Inverse methods for sound speed estimation in the ocean

Tao Lin
New Jersey Institute of Technology

Follow this and additional works at: https://digitalcommons.njit.edu/dissertations

Part of the Mathematics Commons

Recommended Citation
Lin, Tao, "Inverse methods for sound speed estimation in the ocean" (2014). Dissertations. 160.
https://digitalcommons.njit.edu/dissertations/160

This Dissertation is brought to you for free and open access by the Electronic Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Dissertations by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.
Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement.

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen
The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.
ABSTRACT

INVERSE METHODS FOR SOUND SPEED ESTIMATION IN THE OCEAN

by
Tao Lin

This dissertation presents theoretical and computational approaches for estimating sound-speed in the ocean under different conditions. The first part of the dissertation discusses a fast approach for solving the inverse problem of estimating sediment sound-speed based on the Deift-Trubowitz trace formula. Under certain assumptions, this algorithm can recover the sound speed profile in the seabed using pressure field measurements in the water column at low frequencies. The inversion algorithm requires solving a non-linear integral equation. In the past, Stickler and Zhou employed a first order Born approximation for solving the integral equation. This work introduces two new methods. The first is a modified Born approximation that improves upon a standard first order approximation, yet it is easy to implement; the second one is an approximation based on interpolating the integrand. It is shown that these methods work well with synthetic data in the numerical simulations. Results are compared to those of previously developed methods and demonstrate improvement especially at sharp changes in sound speed. Although the methods are stable and effective with noise-free data, problems arise when noise is considered. Regularization methods are developed to remedy this problem. Finally, we recognize that some assumptions necessary for this algorithm to work may not be realistic; several possibilities are presented to relax these limitations.

In the second part, a method is developed for the estimation of source location and sound speed in the water column relying on linearization. The Jacobian matrix, necessary for the proposed linearization approach, includes derivatives with respect to Empirical Orthogonal Function coefficients instead of sound speed directly. First, the
inversion technique is tested on synthetic arrival times, using Gaussian distributions for the errors in the considered arrival times. The approach is efficient, requiring a few iterations, and produces accurate results. Probability densities of the estimates are calculated for different levels of noise in the arrival times. Subsequently, particle filtering is employed for the estimation of arrival times from signals recorded during the Shallow Water 06 experiment. It has been shown in the past that particle filtering can be employed for the successful estimation of multipath arrival times from short-range data and, consequently, in geometry, bathymetry, and sound speed inversion. Here, probability density functions of arrival times computed via particle filtering are propagated backwards through the proposed inversion process. Inversion estimates are consistent with values reported in the literature for the same quantities. Lastly, it is shown that the results and estimates from fast simulated annealing applied to the same arrival times are very similar.
INVERSE METHODS FOR SOUND SPEED ESTIMATION IN THE OCEAN

by

Tao Lin

Advisor: Dr. Zoi-Heleni Michalopoulou

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology and
Rutgers, The State University of New Jersey – Newark
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mathematical Sciences

Department of Mathematical Sciences, NJIT
Department of Mathematics and Computer Science, Rutgers-Newark

May 2014
APPROVAL PAGE

INVERSE METHODS FOR SOUND SPEED ESTIMATION IN THE OCEAN

Tao Lin

Dr. Zoi-Heleni Michalopoulou, Dissertation Advisor
Professor of Mathematical Sciences, NJIT

Dr. Sunil K. Dhar, Committee Member
Professor of Mathematical Sciences, NJIT

Dr. Hongya Ge, Committee Member
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. David J. Horntrop, Committee Member
Associate Professor of Mathematical Sciences, NJIT

Dr. Jonathan H. Luke, Committee Member
Professor of Mathematical Sciences, NJIT
BIOGRAPHICAL SKETCH

Author: Tao Lin
Degree: Doctor of Philosophy
Date: May 2014

Undergraduate and Graduate Education:

- Bachelor of Science,
  New Jersey Institute of Technology, Newark, NJ, 2009
- Doctor of Philosophy in Mathematical Sciences,
  New Jersey Institute of Technology, Newark, NJ, 2014

Major: Mathematical Sciences

Presentations and Publications:


"If we do it now, we’ll never run out of time.”

-Anonymous
ACKNOWLEDGMENT

It is with deep appreciation that I express my thanks to my advisor, Dr. Zoi-Heleni Michalopoulou. Her expert knowledge in inverse problems guided me through my dissertation. She also introduced me to many key professionals in the area, who gave me critical advises when I was stuck on difficult problems. This dissertation would not have been possible without the teaching of Dr. Zoi-Heleni Michalopoulou.

I am grateful to people who served on my dissertation committee: Dr. Jonathan Luke, Dr. David Horntrop, Dr. Hongya Ge, and Dr. Sunil Dhar. Their insightful suggestions and critiques are essential for the improvement and completion of this work.

A special thanks to Dr. David Stickler, who shared many insights and suggestions that served as the basis for one of my projects. I am also grateful for the invaluable help from Dr. Ross Chapman for his suggestion on the sediment inverse problem. I would also like to thank Ms. Clarisa Gonzalez-Lenahan for her professional help in proof-reading the dissertation.

Finally, I wish to express my love and thanks to all of the members of my family. My parents, Arthur Church and Xiu Church, and my brother Avery had always been supportive no matter what I do. I could not have done it without my wife, Liping Lin, who spent many late nights alongside me while I work on the research, and proofread my dissertation. I dedicate my work to them.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Overview of the Dissertation</td>
<td>4</td>
</tr>
<tr>
<td>2 THE FORWARD PROBLEM</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Point Source in a Two-Fluid Half-space</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Fluid Half-space with a Pressure-Release Interface</td>
<td>10</td>
</tr>
<tr>
<td>2.3 Fluid Half-space in a Pekeris Waveguide</td>
<td>14</td>
</tr>
<tr>
<td>2.4 The Jost Function when $q(z)$ is a Step Function</td>
<td>16</td>
</tr>
<tr>
<td>3 EXPERIMENTS FOR RECOVERING THE REFLECTION COEFFICIENT</td>
<td>21</td>
</tr>
<tr>
<td>3.1 The Imaginary Vertical Wavenumber</td>
<td>27</td>
</tr>
<tr>
<td>3.2 Methods for Recovering $v(\lambda, z)$ for All Positive Values of $\lambda$</td>
<td>28</td>
</tr>
<tr>
<td>4 RECOVERING THE SOUND SPEED PROFILE USING THE REFLECTION COEFFICIENT</td>
<td>30</td>
</tr>
<tr>
<td>4.1 The Deift-Trubowitz Trace Formula</td>
<td>31</td>
</tr>
<tr>
<td>4.2 Iterative Methods for Solving the Trace Formula</td>
<td>37</td>
</tr>
<tr>
<td>4.3 Improving the Born Approximation</td>
<td>39</td>
</tr>
<tr>
<td>4.3.1 The Hardy Projection</td>
<td>40</td>
</tr>
<tr>
<td>4.3.2 Approximation Using an Interpolated $m(z, \lambda)$</td>
<td>41</td>
</tr>
<tr>
<td>5 NUMERICAL TECHNIQUES FOR THE SEDIMENT SOUND SPEED INVERSE PROBLEM</td>
<td>47</td>
</tr>
<tr>
<td>5.1 The Scattering Operator $S(\lambda)$</td>
<td>47</td>
</tr>
<tr>
<td>5.2 The Hankel Transform Integral</td>
<td>48</td>
</tr>
<tr>
<td>5.3 Integral with a Highly Oscillatory Integrand</td>
<td>50</td>
</tr>
<tr>
<td>6 ALGORITHM TESTING WITH SYNTHETIC NOISY DATA</td>
<td>52</td>
</tr>
<tr>
<td>6.1 Data Generation</td>
<td>52</td>
</tr>
<tr>
<td>6.2 Effect of Noise on the Recovery of the Reflection Coefficient</td>
<td>53</td>
</tr>
<tr>
<td>6.3 Simulation Results</td>
<td>56</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>7</td>
<td>INVERSION FOR SOUND SPEED IN THE WATER COLUMN USING EMPIRICAL ORTHOGONAL FUNCTIONS AND LINEARIZATION</td>
</tr>
<tr>
<td>7.1</td>
<td>Empirical Orthogonal Functions</td>
</tr>
<tr>
<td>7.2</td>
<td>The Linear System</td>
</tr>
<tr>
<td>7.3</td>
<td>Solving the Forward Problem</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Evaluating the Ray Parameter ( p )</td>
</tr>
<tr>
<td>7.3.2</td>
<td>A Turning Point in the Direct Ray Path</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Surface and Bottom Reflected Paths</td>
</tr>
<tr>
<td>7.4</td>
<td>Time Derivatives with respect to Source Range, Source Depth, and Water Column Depth</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Arrival Time Derivatives with respect to Sound Speed</td>
</tr>
<tr>
<td>7.5</td>
<td>The Complete Inverse Algorithm</td>
</tr>
<tr>
<td>7.6</td>
<td>The Regularization Method</td>
</tr>
<tr>
<td>8</td>
<td>NUMERICAL SIMULATIONS FOR THE EOF LINEARIZATION INVERSION METHOD</td>
</tr>
<tr>
<td>8.1</td>
<td>Testing the Inversion Method using Synthetic Data</td>
</tr>
<tr>
<td>8.2</td>
<td>Inversion Results from Real Data</td>
</tr>
<tr>
<td>9</td>
<td>CONCLUSIONS AND FUTURE WORK</td>
</tr>
<tr>
<td>9.1</td>
<td>Conclusions</td>
</tr>
<tr>
<td>9.2</td>
<td>Future Work</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>96</td>
</tr>
<tr>
<td>Table</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>8.1</td>
<td>83</td>
</tr>
<tr>
<td>8.2</td>
<td>83</td>
</tr>
<tr>
<td>8.3</td>
<td>88</td>
</tr>
<tr>
<td>8.4</td>
<td>91</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>A two-fluid half-space problem.</td>
</tr>
<tr>
<td>2.2</td>
<td>Fluid half-space with a pressure-release interface.</td>
</tr>
<tr>
<td>2.3</td>
<td>An example of a Pekeris waveguide with a fluid half-space with a pressure-release interface.</td>
</tr>
<tr>
<td>3.1</td>
<td>The geometry of Stickler’s experiment.</td>
</tr>
<tr>
<td>4.1</td>
<td>The contour of integration.</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison of three different ways for recovering $q(z)$. The potential $q(z)$ is a Pekeris profile.</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison of three different ways for recovering $q(z)$. The potential $q(z)$ is a two-step profile.</td>
</tr>
<tr>
<td>4.4</td>
<td>Comparison of three different ways for recovering $q(z)$. The potential $q(z)$ is chosen to be an artificial profile to show that the methods can capture discontinuities.</td>
</tr>
<tr>
<td>4.5</td>
<td>Using the DT inverse algorithm to recover $c(z)$. The graphs are for (a) a simple Perkeris profile and (b) a two-step profile.</td>
</tr>
<tr>
<td>5.1</td>
<td>The highly oscillatory integrand, $\lambda R(\lambda)e^{2i\lambda z}m^2(z,\lambda)$, from the DT formula. The potential is that of a Pekeris profile. Here, $z = 5$, $q^* = -2.63 \times 10^{-4}$, and $L = 40$.</td>
</tr>
<tr>
<td>6.1</td>
<td>The imaginary part of the noise-free Green’s function $g(k)$. The function crosses zero several times. The variables $z$ and $z^*$ are fixed.</td>
</tr>
<tr>
<td>6.2</td>
<td>Function $</td>
</tr>
<tr>
<td>6.3</td>
<td>Recovering the scattering operator $S(k)$ using $</td>
</tr>
<tr>
<td>6.4</td>
<td>Simulations for the Pekeris profile with different noise levels. The plots show the recovered SSP results for SNR levels of 50, 40, and 30 dB, as well as the true SSP.</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>6.5</td>
<td>Simulations for the step profile with different noise levels. The plots show the recovered SSP results for SNR levels of 50, 40, and 30 dB, as well as the true SSP.</td>
</tr>
<tr>
<td>6.6</td>
<td>Simulations for the test profile with different noise levels. The plots show the recovered SSP results for SNR levels of 50, 40, and 30 dB, as well as the true SSP.</td>
</tr>
<tr>
<td>7.1</td>
<td>Multipath arrivals in a shallow water oceanic waveguide. Three rays are represented in the figure: direct (D), surface reflection (SR), and bottom reflection (BR).</td>
</tr>
<tr>
<td>7.2</td>
<td>The L-curve, based on which the regularization coefficient ( \lambda^2 ) is selected. The error in the arrival times was normally distributed with zero mean and a standard deviation of 200 ms.</td>
</tr>
<tr>
<td>8.1</td>
<td>Linearization estimates vs. iteration: (a) source range, (b) source depth, (c) water depth, (d) ( \mu_1 ), (e) ( \mu_2 ), and (f) ( \mu_3 ).</td>
</tr>
<tr>
<td>8.2</td>
<td>PDFs of (a) source range, (b) source depth, (c) water column depth, (d) ( \mu_1 ), (e) ( \mu_2 ), and (f) ( \mu_3 ).</td>
</tr>
<tr>
<td>8.3</td>
<td>PDFs of (a) source range, (b) source depth, (c) water column depth, (d) ( \mu_1 ), (e) ( \mu_2 ), and (f) ( \mu_3 ) for a smaller value of ( \lambda^2 ).</td>
</tr>
<tr>
<td>8.4</td>
<td>(a) The SSP used for the simulations (curve with squares) and the SSP constructed with the MAP estimates obtained from 200 realizations. (b) The profiles of (a) with superimposed SSPs estimated from a few realizations (solid curves).</td>
</tr>
<tr>
<td>8.5</td>
<td>PDFs of (a) source range, (b) source depth, (c) water column depth, (d) ( \mu_1 ), (e) ( \mu_2 ), and (f) ( \mu_3 ) for a different set of prior values.</td>
</tr>
<tr>
<td>8.6</td>
<td>PDFs of (a) source range, (b) source depth, (c) water column depth, (d) ( \mu_1 ), (e) ( \mu_2 ), and (f) ( \mu_3 ) for an increased noise level.</td>
</tr>
<tr>
<td>8.7</td>
<td>Posterior PDFs for multipath arrival times for the SW06 data.</td>
</tr>
<tr>
<td>8.8</td>
<td>PDFs of (a) source range, (b) source depth, (c) water column depth, (d) ( \mu_1 ), (e) ( \mu_2 ), and (f) ( \mu_3 ) for real data.</td>
</tr>
<tr>
<td>8.9</td>
<td>The L-curve for the real data, based on which the regularization coefficient ( \lambda^2 ) is selected.</td>
</tr>
<tr>
<td>8.10</td>
<td>Time series at 14 hydrophones. The real data (solid lines) were collected during the SW06 experiment. Synthetic time series (dotted lines) are generated using the linearization estimates.</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>8.11</td>
<td>(a) The mean SSP as calculated by CTD measurements (squares) and the SSP constructed with the MAP estimates obtained from the extracted arrival time PDFs (circles). (b) The profiles of (a) with superimposed SSPs constructed from distinct arrival time particles (solid curves).</td>
</tr>
<tr>
<td>8.12</td>
<td>Fast simulated annealing results vs. iteration for one set of arrival time estimates for (a) source range, (b) source depth, (c) water column depth, (d) $\mu_1$, (e) $\mu_2$, and (f) $\mu_3$ for real data.</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Inverse problems in ocean acoustics are of particular interest to scientists, because of the high effectiveness of water as a medium for acoustic signal propagation. Acoustic signals carry information about environmental parameters of the ocean propagation medium, that are important to estimate for defense applications and environmental monitoring. To put it in a different way, environmental parameters such as sound speed profiles (SSPs) in the water column and sediments, attenuation, and density greatly influence the propagation of acoustic signals.

Signal propagation is naturally affected by the water column SSP but is also largely determined by sound speed in the seabed sediments. The effect is particularly pronounced in low frequency propagation in shallow water environments. The acoustic signal penetrates the ocean bottom layer, interacts with sediments, and then propagates back up. Propagation models built with ocean bottom properties in mind naturally achieve higher accuracy than those without. For this reason, more and more applications require the knowledge and integration of sediment SSPs in their acoustic models for better performances. Important applications that can benefit from an accurate knowledge of the sediment SSP include localization and tracking. However, the difficulty in making direct measurements of such properties of the ocean bottom is self-evident. An inverse algorithm coupled with acoustic experiments is a much more efficient, more convenient alternative. The focus of the dissertation is to develop new inverse methods for estimating environmental properties of the ocean propagation medium.

The first part of this work is on the estimation of the SSP in the sediment layer of a shallow water environment. The sound speed in the water column
layer is mainly determined by temperature, salinity, and pressure (or equivalently, depth). The sound speed in the sediment layer is mainly determined by its material composition, such as clay, mud, sand, or a mixture. The sediment SSP is an important environmental property which can be used for detection, localization, and determination of underwater objects. Various geoacoustic inversion techniques for recovering the sediment SSP and attenuation properties have been developed in the recent years.

One popular approach for inversion is matched-field processing (MFP) [39]. This method uses search algorithms that navigate a large parameter space to seek for parameter values that best fit the data. MFP requires a combination of wave propagation modeling for the generation of replica fields at receiving phones and a decision rule that estimates model parameters entering the replica calculation. Inversion is performed by identifying those values of the model parameters that maximize a similarity measure between replica and true acoustic fields. MFP was originally used for source localization and was later adapted and applied to estimation of environmental parameters. MFP, when applied for the estimation of environmental parameters, is often referred to as matched-field inversion (MFI). MFI has produced very good results in both synthetic and real data cases; however, it is computationally intensive. Global optimization techniques have been successful in accelerating matched-field methods [4, 16, 33, 6, 21, 27], but, still, the computational cost of MFI remains a challenging problem.

Another popular algorithm [23, 32] is the perturbative modal inverse method. This approach uses perturbation analysis on the Helmholtz equation to obtain a relation between the change in sound speed and the change in eigenvalues. The perturbative nature of this method implies that the solution to the full problem cannot be obtained in one step; an initial profile is needed and the forward problem needs to be solved several times until the solution converges. Computationally, this
method is a vast improvement over the global optimization methods, but, on the other hand, it may suffer from the presence of local extrema.

The last class of methods, which are the subject of the first part of this dissertation, is referred to as the class of direct methods. These approaches aim to recover the sediment SSP directly, using a measurement called the reflection coefficient $R$. The solution can be obtained in one step and limited prior information is necessary. Direct methods are very computationally efficient compared to other techniques. However, the theory of direct methods is not fully developed and some difficulties need to be addressed before these methods become suitable for practical applications. On this topic, Merab and Frisk [26] have done work on reconstructing the ocean bottom velocity profile using the $R$ data. Their work is based on a discretization of the Gelfand-Levitan method. Stickler and Deift [36] provided the theoretical background for an inverse method for recovering the SSP using the knowledge of $R$ and employing the Deift-Trubowitz (DT) formula. In all these research projects, the difficulty of measuring the necessary data was discussed and some experiments for such measurements were proposed. The present work aims to expand the current theory on direct methods and attempts to construct an algorithm for real applications.

We will follow on Stickler’s footsteps, exploring the existing algorithm for sound-speed inversion and testing the feasibility of the algorithm on synthetic noisy data. We succeeded in improving on the standard technique for approximating the solution to a key component of Sticker’s algorithm - the DT trace formula. Stickler proposed an experimental method to make the measurement of $R$ more feasible [35]. We performed numerical simulations using synthetic data to test the algorithm and performed an error analysis. We found that the algorithm is very sensitive to noise and we developed a technique to compensate for this shortcoming.
The second part of this dissertation presents an inversion method based on ray tracing theory and linearization. The main contribution of this inversion approach is the combination of linearization of the inverse problem with the use of Empirical Orthogonal Functions (EOFs). In classical ray tracing theory, a typical linearization requires the derivation of the Jacobian matrix, which involves finding the derivative of multipath arrival times with respect to the SSP. It is usually difficult to find this derivative, since the SSP is a continuous function of depth. The EOFs simplify the description of the SSP into a linear combination of three eigenvectors, making the derivation of the Jacobian much easier.

1.1 Overview of the Dissertation

The first part of the dissertation spans Chapters 2-6, which discuss the direct method for sediment SSP inversion. Chapter 2 describes the forward problem. We begin with a mathematical statement of this problem, for it is necessary to understand the forward model before attempting to solve the inverse problem. Jost functions are introduced in this chapter, since we will use many existing inverse techniques involving such functions in our sediment sound speed inverse problem.

Chapter 3 discusses how the reflection coefficient can be experimentally obtained. Three experiments are presented, which can be potentially used to measure physical quantities. Those can then be converted into the reflection coefficient. These experiments are important, since many of the current inverse methods reconstruct the SSP using coefficient \( R \), which is difficult to measure. The experiments described here serve as a bridge that connects the theory of direct inverse method based on the reflection coefficient to practical applications.

Chapter 4 discusses the theory of the DT trace formula, which is the major tool used to invert for the SSP. The derivation of the DT formula is first described and then several numerical methods for solving the DT equation are presented.
Chapter 5 is a collection of the numerical techniques used in different parts of the inverse problem. Here, we describe the numerical methods for solving the forward problem, performing a Hankel transform and evaluating integrals with a highly oscillatory integrand.

Chapter 6 discuss the performance of the inverse algorithm using simulated data. Gaussian noise is added to synthetic data and a Monte Carlo method is employed for testing the algorithm. We found that the method is sensitive to noise and developed a technique to handle the problem.

The second part of the dissertation spans Chapters 7-8, which focus on the inversion method using EOFs. Chapter 7 presents the derivation of EOFs as well as the linearization scheme. A regularization technique is also introduced to deal with the resulting ill-conditioned inverse system.

In Chapter 8, the algorithm is tested using both synthetic and real data. We showed that the algorithm works well for both cases and we also compared the results to those of a global optimization method, fast simulated annealing. The methods produced very similar results, with our approach requiring significantly fewer operations.
The determination of the ocean properties from the reflection coefficient relies heavily on the solution of the direct problem. Therefore, it is necessary to discuss the solution of the Helmholtz equation in order to understand the inverse algorithm. In this chapter, we will look at a few simple cases, where the solutions to the Helmholtz equation can be found analytically. These solutions not only provide us with useful insight into the physical system but also serve as an important basis for testing the inversion algorithm. The first part of this chapter presents a mathematical description of the two-fluid half-space problem. This is followed by a fluid half-space problem with a pressure-release interface. Finally, for the half-space problem, a simple case where the SSP is a step function is discussed. The description of the two-fluid half-space problem is included in this chapter for theoretical interest, while the solution to the half-space problem with a pressure-release interface is essential for our inverse algorithm.

We limit our discussion to an ocean environment satisfying the following conditions:

1. The ocean environment is stratified. The sound-speed profile $c$ depends only on the depth $z$ of the ocean. That is, $c = c(z)$ for $0 \leq z < \infty$.

2. The density profile of the ocean is assumed to be constant.

These assumptions simplify our discussion. The first assumption is often a good approximation. Also, the second assumption does not undermine the generality of this work. For the case of a non-constant density profile, a standard change of variable by setting $P = \frac{u}{\rho(z)}$ is usually used to modify the Helmholtz equation to a simplified
form, which is typically similar to that of a constant density profile problem. For this reason, it is acceptable at this stage to focus our attention only on cases of constant density.

### 2.1 Point Source in a Two-Fluid Half-space

We begin with the Helmholtz equation:

\[
\frac{1}{r} \frac{\partial}{\partial r} r \frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial z^2} + k^2(z)u = -\frac{\delta(r)}{2\pi r} \delta(z - z_0). \tag{2.1}
\]

Here, \( k(z) = \frac{\omega}{c(z)} \). The environmental profile discussed in this section is shown in Figure 2.1. The two half-spaces can represent water and ocean bottom half-spaces, or air and water half-spaces. A mathematical description is as follows:

\[
c(z) = \begin{cases} 
  c_1 & \text{if } z \leq 0, \\
  c_2 & \text{if } z > 0.
\end{cases}
\]

For simplicity, we set: \( \rho(z) = 1 \).

A standard method for solving the Helmholtz equation in a stratified medium is by using the Hankel transform. The Hankel transform is essentially a special case of the two-dimensional Fourier transform, where the domain is radially symmetric. The Hankel transform is defined as follows:

\[
u(r) = \int_{\beta=0}^{\infty} v(\beta) J_0(\beta r) \beta d\beta, \tag{2.2}
\]

\[
v(\beta) = \int_{r=0}^{\infty} u(r) J_0(\beta r) rdr, \tag{2.3}
\]

where \( J_0(\beta r) \) is the standard 0th-order Bessel function of the first kind.

Applying the Hankel transform to Equation 2.1, we can reduce the Helmholtz equation from a partial differential equation to an ordinary differential equation,
which is written as:

\[
\frac{d^2 v}{dz^2} + [k^2(z) - \beta^2] v = -\frac{1}{2\pi} \delta(z - z_0).
\]  \hspace{1cm} (2.4)

Using the radiation condition, we can see that the homogeneous solution for the upper half-space with wavenumber \( k_1 = \omega/c_1 \) is:

\[
H_1(\beta, z) = A_1(\beta)e^{-ik_{z,1}z},
\]  \hspace{1cm} (2.5)

and the homogeneous solution for the lower half-space with \( k_2 = \omega/c_2 \) is

\[
H_2(\beta, z) = A_2(\beta)e^{ik_{z,2}z}.
\]  \hspace{1cm} (2.6)

Quantities \( A_1(\beta) \) and \( A_2(\beta) \) are the amplitudes for the waves represented by the terms \( e^{-ik_{z,1}z} \) and \( e^{-ik_{z,2}z} \), respectively. Note that \( k_{z,1} = \sqrt{k_1^2 - \beta^2} \) and \( k_{z,2} = \sqrt{k_2^2 - \beta^2} \). Now we assume that the source in the upper half-space is at depth \( z = z_s \), where \( z_s < 0 \). Then, the free-field depth-dependent Green’s function is:
\[ g(\beta, z, z_s) = -\frac{e^{ik_{z,1}|z-z_s|}}{4\pi ik_{z,1}}. \]  

(2.7)

Combining Equations 2.5, 2.6, and 2.7, we arrive at the Green’s function solution to the two-fluid half-space problem:

\[ G(\beta, z, z_s) = \begin{cases} 
  g(\beta, z, z_s) + A_1(\beta)e^{-ik_{z,1}z} & \text{if } z \leq 0, \\
  A_2(\beta)e^{ik_{z,2}z} & \text{if } z > 0.
\end{cases} \]  

(2.8)

On the interface \(z = 0\), we require the continuity of pressure and vertical displacements, which corresponds to conditions for the first and second derivative of the Green’s function, respectively. As a result, we obtain the following system of equations:

\[ g(\beta, 0, z_s) + A_1(\beta) = A_2(\beta), \]  

(2.9)

\[ k_{z,1}g(\beta, 0, z_s) - k_{z,1}A_1(\beta) = k_{z,2}A_2(\beta). \]  

(2.10)

Solving the system provides us with the following solution:

\[ A_1(\beta) = \frac{k_{z,1} - k_{z,2}}{k_{z,1} + k_{z,2}} g(\beta, 0, z_s), \]  

(2.11)

\[ A_2(\beta) = \frac{2k_{z,1}}{k_{z,1} + k_{z,2}} g(\beta, 0, z_s). \]  

(2.12)

Here, we examine the physical meaning of \(A_1(\beta)\) and \(A_2(\beta)\). Both terms represent the amplitudes of plane waves, as previously mentioned. \(A_1(\beta)\) is the amplitude of the reflected plane wave and \(A_2(\beta)\) is the amplitude of the transmitted wave. Since the term \(g(\beta, 0, z_s)\) represents the complex amplitude at \(z = 0\) of plane waves incident from above, the fractions in Equations 2.11 and 2.12 are called the reflection coefficient \(R\) and transmission coefficient \(T\), respectively. We can obtain
the reflection coefficient by assuming the incident plane wave has an amplitude of one. By setting \( g(\beta, 0, z_s) = 1 \), we have:

\[
R(\beta) = \frac{k_{z,1} - k_{z,2}}{k_{z,1} + k_{z,2}},
\]

(2.13)

\[
T(\beta) = \frac{2k_{z,1}}{k_{z,1} + k_{z,2}}.
\]

(2.14)

As we can see from the above discussion, one can think of \( R \) as the ratio between the amplitude of plane wave \( A_1(\beta) \) and the amplitude of the incident plane wave \( g(\beta, 0, z_s) \).

2.2 Fluid Half-space with a Pressure-Release Interface

We now shift our attention to a fluid half-space with a pressure-release interface. Consider a spherically symmetric point source located at \( z = z_s \) in the environment depicted in Figure 2.2. Similarly to the two-fluid half-space problem, the standard Helmholtz equation for the pressure field function applies. In cylindrical coordinates, we let the \( z \) axis pass through the point source. The pressure field function is only dependent on the range \( r \) and depth \( z \). It follows that the pressure field \( u(r, z) \) satisfies:

\[
\frac{1}{r} \frac{\partial}{\partial r} r \frac{\partial u}{\partial r} + \frac{\partial^2 u}{\partial z^2} + k^2(z)u = -\frac{\delta(r)}{2\pi r} \delta(z - z_s),
\]

(2.15)

\[
u(r, 0) = 0, \quad 0 \leq r,
\]

(2.16)

\[
u(r, z), \quad \text{outward going as } r, z \to \infty,
\]

(2.17)

where \( k(z) = \omega/c(z) \). With our inverse algorithm in mind, we assume that the SSP \( c(z) \) has a constant limiting value: \( c(z) \to c_\infty \). The solution to the above system can be found without this condition, but it is necessary for the convergence of a key formula in the inverse algorithm - the DT integral formula.
Recall the Hankel transform:

\[ u(r) = \int_{\beta=0}^{\infty} v(\beta) J_0(\beta r) \beta d\beta, \]  
\[ v(\beta) = \int_{r=0}^{\infty} u(r) J_0(\beta r) rdr. \]  

Similarly to the Fourier transform, which looks at the frequency domain by decomposing the wave signal in the time domain into sines and cosine, the Hankel transform decomposes the wave in a spherically symmetric space using Bessel functions (which are the eigenfunctions of propagating waves on a plane), looking at the corresponding frequency domain. The Hankel transform is basically a two-dimensional Fourier transform for radial (axisymmetric) functions. The application of the Hankel transform reduces Equations 2.15, 2.16, and 2.17 into the following:

\[ \frac{d^2 v}{dz^2} + [k^2(z) - \beta^2]v = \frac{1}{2\pi} \delta(z - z_s), \]  
\[ v(\beta, 0) = 0, \quad 0 \leq \beta, \]  
\[ v(\beta, z), \quad \text{outward going as } \beta, z \to \infty. \]
The equations above form a second order ODE. The system can be solved using the Green’s function method. Rather than solving for the Green’s function in the usual way, we introduce a change of variable, so that we can use Jost functions. This change of variable will facilitate our introduction of the inverse algorithm. A Jost function is the solution to a second order ODE and has been extensively studied by Deift and Trubowitz [5]. The Jost ODE can be expressed as follows:

\[-f''_{1,2}(z, \lambda) + q(z)f_{1,2}(z, \lambda) = \lambda^2 f_{1,2}(z, \lambda),\]  
\[f_{1,2}(z, \lambda) \sim e^{\pm i \lambda z}, \text{as } z \to \pm \infty.\]  

The solutions \(f_1(z, \lambda), f_2(z, \lambda)\) are referred to as the Jost function pairs. Note that Equation 2.24 contains two boundary conditions. Each boundary condition corresponds to one Jost function. It is assumed that \(q(z) \to 0\) as \(z \to \infty\). For \(z\) large enough, the condition placed on \(f_i(z, \lambda)\) is equivalent to a radiation condition.

Back to Equation 2.20, the change of variable we use is \(q(z) = k^2_{\infty} - k^2(z)\) and also \(\lambda^2 = k^2_{\infty} - \beta^2\), providing us with an equation of the same form as Equation 2.23. We can write \(v(\beta, z)\) in terms of the Jost functions. Quantity \(\lambda\) is the vertical wavenumber. One detail that needs to be mentioned is that, since \(\lambda^2 = k^2_{\infty} - \beta^2\), \(\lambda\) is real only when \(0 < \beta < k_{\infty}\) or when \(\beta\) is an imaginary number. This means that, if we want to know function \(f(z, \lambda)\) for all real values of \(\lambda\), then we must first know function \(v(\beta, z)\) for all values of \(\beta\) satisfying \(0 < \beta < k_{\infty}\) and all imaginary values of \(\beta\). This is a difficult problem to handle, since, in real experiments, we can only measure the pressure field as a function of all real values of \(\beta\), which means that we can only obtain \(f(z, \lambda)\) for values of \(\lambda\) that are either imaginary or \(0 < \lambda < k_{\infty}\). We will discuss several ways for overcoming this problem in the following section.

The Jost functions in Equations 2.23 and 2.24 are typically analyzed by applying another change of variable. Let us set \(m_{1,2}(z, \lambda) = e^{\pm i \lambda z} f_{1,2}(z, \lambda)\). By doing so, we remove the oscillation in the solution of the ODE and the resulting
equation has a much nicer behavior. The equation for \( m_{1,2}(z, \lambda) \) is:

\[
m''_{1,2}(z, \lambda) \pm 2i\lambda m'_{1,2}(z, \lambda) = q(z) m_{1,2}(z, \lambda), \\
m_{1,2}(z, \lambda) \sim 1, \text{ as } z \to \pm \infty.
\] (2.25, 2.26)

Because of the requirement we placed on the system, \( q(z) \to 0 \text{ as } z \to \infty \), we can choose an \( L \) large enough so that the boundary condition in Equation 2.26 can be approximated as:

\[
m_{1,2}(\pm L, \lambda) = 1, \\
m'_{1,2}(\pm L, \lambda) = 0.
\] (2.27, 2.28)

At this point, any standard numerical technique such as the finite difference method or the Runge-Kutta method is sufficient for solving Equations 2.25, 2.27, and 2.28.

We can also write Equation 2.25 in a Volterra integral equation formulation. We will do this for \( m_1(z, \lambda) \); the derivation for \( m_2(z, \lambda) \) is similar. We first multiply the two sides of the ODE by an integrating factor and then integrate:

\[
e^{2i\lambda z} \left( m''_1(z, \lambda) + 2i\lambda m'_1(z, \lambda) \right) = e^{2i\lambda z} q(z) m_1(z, \lambda) \\
\frac{d}{dz} \left( e^{2i\lambda z} m'_1(z, \lambda) \right) = e^{2i\lambda z} q(z) m_1(z, \lambda) \\
-e^{2i\lambda z} m'_1(z, \lambda) = \int_z^\infty e^{2i\lambda s} q(s) m_1(s, \lambda) ds \\
m'_1(z, \lambda) = -\int_z^\infty e^{2i\lambda(s-z)} q(s) m_1(s, \lambda) ds \\
m_1(z, \lambda) = 1 - \int_z^\infty \frac{e^{2i\lambda(s-z)} - 1}{2i\lambda} q(s) m_1(s, \lambda) ds.
\] (2.29, 2.30)
When integrating, we use Leibniz’s rule and the fact that $m_1'(z, \lambda) \to 0$ and $m_1(z, \lambda) \to 1$ as $z \to \infty$. Similarly, we can get the equations for $m_2, f_{1,2}$ as follows:

\[
m_2(z, \lambda) = 1 + \int_{-\infty}^{z} \frac{e^{2i\lambda(z-s)} - 1}{2i\lambda} q(s)m_2(s, \lambda)ds,
\]

\[
f_2(z, \lambda)e^{i\lambda z} = 1 + \int_{-\infty}^{z} \frac{e^{2i\lambda(z-s)} - 1}{2i\lambda} q(s)f_2(s, \lambda)e^{i\lambda s} ds,
\]

\[
f_1(z, \lambda)e^{-i\lambda z} = 1 - \int_{z}^{\infty} \frac{e^{2i\lambda(s-z)} - 1}{2i\lambda} q(s)f_1(s, \lambda)e^{-i\lambda s} ds.
\]

Another interesting technique that can be used to simplify Equation 2.25 even further is to introduce a method borrowed from electrical circuit theory. We define the impedance $Z(z, \lambda) = \frac{m'(z, \lambda)}{m(z, \lambda)}$. We differentiate $Z(z, \lambda)$ with respect to $z$ and use Equation 2.25. We can, then, reduce the previous second order ODE to a first order nonlinear differential equation. The resulting equation is:

\[
Z'(z, \lambda) = q(z) + Z^2(z, \lambda) - 2i\lambda Z(z, \lambda),
\]

\[
Z(z, \lambda) \sim 0, \text{ as } z \to \infty.
\]

Equation 2.34 immediately provides an inverse formula for recovering $q(z)$, since the function $q(z)$ can be separated. In other words, if we can find the impedance function $Z(z, \lambda)$ for any fixed value of $\lambda$, it can be used directly to recover $q(z)$. However, the experimental measurement of the impedance function $Z(z, \lambda)$ remains a challenge.

### 2.3 Fluid Half-space in a Pekeris Waveguide

A closed form analytical solution for Equation 2.20 can be obtained if our SSP $c(z)$ is that of a Pekeris waveguide. Our derivation for the solution will follow those in [20]. We show an example of a Pekeris waveguide in Figure 2.3. The SSP can be described by the following formula:

\[
c(z) = \begin{cases} 
  c_1 & \text{if } z \leq D, \\
  c_2 & \text{if } z > D.
\end{cases}
\]
We assume that the point source is located within the water column, or $0 < z_s < D$. Symbol $D$ represents the water depth. To obtain the Green’s function, we assume that the solution is of the form $e^{ikz}$:

$$G(\beta, z, z_s) = \begin{cases} g(\beta, z, z_s) + A^-_1(\beta)e^{-ik_{z,1}z} + A^+_1(\beta)e^{ik_{z,1}z} & \text{if } z \leq D, \\ A_2(\beta)e^{ik_{z,2}(z-D)} & \text{if } z > D. \end{cases} \tag{2.36}$$

Here, $k_{z,1} = \sqrt{k_1^2 - \beta^2}$, where $k_1 = \omega/c_1$ is the wavenumber in the water at frequency $\omega$. In order to satisfy the radiation condition, the vertical wavenumber $k_{z,2}$ must be defined as follows:

$$k_{z,2} = \begin{cases} \sqrt{k_2^2 - \beta^2}, & \text{if } |\beta| < k_2, \\ i\sqrt{\beta^2 - k_2^2}, & \text{if } |\beta| > k_2, \end{cases} \tag{2.37}$$

where $k_2 = \omega/c_2$. The free-space Green’s function is given by $g(\beta, z, z_s) = e^{ik_{z,1}|z-z_s|}/(4\pi ik_{z,1})$. The three-boundary conditions we apply to Equation 2.36 are the
pressure-release condition and the continuity of pressure and vertical displacement on the fluid-bottom interface. As a result, the following system of equations is obtained:

\[ g(\beta, 0, z_s) + A^-_1(\beta) + A^+_1(\beta) = 0, \]  
\[ g(\beta, D, z_s) + A^-_1(\beta)e^{-ik_{z,1}D} + A^+_1(\beta)e^{ik_{z,1}D} = A_2(\beta), \]  
\[ k_{z,1}g(\beta, D, z_s) - k_{z,1}A^-_1(\beta)e^{-ik_{z,1}D} + k_{z,1}A^+_1(\beta)e^{ik_{z,1}D} = k_{z,2}A_2(\beta). \]  

Or, in matrix form, we can write the system of equations as follows:

\[
\begin{pmatrix}
1 & 1 & 0 \\
e^{ik_{z,1}D} & e^{-ik_{z,1}D} & 1 \\
k_{z,1}e^{ik_{z,1}D} & -k_{z,1}e^{-ik_{z,1}D} & -k_{z,2}
\end{pmatrix}
\begin{pmatrix}
A^+_1(\beta) \\
A^-_1(\beta) \\
A_2(\beta)
\end{pmatrix}
= \begin{pmatrix}
-g(\beta, 0, z_s) \\
g(\beta, D, z_s) \\
-k_{z,1}g(\beta, D, z_s)
\end{pmatrix}.
\]  

Solving the matrix system and assuming that the determinant of the matrix is not zero, we can get the expressions for the amplitude functions:

\[ A^+_1(\beta) = -\frac{(k_{z,1} - k_{z,2})g(\beta, 0, z_s) + (k_{z,1} + k_{z,2})g(\beta, D, z_s)e^{iDk_{z,1}}}{(k_{z,1} - k_{z,2}) + (k_{z,1} + k_{z,2})e^{2iDk_{z,1}}}, \]  
\[ A^-_1(\beta) = -\frac{(k_{z,1} + k_{z,2})g(\beta, 0, z_s)e^{2iDk_{z,1}} + (k_{z,1} + k_{z,2})g(\beta, D, z_s)e^{iDk_{z,1}}}{(k_{z,1} - k_{z,2}) + (k_{z,1} + k_{z,2})e^{2iDk_{z,1}}}, \]  
\[ A_2(\beta) = \frac{2k_{z,1}g(\beta, 0, z_s)e^{Dk_{z,1}} - 2k_{z,1}g(\beta, D, z_s)}{(k_{z,1} - k_{z,2}) + (k_{z,1} + k_{z,2})e^{2iDk_{z,1}}}. \]

This concludes the discussion on the solution of the Helmholtz equation in a fluid half-space with a pressure-release boundary condition and a Pekeris waveguide.

### 2.4 The Jost Function when \( q(z) \) is a Step Function

If we perform a change of variable on a fluid half-space problem with a Pekeris waveguide, we will obtain the Jost function with a step function as its parameter \( q(z) \). An analytical solution to the Jost ODE for this simple case is very useful for testing the inverse algorithm. For more complex SSPs, the reflection coefficient usually needs to be calculated numerically, since no closed form solutions are available.
As mentioned, a closed form solution to the Jost ODE, Equation 2.23, can be derived when \( q(z) \) is a step function. Quantity \( q(z) \) is defined as:

\[
q(z) = \begin{cases} 
q_1 & \text{if } z \leq z^*, \\
0 & \text{if } z > z^*.
\end{cases}
\]

Or, similarly, \( q(z) = q_1 - q_1 u(z - z^*) \). We are mostly interested in the cases where \( q_1 \leq 0 \). The case where \( q_1 > 0 \) is not physically relevant but is still interesting for theoretical reasons. The eigenfunction solution pairs for Equation 2.23 are shown below for convenience. We first look at only one of the Jost equations. For the Jost equation with the other boundary condition the derivation is almost identical.

\[
-f''(z, \lambda) + (q_1 - q_1 u(z - z^*)) f(z, \lambda) = \lambda^2 f(z, \lambda),
\]

\[
f(z, \lambda) \sim e^{i\lambda z}, \text{ as } z \to \infty.
\]

We look at the two regions \( z \leq z^* \) and \( z > z^* \). In each of these regions, Equation 2.23 is simply an ODE with constant coefficients. We can write down the form of the solution explicitly. This form depends on the value of \( \lambda^2 - q_1 \). We consider the following three cases.

**Case 1:** If \( \lambda^2 - q_1 < 0 \), then

\[
f(z, \lambda) = \begin{cases} 
Ae^{\sqrt{q_1 - \lambda^2}z} + Be^{-\sqrt{q_1 - \lambda^2}z} & \text{if } z \leq z^*, \\
e^{i\lambda z} & \text{if } z > z^*.
\end{cases}
\] (2.45)

Judging from Equation 2.23, we can see that \( f''(z, \lambda) = -(q_1 - q_1 u(z - z^*)) f(z, \lambda) - \lambda^2 f(z, \lambda) \) is not continuous at \( z = z^* \), since there is a jump in term \( q(z)f(z, \lambda) \) at that point. We can seek a solution that is continuous in both the first derivative and second derivative at point \( z = z^* \). In other words, we require \( f'(z, \lambda) \) and \( f(z, \lambda) \) to be continuous at \( z = z^* \). Enforcing this condition on Equation 2.45, the following
system (in a matrix form) is obtained:
\[
\begin{bmatrix}
e^{\omega z^*} & e^{-\omega z^*} \\
\omega e^{\omega z^*} & -\omega e^{-\omega z^*}
\end{bmatrix}
\begin{bmatrix} A \\ B \end{bmatrix} =
\begin{bmatrix} e^{i\lambda z^*} \\ i\lambda e^{i\lambda z^*} \end{bmatrix},
\] (2.46)

where \(\omega = \sqrt{q_1 - \lambda^2}\). Solving the previous system, we have:
\[
A = \frac{i\lambda + \omega}{2\omega} e^{i\lambda z^* - \omega z^*},
\] (2.47)
\[
B = \frac{i\lambda - \omega}{2\omega} e^{i\lambda z^* + \omega z^*}.
\] (2.48)

**Case 2**, if \(\lambda^2 - q_1 > 0\), then
\[
f(z, \lambda) = \begin{cases} 
\begin{bmatrix}
A e^{i\sqrt{\lambda^2 - q_1} z} + B e^{-i\sqrt{\lambda^2 - q_1} z} & \\
\lambda + \gamma e^{i(\lambda z^* - \gamma z*)}
\end{bmatrix} & \text{if } z \leq z^*, \\
e^{i\lambda z} & \text{if } z > z^*. 
\end{cases}
\] (2.49)

Similarly, we require \(f'(z, \lambda)\) and \(f(z, \lambda)\) to be continuous at \(z = z^*\). This results in the following system:
\[
\begin{bmatrix}
e^{i\gamma z^*} & e^{-i\gamma z^*} \\
i\gamma e^{i\gamma z^*} & -i\gamma e^{-i\gamma z^*}
\end{bmatrix}
\begin{bmatrix} A \\ B \end{bmatrix} =
\begin{bmatrix} e^{i\lambda z^*} \\ i\lambda e^{i\lambda z^*} \end{bmatrix},
\] (2.50)

where \(\gamma = \sqrt{\lambda^2 - q_1}\). Solving the previous system, we have:
\[
A = \frac{\lambda + \gamma}{2\gamma} e^{i(\lambda z^* - \gamma z*)},
\] (2.51)
\[
B = \frac{-\lambda + \gamma}{2\gamma} e^{i(\lambda z^* + \gamma z*)}.
\] (2.52)

**Case 3**, when \(\lambda^2 = q_1\), we can see that:
\[
f(z, \lambda) = \begin{cases} 
Az + B & \text{if } z \leq z^*, \\
e^{i\gamma_1 z} & \text{if } z > z^*. 
\end{cases}
\] (2.53)
Setting \( f'(z, \lambda), f(z, \lambda) \) to be continuous at \( z = z^* \), we can solve for \( A \) and \( B \):

\[
A = i q_1 e^{i q_1 z^*}, \quad (2.54)
\]

\[
B = (1 - i q_1 z^*) e^{i q_1 z^*}. \quad (2.55)
\]

To summarize, the solution to the system is:

\[
f(z, \lambda) =
\begin{cases}
  i \lambda e^{i \lambda z^*} - \omega z^* & \text{if } z \leq z^* \text{ and } \lambda^2 - q_1 < 0, \\
  i \lambda e^{i \lambda z^*} z - i \lambda e^{i \lambda z^*} z^* + e^{i \lambda z^*} & \text{if } z \leq z^* \text{ and } \lambda^2 - q_1 = 0, \\
  e^{i \lambda z^*} (\cos \gamma (z^* - z) - \frac{i \lambda}{\gamma} \sin \gamma (z^* - z)) & \text{if } z \leq z^* \text{ and } \lambda^2 - q_1 > 0, \\
  e^{i \lambda z} & \text{if } z > z^*.
\end{cases}
\]

where \( \omega = \sqrt{q_1 - \lambda^2} \) and \( \gamma = \sqrt{\lambda^2 - q_1} \). The scattering operator, \( S(\lambda) = f(0, -\lambda)/f(0, \lambda) \), is then given by:

\[
S(\lambda) =
\begin{cases}
  \frac{(\omega - i \lambda) e^{-\omega z^*} - (\omega + i \lambda) e^{-z^*}}{(\omega + i \lambda) e^{-\omega z^*} - (\omega - i \lambda) e^{-z^*}} e^{-2 i \lambda z^*} & \text{if } \lambda^2 - q_1 < 0, \\
  \frac{1 + i \lambda z^*}{1 - i \lambda z^*} e^{-2 i \lambda z^*} & \text{if } \lambda^2 - q_1 = 0, \quad (2.57)
\end{cases}
\]

Two important properties of the scattering operator are:

\[
\overline{S(\lambda)} = S(-\lambda), \quad (2.58)
\]

\[
S(\lambda) \to 1 \text{ as } \lambda \to \infty. \quad (2.59)
\]

Finally, the reflection coefficient \( R \) can be recovered from the scattering operator \( S \).

The basic result (see [5], p. 189) is that:

\[
R^- = (1 - S)^- \quad (2.60)
\]
Symbol $g^-$ denotes the projection of the square integrable function $g$ onto $H^2^-$, the negative Hardy space. A function $g$ can be projected onto $H^2^-$ using the following formulae:

\[
\hat{g}(y) = \frac{1}{\pi} \int_{-\infty}^{\infty} g(k)e^{2iky}dk, \quad (2.61)
\]

\[
g^-(k) = \int_{-\infty}^{0} \hat{g}(y)e^{-2iky}dy. \quad (2.62)
\]
CHAPTER 3

EXPERIMENTS FOR RECOVERING THE REFLECTION COEFFICIENT

Several experiments have been proposed to measure physical quantities, which can be then used to recover the scattering operator $S(k)$ or the reflection coefficient $R(k)$ [26, 35, 36] with the two functions related via Equation 2.60. We will provide a brief description of these experiments and will then select one particular experiment which can be used in our algorithm. All of these experiments attempt to measure the pressure field as a function of range and then employ the measurement to calculate the reflection coefficient or the scattering operator. These experiments are conducted in shallow water with the aim of recovering the sediment SSP. There are some assumptions that are shared in all these proposed experiments:

1. The source frequency has to be low enough to eliminate trapped modes, or, in quantum mechanic terms, bounded states.

This assumption is very important for simplifying the inverse algorithm. It is possible to recover the sediment SSP using sources with higher frequencies; however, detailed information for the trapped modes would have to be measured as well for inversion. This adds an extra layer of complexity onto the experiments, requiring one to not only measure the pressure field function, but also to extract accurate information about each trapped mode. Then an elaborate form of the DT integral equation would have to be used to accommodate the trapped modes, as opposed to a simpler form when dealing with only continuous spectrums. In addition to this difficulty, having trapped modes would make the signal decay slower, from an order of $r^{3/2}$ (without trapped modes) to order $r^{1/2}$ (with trapped modes) [36]. This means that
measurements would have to be made for a larger distance, in order to capture all required information.

2. The ocean is assumed to be a homogeneous and vertically stratified environment. The sound speed function $c(z)$ is only dependent on the depth $z$.

Homogeneity and vertical stratification are two limiting assumptions for a great number of acoustic models. Such an assumption allows for simpler models and the utilization of a greater number of powerful mathematical tools. In practice, these assumptions are good approximations in a great deal of ocean environments. At the same time, they serve as severe limitations, when dealing with environments where horizontal dependence is significant.

3. The shear wave effect is ignored and only the compressional wave is considered in the models.

Typically, in ocean bottom inversion experiments, acoustic sources would generate compressional (P) waves. Upon interactions with the ocean bottom profile, the P waves under some situations would be converted into shear (S) waves. These experiments ignore the effect of S waves in their acoustic models. Several papers suggested that these types of conversions are minimal for frequencies above 3 Hz in ocean sediments [12, 13]. In the experiments we are considering, the source frequencies are usually higher than 3 Hz.

*Experiment 1:*

The first experiment we describe was suggested by Stickler [36]. Only theoretical guidelines were provided; no detailed experimental setup was described in the paper. The first step of the experiment is to adjust the source frequency, so that trapped modes are eliminated. Then, instead of the pressure field, the normal derivative of the pressure field at the pressure-release surface is measured; that is, $\frac{\partial u}{\partial z}(r,z)|_{z=0}$, for $r > 0$, where $u(r,z)$ is the pressure field function.
Theoretically, in order to recover the reflection coefficient, one needs the normal derivative of the pressure field function at the surface of the ocean for all \( r > 0 \), which is very “expensive” to obtain. Acceptable approximations can be made, when the signal decays quickly. Since no trapped modes are present in the proposed experiment, the pressure field decays at a rate of \( r^{-3/2} \). This allows for a smaller termination range \( L \), where \( L \) is the range up to which the measurements have to be made. Stickler then described a method to recover the reflection coefficient using this measurement of the normal derivative of the pressure field.

Several shortcomings of this experiment made it difficult to execute. First, it is difficult to measure the normal derivative of the pressure field. In order to get the normal derivative, comprehensive profiling of the pressure field has to be made first, which is then used to approximate the normal derivative. Second, the method described by Stickler for recovering the reflection coefficient is somewhat difficult to implement numerically, since it contains several highly oscillatory integrals. The accuracy of numerical methods for evaluating oscillatory integrals tends to suffer greatly when the data contain noise, which is very typical when dealing with real experimental data. For these reasons, this experiment is likely to be unsuitable for practical implementations.

**Experiment 2**: Merab and Frisk described an experiment [11, 26], which measures \( R \) as a function of the angle of incidence. The experiment requires an array of two vertical hydrophone receivers to be anchored to the ocean bottom. Then a drifting vessel is used to tow a low frequency pulsed continuous-wave source. As the ship opens range, the hydrophones near the ocean bottom measure the resulting pressure (both the amplitude and the phase).

The recorded information by the hydrophones is interpreted as the field due to the acoustic signal reflected off the bottom. Since the hydrophone location was fixed and the sources moved around, the resulting measured signal is a function of
the angle of incidence. Using the Hankel transform, the plane wave as a function of the angle of incidence can, therefore, be recovered.

Frisk conducted this experiment to measure the complex pressure field as a function of the angle of incidence. Unfortunately, the reflection coefficient $R$ calculated using the experimental data was not consistent with the physical requirement of the model (which requires $|R| < 1$).

Merab proposed an inverse method for the reconstruction of the ocean bottom SSP from a measurement of the reflection coefficient. The method uses the Gelfand-Levitan equation. Our work is mainly concerned with Stickler’s inverse method, which employs the DT equation. One can show that the inverse algorithm using the Gelfand-Levitan equation and the DT equation are equivalent under a first order approximation.

**Experiment 3:**

Stickler [35] proposed a promising experiment for recovering the reflection coefficient as a function of the wavenumber $k$. This experiment is best explained with reference to Figure 3.1. In addition to the low frequency assumption mentioned earlier, it is also assumed that the SSP $c(z)$ is known in the upper layer of the ocean. In other words, $c(z)$ is known for $z < L$, where $L$ is the depth for the ocean bottom interface; we are interested in solving for SSP $c(z)$ for $L < z$. The harmonic point source is placed within the water column layer at $z = z_0 < L$. The pressure field measurement should be made at a fixed depth $z$ for “all” ranges $r$, with $0 < z < L$. This can be done by either placing a horizontal array of equally spaced hydrophones along the same depth or by a vessel towing a hydrophone and moving it horizontally. The pressure field does not have to be measured for a large range, since the signal decays at a rate of $r^{-3/2}$ with an appropriately low frequency source.

We will now describe the theory for recovering $R$ using the pressure field measurement. The pressure field function $p(r, z)$, $r > 0$, can be used to recover
Figure 3.1 The geometry of Stickler's experiment.

the reflection coefficient. The Hankel transform of $p(r, z)$ provides $P(\mu, z)$, $\mu > 0$, where $\mu$ is the Hankel transform variable. Using a change of variable on the function $P(\mu, z)$ as described in Section 2.2, we can obtain the solution to the following ODE:

$$
\frac{d^2 g(z, z_0, k)}{dz^2} + [k^2(z) - q(z)]g = \delta(z - z_0), \quad (3.1)
$$

$$
g(0, z_0, k) = 0. \quad (3.2)
$$

Functions $g(z, z_0, k)$ and $P(\mu, z)$ differ only in their argument, vertical wavenumber $k$, and the Hankel transform variable $\mu$, which are connected through the dispersion relation $k^2 = k^2_\infty - \mu^2$. This introduced some complications, which will be described in the next section.

It is useful to introduce three solutions that satisfy the homogeneous form of Equation 3.1 with the following boundary condition:

$$
U(z, k) \sim e^{ikz} \text{ as } z \to \infty. \quad (3.3)
$$
Function $U$ represents a unit amplitude outgoing wave at infinity. Two other functions that are needed have the following boundary conditions:

\begin{align}
C(0, k) &= 1, C'(0, k) = 0, \\
V(0, k) &= 0, V'(0, k) = 1.
\end{align}

(3.4)
(3.5)

Note that $C(z, k)$ and $S(z, k)$ can be calculated for $z < L$, since we assumed that the sound speed $c(z)$ is known in the water column. We can now write the Green’s function, or the solution to Equation 3.1, in terms of these functions. Because the pressure field at depth $z$, or $g(z, z_0, k)$, is measured and also because we can use the following relation to solve for $|U(0, k)|$:

\[
\Im g(z, z_0, k) = -kV(z_>, k)V(z_<, k) / |U(0, k)|^2,
\]

(3.6)

$|U(0, k)|$ can be employed to recover the reflection coefficient using the following property. Since $U(0, k)$ is an analytic function for all values of $k$, the function $\ln(U(0, k))$ is also analytic. We can write:

\[
U(0, k) = |U(0, k)| e^{i\theta(k)},
\]

\[
\ln(U(0, k)) = \ln|U(0, k)| + i\theta(k).
\]

A useful property of the Hilbert transform is that applying the transform to the real part of an analytic function provides us with the imaginary part of the function. Thus, if we take the Hilbert transform of $\ln|U(0, k)|$, we will obtain function $\theta(k)$. Finally, the scattering operator $S(k) = f(0, -k)/f(0, k)$ can be found using the relation:

\[
S(k) = \frac{f(0, -k)}{f(0, k)} = e^{-i\theta(k)} / e^{i\theta(k)} = e^{-2i\theta(k)}.
\]

(3.7)
Recall Equation 2.60, which can be used to obtain the reflection coefficient $R$ from the scattering operator $S$. The description of the experiment is complete. The advantages of this experiment over the previous two are the following. One, the experiment is straight-forward to setup; it should be simpler than experiment 1 and not more complex than experiment 2. Two, going from the experimental measurement to the reflection coefficient is mathematically simple, unlike experiment 1, where highly oscillatory integrals need to be evaluated in order to recover $R$.

### 3.1 The Imaginary Vertical Wavenumber

As was mentioned in the discussion of Jost functions in Section 2.2, experimental data can only provide functions for real values of the radial wavenumber, which are shown as follows:

$$u(\beta, z), \text{where } \beta \text{ is real and } \beta > 0.$$  

(3.8)

Function $u(\beta, z)$ is the Hankel transform of the pressure field $p(r, z)$ as a function of range $r$ at a fixed depth $z$; physically, range $r$ has a real and positive value. The variable $\beta$ is the Hankel transform variable, also referred to as the horizontal wavenumber or the radial wavenumber. Recall Equation 2.20, where we applied a change of variable necessary for the inversion algorithm. Then, $q(z) = k_\infty^2 - k^2(z)$ and $\lambda^2 = k_\infty^2 - \beta^2$. Function $u(\beta, z)$ becomes $v(\lambda, z)$ in terms of the new variables. This modification will provide us with an equation of the same form as Equation 2.23. The variable $\lambda$ is referred to as the vertical wavenumber. This change of variable creates a complication. Since $\lambda^2 = k_\infty^2 - \beta^2$, $\lambda$ is real only when $0 < \beta < k_\infty$ or when $\beta$ is an imaginary number. This means that, from the experimental data $p(r, z)$, $r > 0$, we can only obtain the following values:

$$v(\lambda, z), \text{where } 0 < \lambda < k_\infty \text{ or } \Re(\lambda) = 0,$$  

(3.9)
where $\Re(\lambda)$ refers to the real part of $\lambda$. In other words, $v(\lambda, z)$ can be directly recovered for some real values of $\lambda$ and all imaginary values of $\lambda$. This is a difficult problem to handle, since we need $v(\lambda, z)$ as a function of all real positive values of $\lambda$. If this limitation is not addressed, it would mean that the experimental data are never sufficient for a full recovery of the reflection coefficient. The experimental data would have to also provide $u(\beta, z)$ for all imaginary values of $\beta$ for $v(\lambda, z)$ on all real positive values of $\lambda$. We will address this problem later on in this work.

### 3.2 Methods for Recovering $v(\lambda, z)$ for All Positive Values of $\lambda$

The first method is to approximate the full $v(\lambda, z)$ using only the known portions of $v(\lambda, z)$ from the real measurements, that is, we only use the function values, where

$$v(\lambda, z), \text{ where } 0 < \lambda < k_{\infty}. \quad (3.10)$$

In other words, we set $v(\lambda, z) = 0$ for $\lambda > k_{\infty}$. Stickler [36] showed that this approximation is sufficient for some simple SSPs, such as the one of the Pekeris waveguide, but it fails for more complicated profiles. The advantage of this method is due to its convenience; no extra steps are required to extract further information for the function $v(\lambda, z)$. However, the approximation is insufficient for an environment with a complex SSP, and the measured information for $v(\lambda, z)$ when $\lambda$ is an imaginary number is wasted.

The second method is to recover $v(\lambda, z)$ for $\lambda > k_{\infty}$. This means that we need to know $u(\beta, z)$ for imaginary values of $\beta$. A theorem by Van Winter [41] must be used to recover $u(\beta, z)$ for imaginary values of $\beta$. We state the theorem below.

**Van Winter’s Theorem 2.9.** A complex-valued function $f$ defined on the upper half-plane belongs to the Hardy space $H^{2+}$ if and only if it is an inverse Mellin transform according to

$$f(re^{i\phi}) = (2\pi)^{-1/2} \int_{-\infty}^{\infty} F(t)(re^{i\phi})^{-it-1/2} dt, \quad (3.11)$$
with a function \( F(t) \) satisfying
\[
\int_{-\infty}^{\infty} (1 + e^{2\pi t}) |F(t)|^2 dt < \infty. \tag{3.12}
\]

The Mellin transform of Equation 3.11 is:
\[
F(t) = e^{-\phi t + \pi/2} (2\pi)^{-1/2} \int_0^\infty f(re^{i\phi}) r^{it-1/2} dr, \tag{3.13}
\]
where the integration is along any ray \( re^{i\phi}, \phi \in [0, \pi] \). This theorem provides a method for finding \( f(re^{i\phi}), \phi \in [0, \pi] \), given \( F(t) \). Since we can obtain \( u(\beta, z) \) on imaginary values of \( \beta \) from real measurements, we can use Van Winter’s theorem to find its value along any rays in the imaginary plane. To do this, the first step is to find \( F(t) \). We set \( \phi = \pi/2 \) and \( f(re^{i\phi}) = f(ir) = u(ir, z), r > 0 \). Substituting in Equation 3.13, we obtain:
\[
F(t) = e^{-\pi t/2 + \pi/4} (2\pi)^{-1/2} \int_0^\infty u(ir, z) r^{it-1/2} dr. \tag{3.14}
\]

Setting \( \phi = 0 \) and using Equation 3.11, we have:
\[
u(\beta, z) = (2\pi)^{-1/2} \int_{-\infty}^{\infty} F(t) \beta^{-it-1/2} dt. \tag{3.15}\]

Equations 3.14 and 3.15 allow us to construct \( u(\beta, z) \) for \( \beta > 0 \) using \( u(ir, z), r > 0 \). With this information, we can also calculate \( v(\lambda, z) \) for all real values of \( \lambda \). Compared to the first method, the approach that uses Van Winter’s theorem provides a complete function of \( v(\lambda, z) \) at the cost of having to evaluate two improper integrals with complex components. Another consideration is that the values of improper integrals are often very sensitive to perturbations in their integrands. Since we plan to use experimental data, which typically include noise, the evaluation of these integrals can vary significantly based on the amount of noise.
CHAPTER 4

RECOVERING THE SOUND SPEED PROFILE USING THE REFLECTION COEFFICIENT

Our goal is the estimation of the SSP \( c(z) \) using measurable quantities from real experiments, namely, pressure field functions. The previous chapter focused on the recovery of the reflection coefficient \( R \) using experimental measurements of the pressure field \( u(r, z) \). The aim of this chapter is to lay down the necessary theoretical foundation for reconstructing the SSP from the knowledge of \( R \).

We have already shown in our initial direct formulation for a fluid half-space problem that Equation 2.15 is equivalent to the Jost equations (Equations 2.25 and 2.26) under a change of variable. The Jost equations represent the solutions of a one-dimensional quantum scattering problem on a line or a solution to a one-dimensional Schroedinger equation. Gelfand and Levitan presented foundational inverse theory for the quantum scattering problem, which was later expanded by Marchenko and Faddeev [14, 1, 10]. They suggested a method which recovers the potential on a line, \( q(z) \), from the reflection coefficient \( R \) using a linear integral equation, which was later termed the Gelfand-Levitan-Marchenko (GLM) equation. Deift and Trubowitz [5] provided a comprehensive treatment of the Jost equations in their discussion of the scattering problem on a line. One of their main results is an inverse formula, a nonlinear integral equation, which relates the potential \( q(z) \) to the reflection coefficient \( R \) and the scattering operator \( S \). Merab and Frisk [26] investigated the inverse problem of recovering the SSP in the ocean and adapted the GLM linear integral equation to solve the acoustic problem. In this work, we look at the application of the DT nonlinear integral equation for our acoustic inverse problem. Compared to the GLM equation, the DT equation is more complex, because
the integral is nonlinear. However, the simplicity of the GLM equation comes at a

cost: instead of the reflection coefficient, it uses its Fourier transform as its input.

Under a first order Born approximation, it can be shown that the inverse algorithm

using the GLM equation is equivalent to the one using the DT equation.

We first show the system of equations for the inverse algorithm using the DT

formula:

\[
q(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} [\lambda(1 - S(\lambda))]^{-e^{2i\lambda z}m^2(z, \lambda)}d\lambda, \quad (4.1)
\]

\[
m''(z, \lambda) + 2i\lambda m'(z, \lambda) = q(z)m(z, \lambda), \quad (4.2)
\]

\[
m(z, \lambda) \sim 1, \text{ as } z \to \infty. \quad (4.3)
\]

We can see that, employing this formulation, the inverse problem is to solve

a nonlinear system, with the DT formula being a nonlinear integral equation

(Equation 4.1), relating \(q(z)\) and \(m(z, \lambda)\). In the rest of this chapter, we will derive

the DT formula and discuss different ways of solving the problem.

4.1 The Deift-Trubowitz Trace Formula

To derive the DT trace formula, we start from Equations 2.25 and 2.26. We first

look at \(m_2(x, k)\), which satisfies:

\[
m''_2(z, \lambda) - 2i\lambda m'_2(z, \lambda) - q(z)m_2(z, \lambda) = 0, \quad (4.4)
\]

\[
m_2(z, \lambda) \sim 1, \text{ as } z \to -\infty, \quad (4.5)
\]

with \(m'(z, \lambda)\) denoting the derivative of \(m(z, \lambda)\) with respect to depth \(z\). The solution

\(m_2(z, \lambda)\) also satisfies the Volterra integral equation:

\[
m_2(z, \lambda) = 1 + \int_{-\infty}^{z} \frac{e^{2i\lambda(z-s)} - 1}{2i\lambda} q(s)m_2(s, \lambda)ds. \quad (4.6)
\]

Recall that the relationship between \(f\) and \(m\) is:

\[
f_{1,2}(z, \lambda) = m_{1,2}(z, \lambda)e^{\mp i\lambda z}, \quad (4.7)
\]

31
where
\[ f''_{1,2}(z, \lambda) = [q(z) - \lambda^2] f_{1,2}(z, \lambda). \] (4.8)

We know that \( f_{1,2}(z, \lambda) \) are a pair of independent solutions to Equation 2.25. We can see that \( f_2(z, \lambda) \) and \( f_2(z, -\lambda) \) are a pair of independent solutions as well, since their Wronskian is given by
\[ W[f_2(z, \lambda), f_2(z, -\lambda)] = f_2(z, \lambda) f'_2(z, -\lambda) - f'_2(z, \lambda) f_2(z, -\lambda). \] (4.9)

Note that the derivative of the Wronskian is:
\[ W' = f'_2(z, \lambda) f'_2(z, -\lambda) + f_2(z, \lambda) f''_2(z, -\lambda) 
- f'_2(z, \lambda) f'_2(z, -\lambda) - f''_2(z, \lambda) f_2(z, -\lambda) \]
\[ = 0. \] (4.10)

Here, we used Equation 4.8 to simplify \( f''(z, \pm \lambda) \). Since \( W' = 0 \), we know that the Wronskian is a constant. We can observe that, for \( z \to \infty \), the Wronskian becomes
\[ W[f_2(z, \lambda), f_2(z, -\lambda)] = \lim_{z \to \infty} [f_2(z, \lambda) f'_2(z, -\lambda) - f'_2(z, \lambda) f_2(z, -\lambda)] \]
\[ = e^{-i\lambda z} i\lambda e^{i\lambda z} + e^{i\lambda z} i\lambda e^{-i\lambda z}. \]
\[ = 2i\lambda \neq 0. \] (4.11)

In other words, \( W[f_2(z, \lambda), f_2(z, -\lambda)] = 2i\lambda \) for all \( z \). It follows that we can write \( f_1(z, \lambda) \) in terms of the independent solutions \( f_2(z, \lambda) \) and \( f_2(z, -\lambda) \). There are unique transmission and reflection coefficients \( T \) and \( R \) such that
\[ T(\lambda) f_1(z, \lambda) = R(\lambda) f_2(z, \lambda) + f_2(z, -\lambda). \] (4.12)
Lemma 2.9 [5] from Deift and Trubowitz’s treatment of the Jost equation provides the asymptotic forms for $m_1(z,\lambda)$, $m_2(z,\lambda)$, and $T(\lambda)$. These are:

\[
m_1(z,\lambda) = 1 + \frac{1}{2i} \int_{z}^{\infty} (e^{2i\lambda(t-z)} - 1)q(t)dt + \frac{1}{2(2i\lambda)^2} \left( \int_{z}^{\infty} q(t)dt \right)^2 + o\left( \frac{1}{\lambda^2} \right), \quad (4.13)
\]

\[
m_2(z,\lambda) = 1 + \frac{1}{2i} \int_{z}^{\infty} (e^{2i\lambda(z-t)} - 1)q(t)dt + \frac{1}{2(2i\lambda)^2} \left( \int_{z}^{\infty} q(t)dt \right)^2 + o\left( \frac{1}{\lambda^2} \right), \quad (4.14)
\]

\[
T(\lambda) = 1 + \frac{1}{2i} \int_{-\infty}^{\infty} q(t)dt + \frac{1}{2(2i\lambda)^2} \left( \int_{-\infty}^{\infty} q(t)dt \right)^2 + o\left( \frac{1}{\lambda^2} \right). \quad (4.15)
\]

The next step of the calculation is to multiply all three functions and collect the most significant terms. We have:

\[
T(\lambda)m_1(z,\lambda)m_2(z,\lambda) = 1 - \frac{2q(z)}{(2i\lambda)^2} + o\left( \frac{1}{\lambda^2} \right). \quad (4.16)
\]

Multiplying by $\lambda$ and integrating, we find the following:

\[
\frac{2i}{\pi} \int_{C_a} \lambda(Tm_1m_2 - 1)d\lambda = - \int_{C_a} \frac{q(z)}{i\lambda} d\lambda + \int_{C_a} o\left( \frac{1}{\lambda} \right)d\lambda \quad (4.17)
\]

\[
= q(z) + \int_{C_a} o\left( \frac{1}{\lambda} \right)d\lambda. \quad (4.18)
\]

The contour of integration is shown in Figure 4.1. Contour $C_a$ is semicircular and clockwise with radius $a$. When there are no trapped modes in $T(\lambda)$, there is no
singularity between \( C_a \) and \( C_0 \). In such cases, using Cauchy’s theorem, the contour integration of \( C_a \) can be replaced by integration on \( C_0 \), which is on the real axis. Therefore, we have:

\[
q(z) = \frac{2i}{\pi} \int_{-a}^{a} \lambda(Tm_1m_2 - 1) d\lambda + o(\frac{1}{a}). \tag{4.19}
\]

Setting \( a \to \infty \):

\[
q(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} \lambda(Tm_1m_2 - 1) d\lambda. \tag{4.20}
\]

We need to use Equation 4.12, which can be written as:

\[
T(\lambda)m_1(z, \lambda)e^{-i\lambda z} = R(\lambda)m_2(z, \lambda)e^{i\lambda z} + m_2(z, -\lambda)e^{i\lambda z}. \tag{4.21}
\]

Substituting \( Tm_1 \) into Equation 4.20, we get

\[
q(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} \lambda[R(\lambda)m_2(z, \lambda)e^{2i\lambda z} + m_2(z, -\lambda)e^{2i\lambda z}]m_2(z, \lambda) - 1) d\lambda
\]

\[
= \frac{2i}{\pi} \int_{-\infty}^{\infty} \lambda R(\lambda)m_2^2(z, \lambda)e^{2i\lambda z} + \lambda|m_2(z, \lambda)|^2 e^{2i\lambda z} - \lambda d\lambda
\]

\[
= \frac{2i}{\pi} \int_{-\infty}^{\infty} \lambda R(\lambda)m_2^2(z, \lambda)e^{2i\lambda z} d\lambda. \tag{4.22}
\]

The integration of term \( \lambda|m_2(z, \lambda)|^2 e^{2i\lambda z} - \lambda \) leads to zero, since this term is an odd function over the real axis. Equation 4.22 is referred to as the Deift-Trubowitz formula, also called the trace formula. This formula relates the potential \( q(z) \) to the reflection coefficient \( R(\lambda) \). The integral is nonlinear, since the term \( m_2(z, \lambda) \) is dependent on \( q(z) \) via Equation 4.4.

The DT formula can be reduced further, using a property of Hardy spaces, which we introduce below. Roughly, Hardy spaces result from an orthogonal deconstruction of the \( L^2 \) function space, where the set of square integrable \( L^2 \) functions is divided into two orthogonal sets, each with desirable properties. These smaller sets are the Hardy spaces. A proper definition of the Hardy spaces is the
following. Let $H^{2+}$ denote the Hardy space of function $h(k)$, which is analytic in $\Im k > 0$ with

$$\sup_{b>0} \int_{-\infty}^{\infty} |h(a+ib)|^2 da < \infty. \quad (4.23)$$

In $L^2(-\infty, \infty)$, function $h(k) \in H^{2+}$ assumes boundary value $h(a) = \lim_{\epsilon \to 0} h(a+i\epsilon)$. An alternative definition of $H^{2+}$ is the following:

$$H^{2+} = \{ h(k) \in L^2(-\infty, \infty) : \text{supp } \hat{h} \subset (-\infty, 0) \}. \quad (4.24)$$

Here, $\text{supp } \hat{h}$ denotes the support of $\hat{h}$ or the set of points on which $\hat{h}$ is nonzero; $\hat{h}$ denotes the Fourier transform of $h$, defined as:

$$\hat{h}(y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{2i ky} h(k) dk, \quad (4.25)$$

$$h(k) = \int_{-\infty}^{\infty} e^{-2i ky} \hat{h}(y) dy. \quad (4.26)$$

Similarly, $H^{2-}$ is the Hardy space of functions that are analytic in $\Im k < 0$ and

$$H^{2-} = \{ h(k) \in L^2(-\infty, \infty) : \text{supp } \hat{h} \subset (0, \infty) \}. \quad (4.27)$$

Spaces $H^{2+}$ and $H^{2-}$ are orthogonal and their complement is space $L^2$. By the above definition, a function $h(k)$ can be projected onto $H^{2-}$ or $H^{2+}$ using the following equations:

$$\tilde{h}(y) = \frac{1}{\pi} \int_{-\infty}^{\infty} e^{2i ky} h(k) dk, \quad (4.28)$$

$$h^-(k) = \int_{0}^{\infty} e^{-2iky} \tilde{h}(y) dy, \quad (4.29)$$

$$h^+(k) = \int_{-\infty}^{0} e^{-2iky} \tilde{h}(y) dy. \quad (4.30)$$

If a function is analytic only in the upper half complex plane, then this function is in the Hardy space $H^{2+}$ by definition. Integration of such a function on the real
axis vanishes as the integration limits go to infinity. An example of such a function is \( f(k) = e^{2ik} \). Observe that:

\[
\tilde{f}(y) = \frac{1}{\pi} \int_{-\infty}^{\infty} e^{2iky+2ik} dk = \delta(y + 1),
\]

\[
f^-(k) = \int_0^{\infty} e^{-2iky} \delta(y + 1) dy = 0, \tag{4.32}
\]

\[
f^+(k) = \int_{-\infty}^0 e^{-2iky} \delta(y + 1) dy = e^{2ik}. \tag{4.33}
\]

In other words, \( f(k) = e^{2ik} \) is in the \( H^{2+} \) space. A complex integral of \( f(k) \) would vanish exponentially in the upper half complex plane. Using the same contour of integration in Figure 4.1, we see that

\[
\lim_{a \to \infty} \int_{-a}^{a} e^{2ik} dk = \lim_{a \to \infty} -\int_{0}^{\pi} e^{2ia \cos \theta - 2R \sin \theta} Re^{i\theta} d\theta,
\]

\[
\leq \lim_{a \to \infty} \int_{0}^{\pi} |Re^{-2R}| d\theta = 0.
\]

Even though we only presented one example, it can be shown that contour integrals of functions in the \( H^{2+} \) space in the upper half complex plane would vanish; contour integrals of functions in the \( H^{2-} \) space in the lower half complex plane would vanish as well. For more information on Hardy spaces, we refer the reader to [9].

As previously mentioned, by using a Hardy space projection, the DT formula can be simplified further. It can be shown that only the Hardy projection onto the \( H^{2-} \) is needed. We first project \( \lambda R(\lambda) \) onto the two orthogonal Hardy spaces:

\[
q(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} \lambda R(\lambda)m_2^+(z, \lambda)e^{2i\lambda z} d\lambda \tag{4.34}
\]

\[
= \frac{2i}{\pi} \int_{-\infty}^{\infty} ([\lambda R(\lambda)]^+ + [\lambda R(\lambda)]^-)m_2^+(z, \lambda)e^{2i\lambda z} d\lambda. \tag{4.35}
\]

We quote Lemma 1.1 in Deift and Trubowitz [5], which shows that \( m_2(z, \lambda) \) is in \( H^{2+} \). We have shown earlier that \( e^{2i\lambda z} \) is also in \( H^{2+} \). This means that the
term $[\lambda R(\lambda)]^+ m_2^2(z, \lambda) e^{2i\lambda z}$ is in $H^{2+}$ and should integrate to zero. Then, the above integral can be reduced to:

$$q(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} [\lambda R(\lambda)]^{-} m_2^2(z, \lambda) e^{2i\lambda z} d\lambda.$$  \hspace{1cm} (4.36)

The reflection coefficient $R$ and the scattering operator $S$ can be related using the following method. Setting $z = 0$ in Equation 4.21, we have

$$T(\lambda) m_1(0, \lambda) = R(\lambda) m_2(0, \lambda) + m_2(0, -\lambda)$$  \hspace{1cm} (4.37)

$$\frac{T(\lambda) m_1(0, \lambda)}{m_2(0, \lambda)} = R(\lambda) + \frac{m_2(0, -\lambda)}{m_2(0, \lambda)}$$  \hspace{1cm} (4.38)

$$\frac{T(\lambda) m_1(0, \lambda)}{m_2(0, \lambda)} = R(\lambda) + S(\lambda),$$  \hspace{1cm} (4.39)

where we used the definition of the scattering operator $S(\lambda) = \frac{m_2(0, -\lambda)}{m_2(0, \lambda)}$. Also using the fact that $\frac{T(\lambda) m_1(0, \lambda)}{m_2(0, \lambda)} - 1 \in H^{2+}$, we have $R^- = [1 - S]^-$.

Using this relationship between $R$ and $S$, we finally obtain:

$$q(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} [\lambda[1 - S(\lambda)]^{-} m_2^2(z, \lambda) e^{2i\lambda z} d\lambda.$$  \hspace{1cm} (4.40)

### 4.2 Iterative Methods for Solving the Trace Formula

To streamline the following discussion, we will drop the subscript in $m_2(z, \lambda)$ and refer to it simply as $m(z, \lambda)$. The subscript in this section will instead represent the number of iteration steps.

The DT formula is a nonlinear integral equation. Assuming that the only data we have is $S(\lambda)$, which is typically the case, we still do not know $m(z, \lambda)$. We need to either have an initial guess or use an approximation. One method for addressing this problem is the Born approximation. It is a standard technique in quantum scattering problems for solving similar types of nonlinear integral equations. The idea of the Born approximation is the following. Since we know the far-field behavior for $m(z, \lambda)$ (which is 1 as $z \to \infty$), we can use this boundary condition as the initial
guess, setting $m_0(z, \lambda) = 1$. Physically, this is essentially the assumption that our total scattering field only consists of the incident field. Using this approximation for $m(z, \lambda)$, Equation 4.40 becomes:

$$q_0(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} \lambda [1 - S(\lambda)] e^{2i\lambda z} d\lambda. \quad (4.41)$$

This is now a linear integral equation and can be solved directly using the trapezoid rule or other numerical methods.

*Iterative algorithm 1:* The first order Born approximation can also be used as the starting point for an iterative algorithm. Substituting the first guess $q_0$ into the Jost ODE to solve for $m(z, \lambda)$, we can construct an iterative scheme as follows:

$$m''_n(z, \lambda) - 2i \lambda m'_n(z, \lambda) = q_n(z)m_n(z, \lambda) \quad n = 0, 1, 2, \ldots, \quad (4.42)$$

$$m_n(z, \lambda) \sim 1, \text{ as } z \to \infty, \quad (4.43)$$

$$q_{n+1}(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} \lambda [1 - S(\lambda)] e^{-2i\lambda z} m^2_n(z, \lambda) d\lambda. \quad (4.44)$$

This is a scheme that uses the first order Born approximation as the initial input.

*Iterative algorithm 2:* Another iterative scheme uses the Volterra integral equation for Jost function $m(z, \lambda)$. We set the initial guess for $m$ to be $m_0(z, \lambda) = 1$ and use Equation 4.41 as our starting point. The Volterra integral equation (Equation 4.6) is used to obtain the following:

$$m_1(z, \lambda) = 1 + \int_{-\infty}^{z} \frac{e^{2i\lambda(z-s)} - 1}{2i\lambda} q_0(s) ds. \quad (4.45)$$

Function $m_1(z, \lambda)$ is now an improved approximation compared to $m_0 = 1$. We can use this new value in the DT formula to calculate $q(z)$. Repeating this process, one can assemble the following iterative algorithm:

$$m_{n+1}(z, \lambda) = 1 + \int_{-\infty}^{z} \frac{e^{2ik(z-s)} - 1}{2ik} q_n(s)m_n(z, \lambda) ds \quad n = 0, 1, 2, \ldots, \quad (4.46)$$

$$q_{n+1}(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} \lambda [1 - S(\lambda)] e^{-2i\lambda z} m^2_{n+1}(z, \lambda) d\lambda, \quad (4.47)$$
where \( m_0(z, \lambda) = 1 \) and \( q_0 \) is given by Equation 4.41. Compared to the iterative algorithm 1, this algorithm is perhaps more suitable for numerical calculations. Many techniques for solving the Volterra integral equations can be applied here, such as the collocation and Galerkin methods [2, 3].

There are other iterative schemes for solving the DT system. Stickler suggested two algorithms in his work [36, 34]. However, all these iterative schemes involve solving the full Jost ODE or the Volterra integral equation at each iteration step and the computational cost can grow substantially. Especially when running large scale Monte Carlo simulations with noisy data, the iterative schemes can be very costly. In the next section we will focus on approximation techniques that do not require many iteration steps; we introduce approaches that aim at obtaining high accuracy out of the first one or two iterations.

### 4.3 Improving the Born Approximation

It has been shown that the first order Born approximation shown in Equation 4.41, while having only a first order accuracy, still captures the discontinuities in the inverse solution [43]. This simple approximation is sufficient in revealing the sharp changes and jumps in the SSP. It is the objective of inverse experiments to discover the location of such discontinuities in the SSP, because they represent the changes in sound speed in adjacent layers of the seabed. Often, the accuracy and the information about locations of discontinuities provided by the Born approximation is sufficient for practical uses. Due to its simplicity, the Born approximation is very computationally efficient and much faster than the iterative methods proposed in the previous section. For many applications where computational efficiency is a concern, the Born approximation is a good choice for improving calculation time, while still capturing useful information. With this advantage in mind, we introduce
three methods that are just as computationally efficient, while still improving upon numerical accuracy.

### 4.3.1 The Hardy Projection

Let us consider the DT formula (Equation 4.40), where a Hardy space projection is used to simplify the original form. A direct evaluation of the projection of \( R(k) \) onto the Hardy space would involve using the Hardy transform of Equation 4.30. However, we can still utilize Equation 4.40 without having to evaluate the transform formulae. For an even, real function \( g(x) = g(-x) \), the projections onto \( H^2^+ \) and \( H^2^- \) are the complex conjugates of each other. This fact can be proved as follows.

Let us consider the Hardy transform of \( g(-x) \) using Equation 4.30:

\[
\hat{g}(-y) = \frac{1}{\pi} \int_{-\infty}^{\infty} g(-k) e^{-2iky} \, dk = (\hat{g}(y))^* \\
g^-(k) = \int_{0}^{\infty} \hat{g}(y) e^{-2iky} \, dy \\
= \int_{0}^{-\infty} \hat{g}(-y) e^{2iky} \, d(-y) \\
= \int_{-\infty}^{0} (\hat{g}(y) e^{-2iky})^* \, dy \\
= (g^+ k)^*.
\]

This concludes our proof.

Recall the property of the scattering operator \( S(\lambda) \), where \( S^*(\lambda) = S(-\lambda) \), with \(^*\) denoting complex conjugate. We have shown that the Hardy projections of an even, real function are complex conjugates. Since Hardy spaces are orthogonal and complete complementary spaces, it must follow that \( \Re(g^-(k)) = \Re(g^+(k)) = \frac{1}{2} g(k) \).

By applying this observation to Equation 4.40 and letting \( R(k) = 1 - S(k) \), a modified
Born approximation can be constructed as follows:

\[
q(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} \lambda[R(\lambda)]^{-2i\lambda z} d\lambda
\]

\[
= -\frac{4}{\pi} \int_{0}^{\infty} \lambda\Im[R(\lambda)]^{-2i\lambda z} d\lambda
\]

\[
= -\frac{4}{\pi} \int_{0}^{\infty} \lambda\Re[R(\lambda)] \sin(2\lambda z) + \lambda\Im[R(\lambda)] \cos(2\lambda z) d\lambda
\]

\[
q(z) \approx -\frac{4}{\pi} \int_{0}^{\infty} \frac{\lambda}{2} \Re[1 - S(\lambda)] \sin(2\lambda z) + \lambda\Im[1 - S(\lambda)] \cos(2\lambda z) d\lambda.
\] (4.48)

We used the symmetry of \( R(\lambda) \) to simplify the integral formula. Quantity \( \Re[R(\lambda)] \sin(2\lambda z) \) is evaluated using Equation 4.48. On the other hand, \( \Im[R(\lambda)] \) is an odd function and can only be evaluated using the Hardy transform formulae; we want to obtain an approximation, so \( R(\lambda) \) is simply replaced by \( 1 - S(\lambda) \).

This modified Born approximation using a Hardy projection can be shown to be more accurate than the first order Born approximation given by Equation 4.41. Figures 4.2, 4.3, and 4.4 show the performance of this inverse method compared with other approximations. We can see that its accuracy is superior to that of the Born approximation, yet the numerical efficiency is similar.

### 4.3.2 Approximation Using an Interpolated \( m(z, \lambda) \)

The second method exploits our knowledge of the reflection coefficient \( R(\lambda) \) to improve upon the Born approximation. Recall that the Born approximation simplifies the DT formula by replacing \( m(z, \lambda) \) by its limiting value 1 when \( z \to \infty \). This approximation fails when \( z \) is close to zero. However, we do have information about the behavior of \( m(z, \lambda) \), when \( z \) is small. In fact, some knowledge about \( m(z, \lambda) \) when \( z = 0 \) is contained in the scattering operator \( S(\lambda) = f(0, -\lambda)/f(0, \lambda) \).

Consider the relation:

\[
S(\lambda) = \frac{|f(0, -\lambda)| e^{i\theta(\lambda)}}{|f(0, \lambda)| e^{-i\theta(\lambda)}} = e^{-2i\theta(\lambda)},
\] (4.49)

\[
m(0, \lambda) = |m(0, \lambda)| e^{i\theta(\lambda)}.
\] (4.50)
This relation shows that, given the scattering operator $S(k)$, we only need $|m(0, \lambda)|$ in order to recover $m(0, \lambda)$. Observe that

$$
\ln m(0, \lambda) = \ln |m(0, \lambda)| + i\theta(\lambda). 
$$

(4.51)

We now use the property of the Hilbert transform for analytic functions. Note that $|m(0, \lambda)|$ can be obtained by applying the Hilbert transform to $\theta(\lambda)$ to obtain $\ln |m(0, \lambda)|$, since $\theta(\lambda)$ and $\ln |m(0, \lambda)|$ are harmonic conjugates of each other. Once we recover $m(0, \lambda)$, we can use it to interpolate the behavior of $m(z, \lambda)$, when $z$ is small. We can define an interpolating function, $m^*(z, \lambda)$, as follows:

$$
m^*(z, \lambda) = \begin{cases} 
m(0, \lambda) + [1 - m(0, \lambda)](\frac{z}{L})^\alpha & \text{if } z \leq L , \\
1 & \text{if } z > L.
\end{cases}
$$

The above definition satisfies the condition $m(z, \lambda) \to 1$ as $z \to \infty$. For a choice of $L$ that is large enough, this definition forms a polynomial interpolation using $m(0, \lambda)$, where the shape of the interpolation is determined by the shape parameter $\alpha$. In our numerical simulation, the value of $\alpha$ is chosen to be 2 and $L$ is chosen to be 5, in order to approximate the true value of the function $m(z, \lambda)$. In practice, $\alpha$ and $L$ are the two parameters that need to be determined from a priori knowledge of the ocean. Given $m^*(z, \lambda)$, we can use it in the DT formula and evaluate the potential $q(z)$ using the following equation:

$$
q(z) = \frac{2i}{\pi} \int_{-\infty}^{\infty} \lambda [1 - S(\lambda)] e^{-2i\lambda z} m^2(z, \lambda) d\lambda. 
$$

(4.52)

Figures 4.2, 4.3, and 4.4 illustrate a comparison of the results obtained with the simple Born approximation, the Born approximation plus the Hardy space projection modification (Projection), and the method described in this section (Interpolant). These figures show that, with an appropriate choice of values for parameters $\alpha$
Figure 4.2 Comparison of three different ways for recovering \( q(z) \). The potential \( q(z) \) is a Pekeris profile.

and \( L \), the resulting solution is a great improvement over that of the two Born approximations. However, this improvement comes at the cost of having to specify these two parameters. In practice, if we could find two parameters that work for a particular type of environment, then this method could be an accurate and cost-effective way for extracting \( q(z) \).

The next step is to relate the potential \( q(z) \) with the SSP \( c(z) \). The relation between \( q(z) \) and \( c(z) \) is provided by the following simple change of variable:

\[
q(z) = \omega^2 (c_\infty^{-2} - c(z)^{-2}).
\]  

(4.53)

Solving for \( c(z) \) we have

\[
c(z) = (c_\infty^{-2} - q(z)\omega^{-2})^{-\frac{1}{2}},
\]  

(4.54)
Figure 4.3  Comparison of three different ways for recovering $q(z)$. The potential $q(z)$ is a two-step profile.

where $\omega$ is the angular frequency and $c_\infty$ is the constant that $c(z)$ approaches as $z$ goes to infinity. By using appropriate values for $\omega$ and $c_\infty$, this transformation can lead to recovering $c(z)$ using the potential $q(z)$. Examples of two results are shown in Figure 4.5. We omit the Born approximation here and only show the other two approximation methods.
Figure 4.4  Comparison of three different ways for recovering $q(z)$. The potential $q(z)$ is chosen to be an artificial profile to show that the methods can capture discontinuities.
Figure 4.5  Using the DT inverse algorithm to recover $c(z)$. The graphs are for (a) a simple Perkeris profile and (b) a two-step profile.
CHAPTER 5

NUMERICAL TECHNIQUES FOR THE SEDIMENT SOUND SPEED INVERSE PROBLEM

The aim of this chapter is to describe some of the numerical techniques used in evaluating various integrals in the sediment sound speed inverse problem. We will first describe the numerical techniques for recovering the reflection coefficient. We will then discuss how to evaluate the Hankel transform integrals and, finally, the DT integral formula.

5.1 The Scattering Operator \( S(\lambda) \)

For simulations, we first assume an SSP \( c(z) \), which is used to find the scattering operator \( S(\lambda) = f(0, -\lambda)/f(0, \lambda) \). The reflection coefficient is not found directly, because the scattering operator \( S(\lambda) \) is sufficient for the DT formula and, at the same time, is easier to evaluate numerically.

To evaluate \( S(\lambda) = f(0, -\lambda)/f(0, \lambda) \), we start from the Jost equation. The potential \( q(z) \) is related to the SSP by the following equation:

\[
q(z) = \omega^2 (c_{\infty}^{-2} - c(z)^{-2}),
\]

where \( \omega \) is the source frequency and \( c_{\infty} \) is the limiting value of \( c(z) \) as \( z \to \infty \). The Jost equation is:

\[
m_1''(z, \lambda) + 2i\lambda m_1'(z, \lambda) = q(z)m_1(z, \lambda), \quad (5.2)
\]

\[
m_1(z, \lambda) \sim 1, \text{ as } z \to \infty. \quad (5.3)
\]

The simplest way to solve this is by using the finite difference (FD) method. Before proceeding with the FD scheme, it is worth noting that the boundary condition of our
ODE cannot be used directly. We approximate the boundary condition by setting a value \( L \) to be large enough, so that \( m_1(L, \lambda) = 1 \) and \( m'_1(L, \lambda) = 0 \). We would then have \( z \) moving from \( L \) to 0. In this case, a convenient change of variable would be \( \zeta = L - z \) and the new ODE in terms of this variable would be:

\[
\begin{align*}
   m''_1(\zeta, \lambda) - 2i\lambda m'_1(\zeta, \lambda) &= q(L - \zeta)m_1(\zeta, \lambda), \\
   m_1(\zeta = 0, \lambda) &= 1, \\
   m'_1(\zeta = 0, \lambda) &= 0.
\end{align*}
\] (5.4)

(5.5)

(5.6)

Setting \( m'' = (m^{i+1} - 2m^i + m^{i-1})/\Delta \zeta \) and \( m' = (m^{i+1} - m^{i-1})/(2\Delta \zeta) \), we obtain the following FD scheme:

\[
(1 - i\Delta z \lambda)m^{i+1} - (2 + \Delta z^2 q^i)m^i + (1 + i\Delta z \lambda)m^{i-1} = 0. 
\] (5.7)

The initial condition is \( m^0 = m^1 = 1 \). Here, \( m^i \) represents \( m_1(\zeta = i\Delta \zeta, \lambda) \) and \( q^i \) is \( q(z = L - i\Delta \zeta), i = 0, 1, 2, ..., N \), where \( N = L/\Delta \zeta \). The last term calculated using this FD scheme is \( m^N = m_1(0, \lambda) \) and the scattering operator can be computed using \( S(k) = m_1(0, \lambda)/m_1(0, \lambda) \).

If higher accuracy is required, the step size \( \Delta \zeta \) needs to be chosen to be very small, in which case a higher order scheme, such as the Runge-Kutta method, can provide more accurate results.

### 5.2 The Hankel Transform Integral

The use of the Hankel transform was a critical step in the inverse algorithm, transforming the Helmholtz equation into a two-dimensional ODE. This simplification comes at the cost of having to evaluate the Hankel transform integral. There are many available numerical packages that provide Fast Fourier Transform (FFT) algorithms but very few compute the Hankel transform. Fortunately, there are several ways to modify the Hankel transform in such a way that the FFT can be utilized to
speed up the computation. Our discussion in this section will follow work presented by Guizar-Sicairos [17]. Recall the definitions of the Hankel transform and the inverse Hankel transform:

\[
\begin{align*}
  u(r) &= \int_{\beta=0}^{\infty} v(\beta) J_0(\beta r) \beta d\beta, \quad (5.8a) \\
  v(\beta) &= \int_{r=0}^{\infty} u(r) J_0(\beta r) r dr. \quad (5.8b)
\end{align*}
\]

Variables \( r \) and \( \beta \) have to first be limited in a finite region, as required for the numerical computations. Therefore, we specify that

\[
u(r \geq R) = 0, v(\beta \geq B) = 0. \quad (5.9)\]

Here, \( R \) and \( B \) are the truncated radii of the spatial variable \( r \) and the frequency variable \( \beta \), respectively. The next step is to expand \( u(r) \) in terms of the 0\textsuperscript{th}-order Bessel series:

\[
u(r) = \sum_{m=1}^{\infty} c_m J_0(\alpha_m \frac{r}{a}), 0 \leq r \leq a, \quad (5.10)\]

where \( \alpha_m \) is the \( m \)th root of the 0\textsuperscript{th}-order Bessel function and the coefficient \( c_m \) is determined by

\[
c_m = \frac{1}{a^2 J_1(\alpha_m)} \int_0^{a} u(r) J_0(\alpha_m \frac{r}{a}) r dr. \quad (5.11)\]

If the radius \( r \) is evaluated at \( \alpha_n/(2\pi B) \) and the frequency \( \beta \) at \( \alpha_m/(2\pi R) \), then we can approximate Equation 5.8 by the following discrete sums:

\[
u\left( \frac{\alpha_n}{2\pi B} \right) = \frac{1}{\pi R^2} \sum_{m=1}^{N} \frac{v(\alpha_m/(2\pi R))}{J_1^2(\alpha_m)} J_0(\alpha_n \alpha_m \frac{\alpha_n}{S}), \quad (5.12a)\]

\[
u\left( \frac{\alpha_m}{2\pi R} \right) = \frac{1}{\pi B^2} \sum_{n=1}^{N} \frac{u(\alpha_n/(2\pi B))}{J_1^2(\alpha_n)} J_0(\alpha_n \alpha_m \frac{\alpha_n}{S}), \quad (5.12b)\]
where \( S = 2\pi RB \). We can rewrite these equations by defining vectors

\[
F_u(n) = u(\alpha_n/(2\pi B))|J_1(\alpha_n)|^{-1}R, \quad (5.13a)
\]
\[
F_v(m) = v(\alpha_m/(2\pi R))|J_1(\alpha_m)|^{-1}B. \quad (5.13b)
\]

These reduce Equation 5.8 to

\[
F_u(n) = \sum_{n=1}^{N} T_{mn} F_v(m), \quad (5.14a)
\]
\[
F_v(m) = \sum_{m=1}^{N} T_{nm} F_u(n), \quad (5.14b)
\]

where

\[
T_{mn} = \frac{2J_0(\alpha_m\alpha_n/S)}{|J_1(\alpha_m)||J_1(\alpha_n)|S} \quad (5.15)
\]

represents an element of an \( N \times N \) transformation matrix \( T \). A convenient feature of this matrix is that its values can be pre-computed for a fixed number of points. Once the matrix is computed, a Hankel transform or inverse Hankel transform can be calculated via a simple matrix multiplication.

### 5.3 Integral with a Highly Oscillatory Integrand

The evaluation of the DT integral formula requires some care. We show the graph of a typical integrand of the DT integrand in Figure 5.1. For such a highly oscillatory function, a direct application of lower order methods, such as the trapezoid and Simpson’s methods, would give poor results or require a very fine grid size for an acceptable performance. A standard method relying on quadrature will also fail.

In general, we want to evaluate integrals of the following form:

\[
I(k) = \int_{-\infty}^{\infty} e^{ikx} f(x) dx. \quad (5.16)
\]
The highly oscillatory integrand, $\lambda R(\lambda) e^{2i\lambda z} m^2(z, \lambda)$, from the DT formula. The potential is that of a Pekeris profile. Here, $z = 5$, $q^* = -2.63 \times 10^{-4}$, and $L = 40$.

This integral converges if $\int_{-\infty}^{\infty} f(x)dx$ converges absolutely. When $k$ has a large value, the integrand oscillates with high frequency and normal quadrature methods fail. To deal with this problem, observe that the integrand has period $2\pi/k$. We can write Equation 5.16 as:

$$I(k) \approx \int_{-L}^{L} e^{i k x} f(x) dx,$$

(5.17)

$$= \sum_{n=-N}^{N-1} \int_{2n\pi/k}^{2(n+1)\pi/k} e^{i k x} f(x) dx,$$

(5.18)

where $N = \frac{Lk}{2\pi N}$. Parameter $L$ is usually chosen in such a way that $N$ is an integer. Each of the summation terms can then be evaluated using a standard quadrature method.
In this chapter, we will test the inversion algorithm with synthetic noisy data. The data are generated using experiment 3 described in Chapter 3. First, we will discuss the data generation. Then, we will present simulation results with the synthetic data and discuss the sensitivity problems we encounter. Finally, we will present some ideas that can help address the sensitivity concerns.

6.1 Data Generation

A Monte Carlo simulation using synthetic noise was carried out to evaluate the performance of the inverse algorithm. We start the data generation by solving the forward problem described in Equation 3.1. In this equation, the function \( g(z, z_0, k) \) was obtained by taking the Hankel transform of the pressure field function \( p(r, z) \). We do not calculate \( p(r, z) \) directly. Instead, we find \( g(z, z_0, k) \) numerically; its Hankel transform provides \( p(r, z) \). In our numerical experiment, the SSP \( c(z) \) for depth \( z < 20 \text{ m} \) is assumed to be known. The source and receiver locations are set to be at the same depth, \( z = 20 \text{ m} \). The source frequency is chosen to be 10 Hz. The measurement of the pressure field is made at receiver \( z = 20 \text{ m} \) for all \( r \).

The next step is to add noise to these generated data. Normally distributed noise with zero mean and different values for variance are added to function \( g(z, z_0, k) \). We do not add the noise to the pressure field function \( p(r, z) \) and then take the Hankel transform to obtain \( g(z, z_0, k) \), which would have been the case if real data were used. The reasons for this are the following. One, having to perform the Hankel transform for every realization would take a significant amount of time; two, Mook [30] has shown that the addition of zero-mean stationary white Gaussian noise to the point source pressure field would result in a Hankel transform with a non-stationary
variance. It is a good approximation in our case to generate \( g(z, z_0, k) \) with zero mean Gaussian noise.

Merab [26] performed a similar analysis by adding Gaussian noise directly to the reflection coefficient. He showed that his algorithm is stable using those data. However, Gaussian noise added to the reflection coefficient does not correspond to Gaussian noise in the pressure field function \( p(r, z) \); thus, this analysis may not be helpful in practical applications. In a real experiment, we would collect data with noise in the pressure field measurement. To see the effect of noise on the reflection coefficient \( R \), we would have to examine the process of recovering \( R \) as described in experiment 3 of Chapter 3.

6.2 Effect of Noise on the Recovery of the Reflection Coefficient

To consider the effect of additive noise on the Green’s function on the recovery of the reflection coefficient, we start by using Equation 3.6. We can rewrite the equation as:

\[
|U(0, k)| = \sqrt{-kV(z_>, k)V(z_<_k, k)} / \Im g(z, z^*, k). \tag{6.1}
\]

Now we replace \( \Im g \) with \( \Im g + \delta g \), where \( \delta g \) is the noise on the measured data and \( \Im g \) is the exact true value of the data. This results in a new \( \hat{U} \). We can then use a Taylor series expansion to obtain the following:

\[
|\hat{U}(0, k)| - |U(0, k)| = \sqrt{-kV(z_>, k)V(z_<_k, k)\delta g(z, z^*, k)} / \Im g(z, z^*, k)^2 + O(\delta g). \tag{6.2}
\]

The recovery of \( |U(0, k)| \) becomes problematic when the denominator of the right hand side of Equation 6.2 is small. As an example, Figure 6.1 shows the shape of the imaginary part of \( g(z, z_0, k) \) as a function of the wavenumber \( k \), with fixed receiver depth \( z \) and source depth \( z^* \). At about \( k = 7 \), \( \Im g(z, z^*, k) \) is close to zero; a slight perturbation on \( \Im g(z, z^*, k) \) can bring it to zero and, as a result, the effort to recover
$|U(0, k)|$ would encounter a singular point. As we can see, $\Im g(z, z^*, k)$ oscillates around zero at several locations for larger values of $k$. Figure 6.2 illustrates the $|U(0, k)|$ recovered using Equation 6.1 directly, where large spikes appear whenever $\Im g(z, z^*, k)$ crosses zero. Figure 6.3 shows the result for the scattering operator. This result is unacceptable as input for the DT formula.

![Figure 6.1](image)

**Figure 6.1** The imaginary part of the noise-free Green’s function $g(k)$. The function crosses zero several times. The variables $z$ and $z^*$ are fixed.

To reduce the disruptive effect of the singular points, we can use the following method. We first rewrite Equation 6.1 as:

$$W(k) = \frac{1}{|U(0, k)|^2} = \frac{\Im g(z, z^*, k)}{-kV(z^+, k)V(z^-, k)}.$$  \hspace{1cm} (6.3)

This form is easier to handle, since, when $\Im g(z, z^*, k)$ is in the numerator, we do not have a singular point. We apply two restrictions to Equation 6.3 when noise is added to $\Im g(z, z^*, k)$. First, we force $W(k)$ to be positive to ensure $|U(0, k)|$ is strictly a real function. This can be done by simply taking the absolute value of $W(k)$. Second,
Figure 6.2 Function $|m(0,k)|$ recovered using a noisy Green’s function measurement. The measurement was simulated by adding a small normal noise component with zero mean and variance $1 \times 10^{-3}$ to the noise-free $\Im g$. Here we show the difference between recovering $m(0,k)$ with regularization and without for data with 30 dB noise.

we force the function $W(k)$ to smoothly decay to one as $k \to \infty$. This can be done by choosing a value $k_0$, where $k > k_0$. Then, $W(k)$ is replaced by a sigmoid function that smoothly goes to one. One such choice is the following:

$$W(k) = 2 - \frac{1}{1 + e^{b - ak}} \quad \text{for} \quad k > k_0,$$

where the one free parameter $a$ can be adjusted to control the decay rate of the fitting function. We chose $a = 30$ and $k_0 = 0.13$ for our numerical simulations. By applying these two steps, the algorithm can handle significantly more noise, but at the cost of using a interpolating function that may not represent the true data. Finally, we
Figure 6.3  Recovering the scattering operator $S(k)$ using $|m(0, k)|$ from Figure 6.2. Large spikes occur near the singular points. Here we show the difference between recovering $S(k)$ with regularization and without for data with 30 dB noise.

retrieve $|U(0, k)|$ by setting:

$$|U(0, k)| = \frac{1}{\sqrt{W(k)}}. \quad (6.5)$$

The scattering operator $S(k)$ can then be recovered using the Hilbert transform.

6.3 Simulation Results

We will now show the performance of the algorithm with three examples. We synthetically generate data with signal-to-noise ratio (SNR) of 50, 40, and 30 dB. We run 300 realizations for each data set. Without the “regularization” method, the algorithm will fail to produce sensible results even for data with SNR of 50 dB. We will only show results produced using the regularization techniques described in the previous section.
Three representative profiles are chosen for the testing of our algorithm. We define them as follows:

(a) Pekeris profile:

The Pekeris profile is defined by the following function \( c(z) \):

\[
c(z) = \begin{cases} 
  c_1, & 0 < z < L, \\
  c_\infty, & L < z.
\end{cases}
\]  

(6.6)

In our simulations we used \( c_1 = 1500 \text{ m/s} \), \( c_\infty = 2000 \text{ m/s} \), and \( L = 50 \text{ m} \). The analytical solution for this profile is given in Section 2.3. We can see the simulation results in Figure 6.4.

(b) Step profile:

The Step profile is an extension of the Pekeris Profile, defined as:

\[
c(z) = \begin{cases} 
  c_1, & 0 < z < L_1, \\
  c_2, & L_1 < z < L_2, \\
  c_3, & L_2 < z < L_3, \\
  c_\infty, & L_3 < z.
\end{cases}
\]  

(6.7)

Our simulation uses the following parameters: \( c_1 = 1500 \text{ m/s} \), \( c_2 = 1600 \text{ m/s} \), \( c_3 = 1800 \text{ m/s} \), \( c_\infty = 2000 \text{ m/s} \), \( L_1 = 30 \text{ m} \), \( L_2 = 40 \text{ m} \), and \( L_3 = 50 \text{ m} \). The simulation results are shown in Figure 6.5.

(c) Test profile:

In the third example, we define

\[
c(z) = \begin{cases} 
  c_1, & 0 < z < L_1, \\
  c_1 + (c_2 - c_1) \frac{z - L_1}{L_2 - L_1}, & L_1 < z < L_2, \\
  c_2, & L_2 < z < L_3, \\
  c_\infty, & L_3 < z.
\end{cases}
\]  

(6.8)
where \( c_1 = 1800 \, \text{m/s} \), \( c_2 = 1700 \, \text{m/s} \), \( c_\infty = 2000 \, \text{m/s} \), \( L_1 = 20 \, \text{m} \), \( L_2 = 25 \, \text{m} \), and \( L_3 = 30 \, \text{m} \). The simulation results are illustrated in Figure 6.6.

**Figure 6.4** Simulations for the Pekeris profile with different noise levels. The plots show the recovered SSP results for SNR levels of 50, 40, and 30 dB, as well as the true SSP.

We only show the simulation results for the SSP for \( z > 20 \), since we assume the profile to be known in the region \( z < 20 \). In these figures, the estimated SSPs are not exact replicas in the true SSP, even if there is no noise. The reason is two-fold: one, a linear approximation is applied as described in Section 4.3.2 for solving the DT trace formula. Two, in our regularization method, another layer of approximations is added. As a result, the accuracy of the recovered SSP is sacrificed (but noise is easier to handle). However, the discontinuities and general shape of the SSP are still retained even when the data are distorted by noise. Further numerical treatments
Figure 6.5 Simulations for the step profile with different noise levels. The plots show the recovered SSP results for SNR levels of 50, 40, and 30 dB, as well as the true SSP.

are necessary to create algorithms that can produce results with higher accuracy while also handling lower SNRs.
Figure 6.6  Simulations for the test profile with different noise levels. The plots show the recovered SSP results for SNR levels of 50, 40, and 30 dB, as well as the true SSP.
CHAPTER 7

INVERSION FOR SOUND SPEED IN THE WATER COLUMN USING EMPIRICAL ORTHOGONAL FUNCTIONS AND LINEARIZATION

In the past, inversion for the estimation of ocean properties was typically performed using methods falling under the category of matched-field inversion (MFI), as mentioned in the introduction. MFI approaches are based on the signal processing technique known as matched-field processing, which was originally used to locate a passive acoustic source using an array of receivers. In MFI, the forward problem is solved repeatedly and model parameters are adjusted, until the forward problem solution replicates the data. The estimates of the unknown parameters are those values that provide the best match. This is an optimization problem with the goal to find a set of parameter values in a multi-dimensional search space. There are many search techniques developed for MFI, including neural networks [31], simulated annealing [4], genetic algorithms [15, 16], Gibbs Sampling [6], and Tabu [27]. Although such strategies accelerate the search, thus, improving efficiency, techniques that are based on MFI inherently require intensive calculations: the common challenge they face is computational cost.

In this chapter, we look into a different method for solving the inverse problem in ocean acoustics, specifically for estimating the sound speed in the water column along with the water column depth and source location. The proposed method is computationally efficient requiring simple calculations and a few iterations. It employs linearization and Empirical Orthogonal Functions (EOFs).
7.1 Empirical Orthogonal Functions

The concept of EOFs was introduced by LeBlanc and Middleton [22]. EOFs are eigenvectors of the sound speed covariance matrix. Roughly speaking, when many sound speed measurements are available, their mean can be calculated and subtracted from each measured SSP. The matrix resulting from the multiplication of such differences is the covariance matrix to which we are referring. To facilitate the discussion of EOFs, we define the following terms:

\[ Z = [z_1, z_2, \ldots, z_N], \quad (7.1) \]

\[ C = \begin{pmatrix} C_1 \\
C_2 \\
\vdots \\
C_N \end{pmatrix}. \quad (7.2) \]

Vector \( Z \) contains \( N \) predetermined standard depths. Vector \( C \) is the sound speed for the \( N \) depths: each \( C_i \) is the sound speed at depth \( z_i \). Let \( \overline{C} \) be the mean SSP vector. We define \( S \) to be the excess SSP, which is the vector after the mean \( \overline{C} \) is subtracted:

\[ S = C - \overline{C}. \quad (7.3) \]

To derive the EOFs, we express the excess SSP vector \( S \) using the following decomposition,

\[ S = HA, \quad (7.4) \]

where the columns of \( H \) form a set of \( N \) orthonormal basis vectors spanning the space of the columns of \( S \) and \( A \) is a set of linear weighting coefficients corresponding to each of the basis vectors. Since \( H \) contains a set of orthonormal vectors, it satisfies:

\[ H^T H = H H^T = I, \quad (7.5) \]
where $I$ is the identity matrix. To compute $A$, we note that

$$H^T S = H^T H A = A. \quad (7.6)$$

By the above definition, the orthonormal basis in $H$ is not unique. To determine a unique basis, we employ a covariance matrix. Let $\langle AA^T \rangle$ denote the covariance matrix of $A$. Statistical independence will be achieved if:

$$\langle AA^T \rangle = D. \quad (7.7)$$

Let $\Sigma$ denote the covariance matrix of the excess SSP data:

$$\Sigma = \langle SS^T \rangle. \quad (7.8)$$

We can combine Equations 7.4 and 7.8 to obtain:

$$\Sigma = \langle HA(HA)^T \rangle = H\langle AA^T \rangle H^T, \quad (7.9)$$

$$D = H\Sigma H^T. \quad (7.10)$$

Equations 7.8 and 7.10 provide the conditions we enforce to ensure that the orthonormal vector set in $H$ is unique. Note that we can use Singular Value Decomposition in Equation 7.10, which will yield $N$ pairs of eigenvalues and eigenvectors. In several applications (similar to ours), it can be shown that the first three or four eigenvectors can account for most of the total variance. Considering three eigenvectors, we can approximate our SSP vector $c$ as follows:

$$c = \bar{c} + \mu_1 v_1 + \mu_2 v_2 + \mu_3 v_3, \quad (7.11)$$

where $\bar{c}$ is the mean SSP, $v_1$, $v_2$, and $v_3$ are the first three computed eigenvectors, or EOFs, and $\mu_1$, $\mu_2$, and $\mu_3$ are the corresponding eigenvector coefficients. This
approximation of the SSP makes inversion (with linearization, in our case) much simpler: instead of possible vector SSP variations along all depths $z$, now the variation is only dependent on three scalar unknowns, $\mu_1, \mu_2,$ and $\mu_3$.

### 7.2 The Linear System

For the linearization inversion method, we consider three ray paths for source localization, water column depth estimation, and SSP inversion: the direct ray (D), the first surface reflection path (SR), and the first bottom reflection path (BR). A sketch of our problem with the three considered paths is shown in Figure 7.1. Previously, similar methods had been employed for the estimation of bias in sound speed [8] or for isovelocity profile estimation, but not for complete sound speed velocity profile estimation in general situations. The reason is that it is difficult to invert for an entire depth-dependent SSP. To use a linearization scheme, one must incorporate the features of the SSP into the Jacobian matrix, but it is difficult to do so when the SSP is a complete function of depth. We bypass this problem by considering the SSP as a linear combination of EOFs. By doing so, the elements in the Jacobian matrix would be just derivatives with respect to the coefficients of the EOFs.

As also explained in Refs. [8] and [29], Equation 7.12 below reflects the fact that the arrival time $t$ of each path is a function of the source location (range $r$ and source depth $z_s$), water column depth $WD$, sound speed $c(z)$, and transmission instant $t_0$ for the underwater problem of interest:

$$ t = \tau(r, z_s, WD, c(z)) + t_0, \quad (7.12) $$

where $\tau$ represents the ray travel time.

Generalizing, we can write:

$$ t = f(q), \quad (7.13) $$
where \( t \) is the vector of measured/estimated travel times. Function \( f \) represents the forward or acoustic model that relates the measurements to a set of parameters and \( q \) represents the vector containing the parameters to be estimated:

$$ q = [r, z_s, WD, c(z), t_0]. $$

(7.14)

Other parameters such as sound speed and thickness of sediments enter Equation 7.12, when rays that have interacted with the sediments are considered.

For our problem, the signal is received at \( N_h \) hydrophones of a vertical line array (VLA). If the three characteristic ray paths of Figure 7.1 (D, SR, and BR) are employed, that is, \( K = 3 \), where \( K \) is the number of paths, there will be a total of \( KN_h = 3N_h \) arrival time measurements, which serve as our data.

For our acoustic inverse problem, vector \( q \) is estimated using the measured travel times and forward model \( f \), relying on ray theory. Because of the nature of our data (time) and its relationship to the geometry of the problem and also water column
sound speed, the inverse process in our case is quasi-linear. An effective approach to solving the inverse problem in this case is local linearization and iteration. Linearizing the problem locally has been shown to lead to accurate solutions in an efficient manner [8, 7, 29].

A linear approximation to Equation 7.13 can be obtained as:

$$t = f(q_0) + J\delta q,$$  \hspace{1cm} (7.15)

where $q_0$ is a vector of initial conditions for $q$, $\delta q$ is the model perturbation that provides a “correction” to the initial model parameters as will be explained below, and $J$ is the Jacobian matrix which contains the partial derivatives of time with respect to the unknown parameters for each path. That is:

$$J = \begin{pmatrix}
\frac{\partial t_1}{\partial q_1} & \frac{\partial t_1}{\partial q_2} & \cdots & \frac{\partial t_1}{\partial q_M} \\
\frac{\partial t_2}{\partial q_1} & \frac{\partial t_2}{\partial q_2} & \cdots & \frac{\partial t_2}{\partial q_M} \\
\cdots & \cdots & \cdots & \cdots \\
\frac{\partial t_{KN_h}}{\partial q_1} & \frac{\partial t_{KN_h}}{\partial q_2} & \cdots & \frac{\partial t_{KN_h}}{\partial q_M}
\end{pmatrix}.$$  \hspace{1cm} (7.16)

Vector $q$ consists of $M$ variables.

By introducing $\delta t = t - f(q_0)$, Equation 7.15 leads to

$$J\delta q = \delta t.$$  \hspace{1cm} (7.17)

Equation 7.17 reflects a linear relationship between arrival time differences and perturbations of parameters $q$.

More specifically, quantities $\delta t$ of Equation 7.17 represent the differences between path arrivals in the real signals and replica signals generated using ray theory for a set of initial values for the unknown parameters. Through Equation 7.17, corrections $\delta q$ for the unknown parameters are obtained that provide a better match between real and replica times. The system needs to be solved iteratively until
the solution converges. At every step, the results from the previous iteration are employed as the new initial conditions.

Equation 7.17 generally leads to an overdetermined linear system (for our data we consider $N_h=14$ hydrophones and $K = 3$ arrival times for each phone; we are inverting for seven parameters - source range and depth, water depth, three EOF coefficients, and time instant for synthetic data and we also include tilt for the real data inversion). Least squares can be used for the solution of the system; some regularization methods can be combined with the least squares to combat the instability brought by the noise in the data.

In the following sections, we will describe how to solve the forward problem, as well as how to calculate the terms in the Jacobian matrix.

### 7.3 Solving the Forward Problem

The previous section mentioned the necessity to solve the quasi-linear system iteratively until we have convergence. Thus, the forward problem also needs to be solved for several iterations. Specifically, we need to generate the arrival time vector $t$ when the parameter values in vector $q$ are known. Recall from the standard theory of ray methods in stratified media [20], the range $r$ satisfies the following expression:

$$r(z) = r(z_0) + \int_{z_0}^{z} \frac{pc(s)}{\sqrt{1 - p^2 c^2(s)}} ds,$$

(7.18)

where $c(z)$ is the sound speed as a function of depth $z$, $p$ is the ray parameter, $z_0$ is the depth for the sound source, and $r(z)$ is the range for the receiver at depth $z$. To derive the phase equation, which has a similar representation as an integral with respect to $z$, we use the following fact:

$$ds = \sqrt{(dr)^2 + (dz)^2} = \frac{1}{\sqrt{1 - p^2 c^2(z)}} dz.$$  

(7.19)
Quantity \( dr/dz \) can be obtained by differentiating Equation 7.18, leading to:

\[
ds = \sqrt{1 + \left( \frac{dr}{dz} \right)^2} dz.
\]  

(7.20)

We quote the standard phase equation:

\[
\tau(s) = \tau(0) + \int_0^s \frac{1}{c(s')} ds',
\]  

(7.21)

which becomes

\[
\tau(z) = \tau(z_0) + \int_{z_0}^z \frac{1}{c(z')} \sqrt{1 - p^2 c^2(z')} dz'.
\]  

(7.22)

We employ Equations 7.18 and 7.22 for generating arrival time vector data \( t \), when vector \( q \) is known.

### 7.3.1 Evaluating the Ray Parameter \( p \)

An important step in the inverse algorithm is to evaluate the ray parameter \( p \) for a particular ray path (where the range, source depth, and receiver depth are known). We know that these parameters are related via Equations 7.18 and 7.22. One useful technique for recovering the ray parameter \( p \) is by using Newton’s method [8].

We first consider an initial guess on the value of the ray parameter \( p \). In [8], the SSP \( c(z) \) was set to its harmonic mean \( c_H \):

\[
c_H = \frac{1}{z - z_0} \int_{z_0}^z \frac{1}{c(s)} ds.
\]  

(7.23)

An initial guess for \( p \), \( p_0 \), can be obtained by employing Equation 7.18, with \( c(z) \) replaced by \( c_H \). We then have:

\[
p_0 = \left( c_H \left( \frac{1}{1 + \left( \frac{z - z_0}{r} \right)^2} \right) \right)^{-1}.
\]  

(7.24)
Next, we aim to improve the guess by taking the Taylor expansion of \( r(p) \) and keeping only the first order term:

\[
\begin{align*}
    r(p) &= r(p_0) + \frac{\partial r(p_0)}{\partial p}(p - p_0). \\
    \text{(7.25)}
\end{align*}
\]

Solving for \( p \), we obtain:

\[
\begin{align*}
    p &= p_0 + \left[ \frac{\partial r(p_0)}{\partial p} \right]^{-1}(r(p) - r(p_0)), \\
    \text{(7.26)}
\end{align*}
\]

or, equivalently,

\[
\begin{align*}
    p_{i+1} &= p_i + \left[ \frac{\partial r(p_i)}{\partial p} \right]^{-1}(r(p) - r(p_i)). \\
    \text{(7.27)}
\end{align*}
\]

Derivative \( \frac{\partial r(p_i)}{\partial p} \) is evaluated by differentiating Equation 7.18 according to Leibniz’s rule:

\[
\begin{align*}
    \frac{\partial r(p_i)}{\partial p} &= \int_{z_0}^{z} \frac{c^2(z')}{(1 - p_i^2 c^2(z'))^{3/2}} dz'. \\
    \text{(7.28)}
\end{align*}
\]

We repeat the calculation of Equation 7.27 in order to improve \( p_i \) until a desired tolerance \( \epsilon \) is reached: \( |r(p_{i+1}) - r(p_i)| < \epsilon \). In practice, only a few iterations using the above formulation are needed for convergence.

Sometimes, the process will fail when we start with \( c_H \), because \( 1 - p_0^2 c_H^2 \leq 0 \). In such cases, we suggest selecting an initial guess \( p_0 \) as follows:

\[
\begin{align*}
    p_0 &= \frac{1}{\max(c(z))} + \epsilon, \\
    \text{(7.29)}
\end{align*}
\]

where \( \epsilon \) is set to a small number. The rationale is that \( p_0 = \frac{1}{\max(c(z))} \) is the smallest possible value that satisfies \( 1 - p_0^2 c_H^2 \leq 0 \). Doing so will at least start the algorithm successfully and Newton’s iterations will allow it to converge to the correct answer. However, problems can arise when there are turning points in the ray or when the harmonic mean \( c_H \) leads to \( 1 - p_0^2 c_H^2 \leq 0 \). These cases are considered in later sections.
7.3.2 A Turning Point in the Direct Ray Path

The path for the direct ray can be found using the formulae described in the previous section. For some combinations of receiver and source locations, the direct ray path has a concave shape. For such a ray path, there exists a point where \( \frac{\partial r(z)}{\partial z} = 0 \). Equation 7.18 does not apply in this case. Instead, we use a modified version:

\[
r(z) = r(z_0) + \int_{z_m}^{z} \frac{pc(s)}{\sqrt{1 - p^2c^2(s)}} ds + \int_{z_m}^{z_0} \frac{pc(s)}{\sqrt{1 - p^2c^2(s)}} ds. \tag{7.30}
\]

Here, \( z_0 \) denotes source depth and \( z \) denotes receiver depth. Quantity \( z_m \) is the depth of the turning point.

Newton’s method - described in the previous section - does not work, since we do not know the value of the turning point \( z_m \). Instead, we use the bisection method described in Ref. [24] for finding the turning point. The goal is to find the \( z_t \), or the depth of this point. Since the turning point is below both the source depth \( z_s \) and the receiver depth \( z_r \), we know that it is between \( z = 0 \) and \( z = \min(z_s, z_r) \). The algorithm for finding the turning point is summarized below.

We define \( r(z) \) as:

\[
r(z) = \int_{z}^{z_s} \frac{p_1c(z')}{(1 - p_1^2c^2(z'))^{1/2}} dz' + \int_{z}^{z_r} \frac{p_1c(z')}{(1 - p_1^2c^2(z'))^{1/2}} dz'. \tag{7.31}
\]

At the \( i^{th} \) step of the algorithm, where \( z_m \in [z_i, z_{i+1}] \), we have:

\[
p_i = \frac{1}{c(z_{i+1}/2)}, \quad z_{i+1/2} = \frac{z_i + z_{i+1}}{2}, \quad r_i = r(z_{i+1/2}). \tag{7.32}
\]

The bisection method here compares value \( r_i \) with the required range value \( r(p) \) and the next iteration is determined in the following way: If \( r_i - r(p) > 0 \), then \( z_m \in [z_i, z_{i+1/2}] \); else, \( z_m \in [z_{i+1/2}, z_{i+1}] \). The algorithm stops when \( |r_i - r(p)| < \epsilon \), with \( \epsilon \) being the desired tolerance.

The only question remaining now is how to choose the initial interval for starting the bisection method. One choice is to use \([0, \min(z_s, z_r)]\) as the starting
region. Assuming \(\min(z_s, z_r) = 20\), the bisection method would require about 15 iterations to converge to a tolerance of \(1 \times 10^{-4}\). Another way for selecting the region is by utilizing the following facts:

- \(r(z)\) in Equation 7.30 is a monotonically decreasing function of \(z_m\), since \(\frac{\partial r}{\partial z_m} \leq 0\).
- If \(z > z_m\), then \(r(z) - r(z_m) < 0\). If \(z < z_m\), then \(r(z) - r(z_m) > 0\). The sign is different for \(r(z) - r(z_m)\) when \(z\) is on different sides of \(z_m\).

Let \(z^* = \min(z_s, z_r)\) and \(z_n = z^* - n\), where \(n = 0, 1, 2, \ldots\). Going through the depth elements, we evaluate the function \(f(z_n) = r(z_n) - r(z_m)\). If we detect \((f(z_n) < 0\ and\ f(z_{n+1}) > 0)\) or \((f(z_n) > 0\ and\ f(z_{n+1}) < 0)\), that is, a sign change, then we can safely assume that the turning point \(z_m\) is in the region \([z_{n+1}, z_n]\). This interval is used as the starting interval for the algorithm search for the turning point depth \(z_m\). Once \(z_m\) is found, the ray parameter can be determined via \(p = \frac{1}{c(z_m)}\).

### 7.3.3 Surface and Bottom Reflected Paths

For the surface reflection ray parameter, we can still use Newton’s method as described in Equation 7.27. First, we need to use a different equation for range:

\[
r(p_i) = \int_0^{z_s} \frac{p_i c(z')}{(1 - p_i^2 c^2(z'))^{1/2}} d z' + \int_0^{z_r} \frac{p_i c(z')}{(1 - p_i^2 c^2(z'))^{1/2}} d z',
\]

and

\[
\frac{\partial r(p_i)}{\partial p} = \int_0^{z_s} \frac{c(z')}{(1 - p_i^2 c^2(z'))^{3/2}} d z' + \int_0^{z_r} \frac{c(z')}{(1 - p_i^2 c^2(z'))^{3/2}} d z'.
\]

The initial guess \(c_H\) has to be modified as well. We choose

\[
c_H = \max\left(\frac{1}{z_s} \int_0^{z_s} \frac{1}{c(s)} ds, \frac{1}{z_r} \int_0^{z_r} \frac{1}{c(s)} ds\right),
\]

and calculate \(p_0\) according to Equation 7.24.
For the bottom reflection, we need to consider the water column depth parameter $WD$. The equations become:

$$r(p_i) = \int_{z_s}^{WD} \frac{p_i c(z')}{(1 - p_i^2 c^2(z'))^{1/2}} dz' + \int_{z_r}^{WD} \frac{p_i c(z')}{(1 - p_i^2 c^2(z'))^{1/2}} dz', \quad (7.36)$$

$$\frac{\partial r(p_i)}{\partial p} = \int_{z_s}^{WD} \frac{c(z')}{(1 - p_i^2 c^2(z'))^{3/2}} dz' + \int_{z_r}^{WD} \frac{c(z')}{(1 - p_i^2 c^2(z'))^{3/2}} dz', \quad (7.37)$$

and

$$c_H = \max\left(\frac{1}{WD - z_s} \int_{z_s}^{WD} \frac{1}{c(s)} ds, \frac{1}{WD - z_r} \int_{z_r}^{WD} \frac{1}{c(s)} ds\right). \quad (7.38)$$

The remaining procedure is the same as the one for the surface reflection case.

There is a simple “trick” for combining all three cases (corresponding to D, SR, and BR) into one single case by using the mirror method. Namely, we define a new SSP $c^*(z)$, which satisfies the following condition:

$$c^*(z) = \begin{cases} 
    c(z) & \text{if } z \in [0, WD] \\
    c(-z) & \text{if } z \in [-WD, 0] \\
    c(WD - z) & \text{if } z \in [WD, \infty] \end{cases}.$$  

With this definition, we can set the SR range function as follows:

$$r(p_i) = \int_{-z_r}^{z_s} \frac{p_i c^*(z')}{(1 - p_i^2 c^2(z'))^{1/2}} dz', \quad (7.39)$$

$$\frac{\partial r(p_i)}{\partial p} = \int_{-z_r}^{z_s} \frac{c^*(z')}{(1 - p_i^2 c^2(z'))^{3/2}} dz', \quad (7.40)$$

and the BR range function as:

$$r(p_i) = \int_{z_r}^{2WD - z_s} \frac{p_i c^*(z')}{(1 - p_i^2 c^2(z'))^{1/2}} dz', \quad (7.41)$$

$$\frac{\partial r(p_i)}{\partial p} = \int_{z_r}^{2WD - z_s} \frac{c^*(z')}{(1 - p_i^2 c^2(z'))^{3/2}} dz'. \quad (7.42)$$
By employing $c^*(z)$ and the above formulation, one can write a more condensed algorithm by combining the calculation of D, SR, and BR rays into one class that takes $(z_{\text{lower limit}}, z_{\text{upper limit}}, WD)$ as arguments. This approach makes the algorithm more straightforward to implement.

7.4 Time Derivatives with respect to Source Range, Source Depth, and Water Column Depth

The linearization approach requires the computation of ray travel time derivatives with respect to the parameters in $q$. Differentiation of time with respect to source and receiver locations for the direct ray path is presented analytically in Refs. [8, 24]; bottom depth is also considered in the latter reference. The time derivatives with respect to source range $r$, source depth $z_s$, and water column depth $WD$ are as follows [24]:

First dealing with the direct path, we recognize that

$$\frac{\partial t}{\partial WD} = 0,$$  \hspace{1cm} (7.43)

because there is no interaction with the water-sediment interface. From ray theory, the derivatives for range and source depth are shown to be:

$$\frac{\partial t}{\partial r} = p,$$  \hspace{1cm} (7.44)

$$\frac{\partial t}{\partial z_s} = -\sqrt{1 - p^2c^2(z_s)} \frac{1}{c(z_s)}.$$  \hspace{1cm} (7.45)

For the surface reflection, we have:

$$\frac{\partial t}{\partial WD} = 0,$$  \hspace{1cm} (7.46)

$$\frac{\partial t}{\partial r} = p,$$  \hspace{1cm} (7.47)
\[
\frac{\partial t}{\partial z_s} = \frac{\sqrt{1 - p^2c^2(z_s)}}{c(z_s)}.
\] (7.48)

For the bottom reflected path, the derivatives are:

\[
\frac{\partial t}{\partial WD} = \frac{2\sqrt{1 - p^2c^2(WD)}}{c(WD)},
\] (7.49)

\[
\frac{\partial t}{\partial r} = p,
\] (7.50)

\[
\frac{\partial t}{\partial z_s} = -\frac{\sqrt{1 - p^2c^2(z_s)}}{c(z_s)}.
\] (7.51)

Notice that the expression for \( \frac{\partial t}{\partial r} \) is the same for different ray paths. Also, there are only sign differences for both \( \frac{\partial t}{\partial z_s} \) and \( \frac{\partial t}{\partial z_r} \) between the surface and bottom reflected path calculations.

If receiver array tilt is also considered, we can find its derivative as follows. We first assume that the receiver array is linear:

\[
r(z_r) = z_0 + \alpha z_r,
\] (7.52)

where \( r_0 \) is the range for the receiver array at \( z = 0 \); a receiver in the array is located at depth \( z_r \) and range \( r(z_r) \); \( \alpha \) represents the slope or tilt of the entire array. Note that, even though \( \alpha \) here represents line slope, it is approximately the radian angle when the angle is small. The derivative with respect to tilt is:

\[
\frac{\partial t}{\partial \alpha} = \frac{\partial t}{\partial r} \frac{\partial r}{\partial \alpha} = p z_r.
\] (7.53)

Derivatives for all paths with respect to time instant \( t_0 \) are 1.

Finally, we need to include information about sound speed \( c(z) \) in the Jacobian matrix. It is difficult to include the derivative with respect to the entire function in the Jacobian, so we approximate it by considering the SSP \( c(z) \) as a linear combination of EOFs.
7.4.1 Arrival Time Derivatives with respect to Sound Speed

The arrival time derivatives with respect to EOF coefficients can be derived in the following way. The SSP is described as:

\[ c = c_m + \sum_{i=1}^{N_e} \mu_i v_i, \quad (7.54) \]

where \( c_m \) is the mean SSP vector obtained from CTD measurements, \( v_i, i = 1, \ldots, N_e \), are the eigenvectors of the sound speed covariance matrix as previously mentioned, and \( \mu_i, i = 1, \ldots, N_e \) are the eigenvector coefficients. Quantity \( N_e \) is the number of EOFs included in the system. In our case \( N_e = 3 \), because of prior information. Linear interpolation is used on these vectors to obtain the full profile:

\[ c(z) = c_m(z) + \sum_{i=1}^{N_e} \mu_i v_i(z). \quad (7.55) \]

Next, we derive \( \frac{\partial t}{\partial \mu_i} \) for \( i = 1, 2, \ldots N_e \). We can write:

\[ t = \int_{z_s}^{z_r} \frac{1}{c(z) \sqrt{1 - p^2 c^2(z)}} \, dz. \quad (7.56) \]

Note that \( t \) is dependent on both sound speed \( c \) and the ray parameter \( p \). Differentiating \( t \) with respect to \( \mu_i \) by applying the chain rule to Equation 7.56, and with \( \frac{\partial c}{\partial \mu_i} = v_i \) given by Equation 7.55, we obtain:

\[
\frac{\partial t}{\partial \mu_i} = \frac{\partial t}{\partial c} \frac{\partial c}{\partial \mu_i} + \frac{\partial t}{\partial p} \frac{\partial p}{\partial \mu_i} \\
= \int_{z_s}^{z_r} \frac{(2p^2c^2(z) - 1)v_i(z)}{c^2(z)(1 - p^2c^2(z))^{3/2}} \, dz \\
+ \frac{\partial p}{\partial \mu_i} \int_{z_s}^{z_r} \frac{pc(z)}{(1 - p^2c^2(z))^{3/2}} \, dz.
\quad (7.57)
\]

To simplify Equation 7.57, we need to know \( \frac{\partial p}{\partial \mu_i} \). We can use the following relation for range:

\[ r = \int_{z_s}^{z_r} \frac{pc(z)}{\sqrt{1 - p^2c^2(z)}} \, dz. \quad (7.58) \]
We also use the fact that range is independent of change in ray path; it does not depend on the eigenvector coefficients \( \mu_i \). It follows that:

\[
\frac{\partial r}{\partial \mu_i} = 0
\]

\[
= \frac{\partial r}{\partial c} \frac{\partial c}{\partial \mu_i} + \frac{\partial r}{\partial p} \frac{\partial p}{\partial \mu_i}
\]

\[
= \int_{z_s}^{z_r} \frac{pv_i(z)}{(1 - p^2c^2(z))^{3/2}} dz
\]

\[
+ \frac{\partial p}{\partial \mu_i} \int_{z_s}^{z_r} \frac{c(z)}{(1 - p^2c^2(z))^{3/2}} dz.
\]

Using Equation 7.59 to cancel out the \( \frac{\partial p}{\partial \mu_i} \) term in Equation 7.57, we obtain:

\[
\frac{\partial t}{\partial \mu_i} = \int_{z_s}^{z_r} \frac{-v_i(z)}{c^2(z)\sqrt{1 - p^2c^2(z)}} dz.
\]

These derivatives are included in the Jacobian matrix.

We next derive \( \frac{dt}{d\alpha} \), where \( \alpha \) represents vertical tilt for the receiver array and:

\[
r(z_r) = r_0 + \alpha z_r.
\]

The derivative is:

\[
\frac{dt}{d\alpha} = \frac{dt}{dr} \frac{dr}{d\alpha}
\]

\[
= pz_r.
\]

### 7.5 The Complete Inverse Algorithm

In this section, we collect all our previous derivations together and present the full inverse algorithm. Assuming the arrival time data vector \( t \) is of size \( 3N_h \), and \( q \) is a vector containing the following terms:

\[
q = [r, z_s, WD, \mu_1, \mu_2, \mu_3, t_0].
\]

Quantity \( t_0 \) is the time instant of the source. We also assume that the receiver array depth \( z_r = \{z_{r1}, z_{r2}, z_{r3}, \ldots z_{rN_h}\} \) is known. With an initial guess \( q_0 \), we can generate
a replica of the sound arrival time data \( f(q_0) \) by using the technique described in Section 7.3. The Jacobian matrix \( J \) evaluated at \( q_0 \) is:

\[
J = \begin{pmatrix}
\frac{\partial t_1}{\partial r} & \frac{\partial t_1}{\partial z_s} & \frac{\partial t_1}{\partial W_D} & \frac{\partial t_1}{\partial \mu_1} & \frac{\partial t_1}{\partial \mu_2} & \frac{\partial t_1}{\partial \mu_3} & \frac{\partial t_1}{\partial t_0} \\
\frac{\partial t_2}{\partial r} & \frac{\partial t_2}{\partial z_s} & \frac{\partial t_2}{\partial W_D} & \frac{\partial t_2}{\partial \mu_1} & \frac{\partial t_2}{\partial \mu_2} & \frac{\partial t_2}{\partial \mu_3} & \frac{\partial t_2}{\partial t_0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial t_{3N_h}}{\partial r} & \frac{\partial t_{3N_h}}{\partial z_s} & \frac{\partial t_{3N_h}}{\partial W_D} & \frac{\partial t_{3N_h}}{\partial \mu_1} & \frac{\partial t_{3N_h}}{\partial \mu_2} & \frac{\partial t_{3N_h}}{\partial \mu_3} & \frac{\partial t_{3N_h}}{\partial t_0} 
\end{pmatrix}.
\]

We can now improve our initial assumption using the linear approximation of Equation 7.15. We need to solve the system by taking the inverse of the involved matrix. Since we have more data than parameters, this is an overdetermined system. Applying the least squares method, we obtain:

\[
\delta q = (J^TJ)^{-1}J^T(t - f(q_0)).
\]

Solving the inverse problem with this matrix equation is referred to as the creeping method, because, in each iteration, we are solving for the difference in parameters, \( \delta q \). By adding this difference to our initial guess \( q_0 \), we essentially “creep” towards the correct solution.

Another closely related formulation of the linear approximation of Equation 7.15 is the following:

\[
t = f(q_0) + J(q - q_0),
\]

\[
q = (J^TJ)^{-1}J^T(t - f(q_0) + Jq_0).
\]

Instead of solving for the difference of parameter values in successive iterations, this formula directly solves for the next parameter vector \( q \). In other words, it “jumps” from the initial assumption to an improved solution and then to a subsequent one, and so on. This is the reason why Equation 7.67 is referred to as the jumping method.

Both methods provide the exact same answer for noise-free cases. However, as we
shall see, the jumping method is superior for noisy data, because of its convenient form for incorporating \textit{a priori} knowledge directly into a regularization scheme.

### 7.6 The Regularization Method

In our linearization inverse problem, we have an overdetermined system as mentioned above. The standard least squares method is used to solve such a system, by finding the parameter values that minimize the misfit $\chi^2$, which is defined to be

$$
\chi^2 = |J\delta q - \delta t|^2. \tag{7.68}
$$

For a vector $x = [x_1, x_2, \ldots, x_N]$ the Euclidean distance is defined as

$$
||x|| = \sqrt{\sum_{n=1}^{N} |x_n|^2}. \tag{7.69}
$$

The least squares method assumes that the data follow Gaussian statistics. In this case, it can be shown that the parameter values that maximize the likelihood function formulated under the Gaussian assumption are also those that minimize the error in a least squares sense [38, 25].

In discretization of inverse problems, the resulting systems of linear equations are usually highly ill-conditioned. The least squares method is sensitive to noise for these types of systems, which can lead to very unstable solutions. In order to compute stable and useful solutions, it is necessary to apply regularization methods. While some regularization methods aim to reduce the condition number of the ill-conditioned matrix, others attempt to incorporate \textit{a priori} information about the environment into the computation. In this section we will look at a method that belongs in the latter category.

We define a new objective function [8] that takes \textit{a priori} information $q_p$ into account,

$$
\phi = |G(Jq - t)|^2 + \lambda^2|H(q - q_p)|^2, \tag{7.70}
$$
where $G$ is a diagonal weighting matrix given by $G = \text{diag}[1/\sigma_1, 1/\sigma_2, \cdots, 1/\sigma_N]$ and $H$, the regularization matrix, is $H = \text{diag}[1/\xi_1, 1/\xi_2, \cdots, 1/\xi_M]$. Quantity $\xi_j$ represents the uncertainty for the $j$th parameter estimate in the vector $q_p$. Minimizing this objective function with respect to $q$, we obtain the regularized solution

$$q = [J^T G^T G J + \lambda^2 H^T H]^{-1} [J^T G^T G t + \lambda^2 H^T H q_p].$$  \hspace{1cm} (7.71)$$

The term $\lambda^2 H^T H$ ensures that the matrix to be inverted is well-conditioned and that the prior information is appropriately accounted for. The term $\lambda^2 H^T H q_p$ fits the solution towards the *a priori* model $q_p$. The value for $\lambda^2$ can be selected so that the misfit $\chi^2$ is equal to the expected value of $\langle \chi^2 \rangle = KN_h$. The value for $\lambda$ can also be chosen using the L-curve, which is a plot of $|G(q - q_p)|^2$ vs. $|G(f(q) - t)|^2$. The L-curve reveals the trade-off between regularization error and data fitting with *a priori* information. It is suggested that the optimal value for $\lambda^2$ leads to a point on the L-curve slightly to the right of the corner [18, 29], as shown in Figure 7.2.

**Figure 7.2** The L-curve, based on which the regularization coefficient $\lambda^2$ is selected. The error in the arrival times was normally distributed with zero mean and a standard deviation of 200 ms.
CHAPTER 8
NUMERICAL SIMULATIONS FOR THE EOF LINEARIZATION INVERSION METHOD

In this chapter we will first evaluate the performance of the EOF linearization inverse method using Monte Carlo simulations with synthetic data. The arrival time data were generated for the three paths considering a VLA and an environment similar to that of the Shallow Water 2006 experiment (SW06) [19, 42]. We will then apply the same technique to real arrival time data, which are estimated using particle filtering from SW06 time series (see Ref. [28]).

8.1 Testing the Inversion Method using Synthetic Data
As just mentioned, we first assume a true profile and VLA similar to those of the SW06 experiment. Two hundred noisy realizations of arrival times were generated for the three paths; the noise was additive, Gaussian, and zero mean with a standard deviation of 200 µs. Coefficient $\lambda^2$ of Equation 7.71 was chosen by forming an L-curve as shown in Figure 7.2. As recommended in [18], a good value of $\lambda^2$ is at a point on the L-curve slightly to the right of the corner. However, there is currently no way to find a unique optimum value. The essence is that $\lambda^2$ is chosen so that there is a balance between deviation of the solution from the prior information and minimization of the least-squares error. Performing the inversion this way prevents us from overfitting the data (that is, fitting the noise).

We show in Figure 8.1 parameter estimates vs. iteration number for a single realization. It can be seen that the algorithm requires only a few iterations for convergence. The probability density functions (PDFs) for source range, source depth, water column depth, and three EOF coefficients from Monte Carlo runs with 200 realizations are shown in Figure 8.2.
Figure 8.1  Linearization estimates vs. iteration: (a) source range, (b) source depth, (c) water depth, (d) $\mu_1$, (e) $\mu_2$, and (f) $\mu_3$.

Figure 8.2  PDFs of (a) source range, (b) source depth, (c) water column depth, (d) $\mu_1$, (e) $\mu_2$, and (f) $\mu_3$.

We performed the estimation process also using a smaller value for $\lambda^2$, which means that the least squares part of the function that is minimized is weighted more now (with the prior information impacting the solution less). Results are shown in
Figure 8.3  PDFs of (a) source range, (b) source depth, (c) water column depth, (d) $\mu_1$, (e) $\mu_2$, and (f) $\mu_3$ for a smaller value of $\lambda^2$.

Figure 8.3. Table 8.1 presents the true values of the parameters that we estimate and the maximum a posteriori (MAP) estimates obtained from the estimation with the two values of $\lambda^2$ by maximizing the computed PDFs of Figures 8.2 and 8.3. It is interesting to observe how the two values of $\lambda^2$ provide slightly different results for the EOF coefficients (the results for source range and depth and bottom depth are almost identical). The results from using a small value for $\lambda^2$ are very close to the true values for the coefficients. Slight discrepancies between true values and estimates are observed when a larger $\lambda^2$ is used. On the other hand, a small $\lambda^2$ results in higher uncertainty/spread in the estimation process as shown from the PDFs of Figure 8.3 and their comparison to those of Figure 8.2. This is the well known interplay between variability in the results (when a small amount of regularization is applied) and bias (when regularization favors the prior information). When the latter happens, the solution is biased toward the prior values, although this is evident only for $\mu_1$ and $\mu_2$ in our results.
Table 8.1  True, Prior, and Estimated Values for the Unknown Parameters for Two Values of $\lambda^2$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Prior</th>
<th>Estimated</th>
<th>Est. smaller $\lambda^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r(m)$</td>
<td>223</td>
<td>230</td>
<td>221.5</td>
<td>221.6</td>
</tr>
<tr>
<td>$z_s(m)$</td>
<td>26</td>
<td>30</td>
<td>25.9</td>
<td>25.7</td>
</tr>
<tr>
<td>$WD(m)$</td>
<td>73</td>
<td>80</td>
<td>73.0</td>
<td>72.9</td>
</tr>
<tr>
<td>$\mu_1$</td>
<td>-35</td>
<td>-70</td>
<td>-40</td>
<td>-35</td>
</tr>
<tr>
<td>$\mu_2$</td>
<td>-10</td>
<td>-30</td>
<td>-8</td>
<td>-11</td>
</tr>
<tr>
<td>$\mu_3$</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 8.2  True and Estimated Values for the Unknown Parameters for a Different Set of Prior Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Prior</th>
<th>Estimated</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r(m)$</td>
<td>223</td>
<td>180</td>
<td>221</td>
</tr>
<tr>
<td>$z_s(m)$</td>
<td>26</td>
<td>30</td>
<td>26</td>
</tr>
<tr>
<td>$WD(m)$</td>
<td>73</td>
<td>80</td>
<td>72</td>
</tr>
<tr>
<td>$\mu_1$</td>
<td>-35</td>
<td>10</td>
<td>-35</td>
</tr>
<tr>
<td>$\mu_2$</td>
<td>-10</td>
<td>10</td>
<td>-8</td>
</tr>
<tr>
<td>$\mu_3$</td>
<td>2</td>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>

We should point out that $H = \text{diag}[1/30, 1/10, 1/40, 1/30, 1/20, 1/10, 0]$ for source range, source depth, water depth, $\mu_1, \mu_2, \mu_3$, and time instant, respectively. This matrix reflects a significant amount of uncertainty regarding the available prior information. That is, the algorithm was not restricted within search intervals tightly surrounding the true parameter values.

Because we cannot express the quality of the complete SSP through a PDF, we show in Figure 8.4(a) the true SSP of the water column in the synthetic environment (squares) and the MAP profile calculated from the 200 estimates we obtained.
Figure 8.4  (a) The SSP used for the simulations (curve with squares) and the SSP constructed with the MAP estimates obtained from 200 realizations. (b) The profiles of (a) with superimposed SSPs estimated from a few realizations (solid curves). (circles) for the larger value of $\lambda^2$. This estimate was calculated by using the MAP estimates for coefficients $\mu_1$, $\mu_2$, and $\mu_3$. Figure 8.4(b) demonstrates the same two profiles, but now a few estimates obtained from distinct realizations (solid curves) are superimposed. Although these results from different realizations do not represent a full PDF, they provide an idea of the spread of the SSPs around the MAP estimate and they complement Figure 8.2.

To determine how sensitive our method is to initial conditions or prior information, we performed estimation using a different prior model (matrix $H$ remained the same). The new prior assumptions are shown in Table 8.2. Results are demonstrated in Figure 8.5 and Table 8.2 and show robustness of the method: two very different sets of prior values produced practically the same results. The PDFs are very similar to those of Figure 8.2 and the MAP estimates of Table 8.2 are excellent. We should mention that the method does not always converge. For example, when we use 100, 90, as 80 as prior values for the EOF coefficients, the
method diverged. However, such a choice reflects a complete lack of information on the involved parameters, which is very rarely the case.

![Figure 8.5](image)

**Figure 8.5** PDFs of (a) source range, (b) source depth, (c) water column depth, (d) $\mu_1$, (e) $\mu_2$, and (f) $\mu_3$ for a different set of prior values.

Lastly, to test our method under more challenging circumstances, we performed estimation employing arrival times with higher uncertainty (the standard deviation was 300 $\mu$s). The PDFs, shown in Figure 8.6, demonstrate a larger spread than those of Figure 8.2; this is expected because of the increased uncertainty. However, the modes of the densities are very close to the true parameter values.
8.2 Inversion Results from Real Data

Using particle filtering, MAP amplitude estimation, and smoothing techniques as described in Ref. [28], we obtained the PDFs for three path arrivals, which are shown in Figure 8.7. The PDFs were constructed from 500 sampled particles and their associated probabilities. These arrival time particles are used as input to the linearized system of Equation 7.66 (data $t$). For every set of three arrival times within a particle, a solution is obtained for the unknown parameters (vector $q$) using Equation 7.71. These multiple solutions/estimates form PDFs for the unknown parameters. PDFs for source range and depth, water column depth, and EOF coefficients $\mu_1$, $\mu_2$, and $\mu_3$ are shown in Figure 8.8. Array tilt was also set as an unknown quantity. MAP parameter estimates along with the considered prior information are listed in Table 8.3. The diagonal elements of the uncertainty matrix $H$ were $1/3$, $1/2$, $1/3$, $1/30$, $1/20$, $1/10$, and $1/2$ for range, source depth, water column depth, $\mu_1$, $\mu_2$, and $\mu_3$, and tilt, respectively. Coefficient $\lambda^2$ was selected by computing the L-curve of Figure 8.9.
Range and source depth and water column depth estimates are very close to the true values provided to us with the data. The estimates we obtained for $\mu_1$ and $\mu_2$ are very similar to the ones obtained for the same coefficients in Refs. [19] and [42]. The estimate for $\mu_3$ somewhat differs from those in Ref. [42] but agrees with estimates reported in Ref. [19] (the values within those references vary among themselves as well). It should be noted that the data collection sites differ for the different inversions. In Figure 8.10, we demonstrate the fit between the true time series (solid lines) and the synthetic time series, generated using the estimates shown in Table 8.3 (dotted lines). The arrival times of the synthetic time series were created using ray tracing for the MAP parameter estimates of the table. The fit appears to be very good, indicating that the inversion was successful. A perfect match was not expected because the linearization process provides an approximation. The inversion was repeated with alternative prior information to the one presented in Table 8.3. Results from all inversions were very close.

Figure 8.11(a) shows the mean SSP of the water column calculated from CTD measurements (squares) and the MAP profile calculated from arrival time estimates.
Table 8.3  Prior Information and MAP Parameter Estimates for Source Range and Depth, Water Column Depth, EOF Coefficients $\mu_1, \mu_2, \mu_3$, and Tilt

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Prior</th>
<th>MAP Estimated</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$ ($m$)</td>
<td>230</td>
<td>230</td>
</tr>
<tr>
<td>$z_s$ ($m$)</td>
<td>25</td>
<td>26.8</td>
</tr>
<tr>
<td>$WD$ ($m$)</td>
<td>79</td>
<td>76.9</td>
</tr>
<tr>
<td>$\mu_1$</td>
<td>-85</td>
<td>-50</td>
</tr>
<tr>
<td>$\mu_2$</td>
<td>-55</td>
<td>-7</td>
</tr>
<tr>
<td>$\mu_3$</td>
<td>2</td>
<td>1.5</td>
</tr>
<tr>
<td>Tilt ($^\circ$)</td>
<td>0</td>
<td>-0.2</td>
</tr>
</tbody>
</table>

and linearization (circles). Figure 8.11(b) demonstrates the same two profiles, but now SSPs from distinct arrival time particles (solid curves) are superimposed.

To further validate the potential and accuracy of our linearization method, we compared our results to estimates obtained from a global optimization technique for the same arrival time particles that were used for our inversion. For global optimization, we used fast simulated annealing [37]. The process searched for the set of unknown parameters that minimize the mean squared error between the true arrival times (that is, the arrivals extracted from the SW06 time series with the particle filter) and replica arrival times calculated with ray tracing for that set of parameters, similarly to Ref. [40]. The search intervals were the same as those employed in Ref. [19]. Table 8.4 presents inversion results for two sets of arrival time estimates using linearization and simulated annealing. There is a very good agreement between the results for both cases, indicating that both methods are successful (being consistent among themselves and with values reported in our references). This was expected for the global optimization-matching process because it relies on calculating the arrival times of a replica signal for multiple sets of
unknown parameter values, optimizing the search for identifying the best set; no approximation to the forward model is performed. It appears that the linearization process, although it is based on an approximation and uses only a few calculations, performs equally well. Specifically, in terms of efficiency, the linearization method required six iterations (typically four to converge, but we continued to six), whereas the annealing process in our case involved seven ray tracing runs, one for each unknown parameter, for a large number of iterations (between a few hundred to a few thousand). This significant difference demonstrates the efficiency of our method, which does not come at the expense of accuracy. For demonstration purposes, we show in Figure 8.12 simulated annealing results vs. iteration by using one set of arrival time estimates obtained by the particle filter.

Figure 8.8  PDFs of (a) source range, (b) source depth, (c) water column depth, (d) $\mu_1$, (e) $\mu_2$, and (f) $\mu_3$ for real data.
Figure 8.9  The L-curve for the real data, based on which the regularization coefficient $\lambda^2$ is selected.

Figure 8.10  Time series at 14 hydrophones. The real data (solid lines) were collected during the SW06 experiment. Synthetic time series (dotted lines) are generated using the linearization estimates.
Table 8.4  Parameter Estimates Using the Linearization/Particle Filter (PF) Method for Two Sets of Arrival Times and Corresponding Simulated Annealing (SA) Results From the Same Arrival Times

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Lin./PF</th>
<th>SA Est.</th>
<th>Lin./PF</th>
<th>SA Est.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r(m) )</td>
<td>228</td>
<td>231</td>
<td>232</td>
<td>231</td>
</tr>
<tr>
<td>( z_s(m) )</td>
<td>26</td>
<td>26</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>( WD(m) )</td>
<td>76</td>
<td>76</td>
<td>77</td>
<td>77</td>
</tr>
<tr>
<td>( \mu_1 )</td>
<td>-54</td>
<td>-56</td>
<td>-42</td>
<td>-43</td>
</tr>
<tr>
<td>( \mu_2 )</td>
<td>-10</td>
<td>-12</td>
<td>-7</td>
<td>-7</td>
</tr>
<tr>
<td>( \mu_3 )</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>-2</td>
</tr>
<tr>
<td>Tilt (°)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 8.11  (a) The mean SSP as calculated by CTD measurements (squares) and the SSP constructed with the MAP estimates obtained from the extracted arrival time PDFs (circles). (b) The profiles of (a) with superimposed SSPs constructed from distinct arrival time particles (solid curves).
Figure 8.12  Fast simulated annealing results vs. iteration for one set of arrival time estimates for (a) source range, (b) source depth, (c) water column depth, (d) $\mu_1$, (e) $\mu_2$, and (f) $\mu_3$ for real data.
CHAPTER 9
CONCLUSIONS AND FUTURE WORK

9.1 Conclusions
In this dissertation we have examined two inverse methods for estimating the SSP in the ocean. The focus of the first inverse approach (Chapters 2-6) is on the estimation of the sediment SSP. We discussed Stickler’s experiment for recovering the reflection coefficient and the SSP inversion using the DT formula. While traditional methods for estimating the sediment SSP require global optimization techniques, this inverse method takes a much more computationally efficient approach. Once the reflection coefficient is known, the solution is usually obtained in one integral evaluation. This technique still suffers from high sensitivity to noise and more work is required to reduce the sensitivity to a practical level. At this stage, it is difficult to evaluate the usefulness of the method without any experimental data, but results appear promising.

The second inverse method (Chapters 7-8) focuses on source localization, water column depth estimation, and SSP estimation in the ocean using arrival time data and a priori information regarding the environment and geometry of the experiment. With both accuracy and computational efficiency as goals, the method used a linearized system and EOFs. Its results are comparable to those of the more computationally intensive simulated annealing method and to ground-truth information. This comparison and, especially, the success of the approach with real data demonstrate its potential for accurate and efficient inversion.

9.2 Future Work
To further improve on the first inverse method, the following considerations are of interest:
1. On the experimental front, the implementation of Stickler’s experiment should be a main priority. The low frequency source required by the algorithm remains a challenge.

2. On the numerical front, the evaluation of the nonlinear DT trace formula using the iterative method is slow. Using the Born approximation, we save considerable computation time but at the cost of accuracy. Even though the Born approximation did capture the discontinuities in the SSP solution, methods with higher accuracy are desirable and necessary in order to resolve the details of the ocean environmental parameters. It is useful to investigate fast numerical methods for solving the DT formula using iterative schemes.

3. The algorithm requires a sound speed limit for a large depth, $c_\infty$, to be a known quantity. This is a theoretical requirement for the convergence of the DT trace formula. However, it is not always possible to know $c_\infty$ beforehand. It is useful to conduct a study to test the sensitivity of the method with respect to the assumed value $c_\infty$. Numerical simulations and analysis can be performed to investigate the case when the real value of $c_\infty$ is different from the assumption. Ultimately, we would like to treat $c_\infty$ as an unknown quantity in the inverse algorithm.

As to the inverse method using EOFs and linearization, the results after application to real data were compared to ground-truth information, estimates reported by other authors, and estimates from fast simulated annealing. All results were found to be similar. The new algorithm has significant advantages in terms of efficiency: convergence required only a few iterations per inversion, whereas simulated annealing involved many more. That is, accurate estimates are obtained without the need for extensive prior information or onerous computations. The inverse method can be extended to include more uncertain parameters such as hydrophone locations.
Testing on different environments and more experimental data - preferably with a higher noise level - is also desirable to further validate the method.
BIBLIOGRAPHY


