
<td>-0.1146</td>
<td>0.5972</td>
<td>0.1945</td>
<td>-0.0848</td>
<td>-0.0947</td>
<td>-0.2286</td>
<td>-0.1424</td>
</tr>
<tr>
<td>$x_{s_5}$</td>
<td>1.8453</td>
<td>-0.9667</td>
<td>-0.1066</td>
<td>0.3142</td>
<td>0.0283</td>
<td>0.0805</td>
<td>0.0984</td>
<td>0.1549</td>
</tr>
<tr>
<td>$x_{s_6}$</td>
<td>0.4747</td>
<td>-0.2483</td>
<td>-0.0031</td>
<td>-0.0326</td>
<td>0.0446</td>
<td>0.1242</td>
<td>0.1608</td>
<td>-0.0335</td>
</tr>
<tr>
<td>$x_{s_7}$</td>
<td>0.5431</td>
<td>-0.2018</td>
<td>0.0210</td>
<td>-0.0333</td>
<td>0.0575</td>
<td>0.1466</td>
<td>0.1905</td>
<td>-0.0231</td>
</tr>
<tr>
<td>$x_{s_8}$</td>
<td>-0.0977</td>
<td>-1.0368</td>
<td>0.469</td>
<td>-0.0315</td>
<td>-0.1899</td>
<td>-0.0957</td>
<td>-0.0835</td>
<td>-0.0869</td>
</tr>
<tr>
<td>$x_{s_9}$</td>
<td>0.8805</td>
<td>-0.2399</td>
<td>0.2039</td>
<td>0.5844</td>
<td>-0.1899</td>
<td>0.0896</td>
<td>0.0274</td>
<td>0.1356</td>
</tr>
<tr>
<td>$x_{s_{10}}$</td>
<td>0.1179</td>
<td>-0.4319</td>
<td>0.0252</td>
<td>0.0117</td>
<td>-0.0387</td>
<td>0.0771</td>
<td>0.0444</td>
<td>-0.433</td>
</tr>
<tr>
<td>$x_{s_{11}}$</td>
<td>0.8696</td>
<td>-1.0619</td>
<td>0.2531</td>
<td>0.1376</td>
<td>-0.2830</td>
<td>0.2200</td>
<td>0.0300</td>
<td>-0.0827</td>
</tr>
<tr>
<td>$x_{s_{12}}$</td>
<td>-0.2406</td>
<td>-0.2424</td>
<td>0.5316</td>
<td>-0.1284</td>
<td>0.1092</td>
<td>0.6478</td>
<td>0.2102</td>
<td>0.3620</td>
</tr>
<tr>
<td>$x_{s_{13}}$</td>
<td>1.6019</td>
<td>1.3478</td>
<td>-0.6447</td>
<td>1.0550</td>
<td>1.0423</td>
<td>1.0158</td>
<td>0.7855</td>
<td>0.7111</td>
</tr>
<tr>
<td>$x_{s_{14}}$</td>
<td>0.4593</td>
<td>0.1288</td>
<td>-0.1494</td>
<td>0.2635</td>
<td>0.0919</td>
<td>0.2704</td>
<td>0.4176</td>
<td>0.1820</td>
</tr>
<tr>
<td>$x_{s_{15}}$</td>
<td>0.0344</td>
<td>0.0460</td>
<td>-0.1630</td>
<td>0.0242</td>
<td>0.1021</td>
<td>-0.0859</td>
<td>-0.1456</td>
<td>0.1674</td>
</tr>
<tr>
<td>$x_{s_{16}}$</td>
<td>0.0176</td>
<td>-0.1968</td>
<td>0.1146</td>
<td>0.1105</td>
<td>-0.0691</td>
<td>0.1106</td>
<td>-0.1739</td>
<td>0.2518</td>
</tr>
<tr>
<td>$x_{s_{17}}$</td>
<td>0.0017</td>
<td>0.1702</td>
<td>-0.3164</td>
<td>-0.0141</td>
<td>0.5191</td>
<td>-0.1783</td>
<td>0.1479</td>
<td>-0.4714</td>
</tr>
<tr>
<td>$x_{s_{18}}$</td>
<td>0.0999</td>
<td>0.0611</td>
<td>0.0087</td>
<td>-0.0046</td>
<td>-0.0336</td>
<td>0.0093</td>
<td>0.0669</td>
<td>-0.1386</td>
</tr>
<tr>
<td>$x_{s_{19}}$</td>
<td>0.1085</td>
<td>0.0527</td>
<td>0.0101</td>
<td>-0.0038</td>
<td>-0.0422</td>
<td>0.0252</td>
<td>0.0570</td>
<td>-0.1468</td>
</tr>
<tr>
<td>$x_{s_{20}}$</td>
<td>-0.0600</td>
<td>-0.1055</td>
<td>0.1326</td>
<td>0.0527</td>
<td>0.0404</td>
<td>0.0344</td>
<td>-0.0830</td>
<td>0.2622</td>
</tr>
<tr>
<td>$x_{s_{21}}$</td>
<td>-0.0096</td>
<td>0.1402</td>
<td>-0.0483</td>
<td>0.0519</td>
<td>0.1067</td>
<td>0.0088</td>
<td>0.0838</td>
<td>0.2580</td>
</tr>
<tr>
<td>$x_{s_{22}}$</td>
<td>0.0442</td>
<td>0.0888</td>
<td>-0.0285</td>
<td>0.0052</td>
<td>0.0273</td>
<td>-0.0874</td>
<td>0.0770</td>
<td>-0.0325</td>
</tr>
<tr>
<td>$x_{s_{23}}$</td>
<td>0.4931</td>
<td>0.0607</td>
<td>0.1658</td>
<td>-0.5358</td>
<td>0.1377</td>
<td>0.1212</td>
<td>0.0027</td>
<td>0.1601</td>
</tr>
<tr>
<td>$x_{s_{24}}$</td>
<td>0.0505</td>
<td>0.0886</td>
<td>-0.0204</td>
<td>0.0027</td>
<td>0.0193</td>
<td>-0.0774</td>
<td>0.0832</td>
<td>-0.0514</td>
</tr>
<tr>
<td>$x_{s_{25}}$</td>
<td>0.7667</td>
<td>0.6305</td>
<td>-0.6093</td>
<td>0.3896</td>
<td>0.3922</td>
<td>0.3576</td>
<td>0.2272</td>
<td>0.1803</td>
</tr>
<tr>
<td>$x_{s_{26}}$</td>
<td>0.1504</td>
<td>0.0745</td>
<td>-0.1716</td>
<td>0.0217</td>
<td>-0.0396</td>
<td>-0.1079</td>
<td>-0.1417</td>
<td>-0.3489</td>
</tr>
<tr>
<td>$x_{s_{27}}$</td>
<td>0.0829</td>
<td>0.1136</td>
<td>-0.0289</td>
<td>0.1311</td>
<td>0.0898</td>
<td>0.0807</td>
<td>-0.3038</td>
<td>0.1404</td>
</tr>
</tbody>
</table>
Figure 4.15: The bisecting line segments at the scatterer \( y_0 \), colored according to the optimization weights with \( \lambda_0 = 500 \). The horizontal axis is \( k_1 \) and the vertical axis is \( k_3 \). The units on each axis are \( \text{rad/km} \). Gray-scale corresponds to the weight values on bisecting line segments.

point-spread function with constant weights.

Fig. 4.17 shows the reconstruction without noise in data using the optimization weights for \( \lambda_0 = 500 \). For this choice of \( \lambda_0 \), it successfully constructs the point scatterer. The image shows a better focus at the scatterer location than the image shown in Fig. 4.7 where constant weights are applied. Again, this verifies that optimization weights improve the resolution.

Fig. 4.18 shows the reconstruction with 10\% Gaussian noise in the simulation data. Adding noise in the data gives us a reasonable reconstruction. Although the image looks very noisy, the point scatterer can still be recognized at the center. Fig. 4.19 shows the reconstruction with up-to-50-meter positioning error in the locations of sources and receivers. The highest peak in the image moves to \((4.37, 1.98)\), which is about 0.139 kilometer or 139 meters away from the actual scatterer location. This weight function results in a reconstruction that is sensitive to positioning error.

Comparing the results for \( \lambda_0 = 500 \) with those for \( \lambda_0 = 1 \), we see that the larger
Figure 4.16: Point spread function at the scatterer $y_0$ using optimization weights for $\lambda_0 = 500$. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The units of the point-spread function values are $1/km^2$.

Figure 4.17: Reconstruction without noise using optimization weights for $\lambda_0 = 500$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are $km$. 

Figure 4.18: Reconstruction with 10% noise in scattered waves using optimization weights for $\lambda_0 = 500$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km.

Figure 4.19: Reconstruction with up-to-50-meter positioning error in the locations of sources and receivers using optimization weights for $\lambda_0 = 500$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km.
Figure 4.20: The bisecting line segments at the scatterer \( y_0 \), colored according to the optimization weights with \( \lambda_0 = 10,000 \). The horizontal axis is \( k_1 \) and the vertical axis is \( k_3 \). The units on each axis are rad/km. Note that most bisecting line segments have positive weights.

A regularization parameter results in a weight vector with smaller values. However, \( \lambda_0 = 500 \) successfully form an image of the point scatterer with no corruptions in our computational domain. Even with 10% noise in data, we can still recover the correct location of the scatterer. But the reconstruction doesn’t behave well with positioning error. Next, we choose even larger \( \lambda_0 \) to see if we can get even better results.

### 4.4.5 Reconstruction with \( \lambda_0 = 10,000 \)

Fig. 4.20 shows the bisecting line segments colored by the optimization weights calculated with \( \lambda_0 = 10,000 \). Table 4.4 shows the exact values of the optimization weights. When \( \lambda_0 = 10,000 \), the weight values are even smaller and tend to be positive. Only nine weights are negative, but very small in magnitude. One phenomenon we find in Fig. 4.20 is that the directions with fewer bisectors have higher weight values and the directions with more bisectors have smaller weight values.
Table 4.4: Optimization weights at \( y_0 \) with \( \lambda_0 = 10,000 \).

<table>
<thead>
<tr>
<th>( x_{s1} )</th>
<th>( x_{r1} )</th>
<th>( x_{r2} )</th>
<th>( x_{r3} )</th>
<th>( x_{r4} )</th>
<th>( x_{r5} )</th>
<th>( x_{r6} )</th>
<th>( x_{r7} )</th>
<th>( x_{r8} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3177</td>
<td>0.2514</td>
<td>0.2542</td>
<td>0.2503</td>
<td>0.1905</td>
<td>0.2162</td>
<td>0.4279</td>
<td>0.2722</td>
<td></td>
</tr>
<tr>
<td>0.2427</td>
<td>0.2785</td>
<td>0.3671</td>
<td>0.3153</td>
<td>0.3362</td>
<td>0.4047</td>
<td>0.3694</td>
<td>0.3146</td>
<td></td>
</tr>
<tr>
<td>0.2245</td>
<td>0.2271</td>
<td>0.2511</td>
<td>0.2002</td>
<td>0.2102</td>
<td>0.2384</td>
<td>0.1857</td>
<td>0.1813</td>
<td></td>
</tr>
<tr>
<td>0.0869</td>
<td>0.0738</td>
<td>0.0802</td>
<td>0.0456</td>
<td>0.0329</td>
<td>0.0341</td>
<td>0.0164</td>
<td>0.0119</td>
<td></td>
</tr>
<tr>
<td>0.1295</td>
<td>0.0742</td>
<td>0.0431</td>
<td>0.0613</td>
<td>0.0409</td>
<td>0.0307</td>
<td>0.0064</td>
<td>0.0011</td>
<td></td>
</tr>
<tr>
<td>0.2397</td>
<td>0.1605</td>
<td>0.0982</td>
<td>0.0833</td>
<td>0.0713</td>
<td>0.0647</td>
<td>0.0389</td>
<td>0.0321</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( x_{r9} )</th>
<th>( x_{r10} )</th>
<th>( x_{r11} )</th>
<th>( x_{r12} )</th>
<th>( x_{r13} )</th>
<th>( x_{r14} )</th>
<th>( x_{r15} )</th>
<th>( x_{r16} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2447</td>
<td>0.3499</td>
<td>0.2955</td>
<td>0.2448</td>
<td>0.2935</td>
<td>0.4380</td>
<td>0.3627</td>
<td>0.3511</td>
</tr>
<tr>
<td>0.3549</td>
<td>0.3555</td>
<td>0.3124</td>
<td>0.4192</td>
<td>0.4786</td>
<td>0.4946</td>
<td>0.4567</td>
<td>0.4611</td>
</tr>
<tr>
<td>0.2035</td>
<td>0.1367</td>
<td>0.1475</td>
<td>0.2359</td>
<td>0.1612</td>
<td>0.1054</td>
<td>0.0603</td>
<td>0.0278</td>
</tr>
<tr>
<td>0.0265</td>
<td>0.0215</td>
<td>0.0148</td>
<td>0.0259</td>
<td>0.0462</td>
<td>-0.0014</td>
<td>0.0112</td>
<td>0.0268</td>
</tr>
<tr>
<td>0.0154</td>
<td>0.0073</td>
<td>0.0028</td>
<td>0.0155</td>
<td>0.0355</td>
<td>-0.0266</td>
<td>0.0027</td>
<td>0.0534</td>
</tr>
<tr>
<td>0.0358</td>
<td>0.0167</td>
<td>0.0117</td>
<td>0.0290</td>
<td>0.0453</td>
<td>0.0037</td>
<td>-0.0036</td>
<td>0.0648</td>
</tr>
<tr>
<td>0.0197</td>
<td>0.0124</td>
<td>0.0059</td>
<td>0.0142</td>
<td>0.0173</td>
<td>0.0074</td>
<td>-0.0122</td>
<td>0.0259</td>
</tr>
<tr>
<td>0.0194</td>
<td>0.0119</td>
<td>0.0055</td>
<td>0.0138</td>
<td>0.0164</td>
<td>0.0065</td>
<td>-0.0127</td>
<td>0.0262</td>
</tr>
<tr>
<td>0.0178</td>
<td>0.0088</td>
<td>0.0044</td>
<td>0.0176</td>
<td>0.0415</td>
<td>-0.0221</td>
<td>0.0031</td>
<td>0.0569</td>
</tr>
<tr>
<td>0.1303</td>
<td>0.1218</td>
<td>0.1066</td>
<td>0.1299</td>
<td>0.1473</td>
<td>0.0249</td>
<td>0.0118</td>
<td>0.1219</td>
</tr>
<tr>
<td>0.0218</td>
<td>0.0157</td>
<td>0.0091</td>
<td>0.0175</td>
<td>0.0260</td>
<td>0.0099</td>
<td>-0.0054</td>
<td>0.0247</td>
</tr>
<tr>
<td>0.0521</td>
<td>0.0398</td>
<td>0.0523</td>
<td>0.0211</td>
<td>0.1311</td>
<td>0.0433</td>
<td>0.0458</td>
<td>0.1613</td>
</tr>
<tr>
<td>0.0215</td>
<td>0.0152</td>
<td>0.0087</td>
<td>0.0170</td>
<td>0.0246</td>
<td>0.0100</td>
<td>-0.0066</td>
<td>0.0248</td>
</tr>
<tr>
<td>0.1822</td>
<td>0.2506</td>
<td>0.1875</td>
<td>0.2350</td>
<td>0.3344</td>
<td>0.3553</td>
<td>0.3188</td>
<td>0.2609</td>
</tr>
<tr>
<td>0.0387</td>
<td>0.0321</td>
<td>0.0240</td>
<td>0.0374</td>
<td>0.0721</td>
<td>-0.0141</td>
<td>0.0117</td>
<td>0.0387</td>
</tr>
<tr>
<td>0.0823</td>
<td>0.0775</td>
<td>0.0647</td>
<td>0.0700</td>
<td>0.1154</td>
<td>0.0112</td>
<td>0.0043</td>
<td>0.0980</td>
</tr>
</tbody>
</table>
We have seen from the constant-weights results that choosing the same weight for every direction ruins the resolution in the directions with fewer bisectors. As we see from Fig. 3.4 and Fig. 3.5, intuitively, we hope to assign larger weight values on directions with fewer bisectors to gain good resolution along those directions. The results using optimization weights calculated with \( \lambda_0 = 10,000 \) matches our intuition.

Fig. 4.21 shows the point-spread function at the scatterer using optimization weights for \( \lambda_0 = 10,000 \). It looks pretty similar the point-spread function shown in Fig. 4.11 and Fig. 4.16. These results convince us that we can achieve similar resolution with quite different weight values. We need to do reconstruction experiments to find out which optimization weight function is better for practical use.

Fig. 4.22 shows the reconstruction without noise using optimization weights for \( \lambda_0 = 10,000 \). The point scatterer is well reconstructed with the peak value at the correct location. Fig. 4.23 shows the reconstruction with 10% Gaussian noise in the simulation data. The image just becomes slightly noisier than Fig.
Figure 4.22: Reconstruction without noise using optimization weights for $\lambda_0 = 10,000$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are $km$.

Figure 4.23: Reconstruction with 10% noise in scattered waves using optimization weights for $\lambda_0 = 10,000$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are $km$. 
4.22. Fig. 4.24 shows the reconstruction with up-to-50-meter positioning error in the locations of sources and receivers. Although a couple of bright spots show up in the reconstruction, the highest peak is still located at the right position, which is recognized as the point scatterer. This reconstruction is much better than the reconstruction shown in Fig. 4.19 with $\lambda_0 = 500$.

From the results of this experiment, we find that with $\lambda_0 = 10,000$, we get reconstructions that are much less sensitive to noise and positioning error. These optimization weights can be used in practice.

4.4.6 Reconstruction with $\lambda_0 = 120,000$

Fig. 4.25 shows the bisecting line segments colored by the optimization weights calculated with $\lambda_0 = 120,000$. Table 4.5 shows the exact values of the optimization weights. Note that the optimization weights are all positive and almost constant with $\lambda_0 = 120,000$. Because the value of $\lambda_0$ controls the magnitude of our weight vector $w$, the term $\|w\|^2_2$ becomes dominant in the minimization scheme when $\lambda_0$ is
Table 4.5: Optimization weights at $y_0$ with $\lambda_0 = 120,000$.

<table>
<thead>
<tr>
<th></th>
<th>$x_{r1}$</th>
<th>$x_{r2}$</th>
<th>$x_{r3}$</th>
<th>$x_{r4}$</th>
<th>$x_{r5}$</th>
<th>$x_{r6}$</th>
<th>$x_{r7}$</th>
<th>$x_{r8}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{s1}$</td>
<td>0.0532</td>
<td>0.0532</td>
<td>0.0531</td>
<td>0.0531</td>
<td>0.0530</td>
<td>0.0530</td>
<td>0.0535</td>
<td>0.0531</td>
</tr>
<tr>
<td>$x_{s2}$</td>
<td>0.0531</td>
<td>0.0532</td>
<td>0.0534</td>
<td>0.0533</td>
<td>0.0534</td>
<td>0.0535</td>
<td>0.0534</td>
<td>0.0534</td>
</tr>
<tr>
<td>$x_{s3}$</td>
<td>0.0517</td>
<td>0.0517</td>
<td>0.0514</td>
<td>0.0514</td>
<td>0.0516</td>
<td>0.0518</td>
<td>0.0520</td>
<td>0.0521</td>
</tr>
<tr>
<td>$x_{s4}$</td>
<td>0.0510</td>
<td>0.0511</td>
<td>0.0513</td>
<td>0.0492</td>
<td>0.0489</td>
<td>0.0487</td>
<td>0.0482</td>
<td>0.0476</td>
</tr>
<tr>
<td>$x_{s5}$</td>
<td>0.0518</td>
<td>0.0513</td>
<td>0.0510</td>
<td>0.0504</td>
<td>0.0500</td>
<td>0.0497</td>
<td>0.0481</td>
<td>0.0473</td>
</tr>
<tr>
<td>$x_{s6}$</td>
<td>0.0525</td>
<td>0.0521</td>
<td>0.0516</td>
<td>0.0508</td>
<td>0.0505</td>
<td>0.0503</td>
<td>0.0487</td>
<td>0.0479</td>
</tr>
<tr>
<td>$x_{s7}$</td>
<td>0.0509</td>
<td>0.0508</td>
<td>0.0508</td>
<td>0.0495</td>
<td>0.0491</td>
<td>0.0487</td>
<td>0.0478</td>
<td>0.0471</td>
</tr>
<tr>
<td>$x_{s8}$</td>
<td>0.0509</td>
<td>0.0508</td>
<td>0.0508</td>
<td>0.0496</td>
<td>0.0491</td>
<td>0.0487</td>
<td>0.0478</td>
<td>0.0471</td>
</tr>
<tr>
<td>$x_{s9}$</td>
<td>0.0520</td>
<td>0.0515</td>
<td>0.0511</td>
<td>0.0505</td>
<td>0.0502</td>
<td>0.0498</td>
<td>0.0482</td>
<td>0.0474</td>
</tr>
<tr>
<td>$x_{s10}$</td>
<td>0.0516</td>
<td>0.0514</td>
<td>0.0512</td>
<td>0.0507</td>
<td>0.0506</td>
<td>0.0505</td>
<td>0.0510</td>
<td>0.0513</td>
</tr>
<tr>
<td>$x_{s11}$</td>
<td>0.0509</td>
<td>0.0508</td>
<td>0.0509</td>
<td>0.0494</td>
<td>0.0489</td>
<td>0.0487</td>
<td>0.0479</td>
<td>0.0472</td>
</tr>
<tr>
<td>$x_{s12}$</td>
<td>0.0525</td>
<td>0.0523</td>
<td>0.0522</td>
<td>0.0514</td>
<td>0.0512</td>
<td>0.0511</td>
<td>0.0507</td>
<td>0.0491</td>
</tr>
<tr>
<td>$x_{s13}$</td>
<td>0.0509</td>
<td>0.0508</td>
<td>0.0509</td>
<td>0.0494</td>
<td>0.0490</td>
<td>0.0487</td>
<td>0.0479</td>
<td>0.0472</td>
</tr>
<tr>
<td>$x_{s14}$</td>
<td>0.0529</td>
<td>0.0529</td>
<td>0.0530</td>
<td>0.0530</td>
<td>0.0530</td>
<td>0.0532</td>
<td>0.0535</td>
<td>0.0530</td>
</tr>
<tr>
<td>$x_{s15}$</td>
<td>0.0514</td>
<td>0.0515</td>
<td>0.0517</td>
<td>0.0493</td>
<td>0.0491</td>
<td>0.0491</td>
<td>0.0486</td>
<td>0.0481</td>
</tr>
<tr>
<td>$x_{s16}$</td>
<td>0.0521</td>
<td>0.0521</td>
<td>0.0520</td>
<td>0.0499</td>
<td>0.0498</td>
<td>0.0498</td>
<td>0.0492</td>
<td>0.0491</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$x_{r9}$</th>
<th>$x_{r10}$</th>
<th>$x_{r11}$</th>
<th>$x_{r12}$</th>
<th>$x_{r13}$</th>
<th>$x_{r14}$</th>
<th>$x_{r15}$</th>
<th>$x_{r16}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{s1}$</td>
<td>0.0531</td>
<td>0.0534</td>
<td>0.0533</td>
<td>0.0532</td>
<td>0.0533</td>
<td>0.0536</td>
<td>0.0535</td>
<td>0.0534</td>
</tr>
<tr>
<td>$x_{s2}$</td>
<td>0.0534</td>
<td>0.0534</td>
<td>0.0534</td>
<td>0.0535</td>
<td>0.0536</td>
<td>0.0536</td>
<td>0.0536</td>
<td>0.0536</td>
</tr>
<tr>
<td>$x_{s3}$</td>
<td>0.0521</td>
<td>0.0526</td>
<td>0.0527</td>
<td>0.0528</td>
<td>0.0519</td>
<td>0.0507</td>
<td>0.0487</td>
<td>0.0367</td>
</tr>
<tr>
<td>$x_{s4}$</td>
<td>0.0475</td>
<td>0.0468</td>
<td>0.0467</td>
<td>0.0471</td>
<td>0.0478</td>
<td>0.0421</td>
<td>0.0362</td>
<td>0.0483</td>
</tr>
<tr>
<td>$x_{s5}$</td>
<td>0.0471</td>
<td>0.0450</td>
<td>0.0449</td>
<td>0.0453</td>
<td>0.0436</td>
<td>0.0372</td>
<td>0.0401</td>
<td>0.0494</td>
</tr>
<tr>
<td>$x_{s6}$</td>
<td>0.0477</td>
<td>0.0453</td>
<td>0.0452</td>
<td>0.0456</td>
<td>0.0421</td>
<td>0.0362</td>
<td>0.0418</td>
<td>0.0500</td>
</tr>
<tr>
<td>$x_{s7}$</td>
<td>0.0470</td>
<td>0.0457</td>
<td>0.0456</td>
<td>0.0460</td>
<td>0.0460</td>
<td>0.0402</td>
<td>0.0373</td>
<td>0.0484</td>
</tr>
<tr>
<td>$x_{s8}$</td>
<td>0.0470</td>
<td>0.0457</td>
<td>0.0455</td>
<td>0.0459</td>
<td>0.0459</td>
<td>0.0401</td>
<td>0.0374</td>
<td>0.0484</td>
</tr>
<tr>
<td>$x_{s9}$</td>
<td>0.0472</td>
<td>0.0451</td>
<td>0.0449</td>
<td>0.0454</td>
<td>0.0433</td>
<td>0.0369</td>
<td>0.0405</td>
<td>0.0495</td>
</tr>
<tr>
<td>$x_{s10}$</td>
<td>0.0513</td>
<td>0.0517</td>
<td>0.0518</td>
<td>0.0523</td>
<td>0.0515</td>
<td>0.0495</td>
<td>0.0464</td>
<td>0.0427</td>
</tr>
<tr>
<td>$x_{s11}$</td>
<td>0.0472</td>
<td>0.0461</td>
<td>0.0459</td>
<td>0.0463</td>
<td>0.0466</td>
<td>0.0409</td>
<td>0.0368</td>
<td>0.0483</td>
</tr>
<tr>
<td>$x_{s12}$</td>
<td>0.0482</td>
<td>0.0382</td>
<td>0.0403</td>
<td>0.0461</td>
<td>0.0489</td>
<td>0.0463</td>
<td>0.0476</td>
<td>0.0524</td>
</tr>
<tr>
<td>$x_{s13}$</td>
<td>0.0471</td>
<td>0.0460</td>
<td>0.0458</td>
<td>0.0463</td>
<td>0.0465</td>
<td>0.0408</td>
<td>0.0369</td>
<td>0.0483</td>
</tr>
<tr>
<td>$x_{s14}$</td>
<td>0.0529</td>
<td>0.0531</td>
<td>0.0529</td>
<td>0.0530</td>
<td>0.0533</td>
<td>0.0534</td>
<td>0.0534</td>
<td>0.0532</td>
</tr>
<tr>
<td>$x_{s15}$</td>
<td>0.0481</td>
<td>0.0478</td>
<td>0.0477</td>
<td>0.0482</td>
<td>0.0491</td>
<td>0.0437</td>
<td>0.0362</td>
<td>0.0485</td>
</tr>
<tr>
<td>$x_{s16}$</td>
<td>0.0491</td>
<td>0.0494</td>
<td>0.0494</td>
<td>0.0500</td>
<td>0.0509</td>
<td>0.0467</td>
<td>0.0384</td>
<td>0.0489</td>
</tr>
</tbody>
</table>
Figure 4.25: The bisecting line segments at the scatterer $y_0$, colored according to the optimization weights with $\lambda_0 = 120,000$. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$. Note weights are almost constant.

chosen this large.

Fig. 4.21 shows the point-spread function at the scatterer using optimization weights for $\lambda_0 = 10,000$. Because the optimization weights with $\lambda_0 = 120,000$ are almost constant, the point-spread function has almost the same shape as the one shown in Fig. 4.5 with constant weights, and therefore the resolution is not improved much in this case.

Fig. 4.27 shows the reconstruction without noise using optimization weights for $\lambda_0 = 120,000$. Fig. 4.28 shows the reconstruction with 10% Gaussian noise in the simulation data. Fig. 4.29 shows the reconstruction with up-to-50-meter positioning error in the locations of sources and receivers. The three reconstructions are not very different from the results with constant weights as shown in Fig. 4.7, Fig. 4.8 and Fig. 4.29.
Table 4.6: Optimization weights at $y_0$ with $\lambda_0 = 200,000$.  

<table>
<thead>
<tr>
<th>$x_{s1}$</th>
<th>$x_{r1}$</th>
<th>$x_{r2}$</th>
<th>$x_{r3}$</th>
<th>$x_{r4}$</th>
<th>$x_{r5}$</th>
<th>$x_{r6}$</th>
<th>$x_{r7}$</th>
<th>$x_{r8}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0261</td>
<td>0.0260</td>
<td>0.0260</td>
</tr>
<tr>
<td>$x_{s2}$</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0261</td>
<td>0.0261</td>
<td>0.0261</td>
<td>0.0261</td>
<td>0.0261</td>
<td>0.0261</td>
</tr>
<tr>
<td>$x_{s3}$</td>
<td>0.0257</td>
<td>0.0257</td>
<td>0.0257</td>
<td>0.0257</td>
<td>0.0257</td>
<td>0.0257</td>
<td>0.0258</td>
<td>0.0258</td>
</tr>
<tr>
<td>$x_{s4}$</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0257</td>
<td>0.0253</td>
<td>0.0252</td>
<td>0.0252</td>
<td>0.0251</td>
<td>0.0249</td>
</tr>
<tr>
<td>$x_{s5}$</td>
<td>0.0258</td>
<td>0.0257</td>
<td>0.0256</td>
<td>0.0255</td>
<td>0.0254</td>
<td>0.0254</td>
<td>0.0251</td>
<td>0.0251</td>
</tr>
<tr>
<td>$x_{s6}$</td>
<td>0.0259</td>
<td>0.0258</td>
<td>0.0257</td>
<td>0.0256</td>
<td>0.0255</td>
<td>0.0255</td>
<td>0.0252</td>
<td>0.0252</td>
</tr>
<tr>
<td>$x_{s7}$</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0253</td>
<td>0.0253</td>
<td>0.0252</td>
<td>0.0250</td>
<td>0.0250</td>
</tr>
<tr>
<td>$x_{s8}$</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0254</td>
<td>0.0253</td>
<td>0.0252</td>
<td>0.0250</td>
<td>0.0250</td>
</tr>
<tr>
<td>$x_{s9}$</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0255</td>
<td>0.0255</td>
<td>0.0254</td>
<td>0.0251</td>
<td>0.0251</td>
</tr>
<tr>
<td>$x_{s10}$</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0257</td>
<td>0.0255</td>
<td>0.0255</td>
<td>0.0254</td>
<td>0.0251</td>
<td>0.0251</td>
</tr>
<tr>
<td>$x_{s11}$</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0253</td>
<td>0.0252</td>
<td>0.0252</td>
<td>0.0250</td>
<td>0.0250</td>
</tr>
<tr>
<td>$x_{s12}$</td>
<td>0.0259</td>
<td>0.0259</td>
<td>0.0259</td>
<td>0.0257</td>
<td>0.0257</td>
<td>0.0256</td>
<td>0.0252</td>
<td>0.0252</td>
</tr>
<tr>
<td>$x_{s13}$</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0256</td>
<td>0.0253</td>
<td>0.0252</td>
<td>0.0252</td>
<td>0.0250</td>
<td>0.0250</td>
</tr>
<tr>
<td>$x_{s14}$</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0261</td>
<td>0.0261</td>
</tr>
<tr>
<td>$x_{s15}$</td>
<td>0.0257</td>
<td>0.0257</td>
<td>0.0258</td>
<td>0.0253</td>
<td>0.0253</td>
<td>0.0252</td>
<td>0.0251</td>
<td>0.0251</td>
</tr>
<tr>
<td>$x_{s16}$</td>
<td>0.0258</td>
<td>0.0258</td>
<td>0.0258</td>
<td>0.0254</td>
<td>0.0254</td>
<td>0.0254</td>
<td>0.0253</td>
<td>0.0253</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x_{r9}$</th>
<th>$x_{r10}$</th>
<th>$x_{r11}$</th>
<th>$x_{r12}$</th>
<th>$x_{r13}$</th>
<th>$x_{r14}$</th>
<th>$x_{r15}$</th>
<th>$x_{r16}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0260</td>
<td>0.0261</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0261</td>
<td>0.0261</td>
<td>0.0261</td>
</tr>
<tr>
<td>$x_{s2}$</td>
<td>0.0261</td>
<td>0.0261</td>
<td>0.0261</td>
<td>0.0261</td>
<td>0.0261</td>
<td>0.0261</td>
<td>0.0261</td>
</tr>
<tr>
<td>$x_{s3}$</td>
<td>0.0258</td>
<td>0.0259</td>
<td>0.0259</td>
<td>0.0260</td>
<td>0.0258</td>
<td>0.0256</td>
<td>0.0252</td>
</tr>
<tr>
<td>$x_{s4}$</td>
<td>0.0249</td>
<td>0.0248</td>
<td>0.0248</td>
<td>0.0248</td>
<td>0.0250</td>
<td>0.0238</td>
<td>0.0223</td>
</tr>
<tr>
<td>$x_{s5}$</td>
<td>0.0249</td>
<td>0.0244</td>
<td>0.0244</td>
<td>0.0245</td>
<td>0.0241</td>
<td>0.0226</td>
<td>0.0233</td>
</tr>
<tr>
<td>$x_{s6}$</td>
<td>0.0250</td>
<td>0.0245</td>
<td>0.0245</td>
<td>0.0246</td>
<td>0.0238</td>
<td>0.0223</td>
<td>0.0237</td>
</tr>
<tr>
<td>$x_{s7}$</td>
<td>0.0248</td>
<td>0.0246</td>
<td>0.0245</td>
<td>0.0246</td>
<td>0.0246</td>
<td>0.0234</td>
<td>0.0226</td>
</tr>
<tr>
<td>$x_{s8}$</td>
<td>0.0248</td>
<td>0.0246</td>
<td>0.0245</td>
<td>0.0246</td>
<td>0.0246</td>
<td>0.0233</td>
<td>0.0226</td>
</tr>
<tr>
<td>$x_{s9}$</td>
<td>0.0249</td>
<td>0.0244</td>
<td>0.0244</td>
<td>0.0245</td>
<td>0.0240</td>
<td>0.0225</td>
<td>0.0234</td>
</tr>
<tr>
<td>$x_{s10}$</td>
<td>0.0257</td>
<td>0.0258</td>
<td>0.0258</td>
<td>0.0259</td>
<td>0.0257</td>
<td>0.0253</td>
<td>0.0247</td>
</tr>
<tr>
<td>$x_{s11}$</td>
<td>0.0249</td>
<td>0.0246</td>
<td>0.0246</td>
<td>0.0247</td>
<td>0.0247</td>
<td>0.0235</td>
<td>0.0225</td>
</tr>
<tr>
<td>$x_{s12}$</td>
<td>0.0251</td>
<td>0.0228</td>
<td>0.0233</td>
<td>0.0246</td>
<td>0.0252</td>
<td>0.0247</td>
<td>0.0250</td>
</tr>
<tr>
<td>$x_{s13}$</td>
<td>0.0249</td>
<td>0.0246</td>
<td>0.0246</td>
<td>0.0247</td>
<td>0.0247</td>
<td>0.0235</td>
<td>0.0225</td>
</tr>
<tr>
<td>$x_{s14}$</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0260</td>
<td>0.0261</td>
</tr>
<tr>
<td>$x_{s15}$</td>
<td>0.0250</td>
<td>0.0250</td>
<td>0.0250</td>
<td>0.0251</td>
<td>0.0252</td>
<td>0.0241</td>
<td>0.0223</td>
</tr>
<tr>
<td>$x_{s16}$</td>
<td>0.0252</td>
<td>0.0253</td>
<td>0.0253</td>
<td>0.0254</td>
<td>0.0256</td>
<td>0.0247</td>
<td>0.0229</td>
</tr>
</tbody>
</table>
Figure 4.26: Point spread function at the scatterer $y_0$ using optimization weights for $\lambda_0 = 120,000$. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are km. The units of the point-spread function values are $1/km^2$.

4.4.7 Reconstruction with $\lambda_0 = 200,000$

Now we choose an even larger regularization parameter. Fig. 4.30 shows the bisecting line segments colored by the optimization weights calculated with $\lambda_0 = 200,000$. Table 4.6 shows the exact values of the optimization weights. When $\lambda_0 = 200,000$, the optimization weights are almost constant just like the case as $\lambda_0 = 120,000$. However, the optimization weights have smaller magnitudes because $\|w\|^2$ becomes more dominant in the minimization problem as $\lambda_0$ becomes larger. We expect the optimization weights tend to the constant weights as $\lambda_0$ goes to infinity.

Fig. 4.31 shows the point-spread function at the scatterer using optimization weights for $\lambda_0 = 200,000$. This point-spread function is even closer to the one calculated with constant weights as shown in Fig. 4.6. This is expected because the optimization weights are almost constant. We will not show reconstruction results with the optimization weights since the results are not too different from constant
Figure 4.27: Reconstruction without noise using optimization weights for $\lambda_0 = 120,000$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km.

Weights case just like the results with $\lambda_0 = 120,000$.

Fig. 4.32 shows the plot of all the singular values of matrix $A$ at the scatterer point $y_0 = (4.3, 2.1)$. Note that with different $\lambda_0$, the singular values of matrix $A$ at $y_0$ remain the same. Fig. 4.33 shows the plot of the logarithm of these singular values of matrix $A$ at the scatterer point $y_0 = (4.3, 2.1)$. The five regularization parameters correspond to $\lambda_0 = 1$, $\lambda_0 = 500$, $\lambda_0 = 10,000$, $\lambda_0 = 120,000$ and $\lambda_0 = 200,000$. The size of the regularization parameter relative to the singular values can be easily seen in the two figures. The exact values of these singular values can be found in Table C.1 and Table C.2. Note that both $\lambda_0 = 10,000$ and $\lambda_0 = 200,000$ are larger than any singular value. The optimization weights with $\lambda_0 = 120,000$ and $\lambda_0 = 200,000$ are almost constant. Thus, the point-spread functions with $\lambda_0 = 120,000$ and $\lambda_0 = 200,000$ are almost the same as with the constant weights.
With 10% noise in data ($\lambda_0 = 120,000$)

Figure 4.28: Reconstruction with 10% noise in scattered waves using optimization weights for $\lambda_0 = 120,000$. The horizontal axis is $x_1$ and the vertical axis is $x_3$. The units on each axis are km.

4.4.8 Summary

We have shown point-spread functions and reconstructions with constant weights and optimization weights calculated with $\lambda_0 = 1$, $\lambda_0 = 500$, $\lambda_0 = 10,000$, $\lambda_0 = 120,000$ and $\lambda_0 = 200,000$. Except for the optimization weight function with $\lambda_0 = 10,000$ and $\lambda_0 = 200,000$, the other three optimization weight functions all give better resolution at the scatterer than the constant weights. However, with $\lambda_0 = 1$, the reconstruction away from the scatterer is unacceptable. Both reconstructions for $\lambda_0 = 500$ and $\lambda_0 = 10,000$ result in a better image than the constant-weights case. But the reconstruction with optimization weights for $\lambda_0 = 500$ is much more sensitive to noise and positioning error. The reconstruction with the optimization weights for $\lambda_0 = 10,000$ is much more stable. This is because the weight values have smaller magnitudes which do not amplify the noise or error much. We notice that when $\lambda_0$ is increased, the optimization weights get smaller and tend to be positive. This suggests a criterion we could use to choose $\lambda_0$, namely that $\lambda_0$ should be chosen so that most of the optimization weight values are positive. However, $\lambda_0$ shouldn’t
be too large. As we see from the results with $\lambda_0 = 120,000$ and $\lambda_0 = 200,000$, the optimization weights are almost constant, therefore the point-spread functions with $\lambda_0 = 120,000$ and $\lambda_0 = 200,000$ are almost the same as with the constant weights. The simulation results we show in the next section were calculated from values of $\lambda_0$ chosen using this criterion.

### 4.5 Using Optimization Weights to Improve Resolution

We test the optimization scheme with three different source-receiver geometries by calculating the point-spread functions and half-maximum-value areas at chosen imaging points using optimization weights. We then compare them with the results using constant weights to see how much improvement we gain by optimization weights. Here, we list all the parameters used in the calculation of the optimization weights in Table 4.7.
Figure 4.30: The bisecting line segments at the scatterer $y_0$, colored according to the optimization weights with $\lambda_0 = 200,000$. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are rad/km. Note that weights are almost constant.

Table 4.7: List of parameters. All reconstructions were done with 16 sources and 16 receivers, and the wave speed $c$ was 5km/s.

<table>
<thead>
<tr>
<th>Geometry</th>
<th>Frequency</th>
<th>$z$-computational domain</th>
<th>$\Delta z$</th>
<th>$\lambda_0$</th>
<th>$\lambda_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry 1</td>
<td>5Hz-50Hz</td>
<td>$|z|_{\infty} \leq 0.6$</td>
<td>0.01km</td>
<td>$10^4$</td>
<td>20</td>
</tr>
<tr>
<td>Geometry 2</td>
<td>5Hz-10Hz</td>
<td>$|z|_{\infty} \leq 6$</td>
<td>0.1km</td>
<td>32</td>
<td>20</td>
</tr>
<tr>
<td>Geometry 3</td>
<td>30Hz-50Hz</td>
<td>$|z|_{\infty} \leq 0.6$</td>
<td>0.01km</td>
<td>$10^4$</td>
<td>20</td>
</tr>
</tbody>
</table>

4.5.1 Geometry 1

Fig. 4.34 shows the locations of sources and receivers of geometry 1. In this geometry, sources and receivers are non-equally spaced in the computational domain. This geometry is analogous to the geophysical problem in which a nearly vertical fault is located at around $x = 3.3km$. Micro earthquakes occur frequently at locations near the fault. These low intensity earthquakes can be modelled as point-like wave sources. Geophones are buried near the surface of earth in a non-
uniform way because of the geographical complexities such as hard rocks, lakes and swamps, etc.

We calculate the point-spread functions at the locations represented by green pentagrams and squares shown in Fig. 4.34. In total, $7 \times 5$ imaging points have been chosen, i.e., $7 \times 5$ point-spread functions are shown. The weights for each location are calculated separately. Fig. 4.35 and Fig. 4.36 show the point-spread functions with constant weights and optimization weights respectively. The $7 \times 5$ point-spread functions in these figures are placed in the same manner as they are located in Fig. 4.34, i.e., the $x_1$ coordinate of the corresponding imaging point increases from left to right, and the $x_3$ coordinate increases from top to bottom. The adjacent imaging points are separated by 1$km$ in both horizontal and vertical directions. Point-spread functions will be placed in the same way for other geometries that will be shown later.

Comparing the results with both weight functions, we find that the point-
Figure 4.32: Singular values of matrix \( A \) at \( y_0 \). The five lines correspond to the five regularization parameters \( \lambda_0 = 1 \), \( \lambda_0 = 500 \), \( \lambda_0 = 10,000 \), \( \lambda_0 = 120,000 \) and \( \lambda_0 = 200,000 \). The \( \lambda_0 = 1 \) line and the \( \lambda_0 = 500 \) line are very close to the horizontal axis.

Figure 4.33: Logarithm of the singular values of matrix \( A \) at \( y_0 \). The five lines correspond to the logarithm of the five regularization parameters \( \lambda_0 = 1 \), \( \lambda_0 = 500 \), \( \lambda_0 = 10,000 \), \( \lambda_0 = 120,000 \) and \( \lambda_0 = 200,000 \).
Figure 4.34: Sources (red stars) and receivers (blue circles) locations of geometry 1. The horizontal axis is $y_1$ and the vertical axis is $y_3$. Green pentagrams and squares represent the chosen imaging points, where optimization weights and point-spread functions are calculated. Bisecting line segments plots are shown only at green pentagram locations.

Spread functions with the optimization weights look much better than the ones calculated with constant weights. Especially for the imaging points close to the center of our computational domain, point-spread functions with constant weights are more blurry and stretched, while for the case with optimization weights, the point-spread functions are sharper and more focused. Optimization weights improve the point-spread function in this geometry.

We quantify how resolution is improved by the half-maximum-value area as shown in Fig. 4.37. We show results for $7 \times 5$ imaging points designated by either green pentagrams or squares in Fig. 4.34. We plot the half-maximum-value curves
Figure 4.35: Plots of the point-spread functions of geometry 1 using constant weights. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are km. The units of the point-spread function values are $1/km^2$. The locations correspond to the $7 \times 5$ different imaging points designated by green pentagrams and squares in Fig. 4.34.

Figure 4.36: Plots of point-spread functions of geometry 1 using optimization weights. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are km. The units of the point-spread function values are $1/km^2$. The locations are the same as those in Fig. 4.35.
Figure 4.37: Half-maximum-value contours of the point-spread functions calculated with optimization weights (the contours filled with black) and constant weights (the outer contours) of geometry 1. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are $km$. The $7 \times 5$ different imaging points correspond to both green pentagrams and squares in Fig. 4.34. The number shown outside the parenthesis is half-maximum-value area with optimization weights. The number shown inside the parenthesis is half-maximum-value area with constant weights.

of point-spread functions calculated with both optimization weights (the contour filled with black) and constant weights (the outer contour). The half-maximum-value area is the area of the region enclosed by the half-maximum-value contour. The half-maximum-value area with the optimization weights is shown outside the parenthesis. The half-maximum-value area with the constant weights is shown inside the parenthesis. In general, the smaller the area is, the better the resolution we obtain. For example, for the imaging point shown at the center, the half-maximum-value area for constant weights is $15102m^2$, and using the optimization weights instead improve it to $1649m^2$. This point is a good example to illustrate the fact that constant weights could result in very poor resolution, while optimization weights can improve the resolution dramatically. This is true not only for the center point but also for the other points.
Figure 4.38: Bisecting line segments plots of $4 \times 3$ different imaging points corresponding to the green pentagrams in Fig. 4.34 of geometry 1. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $\text{rad/km}$.

In Fig. 4.37, we can roughly see the resolution of an imaging point in every direction. One way to quantify resolution in one direction is to calculate the diameter of the half-maximum-value contour in that direction. For example, for the imaging point at the center of Fig. 4.37, the half-maximum-value contour from optimization weights has an ellipse shape. The best resolution is in the direction of the minor axis which is about 30 meters. The worst resolution is in the direction of the major axis which is about 70 meters. Note that in this example, the minimum wavelength is 100 meters. Thus we obtain sub-wavelength resolution, which is roughly consistent with the resolution limit of one fourth the minimum wavelength [1].

Fig. 4.38 shows the bisecting line segments at the $4 \times 3$ imaging points corresponding to the green pentagrams in Fig. 4.34. Fig. 4.39 shows the same bisecting line segments colored according to the optimization weights. In the two figures, the adjacent imaging points are separated by $2 \text{km}$ in both horizontal and vertical directions. The directions with larger and smaller weights can be easily seen from these two figures.
4.5.2 Geometry 2

We have shown results for geometry 1, where sources and receivers are non-equally spaced. Our calculated optimization weights enable us to combine the information contained in each source-receiver pair in a proper way to achieve better resolution.

This method applies also to the case of a regular grid. For example, for geometry 2 as shown in Fig. 4.40, sources and receivers are evenly located on the top of our computational region. If the data is densely sampled, then the recorded data is three dimensional, where one degree of freedom is from frequency and the other two are from parameterizing the source manifold and the receiver manifold respectively. Since we are only forming a two dimensional image of $\alpha(y)$, the dimension of the data exceeds the dimension of the image. But on the other hand, the more information we have, the better image we can potentially reconstruct. We can use our method to choose optimization weights to combine all the data to achieve better resolution.

For this geometry, we use the lower frequency band 5Hz-10Hz, which in the
Figure 4.40: Sources and receivers locations of geometry 2. Sources and receivers have exact same locations at blue circles. The horizontal axis is $y_1$ and the vertical axis is $y_3$. Green pentagrams and squares represent the chosen imaging points, where optimization weights and point-spread functions are calculated. Bisecting line segments plots are shown only at green pentagram locations.

The geophysics case corresponds to using less expensive narrowband receivers.

Fig. 4.41 and Fig. 4.42 show the point-spread functions calculated with constant weights and optimization weights respectively. In this experiment, the optimization weights improve the point-spread functions: they become more focused and peaky than in Fig. 4.41. If we compare Fig. 4.41 and Fig. 4.42 with Fig. 4.36 from geometry 1, the point-spread functions in this example become much wider in all directions. This is not because the source-receiver geometry in this example degrades the resolution. Rather, it is because we have used a much lower frequency
Figure 4.41: Plots of point-spread functions of geometry 2 using constant weights. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are km. The units of the point-spread function values are $1/km^2$. Again the locations correspond to the $7 \times 5$ points marked by green pentagrams and squares in Fig. 4.40.

Figure 4.42: Plots of point-spread functions of geometry 2 using optimization weights. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are km. The units of the point-spread function values are $1/km^2$. The locations are the same as those in Fig. 4.41.
in geometry 2 than in geometry 1. This implies that the bisecting line segments in this case are much shorter, and consequently the data-collection manifold in k-space becomes much smaller. In fact, the point-spread function can be considered as the fourier transform of a characteristic function supported on the data-collection manifold, or mathematically \( \int_{\mathbb{R}^3} F(\mathbf{k})e^{-i\mathbf{k} \cdot \mathbf{x}}d\mathbf{k} \), where \( F(\mathbf{k}) \) is the characteristic function of the data-collection manifold. For the high-frequency case, \( F(\mathbf{k}) \) covers a broader area. Consequently, to produce an image with better resolution, higher frequencies should be used. Moreover, if we want to form an image with good resolution in all directions, we need to record data so that the bisecting directions cover all directions from the imaging point.

Fig. 4.43 shows the half-maximum-value contours and half-maximum-value areas at 7 × 5 imaging points corresponding to both green pentagrams and squares.
Figure 4.44: Bisecting line segments plots of $4 \times 3$ different imaging points corresponding to the green pentagrams in Fig. 4.40 of geometry 2. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are rad/km.

in Fig. 4.40. We plot the half-maximum-value contours calculated with optimization weights (the contours filled with black) and constant weights (the outer contours). The half-maximum-value areas with the optimization weights are shown outside the parentheses. The half-maximum-value areas with constant weights are shown inside the parentheses. Although the optimization weights don’t decrease the half-maximum-value area much in this example, the shape of point-spread function is actually improved a lot in some directions. Again, these results show that lower frequency results in lower resolution, as compared with results shown in Fig. 4.37 from geometry 1.

Fig. 4.44 shows the bisecting line segments at the $4 \times 3$ imaging points corresponding to the green pentagrams in Fig. 4.40. Fig. 4.45 shows the same bisecting line segments colored according to the optimization weights.

4.5.3 Geometry 3

Fig. 4.34 shows the locations of sources and receivers of geometry 3. This source-receiver geometry corresponds to cross-well seismic experiments in geophysics.
As in geometry 2, the dimension of the data exceeds the dimension of the image. Thus, we can use optimization weights to improve resolution. In this example, we consider frequency band 30Hz-50Hz.

Fig. 4.47 and Fig. 4.48 show the point-spread functions calculated with constant weights and optimization weights respectively.

Fig. 4.49 shows the half-maximum-value contours and half-maximum-value areas at 7×5 imaging points corresponding to both green pentagrams and squares in Fig. 4.46. We plot the half-maximum-value contours calculated with optimization weights (the contours filled with black) and constant weights (the outer contours). The half-maximum-value area with optimization weights is shown outside the parenthesis. The half-maximum-value area with constant weights is shown inside the parenthesis.

We notice that resolution is much better in the vertical directions than the horizontal directions. This is because most bisectors point almost vertically in this type of source-receiver geometry. In order to get good resolution in the horizon-
Figure 4.46: Sources (red stars) and receivers (blue circles) locations of geometry 3. The horizontal axis is $y_1$ and the vertical axis is $y_3$. Green pentagrams and squares represent the chosen imaging points, where optimization weights and point-spread functions are calculated. Bisecting line segments plots are shown only at green pentagram locations.

tal direction, we need horizontal bisectors which are not present for this example. Consequently, this source-receiver geometry is not a good choice to image vertical structures.

Fig. 4.50 shows the bisecting line segments at the $4 \times 3$ imaging points corresponding to the green pentagrams in Fig. 4.46. Fig. 4.51 shows the same bisecting line segments colored according to the optimization weights. Note that the unit length for horizontal direction and vertical direction are quite different in the two figures. In fact, the bisecting line segments are more inclined to the vertical direction than the horizontal direction.
Figure 4.47: Plots of point-spread functions of geometry 3 using constant weights. The units on each axis are km. The units of the point-spread function values are $1/km^2$. Again the locations correspond to the $7 \times 5$ imaging points designated by both green pentagrams and squares in Fig. 4.46.

Figure 4.48: Plots of point-spread functions of geometry 3 using optimization weights. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are km. The units of the point-spread function values are $1/km^2$. The locations are the same as those in Fig. 4.47.
Figure 4.49: Half-maximum-value contours of point-spread functions calculated with optimization weights (the contours filled with black) and constant weights (the outer contours) of geometry 3. The horizontal axis is $z_1$ and the vertical axis is $z_3$. The units on each axis are km. The $7 \times 5$ different imaging points correspond to both green pentagrams and squares in Fig. 4.46. The number shown outside the parenthesis is the half-maximum-value area with optimization weights. The number shown inside the parenthesis is half-maximum-value area with constant weights.

4.5.4 Summary

In this section, we have shown point-spread functions at different imaging points using different source-receiver geometries. Different frequency bands have also been considered for different geometries. We have shown simulations to illustrate how resolution is affected by the k-space data-collection manifold. We also have shown plots of the optimization weights at the chosen imaging points in these examples. The directions with high and low weights at these points can be easily seen. Using optimization weights improves resolution relative to the trivial constant weights. Actually, the degree of the improvement can be predicted from the bisecting-line-segments plot. For example, in geometry 1 (see Fig. 4.38), where bisecting line segments are not well distributed in k-space at some imaging points, we anticipate the constant weights will perform poorly. Thus, the optimization weights
Figure 4.50: Bisecting line segments plots of $4 \times 3$ different imaging points corresponding to the green pentagrams in Fig. 4.46 of geometry 3. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$.

Figure 4.51: Bisecting line segments plots of $4 \times 3$ different imaging points corresponding to the green pentagrams in Fig. 4.46 of geometry 3. Each bisecting line segment is colored according to the optimization weights. The horizontal axis is $k_1$ and the vertical axis is $k_3$. The units on each axis are $rad/km$. 
dramatically improve resolution at those points. But in geometry 2 and geometry 3 (see Fig. 4.44 and Fig. 4.50), the bisecting line segments are more evenly distributed. In this situation, the constant weights work relatively well, which indicates that we don’t obtain much improvement in resolution with the optimization weights.
CHAPTER 5
Conclusion and Future Work

Our method provides a way to calculate optimization weights that improve resolution relative to reconstructions using constant weights. In order to gain stability in the reconstruction, we need to choose regularization parameters properly so that optimization weight values are mostly positive. We have also investigated how frequency affects resolution. In order to reconstruct small objects, we need to use high frequencies. Resolution is also affected by source-receiver geometry, and a given geometry can have quite different resolution in different directions. If we have prior information about the objects, we can choose the source-receiver geometry to form better images.

Many issues remain for the future. First, we need to apply our optimization weight function to real data.

One problem we notice in Fig. 4.12 with large magnitude weights is the corruption of reconstruction at imaging points far apart from the center of the scatterer. The reconstruction with other weight functions do not show the same problem, probably due to the small computational domain we have chosen. We need to investigate if the same problem comes up when the computational domain is big enough. If this is indeed a problem, it could perhaps be dealt with by, for example, splitting up the computational domain into small pieces and treating each piece separately.

In our optimization problem, we found that choosing $\lambda_0$ large has several effects: it makes the backprojection stable, it makes most of the weights positive, and it decreases the magnitudes of the weights. Alternatively, we can try adding in a non-negativity constraint to our minimization problem. With this inequality constraint, the minimization problem can be solved by an iterative method. In order to compare the resulting non-negative weight function with the one produced without a non-negativity constraint, an analysis of the errors in the numerical backprojection needs to be done.

In this paper, we have tried a few different regularization parameters in our
simulations. But we don’t have an approach to determine which value of the parameter is the “best”.

So far, our simulations were done with real signals, which reduce our minimization problem to a real-valued problem. However, there are applications that use analytic signals, such as MRI and radar problems. In this situation, only the positive frequency band needs to be considered, which implies that optimization weight function could be complex. We need to investigate stability issue of backprojection reconstruction with complex weight functions to get the answer.

One assumption we have made in our model is that the weight function is independent of frequency. But allowing it to depend on frequency means that the search space is much larger. This implies that the resulting weight function should be better, but it also increases the number of variables in our problem, and therefore increases the computational cost in solving the minimization problem.

We can also tweak our minimization problem by choosing a different set of test functions. The size of our problem not only depends on the number of source-receiver pairs, but also depends on the number of test functions. If we can take fewer test functions to achieve the same improvement of the point-spread function and stability in the backprojection reconstruction, we can decrease the computational cost.

Another possible direction is to try a different norm. We take the $L^2$ norm in our formulation to reduce our minimization problem to solving a linear system. With other norms, we have to solve the minimization problem by iterative methods. One question is: does any other norm give better results? The $L^1$ norm, with its connection to sparsity of the reconstruction, would be a particularly interesting choice.
LITERATURE CITED


APPENDIX A
Approximation of Green’s Function by the Eikonal Equation and Transport Equation

For high-frequency problems, we can approximate the Helmholtz Green’s function using ray theory. The incident wave can be represented by the WKBJ series,

\[
    u(x, \omega) \simeq \omega^\beta e^{i\omega \tau(x)} \sum_{j=0}^{\infty} \frac{A_j(x)}{(i\omega)^j}
\]

(A.1)

where \( \tau(x) \) is travel time and \( A_j(x) \)'s are amplitude functions. We substitute (A.1) into the homogeneous Helmholtz equation

\[
    \nabla^2_x u(x, \omega) + \frac{\omega^2}{c^2(x)} u(x, \omega) = 0,
\]

(A.2)

where \( c(x) \) is the smooth background wave speed. We get

\[
    \omega^\beta \nabla^2_x (e^{i\omega \tau(x)} \sum_{j=0}^{\infty} \frac{A_j(x)}{(i\omega)^j}) - \omega^\beta e^{i\omega \tau(x)} \sum_{j=0}^{\infty} \frac{A_j(x)}{(i\omega)^j-2} = 0.
\]

(A.3)

The gradient field of \( e^{i\omega \tau(x)} \sum_{j=0}^{\infty} \frac{A_j(x)}{(i\omega)^j} \) is

\[
    \nabla_x (e^{i\omega \tau(x)} \sum_{j=0}^{\infty} \frac{A_j(x)}{(i\omega)^j}) = e^{i\omega \tau(x)} \nabla_x \tau(x) \sum_{j=0}^{\infty} \frac{A_j(x)}{(i\omega)^j-1} + e^{i\omega \tau(x)} \sum_{j=0}^{\infty} \frac{\nabla_x A_j(x)}{(i\omega)^j}.
\]

(A.4)
The Laplacian of \( e^{i \omega \tau(x)} \sum_{j=0}^{\infty} \frac{A_j(x)}{(i \omega)^j} \) is

\[
\nabla_x^2 (e^{i \omega \tau(x)} \sum_{j=0}^{\infty} \frac{A_j(x)}{(i \omega)^j}) = e^{i \omega \tau(x)} \| \nabla_x \tau(x) \|^2 \sum_{j=0}^{\infty} \frac{A_j(x)}{(i \omega)^j} + 2e^{i \omega \tau(x)} \sum_{j=0}^{\infty} \frac{\nabla_x^2 A_j(x)}{(i \omega)^j},
\]

\[
+ 2e^{i \omega \tau(x)} \sum_{j=0}^{\infty} \frac{\nabla_x \tau(x) \cdot \nabla_x A_j(x)}{(i \omega)^j} \sum_{j=0}^{\infty} \frac{A_j(x)}{(i \omega)^j}.
\]

We substitute (A.5) into (A.3) to obtain

\[
\| \nabla_x \tau(x) \|^2 \sum_{j=0}^{\infty} \frac{A_j(x)}{(i \omega)^j} + \nabla_x^2 \tau(x) \sum_{j=0}^{\infty} \frac{A_j(x)}{(i \omega)^j} - 2 \sum_{j=0}^{\infty} \frac{\nabla_x \tau(x) \cdot \nabla_x A_j(x)}{(i \omega)^j} + 2 \sum_{j=0}^{\infty} \frac{\nabla_x^2 A_j(x)}{(i \omega)^j} = 0. \quad (A.6)
\]

To solve (A.6), we set the coefficients of each powers of \( \omega \) to zero. For example, equating the coefficient of \( \omega^2 \) to zero yields

\[
\| \nabla_x \tau(x) \|^2 A_0(x) - \frac{1}{c^2(x)} A_0(x) = 0. \quad (A.7)
\]

Here, choosing \( A_0(x) \neq 0 \) to get a nontrivial solution, we derive Eikonal equation

\[
\| \nabla_x \tau(x) \|^2 = \frac{1}{c^2(x)}. \quad (A.8)
\]

We use the Eikonal equation in (A.6) to obtain the simplified equation

\[
\nabla_x^2 \tau(x) \sum_{j=0}^{\infty} \frac{A_j(x)}{(i \omega)^j} + 2 \sum_{j=0}^{\infty} \frac{\nabla_x \tau(x) \cdot \nabla_x A_j(x)}{(i \omega)^j} + \sum_{j=0}^{\infty} \frac{\nabla_x^2 A_j(x)}{(i \omega)^j} = 0. \quad (A.9)
\]

Next step, to make the coefficient of \( \omega \) vanish, we solve

\[
\nabla_x^2 \tau(x) A_0(x) + 2 \nabla_x \tau(x) \cdot \nabla_x A_0(x) = 0. \quad (A.10)
\]
This equation is called a transport equation. Note that in order to solve transport equation for \( A_0(\mathbf{x}) \), the Eikonal equation needs to be solved first to get \( \tau(\mathbf{x}) \). We follow the same procedure for the coefficient of \( \omega^j \) and find that \( A_j(\mathbf{x}) \) for \( j \geq 1 \) is governed by

\[
\nabla_x^2 \tau(\mathbf{x}) A_j(\mathbf{x}) + 2 \nabla_x \tau(\mathbf{x}) \cdot \nabla_x A_j(\mathbf{x}) + \nabla_x^2 A_{j-1}(\mathbf{x}) = 0, \tag{A.11}
\]

or, equivalently,

\[
\nabla_x \cdot (A_j^2(\mathbf{x}) \nabla_x \tau(\mathbf{x})) = -\nabla_x^2 A_{j-1}(\mathbf{x}). \tag{A.12}
\]

Thus, \( \{ A_j(\mathbf{x}) \}^{\infty}_{j=0} \) can be obtained by recursively solving (A.10) and (A.11) or (A.12).

In general, the Eikonal equation and transport equations have an infinite number of solutions. However, we only look for the particular solution that gives the Green’s function. For the case of constant wave speed \( c(\mathbf{x}) = c \) in 3-D, \( \tau(\mathbf{x}) = \frac{|\mathbf{x}|}{c} \), \( A_0(\mathbf{x}) = \frac{1}{4\pi |\mathbf{x}|} \) and \( A_j(\mathbf{x}) = 0 \) for \( j \geq 1 \) are solutions to the Eikonal equation and transport equations. It can be proved that

\[
u(\mathbf{x}, \omega) = \frac{e^{-i\omega \frac{|\mathbf{x}|}{c}}}{4\pi |\mathbf{x}|} \tag{A.13}\]

is actually the weak solution to the Helmholtz equation. Therefore, it is the Green’s function.

In a varying wave speed scenario, we approximate the Green’s function by

\[
G_0(\mathbf{x}, \omega) = A_0(\mathbf{x}) e^{i\omega \tau(\mathbf{x})}. \tag{A.14}\]
APPENDIX B

Scaling the Helmholtz Equation

Let \( \bar{c} > 0 \) be a constant scaling factor. In this thesis we choose it to be the average of the background velocity in the computational domain \( \Omega \), i.e.,

\[
\bar{c} = \frac{\int_{\Omega} c(x) dx}{\int_{\Omega} d}\text{.} 
\]  

We scale the actual velocity \( v(x) \), background velocity \( c(x) \) and frequency \( \omega \) by \( \frac{1}{\bar{c}} \). Define

\[
\begin{align*}
\tilde{v}(x) &= \frac{v(x)}{\bar{c}}, \\
\tilde{c}(x) &= \frac{c(x)}{\bar{c}}, \\
\tilde{\omega} &= \frac{\omega}{\bar{c}}. 
\end{align*}
\]

We have defined the perturbation function \( \alpha(x) \) by formula

\[
\frac{1}{v^2(x)} = \frac{1}{c^2(x)}(1 + \alpha(x)),
\]

which implies the new perturbation function \( \tilde{\alpha}(x) \) satisfies

\[
\tilde{\alpha}(x) = \alpha(x). 
\]

Replace the \( v(x) \), \( \omega \), by \( \tilde{v}(x) \), \( \tilde{\omega} \) in Helmholtz equation

\[
\nabla^2_{x} u(x, x_s, \omega) + \frac{\omega^2}{v^2(x)} u(x, x_s, \omega) = -\delta(x - x_s),
\]

to get

\[
\nabla^2_{x} u(x, x_s, \tilde{c}\tilde{\omega}) + \frac{\tilde{\omega}^2}{\tilde{v}^2(x)} u(x, x_s, \tilde{c}\tilde{\omega}) = -\delta(x - x_s)
\]
Define
\[ \tilde{u}(x, x_s, \tilde{\omega}) = u(x, x_s, \tilde{\omega}). \] \hspace{1cm} (B.9)

We get the scaled Helmholtz equation
\[ \nabla^2_x \tilde{u}(x, x_s, \tilde{\omega}) + \frac{\tilde{\omega}^2}{\tilde{v}^2(x)} \tilde{u}(x, x_s, \tilde{\omega}) = -\delta(x - x_s). \] \hspace{1cm} (B.10)

Note that the only difference between (B.10) and (B.7) is in notation. This means we only need to replace \( u, \omega \) and \( v \) in the results from (B.7) by \( \tilde{u}, \tilde{\omega} \) and \( \tilde{v} \) to get results from (B.10). Now, the scaled the Eikonal equation from (B.10) is
\[ \| \nabla_x \tilde{\tau}(x) \| \frac{1}{c^2(x)} = 1, \] \hspace{1cm} (B.11)
\[ \Rightarrow \| \nabla_x \left( \frac{\tilde{\tau}(x)}{c} \right) \| \frac{1}{c^2(x)} = 1, \] \hspace{1cm} (B.12)
which implies
\[ \tilde{\tau}(x) = \bar{c} \tau(x). \] \hspace{1cm} (B.13)

The scaled transport equation is
\[ \| \nabla_x \tilde{\tau}(x) \| \frac{1}{c^2(x)} \tilde{A}_0(x) - \frac{1}{c^2(x)} \tilde{A}_0(x) = 0, \] \hspace{1cm} (B.14)
\[ \Rightarrow \| \nabla_x \tau(x) \| \frac{1}{c^2(x)} A_0(x) - \frac{1}{c^2(x)} A_0(x) = 0. \] \hspace{1cm} (B.15)
This implies
\[ \tilde{A}_0(x) = A_0(x). \] \hspace{1cm} (B.16)

By (B.13) and (B.16), we conclude
\[ \tilde{\phi}(x_s, x, x_r) = \bar{c} \phi(x_s, x, x_r), \] \hspace{1cm} (B.17)
\[ \tilde{a}(x_s, x, x_r) = a(x_s, x, x_r). \] \hspace{1cm} (B.18)
The scaled scattered wave \( \tilde{u}_D(x_r, x_s, \tilde{\omega}) \) satisfies
\[
\tilde{u}_D(x_r, x_s, \tilde{\omega}) = \tilde{\omega}^2 \int_{\mathbb{R}^3} \frac{\partial(y)}{c^2(y)} \tilde{a}(x_s, y, x_r) e^{i\tilde{\omega}\phi(x_s, y, x_r)} dy, \quad (B.19)
\]
\[
= \omega^2 \int_{\mathbb{R}^3} \frac{\alpha(y)}{c^2(y)} a(x_s, y, x_r) e^{i\omega\phi(x_s, y, x_r)} dy, \quad (B.20)
\]
\[
= u^s_D(x_r, x_s, \omega). \quad (B.21)
\]

We assume \((x_s, x_r)\) can be parameterized by two dimensional variable \(\sigma\). The scaled Beylkin determinant \(\tilde{h}(x, \sigma)\) is
\[
\tilde{h}(x, \sigma) = \det \begin{pmatrix} \nabla_x \tilde{\phi}(x, \sigma) \\ \frac{\partial}{\partial \sigma_1} \nabla_x \tilde{\phi}(x, \sigma) \\ \frac{\partial}{\partial \sigma_2} \nabla_x \tilde{\phi}(x, \sigma) \end{pmatrix}, \quad (B.22)
\]
\[
= \det \begin{pmatrix} \tilde{c} \nabla_x \phi(x, \sigma) \\ \tilde{c} \frac{\partial}{\partial \sigma_1} \nabla_x \phi(x, \sigma) \\ \tilde{c} \frac{\partial}{\partial \sigma_2} \nabla_x \phi(x, \sigma) \end{pmatrix}, \quad (B.23)
\]
\[
= \tilde{c}^3 h(x, \sigma). \quad (B.24)
\]

The new point spread function \(\tilde{K}(x, y)\) satisfies
\[
\tilde{K}(x, y) = \frac{1}{(2\pi)^3} \int |\tilde{h}(x, \sigma)| \int \tilde{\omega}^2 e^{-i\tilde{\omega}\nabla_x \tilde{\phi}(x, \sigma) \cdot (x-y)} d\tilde{\omega} d\sigma, \quad (B.25)
\]
\[
= \frac{1}{(2\pi)^3} \int |h(x, \sigma)| \int \omega^2 e^{-i\omega\nabla_x \phi(x, \sigma) \cdot (x-y)} d\omega d\sigma, \quad (B.26)
\]
\[
= K(x, y). \quad (B.27)
\]

The new image \(\tilde{I}(x)\) satisfies
\[
\tilde{I}(x) = \int \int \frac{c^2(x)}{(2\pi)^3} |\tilde{h}(x, \sigma)| |\tilde{u}^s_D(\sigma, \tilde{\omega})| e^{-i\tilde{\omega}\phi(x, \sigma)} d\tilde{\omega} d\sigma, \quad (B.28)
\]
\[
= \int \int \frac{c^2(x)}{(2\pi)^3} a(x, \sigma) u^s_D(\sigma, \omega) e^{-i\omega\phi(x, \sigma)} d\omega d\sigma, \quad (B.29)
\]
\[
= I(x). \quad (B.30)
\]
We conclude that scaling the problem by $\frac{1}{c}$ doesn’t change the point-spread function or image. However, numerically we don’t know how to calculate $\hat{h}(x, \sigma)$ and we can’t carry out the $\sigma$ integration, because the sources and receivers are non-equally spaced. But we can approximate them in the following way,

$$
\hat{K}_{\hat{W}}(x, y) \approx \frac{1}{(2\pi)^3} \sum_{\sigma} \hat{W}(x, \sigma) \int \hat{\omega}^2 e^{-i\hat{\omega} \nabla_{x} \hat{\phi}(x, \sigma) \cdot (x-y)} d\hat{\omega} 
$$

(B.31)

$$
= \frac{1}{(2\pi \bar{c})^3} \sum_{\sigma} \tilde{W}(x, \sigma) \int \omega^2 e^{-i\omega \nabla_{x} \phi(x, \sigma) \cdot (x-y)} d\omega. 
$$

(B.32)

Since $\hat{K}_{\hat{W}}(x, y)$ is an approximation of $\hat{K}(x, y)$ and $\hat{K}(x, y) = K(x, y)$, we want $\hat{K}_{\hat{W}}(x, y) = K_W(x, y)$. This is the case if we choose

$$
\hat{W}(x, \sigma) = \bar{c}^3 W(x, \sigma).
$$

(B.33)

We can calculate the inversion formula similarly.

$$
\hat{I}_{\hat{W}}(x) \approx \frac{1}{(2\pi)^3} \sum_{\sigma} \hat{W}(x, \sigma) \int \frac{\hat{c}^2(x)}{\hat{a}(x, \sigma)} \hat{U}_D(\sigma, \hat{\omega}) e^{-i\hat{\omega} \hat{\phi}(x, \sigma)} d\hat{\omega}, 
$$

(B.34)

$$
= \frac{1}{(2\pi \bar{c})^3} \sum_{\sigma} \hat{W}(x, \sigma) \int \frac{c^2(x)}{a(x, \sigma)} u_D^s(\sigma, \omega) e^{-i\omega \phi(x, \sigma)} d\omega. 
$$

(B.35)

Note that $\hat{I}_{\hat{W}}(x) = I_W(x)$ because of (B.33).

In (3.17) and (3.21), we are safe to change the notation from $\hat{K}_{\hat{W}}(x, y)$, $\hat{I}_{\hat{W}}(x)$ and $\hat{W}(x, \sigma)$ to $K_W(x, y)$, $I_W(x)$ and $W(x, \sigma)$ because $K_W(x, y) = \hat{K}_{\hat{W}}(x, y)$ and $I_W(x) = \hat{I}_{\hat{W}}(x)$. But the $W(x, \sigma)$ and $\hat{W}(x, \sigma)$ are different by the scaling factor $\frac{1}{c \sigma}$.

We have this factor in both equations to account for this. However, we should keep in mind that the optimization weight function we calculate is actually $\bar{c}^3 W(x, \sigma)$ instead of $W(x, \sigma)$ itself.
### APPENDIX C

List of the Singular Values

Table C.1: List of the singular values $s_1$ to $s_{116}$ of matrix $A$ at $y_0$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$s_n$</th>
<th>$n$</th>
<th>$s_n$</th>
<th>$n$</th>
<th>$s_n$</th>
<th>$n$</th>
<th>$s_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1472e+005</td>
<td>30</td>
<td>1.1458e+004</td>
<td>59</td>
<td>7.0802e+003</td>
<td>88</td>
<td>3.9913e+003</td>
</tr>
<tr>
<td>2</td>
<td>7.0470e+004</td>
<td>31</td>
<td>1.1420e+004</td>
<td>60</td>
<td>6.9597e+003</td>
<td>89</td>
<td>3.9224e+003</td>
</tr>
<tr>
<td>3</td>
<td>5.1586e+004</td>
<td>32</td>
<td>1.1273e+004</td>
<td>61</td>
<td>6.9006e+003</td>
<td>90</td>
<td>3.8856e+003</td>
</tr>
<tr>
<td>4</td>
<td>4.0781e+004</td>
<td>33</td>
<td>1.1071e+004</td>
<td>62</td>
<td>6.8574e+003</td>
<td>91</td>
<td>3.7980e+003</td>
</tr>
<tr>
<td>5</td>
<td>3.6210e+004</td>
<td>34</td>
<td>1.0942e+004</td>
<td>63</td>
<td>6.5790e+003</td>
<td>92</td>
<td>3.7705e+003</td>
</tr>
<tr>
<td>6</td>
<td>3.4606e+004</td>
<td>35</td>
<td>1.0631e+004</td>
<td>64</td>
<td>6.3594e+003</td>
<td>93</td>
<td>3.5663e+003</td>
</tr>
<tr>
<td>7</td>
<td>3.2112e+004</td>
<td>36</td>
<td>1.0589e+004</td>
<td>65</td>
<td>6.3228e+003</td>
<td>94</td>
<td>3.4742e+003</td>
</tr>
<tr>
<td>8</td>
<td>3.1222e+004</td>
<td>37</td>
<td>1.0504e+004</td>
<td>66</td>
<td>6.1306e+003</td>
<td>95</td>
<td>3.4697e+003</td>
</tr>
<tr>
<td>9</td>
<td>2.9716e+004</td>
<td>38</td>
<td>1.0045e+004</td>
<td>67</td>
<td>5.9361e+003</td>
<td>96</td>
<td>3.4168e+003</td>
</tr>
<tr>
<td>10</td>
<td>2.6838e+004</td>
<td>39</td>
<td>9.9200e+003</td>
<td>68</td>
<td>5.9239e+003</td>
<td>97</td>
<td>3.3114e+003</td>
</tr>
<tr>
<td>11</td>
<td>2.3409e+004</td>
<td>40</td>
<td>9.8098e+003</td>
<td>69</td>
<td>5.8563e+003</td>
<td>98</td>
<td>3.2390e+003</td>
</tr>
<tr>
<td>12</td>
<td>2.2576e+004</td>
<td>41</td>
<td>9.6276e+003</td>
<td>70</td>
<td>5.6242e+003</td>
<td>99</td>
<td>3.1483e+003</td>
</tr>
<tr>
<td>13</td>
<td>2.1304e+004</td>
<td>42</td>
<td>9.5082e+003</td>
<td>71</td>
<td>5.4580e+003</td>
<td>100</td>
<td>3.1356e+003</td>
</tr>
<tr>
<td>14</td>
<td>2.0815e+004</td>
<td>43</td>
<td>9.1978e+003</td>
<td>72</td>
<td>5.3497e+003</td>
<td>101</td>
<td>2.9688e+003</td>
</tr>
<tr>
<td>15</td>
<td>2.0092e+004</td>
<td>44</td>
<td>9.0788e+003</td>
<td>73</td>
<td>5.3171e+003</td>
<td>102</td>
<td>2.8599e+003</td>
</tr>
<tr>
<td>16</td>
<td>1.9112e+004</td>
<td>45</td>
<td>8.8655e+003</td>
<td>74</td>
<td>5.2481e+003</td>
<td>103</td>
<td>2.7586e+003</td>
</tr>
<tr>
<td>17</td>
<td>1.7372e+004</td>
<td>46</td>
<td>8.7614e+003</td>
<td>75</td>
<td>5.2211e+003</td>
<td>104</td>
<td>2.7040e+003</td>
</tr>
<tr>
<td>18</td>
<td>1.6937e+004</td>
<td>47</td>
<td>8.5242e+003</td>
<td>76</td>
<td>5.1217e+003</td>
<td>105</td>
<td>2.6815e+003</td>
</tr>
<tr>
<td>19</td>
<td>1.6589e+004</td>
<td>48</td>
<td>8.3214e+003</td>
<td>77</td>
<td>5.0292e+003</td>
<td>106</td>
<td>2.5291e+003</td>
</tr>
<tr>
<td>20</td>
<td>1.5797e+004</td>
<td>49</td>
<td>8.2986e+003</td>
<td>78</td>
<td>4.9125e+003</td>
<td>107</td>
<td>2.4295e+003</td>
</tr>
<tr>
<td>21</td>
<td>1.5181e+004</td>
<td>50</td>
<td>8.1325e+003</td>
<td>79</td>
<td>4.8196e+003</td>
<td>108</td>
<td>2.4013e+003</td>
</tr>
<tr>
<td>22</td>
<td>1.4607e+004</td>
<td>51</td>
<td>8.0024e+003</td>
<td>80</td>
<td>4.7210e+003</td>
<td>109</td>
<td>2.3481e+003</td>
</tr>
<tr>
<td>23</td>
<td>1.4109e+004</td>
<td>52</td>
<td>7.8375e+003</td>
<td>81</td>
<td>4.6222e+003</td>
<td>110</td>
<td>2.2569e+003</td>
</tr>
<tr>
<td>24</td>
<td>1.3624e+004</td>
<td>53</td>
<td>7.6418e+003</td>
<td>82</td>
<td>4.5045e+003</td>
<td>111</td>
<td>2.2375e+003</td>
</tr>
<tr>
<td>25</td>
<td>1.3028e+004</td>
<td>54</td>
<td>7.5918e+003</td>
<td>83</td>
<td>4.4686e+003</td>
<td>112</td>
<td>2.0811e+003</td>
</tr>
<tr>
<td>26</td>
<td>1.2891e+004</td>
<td>55</td>
<td>7.4851e+003</td>
<td>84</td>
<td>4.4141e+003</td>
<td>113</td>
<td>2.0333e+003</td>
</tr>
<tr>
<td>27</td>
<td>1.2572e+004</td>
<td>56</td>
<td>7.3318e+003</td>
<td>85</td>
<td>4.4002e+003</td>
<td>114</td>
<td>1.9901e+003</td>
</tr>
<tr>
<td>28</td>
<td>1.2122e+004</td>
<td>57</td>
<td>7.1810e+003</td>
<td>86</td>
<td>4.3931e+003</td>
<td>115</td>
<td>1.9610e+003</td>
</tr>
<tr>
<td>29</td>
<td>1.1765e+004</td>
<td>58</td>
<td>7.1067e+003</td>
<td>87</td>
<td>4.0906e+003</td>
<td>116</td>
<td>1.9306e+003</td>
</tr>
</tbody>
</table>
Table C.2: List of the singular values $s_{117}$ to $s_{256}$ of matrix $A$ at $y_0$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$s_n$</th>
<th>$n$</th>
<th>$s_n$</th>
<th>$n$</th>
<th>$s_n$</th>
<th>$n$</th>
<th>$s_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>117</td>
<td>1.8492e+003</td>
<td>152</td>
<td>4.9061e+002</td>
<td>187</td>
<td>3.7336e+001</td>
<td>222</td>
<td>8.4815e-002</td>
</tr>
<tr>
<td>118</td>
<td>1.7895e+003</td>
<td>153</td>
<td>4.7831e+002</td>
<td>188</td>
<td>3.3837e+001</td>
<td>223</td>
<td>6.8244e-002</td>
</tr>
<tr>
<td>119</td>
<td>1.7649e+003</td>
<td>154</td>
<td>4.6463e+002</td>
<td>189</td>
<td>3.2229e+001</td>
<td>224</td>
<td>4.9522e-002</td>
</tr>
<tr>
<td>120</td>
<td>1.7133e+003</td>
<td>155</td>
<td>4.5049e+002</td>
<td>190</td>
<td>2.9765e+001</td>
<td>225</td>
<td>3.4914e-002</td>
</tr>
<tr>
<td>121</td>
<td>1.7003e+003</td>
<td>156</td>
<td>4.3570e+002</td>
<td>191</td>
<td>2.5378e+001</td>
<td>226</td>
<td>2.7786e-002</td>
</tr>
<tr>
<td>122</td>
<td>1.6663e+003</td>
<td>157</td>
<td>4.1243e+002</td>
<td>192</td>
<td>2.2084e+001</td>
<td>227</td>
<td>2.1543e-002</td>
</tr>
<tr>
<td>123</td>
<td>1.4709e+003</td>
<td>158</td>
<td>3.9872e+002</td>
<td>193</td>
<td>1.6447e+001</td>
<td>228</td>
<td>1.0035e-002</td>
</tr>
<tr>
<td>124</td>
<td>1.4233e+003</td>
<td>159</td>
<td>3.7554e+002</td>
<td>194</td>
<td>1.5527e+001</td>
<td>229</td>
<td>8.5804e-003</td>
</tr>
<tr>
<td>125</td>
<td>1.3856e+003</td>
<td>160</td>
<td>3.6547e+002</td>
<td>195</td>
<td>1.2847e+001</td>
<td>230</td>
<td>5.0489e-003</td>
</tr>
<tr>
<td>126</td>
<td>1.3511e+003</td>
<td>161</td>
<td>3.3827e+002</td>
<td>196</td>
<td>1.1988e+001</td>
<td>231</td>
<td>4.3992e-003</td>
</tr>
<tr>
<td>127</td>
<td>1.3221e+003</td>
<td>162</td>
<td>3.1310e+002</td>
<td>197</td>
<td>1.0875e+001</td>
<td>232</td>
<td>3.3588e-003</td>
</tr>
<tr>
<td>128</td>
<td>1.2541e+003</td>
<td>163</td>
<td>3.0312e+002</td>
<td>198</td>
<td>1.0524e+001</td>
<td>233</td>
<td>3.1243e-003</td>
</tr>
<tr>
<td>129</td>
<td>1.1726e+003</td>
<td>164</td>
<td>3.0600e+002</td>
<td>199</td>
<td>8.9230e+000</td>
<td>234</td>
<td>3.0694e-003</td>
</tr>
<tr>
<td>130</td>
<td>1.0885e+003</td>
<td>165</td>
<td>2.8492e+002</td>
<td>200</td>
<td>7.3405e+000</td>
<td>235</td>
<td>2.2232e-003</td>
</tr>
<tr>
<td>131</td>
<td>1.0675e+003</td>
<td>166</td>
<td>2.7385e+002</td>
<td>201</td>
<td>5.8726e+000</td>
<td>236</td>
<td>1.5428e-003</td>
</tr>
<tr>
<td>132</td>
<td>1.0212e+003</td>
<td>167</td>
<td>2.5905e+002</td>
<td>202</td>
<td>5.1844e+000</td>
<td>237</td>
<td>1.1369e-003</td>
</tr>
<tr>
<td>133</td>
<td>9.9888e+002</td>
<td>168</td>
<td>2.4618e+002</td>
<td>203</td>
<td>4.5816e+000</td>
<td>238</td>
<td>7.6324e-004</td>
</tr>
<tr>
<td>134</td>
<td>9.5305e+002</td>
<td>169</td>
<td>2.3777e+002</td>
<td>204</td>
<td>3.9484e+000</td>
<td>239</td>
<td>6.1931e-004</td>
</tr>
<tr>
<td>135</td>
<td>9.3624e+002</td>
<td>170</td>
<td>2.1912e+002</td>
<td>205</td>
<td>2.5683e+000</td>
<td>240</td>
<td>4.5532e-004</td>
</tr>
<tr>
<td>136</td>
<td>8.6967e+002</td>
<td>171</td>
<td>2.0073e+002</td>
<td>206</td>
<td>2.4765e+000</td>
<td>241</td>
<td>3.6334e-004</td>
</tr>
<tr>
<td>137</td>
<td>8.6859e+002</td>
<td>172</td>
<td>1.8506e+002</td>
<td>207</td>
<td>1.9388e+000</td>
<td>242</td>
<td>2.9739e-004</td>
</tr>
<tr>
<td>138</td>
<td>7.9816e+002</td>
<td>173</td>
<td>1.8015e+002</td>
<td>208</td>
<td>1.7912e+000</td>
<td>243</td>
<td>2.2295e-004</td>
</tr>
<tr>
<td>139</td>
<td>7.8997e+002</td>
<td>174</td>
<td>1.6652e+002</td>
<td>209</td>
<td>1.4477e+000</td>
<td>244</td>
<td>1.2871e-004</td>
</tr>
<tr>
<td>140</td>
<td>7.8283e+002</td>
<td>175</td>
<td>1.4927e+002</td>
<td>210</td>
<td>1.3028e+000</td>
<td>245</td>
<td>9.3798e-005</td>
</tr>
<tr>
<td>141</td>
<td>7.7725e+002</td>
<td>176</td>
<td>1.3050e+002</td>
<td>211</td>
<td>9.9577e+001</td>
<td>246</td>
<td>8.0347e-005</td>
</tr>
<tr>
<td>142</td>
<td>7.1605e+002</td>
<td>177</td>
<td>1.2608e+002</td>
<td>212</td>
<td>4.9700e+001</td>
<td>247</td>
<td>7.1950e-005</td>
</tr>
<tr>
<td>143</td>
<td>7.0558e+002</td>
<td>178</td>
<td>1.1755e+002</td>
<td>213</td>
<td>4.8182e+001</td>
<td>248</td>
<td>5.3765e-005</td>
</tr>
<tr>
<td>144</td>
<td>6.6710e+002</td>
<td>179</td>
<td>1.0848e+002</td>
<td>214</td>
<td>3.8206e+001</td>
<td>249</td>
<td>3.1342e-005</td>
</tr>
<tr>
<td>145</td>
<td>6.5810e+002</td>
<td>180</td>
<td>9.7709e+001</td>
<td>215</td>
<td>3.5943e+001</td>
<td>250</td>
<td>2.0374e-005</td>
</tr>
<tr>
<td>146</td>
<td>6.4323e+002</td>
<td>181</td>
<td>7.7347e+001</td>
<td>216</td>
<td>3.4569e+001</td>
<td>251</td>
<td>1.0260e-005</td>
</tr>
<tr>
<td>147</td>
<td>5.8727e+002</td>
<td>182</td>
<td>7.0597e+001</td>
<td>217</td>
<td>2.3799e+001</td>
<td>252</td>
<td>5.7407e-006</td>
</tr>
<tr>
<td>148</td>
<td>5.8117e+002</td>
<td>183</td>
<td>6.7189e+001</td>
<td>218</td>
<td>2.1616e+001</td>
<td>253</td>
<td>5.1030e-006</td>
</tr>
<tr>
<td>149</td>
<td>5.5836e+002</td>
<td>184</td>
<td>6.0099e+001</td>
<td>219</td>
<td>1.5289e+001</td>
<td>254</td>
<td>2.2609e-006</td>
</tr>
<tr>
<td>150</td>
<td>5.5011e+002</td>
<td>185</td>
<td>5.1471e+001</td>
<td>220</td>
<td>1.3389e+001</td>
<td>255</td>
<td>1.7362e-006</td>
</tr>
<tr>
<td>151</td>
<td>5.0312e+002</td>
<td>186</td>
<td>4.5098e+001</td>
<td>221</td>
<td>1.1108e-001</td>
<td>256</td>
<td>4.5262e-007</td>
</tr>
</tbody>
</table>