The thesis of Chad M. Smith was reviewed and approved* by the following:

David L. Bradley
Senior Scientist
Professor of Acoustics
Thesis Advisor

R. Lee Culver
Senior Research Associate
Assistant Professor of Acoustics

Thomas B. Gabrielson
Senior Scientist
Professor of Acoustics

Victor W. Sparrow
Interim Chair of the Graduate Program in Acoustics
Professor of Acoustics

*Signatures are on file in the Graduate School.
Abstract

During August of 2008, the Transverse Acoustic Variability EXperiment (TAVEX) was carried out within the northern East China Sea southwest of Jeju Island, South Korea. Water depth of the experimental area varied around 70 to 85 m and the site was located approximately 100 km southwest of Jeju and 300 km northwest of a continental shelf break. Data collected includes towed conductivity, temperature, and depth (CTD) measurements of the water column, moored temperature recordings, bathymetry and acoustical sediment properties, as well as horizontal line array (HLA) recordings of CW and LFM pulsed sources. The focus of this measurement experiment is to understand and model the effects of internal waves on the acoustic signals received by a bottom-mounted HLA in the shallow-water environment within this region. Specifically, to evaluate the effects of the internal wave field on coherence of the HLA. The research presented in this thesis is a portion of this work. An overview of the TAVEX 2008 experiment is given, and the effects of internal waves on acoustic variation across the HLA are considered using a spectral approach. This analysis includes examination of harmonic variations in received acoustic pressure. Internal wave characteristics measured during the CTD tow, modeling of internal wave and acoustic fields, and the comparison of computational models with experimental acoustic data are presented. Techniques used for modeling of the acoustic field include both analytical ray and computational parabolic equation (PE) methods.
# Table of Contents

- **List of Figures** vii
- **List of Tables** xii
- **Acknowledgments** xiii

## Chapter 1

**Introduction** 1
- 1.1 Research Goals .................................................. 1
- 1.2 Experimental Data .................................................... 2
- 1.3 Thesis Outline .......................................................... 2
- 1.4 Acoustic Propagation in the Ocean .............................. 3
- 1.5 Internal Gravity Waves ............................................. 6
- 1.6 Background ............................................................ 7

## Chapter 2

**The TAVEX 2008 Field Experiment** 11
- 2.1 Chapter 2 Overview .................................................. 11
- 2.2 Description of the Experiment .................................... 12
- 2.3 Towed CTD Measurements ......................................... 16
- 2.4 Moored Temperature Measurements ............................... 23
- 2.5 Acoustic Measurements ............................................. 24

## Chapter 3

**Internal Waves** 34
- 3.1 Chapter 3 Overview .................................................. 34
- 3.2 Internal Wave Measurements ..................................... 34
- 3.3 Internal Wave Modeling ............................................. 39
- 3.4 Estimating Internal Wave Propagation Direction .............. 47
# Chapter 4

**Acoustic Modeling**

4.1 Chapter 4 Overview ........................................... 59
4.2 Harmonic Variation Analysis ................................. 59
4.3 Ray Model .................................................. 61
4.4 Parabolic Equation Model ................................. 68

# Chapter 5

**Acoustic Analysis**

5.1 Chapter 5 Overview ........................................... 71
5.2 Acoustic Analysis of KdV Wave Field at Multiple Propagation Angles ................................. 72
5.3 Acoustic Analysis of KdV and CTD Measurement Wave Fields ................................. 76
5.4 Harmonic Variation Analysis of HLA Data and Acoustic Models ................................. 80
5.5 Correlation of Nearby Ship Traffic with Acoustic Events ........................................ 88
5.6 Result of Analysis ........................................... 90

# Chapter 6

**Conclusions and Suggestions for Future Work**

6.1 Conclusions ........................................... 93
6.2 Future Work ........................................... 94

# Appendix A

**Towed CTD Data Collection and Processing Scripts**

A.1 Calibration and Configuration Scripts ...................... 96
A.2 Calibration Script (FULCHAIN.CAL) ...................... 96
A.3 Configuration Script (CONFIG2.CFG) ...................... 102
A.4 Towed CTD Data Processing Scripts ...................... 104
A.5 Converting *.DTA to *.mat Files (load_dta.m) ............. 104
A.6 Separate CTD Data (SeparateDataFiles.m) ............... 105
A.7 Calibration and Filtering (cal_filter_cut.m) ............... 107
A.8 Auto-filter (filter_bad_pts_strddev.m) .................... 115
A.9 Graphically Filter (Gfilter.m) ............................. 123
A.10 Extrapolate (extrapolate.m) ............................... 129
A.11 Calculate Sound Speed and other Fields (calculate_SDc.m) .... 134

# Appendix B

**Range-dependent Acoustic Model (RAM)**

B.1 Modified FORTRAN Code ................................. 142
B.2 Iterative 2D PE Model for MATLAB ..................... 166
### Appendix C

<table>
<thead>
<tr>
<th>Supporting Analysis</th>
<th>178</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1 Harmonic Variation Analysis of Individual Hydrophones</td>
<td>178</td>
</tr>
</tbody>
</table>

| Bibliography | 183 |
## List of Figures

1.1 Generic deep-water sound speed profile, taken from *Computational Ocean Acoustics* by Jensen *et al.* .................................................. 5

1.2 A sound speed field calculated from towed CTD measurements taken during the TAVEX 2008. A single internal wave depression is visible. Color scale in [m/s]. .................................................. 7

2.1 Location of the TAVEX 2008 denoted by red circle. .................. 12

2.2 Enlargement in GPS space of TAVEX 2008 measurement area. .... 13

2.3 Bathymetry and major currents of the East China Sea region. Depth contours in meters. ................................................................. 14

2.4 Bathymetry of CTD tow region measured by R/V Eardo’s depth sounder. ............................................................. 15

2.5 Schematic of the CTD chain measurement system used in the TAVEX 2008. .......................................................... 17

2.6 CTD fin enlargement from the ADM CTD Chain website. ......... 18

2.7 CTD cable/fin assembly stored on drum prior to deployment. ..... 20

2.8 CTD cable/fin assembly deployed through block at stern of ship. 21

2.9 Vfin (type 167) on rear deck of ship just before deployment. ....... 21

2.10 300 and 500 Hz sources and moorings prior to deployment. Several temperature recorders are circled in red to show their placement on the mooring cable. Left, both sources are shown along with a mooring float. Right, a close up of the cable and temperature recorders. ................................. 24

2.11 Temperature measurements made along the 300 Hz source’s mooring cable. Color represents depth shown in legend. Time axis is in hours of August 28th and 29th of 2008. CTD tow starts at 8/28/2008 9:15:16 GMT. .......................................................... 25

2.12 Temperature measurements made along the 500 Hz source’s mooring cable. Color represents depth shown in legend. Time axis is in hours of August 28th and 29th of 2008. CTD tow starts at 8/28/2008 9:15:16 GMT. .......................................................... 25
2.13 Diagram of HLA mooring. Batteries and recording equipment are housed at the central weighted sled and the 32 and 64 hydrophone arrays were laid as straight as possible away from one another. 26

2.14 Mooring diagram for the 300 Hz source. Both the 300 and 500 Hz source were moored in this fashion and placed at approximately 64 m depth. 27

2.15 A portion of the recorded RMS pressure at 300 Hz for hydrophone #96. Average pressure of each CW pulse is obtained by averaging of the center 1 second of each pulse (average value denoted by red markers). 29

2.16 Averaged received signal level (RL) for CW pulses for hydrophone #96. Average level is obtained by averaging of the center 1 second of each pulse. 29

2.17 Average sound speed profile for entire CTD tow, which was used for range independent propagation models. 30

2.18 RAM calculated TL for 300 Hz source with an average water column sound speed profile. Placement of source and receiver is shown as well as the region (encompassed by dashed box) used to approximate average TL at the HLA. 31

2.19 RAM calculated TL for 500 Hz source with an average water column sound speed profile. Placement of source and receiver is shown as well as the region (encompassed by dashed box) used to approximate average TL at the HLA. 31

3.1 Times and locations of SAR images shown in Figure 3.2. The single RADARSAT-2 SAR image was taken about 8 hours prior to CTD deployment, and the two Envisat-1 ASAR images were taken about 35 hours prior. Depth contours in meters. 35

3.2 SAR images superimposed to display internal wave activity and directionality. Visibly apparent internal waves are highlighted in red, and the CTD tow area is highlighted in green. 36

3.3 Track of CTD tow denoted by blue track. Location of internal wave events denoted by red segments. Internal wave events are labeled chronologically. 37

3.4 Sound speed field of IW1 calculated from towed CTD measurements. Color scale in [m/s]. 40

3.5 Sound speed field of IW2 calculated from towed CTD measurements. Color scale in [m/s]. 40

3.6 Sound speed field of IW3 calculated from towed CTD measurements. Color scale in [m/s]. 41
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7</td>
<td>Sound speed field of IW4 calculated from towed CTD measurements. Color scale in [m/s].</td>
<td>41</td>
</tr>
<tr>
<td>3.8</td>
<td>Sound speed field of IW5 calculated from towed CTD measurements. Color scale in [m/s].</td>
<td>42</td>
</tr>
<tr>
<td>3.9</td>
<td>Sound speed field of IW6 calculated from towed CTD measurements. Color scale in [m/s].</td>
<td>42</td>
</tr>
<tr>
<td>3.10</td>
<td>Average density profiles of water column near internal wave measurements and 2-layer model averaging regions.</td>
<td>44</td>
</tr>
<tr>
<td>3.11</td>
<td>Schematic description of physical parameters of the KdV soliton model. Arbitrary units of depth and range.</td>
<td>45</td>
</tr>
<tr>
<td>3.12</td>
<td>Depth-dependent internal wave amplitude function used to give internal wave models realistic amplitude over depth.</td>
<td>48</td>
</tr>
<tr>
<td>3.13</td>
<td>Example of an average sound speed field perturbed by modeled internal waves (arbitrary units of depth and range). Color scale in [m/s].</td>
<td>48</td>
</tr>
<tr>
<td>3.14</td>
<td>Basic geometry used to find internal wave propagation direction.</td>
<td>50</td>
</tr>
<tr>
<td>3.15</td>
<td>Vector connection between ship speed, internal wave speed, and apparent internal wave propagation speed.</td>
<td>50</td>
</tr>
<tr>
<td>3.16</td>
<td>Plot demonstrating angular ambiguity found in Equation 3.12. The cases shown on the right side requires that ship speed be greater than internal wave speed.</td>
<td>52</td>
</tr>
<tr>
<td>3.17</td>
<td>Plot demonstrating angular ambiguity found in Equation 3.12 when wave travel speed is comparable to ship speed.</td>
<td>53</td>
</tr>
<tr>
<td>3.18</td>
<td>Plot demonstrating the angular ambiguity found in Equation 3.12 for the IW1 event.</td>
<td>56</td>
</tr>
<tr>
<td>3.19</td>
<td>Enlargement of Figure 3.3 displaying the approximate direction of each internal wave measurement during CTD tows.</td>
<td>57</td>
</tr>
<tr>
<td>3.20</td>
<td>Geometric relationship of linear wave fronts.</td>
<td>58</td>
</tr>
<tr>
<td>4.1</td>
<td>Drawing (not a ray trace) showing how an internal wave can affect the critical ray $R_c$ while having little impact on other rays that do not encounter the thermocline. $D_{R_c}$ is the critical ray cycle distance.</td>
<td>62</td>
</tr>
<tr>
<td>4.2</td>
<td>Average water column sound speed profile for all towed CTD data with linear approximation used for analytic ray calculations.</td>
<td>65</td>
</tr>
<tr>
<td>4.3</td>
<td>Diagram of analytic ray model components.</td>
<td>67</td>
</tr>
<tr>
<td>4.4</td>
<td>Diagram of critical ray ($\theta_{R_c} = 11.8^\circ$) most influenced by soliton passing for 500 Hz source (Left). Average water column sound speed profile used for computations (right).</td>
<td>69</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Average sound speed field of IW5&amp;IW6 perturbed by analytic KdV model over 300 Hz propagation path. Propagation angle assumed to be $0^\circ$ with acoustic axis. Color scale in m/s.</td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>Average sound speed field of IW5&amp;IW6 perturbed by analytic KdV model over 300 Hz propagation path. Propagation angle assumed to be $45^\circ$ with acoustic axis. Color scale in m/s.</td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>TL for the single range step of the IW5&amp;IW6 sound speed field shown in Figure 5.1. Color scale in dB.</td>
<td></td>
</tr>
<tr>
<td>5.4</td>
<td>TL for the single range step of the IW5&amp;IW6 sound speed field shown in Figure 5.2. Color scale in dB.</td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td>Received level at the range and depth of the HLA for IW5&amp;IW6 KdV wave fields with propagation angles of $0$ and $45^\circ$.</td>
<td></td>
</tr>
<tr>
<td>5.6</td>
<td>HVI of IW5&amp;IW6 propagating PE model for 300 Hz analytic KdV model. Propagation angle assumed to be $0^\circ$ with acoustic axis. Color scale in dB/Hz.</td>
<td></td>
</tr>
<tr>
<td>5.7</td>
<td>HVI of IW5&amp;IW6 propagating PE model for 300 Hz CTD data model. Propagation angle assumed to be $45^\circ$ with acoustic axis. Color scale in dB/Hz.</td>
<td></td>
</tr>
<tr>
<td>5.8</td>
<td>Average sound speed field of IW5&amp;IW6 perturbed by analytic KdV model over 300 Hz propagation path. Color scale in m/s.</td>
<td></td>
</tr>
<tr>
<td>5.9</td>
<td>Average sound speed field of IW5&amp;IW6 perturbed by actual CTD measurements over 300 Hz propagation path. Color scale in m/s.</td>
<td></td>
</tr>
<tr>
<td>5.10</td>
<td>Received level at the range and depth of the HLA for both the analytic KdV and CTD measurement wave fields of IW5&amp;IW6.</td>
<td></td>
</tr>
<tr>
<td>5.11</td>
<td>HVI of the IW5&amp;IW6 based propagating PE model at 300 Hz for analytic KdV internal wave. Color scale in dB/Hz.</td>
<td></td>
</tr>
<tr>
<td>5.12</td>
<td>HVI of the IW5&amp;IW6 based propagating PE model at 300 Hz for CTD data internal wave. Color scale in dB/Hz.</td>
<td></td>
</tr>
<tr>
<td>5.13</td>
<td>Average sound speed field of IW1 perturbed by analytic KdV model over 300 Hz propagation path. Color scale in m/s.</td>
<td></td>
</tr>
<tr>
<td>5.14</td>
<td>Average sound speed field of IW1 perturbed by analytic KdV model over 500 Hz propagation path. Color scale in m/s.</td>
<td></td>
</tr>
<tr>
<td>5.15</td>
<td>Average sound speed field of IW5&amp;IW6 perturbed by analytic KdV model over 300 Hz propagation path (same as Figures 5.1 and 5.8 but placed here for easy reference to other models). Color scale in m/s.</td>
<td></td>
</tr>
<tr>
<td>5.16</td>
<td>Average sound speed field of IW5&amp;IW6 perturbed by analytic KdV model over 500 Hz propagation path. Color scale in m/s.</td>
<td></td>
</tr>
<tr>
<td>5.17</td>
<td>HVI of the IW1 based propagating PE model at 300 Hz for the analytic KdV model. Color scale in dB/Hz.</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>5.18</td>
<td>HVI of the IW1 based propagating PE model at 500 Hz for the analytic KdV model. Color scale in dB/Hz.</td>
<td>85</td>
</tr>
<tr>
<td>5.19</td>
<td>HVI of the IW5&amp;IW6 based propagating PE model at 300 Hz for the analytic KdV model (same as Figure 5.11, but placed here to allow easy comparison with other models). Color scale in dB/Hz.</td>
<td>86</td>
</tr>
<tr>
<td>5.20</td>
<td>HVI of the IW5&amp;IW6 based propagating PE model at 500 Hz for the analytic KdV model. Color scale in dB/Hz.</td>
<td>86</td>
</tr>
<tr>
<td>5.21</td>
<td>Average HVI of received RMS acoustic pressure of 300 Hz averaged CW pulses (average of Figures C.1-C.4 in Appendix C). Color scale in dB/Hz.</td>
<td>91</td>
</tr>
<tr>
<td>5.22</td>
<td>Average HVI of received RMS acoustic pressure of 500 Hz averaged CW pulses (average of Figures C.5-C.8 in Appendix C). Color scale in dB/Hz.</td>
<td>91</td>
</tr>
<tr>
<td>5.23</td>
<td>Spectrogram of acoustic data looking for ship traffic. Color scale in dB/Hz re 1 µPa.</td>
<td>92</td>
</tr>
<tr>
<td>5.24</td>
<td>RL of CW pulses and noise floor from between CW pulses, as well as the SNR from during CTD measurement times.</td>
<td>92</td>
</tr>
<tr>
<td>C.1</td>
<td>HVI of acoustic pressure of 300Hz CW pulses at hydrophone #33. Color scale in dB.</td>
<td>179</td>
</tr>
<tr>
<td>C.2</td>
<td>HVI of acoustic pressure of 300Hz CW pulses at hydrophone #53. Color scale in dB.</td>
<td>179</td>
</tr>
<tr>
<td>C.3</td>
<td>HVI of acoustic pressure of 300Hz CW pulses at hydrophone #75. Color scale in dB.</td>
<td>180</td>
</tr>
<tr>
<td>C.4</td>
<td>HVI of acoustic pressure of 300Hz CW pulses at hydrophone #96. Color scale in dB.</td>
<td>180</td>
</tr>
<tr>
<td>C.5</td>
<td>HVI of acoustic pressure of 500Hz CW pulses at hydrophone #33. Color scale in dB.</td>
<td>181</td>
</tr>
<tr>
<td>C.6</td>
<td>HVI of acoustic pressure of 500Hz CW pulses at hydrophone #53. Color scale in dB.</td>
<td>181</td>
</tr>
<tr>
<td>C.7</td>
<td>HVI of acoustic pressure of 500Hz CW pulses at hydrophone #75. Color scale in dB.</td>
<td>182</td>
</tr>
<tr>
<td>C.8</td>
<td>HVI of acoustic pressure of 500Hz CW pulses at hydrophone #96. Color scale in dB.</td>
<td>182</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Geoacoustic parameters estimated for the tow area by Choi et al.</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>Linear distance of each CTD fin along the tow cable (not depth when ship is in motion) measured from the point that the cable enters the water at the stern of the research vessel</td>
<td>19</td>
</tr>
<tr>
<td>2.3</td>
<td>Bathymetry values used in RAM PE 2D computational models for each source’s acoustic path to the HLA (RAM linearly interpolates between these points). Notice that in general the bathymetry is flat, but is included for model accuracy</td>
<td>32</td>
</tr>
<tr>
<td>3.1</td>
<td>Parameters of 2-layer model for the environment surrounding each internal wave measured during CTD tow</td>
<td>44</td>
</tr>
<tr>
<td>3.2</td>
<td>Internal wave event parameters both measured and calculated.</td>
<td>49</td>
</tr>
<tr>
<td>3.3</td>
<td>Internal wave events and propagation angles. $\theta$ is referenced to the ships course during event measurement</td>
<td>55</td>
</tr>
<tr>
<td>3.4</td>
<td>List of four internal waves and their expected acoustic axis cross-times.</td>
<td>58</td>
</tr>
</tbody>
</table>
Acknowledgments

Much credit is deserved by my thesis advisor Dr. David Bradley for helping give direction and meaning to my research. His extensive experience, professional expertise, and academic support have greatly enhanced my understanding of the field and my working productivity. I would also like to thank my committee members Dr. Lee Culver and Dr. Thomas Gabrielson for their added support, technical assistance, and input to my research. The collaboration of Dr. David Goldstein and Dr. Peter Mignerey from the Naval Research Laboratory (NRL) of Washington, DC is also greatly appreciated.

I would like to acknowledge Dr. Jeff Schindall of NRL, as well as Jim Dorminy of Penn State Applied Research Laboratories (ARL) for their hard work and collaboration during the field measurements for this work. My thanks also include the captain and crew of the Korean research vessel EARDO for their professionalism and most able assistance during the experiment cruise, and the Korean Agency for Defense Development for arranging and coordinating the cruise.

My time in the Graduate Program of Acoustics at The Pennsylvania State University has been both enlightening and enjoyable due to the knowledgeable and friendly nature of the staff, and the diverse but forthcoming and sociable nature of the students. This atmosphere was highly beneficial to my academic and professional success.

The support of one’s family and friends is always an important factor to success in life, and my case is no exception. In particular, my mother Vicky L. Smith, father Alton D. Smith, sister Shannon L. Smith, and aunt Patricia A. Fusco have always been exceptionally supportive of my career interests in science, and have been available for any help that I have needed. For this I am extremely grateful, and I must acknowledge this as a major contributor to any success of mine. Many thanks are also deserved by both personal and professional friends that have been a sense of inspiration and a driving source in my life.

Thanks must also go to the Office of Naval Research (ONR) for funding of the field study as well as this research.

xiii
Dedication

I dedicate this thesis to my family and friends.
Chapter 1

Introduction

1.1 Research Goals

The goal of this thesis is to examine the relationship between measured variability in received acoustic signal at a horizontal line array (HLA), and the variability in the sound speed field due to internal wave perturbations. Internal wave perturbations were measured during conductivity, temperature, and depth (CTD) tows made during the Transverse Acoustic Variability EXperiment of 2008 (TAVEX 2008). It is hypothesized that harmonic variations in received signal level can be used to determine when an internal wave is propagating over the acoustic axis from source to receiver. Using profiles of water column dynamics from towed CTD measurements and acoustic data from the HLA, the practicality of this processing technique as it applies to the TAVEX scenario is analyzed and discussed. This work is in conjunction with research by Peter Mignerey of the Naval Research Laboratory (NRL), who’s interests are designing a modal theory to connect the coherence of the HLA to internal wave perturbations \cite{Mignerey1}. The present work furthers both the current understanding of how propagating internal waves effect acoustic propagation in a shallow-water scenario, and presents a method for using acoustic pressure (an easily measured quantity) to detect the passing of internal waves over a large geographic area.

In order to connect acoustic variability to the passing of internal waves, measurements from the TAVEX 2008 experiment and environmental considerations are first introduced. It is found that a lack of internal wave signatures in moored
temperature measurements during TAVEX makes it necessary to use an analytic internal wave model during propagation direction analysis. After wave direction is known, analytic and computational acoustic models are used to simulate the acoustic variability induced at a single receiver by the internal waves measured during TAVEX. Spectral analysis is used to compare the variability of acoustic computational models using both data-simulated and analytically-simulated internal wave fields (acoustic models using CTD measurements and analytic internal wave models to simulate internal wave perturbations) to measured acoustic data analysis.

1.2 Experimental Data

Data collected during the field experiment includes towed CTD measurements, moored vertical temperature recordings, bathymetry and seafloor properties, as well as HLA recordings of CW (continuous single frequency) and LFM (linear frequency modulated) pulsed sources. LFM pulses have bandwidth of ±10% of the CW pulse frequency, and CW frequencies include 300 and 500 Hz transmitted over approximately 34 and 20 km respectively. Sound speed field information was collected using a high-resolution towed CTD measurement chain. CTD tows were made in directions both parallel and perpendicular to the acoustic propagation path over a 36 hour time frame. It was found that moored temperature recordings were made too deep in the water column (too far below the thermocline to capture internal wave perturbations). Several internal waves were recorded during the CTD tow with periods of 4-42 min and amplitudes of 3-6 m. Acoustic recordings during the experiment display harmonic variations in acoustic pressure similar in period, and received level (RL) variations around 6-10 dB.

1.3 Thesis Outline

The remaining portion of Chapter 1 introduces the basic concepts, importance, and background for this thesis work. Chapter 2 discusses the methods used for data collection, data processing, and general environmental considerations of the experimental region. Chapter 3 covers internal wave measurements, modeling of
internal wave fields from environmental data, and propagation direction analysis of internal wave events. In Chapter 4, analytic and computational models of acoustic variation induced by a traveling wave field will be considered. In Chapter 5, spectral analysis of acoustic data and comparison with models is discussed. Chapter 6 presents final conclusions and suggestions for future work.

1.4 Acoustic Propagation in the Ocean

The ocean is a constantly changing environment, and this dynamic environment can have dramatic effects upon the sound speed field and acoustic propagation through it. At the heart this variation are a few fundamental physical forcing functions: gravitational and rotational forcing (large-scale diurnal and other tidal patterns), thermodynamic forces (convective overturning and turbulent processes), mechanical forces (surface wind stress, atmospheric pressure variations, and seismic seafloor motion), internal fluid forces (pressure and viscosity within the seawater itself), and boundary forcing (surface and seafloor boundary conditions) [3]. These forcing functions are the cause of most ocean dynamics, and each can affect the sound speed field of the water column because sound speed is directly dependent upon pressure, temperature, and salinity [4]. This thesis will examine the impact of sound speed field variability during TAVEX 2008 due to internal waves.

Acoustic radiation has been routinely used as a tool for measurements and communications within the ocean since the Second World War. During this time, much effort was put into understanding the basic principles of how sound waves travel through water, and studies were undertaken to develop the technology known today as sonar (sound navigation and ranging). During this period, sonar technology played a prominent role in naval battles both above and below the ocean’s surface. Today sonar is still used heavily by the military, however, it also has many common uses in the civilian market. Depth sounders, sub-bottom profilers, sidescanning sonar, acoustic speedometers, fish finders, divers aids, position marking, homing beacons, transponders, communication and telemetry, as well as control applications such as acoustic releases all are examples of military and civilian uses of sonar [5]. Although sonar technology has been under development for the past several decades, it is still not without its shortcomings. The accuracy and range of
acoustic measurement and communication systems can be significantly degraded by hydrodynamic variability in water column. The negative impact of this variability upon acoustic technology still motivates research today in areas such as transducer arrays, acoustic signal processing, and physical oceanography. This thesis examines internal wave ocean dynamics, the effects of these dynamics on acoustic propagation, and in turn how to improve acoustic systems’ effectiveness through signal processing and inference of environmental dynamics.

Acoustic energy can be affected in many ways while propagating through the ocean waveguide. Reflections from the ocean surface redirect and scatter acoustic waves, energy is reflected, scattered, and absorbed by the seafloor, and acoustic waves are refracted within the water column. A combination of seafloor absorption and scattering, along with sea surface scattering, causes the majority of acoustic variability in shallow-water propagation scenarios [4, 5]. However, perturbations in the sound speed profile within the water column can also effect acoustic propagation by refracting (redirecting) acoustic energy toward the region of lower sound speed [4, 6]. Although refraction alone does not cause acoustic variability, spatial changes in the sound speed field cause changes in the refraction of energy and in turn alter the acoustic field.

These acoustic phenomena are generally thought of in a negative manner, however, refraction is actually the reason acoustic energy can be used for long-range communications in the ocean. It is the cause of the acoustic SOFAR (SOund Fixing And Ranging) channel that can be found in most of the world’s deep oceans. Low-frequency acoustic energy can propagate large distances with minimal losses because the signal does not contact the ocean’s surface or seabed. Recordings of explosive sources have been made thousands of miles from the source. A typical sound speed profile found in the deep ocean is shown in Figure 1.1 taken from *Computational Ocean Acoustics* by Jensen et al. [4]. A sound speed minimum causes rays within a certain angular aperture to be upward and downward refracted continually around that minimum (the deep sound channel axis). Never coming into contact with the loss mechanisms of the surface and bottom of the ocean, this angular aperture of acoustic energy can propagate extremely long distances (assuming frequencies low enough that there is little attenuation due to chemical relaxation) [4]. The SOFAR channel does not generally exist in shallow-
water shelf regions, thus acoustic energy there continually interacts with the sea’s surface and seafloor, as well as water column variability.

Figure 1.1. Generic deep-water sound speed profile, taken from *Computational Ocean Acoustics* by Jensen *et al*.

Refraction of acoustic waves can cause energy to be redirected toward or away from a receiver, causing unpredictable variations in received acoustic signal unless much is known about the propagation path. In shallow-water applications, the profile in Figure 1.1 is almost never found. Instead, factors such as complicated bathymetry, sharp thermocline and active mixed layer, freshwater drainage, tidal and wind driven forcing, and other water column dynamics have a great impact upon the sound speed field. Propagating internal gravity waves can also be added to this list, for they alter both the spatial and temporal state of the sound speed field. This work is concerned specifically with the complex effects that propagating internal waves had upon the acoustic propagation paths and received acoustic signals during the TAVEX 2008 experiment.
1.5 Internal Gravity Waves

In general, the ocean is considered a horizontally stratified medium, vertically varying in temperature and density and in turn sound speed [4, 6]. This vertical gradient within the water column is caused mostly by solar heating at the sea surface and local pressure increasing with depth. During warmer months solar heating creates a strong thermocline as well as pycnocline within the water column (density is also dependent on temperature, salinity, and pressure [7]). Internal gravity waves (referred to as simply internal waves hence forth) are propagating subsurface waves that travel along this density gradient. As they propagate, they displace water, causing perturbations in the sound speed field in both the horizontal and vertical directions [3, 5, 6]. Displacement of the horizontally stratified sound speed field within the thermocline region can be seen in Figure 1.2, which was measured during TAVEX 2008. This figure shows the warping of the sound speed field caused by a single internal wave depression. Internal waves have maximum displacement amplitude at the highest density gradient of the water column; this amplitude then goes to zero (in ideal theory) at the upper and lower boundaries (sea surface and seafloor) [3].

Internal waves are commonly encountered throughout the world’s oceans in both deep and shallow-water regions [3]. They are often observed propagating away from or toward the continental shelf during the summer, and are associated with large scale water movement (tidal fluctuations) over abrupt changes in bathymetry, as well as strong thermocline conditions due to solar heating [3]. These waves can be relatively large-scale compared to surface waves, with wavelengths ranging from a few hundred meters to kilometers, and amplitudes on the order 10 m [8]. Shallow-water internal waves can cause vertical density, temperature, and salinity fluctuations over much of the water column and are very persistent, propagating long distances over shelf regions [9]. These fluctuations create both vertical and horizontal perturbations in the sound speed field.
1.6 Background

Previous to the 1990’s, little work was done to understand the effects of water column sound speed fluctuations on acoustic propagation in shallow-water [10, 11]. Most of the interest was in deep-water transmission problems where internal wave activity is well characterized by the Garrett-Munk spectrum [12, 13, 14]. One of the earliest indications of how strongly internal waves could affect acoustic propagation in shallow-water environments came from work by Baxter and Orr [15]. In this paper they use water column and bathymetry measurements as inputs to a short-range ray trace code used to display the significant influence of internal wave perturbations on close range high-frequency acoustic propagation. Early work concerning internal wave influence on long-range acoustic propagation, is by Zhou et al. [16]. This experimental paper was written before any long-range acoustic data had been collected concurrent with observations of internal waves in a shallow-water region. They discuss sound propagation measured over several years in an area where seafloor properties and bathymetry were well known. In these recordings, transmission loss (TL) followed standard propagation model
predictions except during summer months. They hypothesized that propagating internal waves were the cause of this deviation from predictions. Using a parabolic equation (PE) model to predict the effects of internal waves on acoustic propagation with a strong thermocline present, they found TL to be sensitive to signal frequency. Then, decomposing the acoustic field into normal modes, they showed that TL variation was linked to mode coupling induced by the internal wave field.

A paper by Rubenstein and Brill [16] notes TL variation over a wide range of time scales during transmission of 400 Hz CW pulses over 18.5 km to a HLA. In this study, moored sound speed time series (from vertical thermistor chain data) are advected at a constant speed through the computational environment to simulate the propagation of an internal wave train. Then using a PE model with both time and range-dependent steps in the sound speed field, they determine that propagating internal wave packets can explain the acoustic variability seen in the 400 Hz signal.

Long before the 1990’s, work began concerning the analytical modeling of ocean internal waves using hydrodynamic physics and data measurement comparisons. Ostrocsky and Stepanyants [17] give a comprehensive overview of early internal wave modeling work directly focused on the ocean and other large bodies of water. Three recent publications, valuable to this research for their deterministic modeling of internal wave fields and comparison with data measurements, were heavily influenced by John Apel. The first, authored by Apel [18], gives a complete but reasonably succinct discussion of the three most common analytic models for deterministic propagating internal wave fields. Apel et al. [19] then discusses these models further and compares with experimental data. The third, an online publication by Jackson [9], gives an overview of these models as well as measurements of internal waves from around the globe. A journal paper by Rubenstein [20] discusses the use of two of these models in computational work comparing with experimental data. All of these works were important in the development of the analytic internal wave model for this research.

Prior to the TAVEX 2008 experiment, there have been a multitude of other experiments and analysis devoted primarily to discerning and understanding the physical effects of oceanic hydrodynamic variability (of which internal waves are part of) on acoustics within a shallow-water shelf region. The following is not an
exhaustive listing of these experiments and research, however, it does provide an overview of related experiments for either the current work or future research by the reader.

Two of the largest and most influential of these experiments are the Shallow-Water Acoustics in a Random Media (SWARM) [21, 22, 23, 24, 25, 26, 27, 28] and Shallow Water ’06 (SW06) [29, 30, 31, 32, 33, 34] experiments. Both SWARM and SW06 took place off the New Jersey coast, and were collaborative efforts between many institutions including the Woods Hole Oceanographic Institution (WHOI), University of Delaware, Naval Research Laboratory (NRL), Naval Post Graduate School (NPS), and the Applied Physics Laboratory of Johns Hopkins University (APL/JHU). The 1996-97 New England Shelfbreak Front PRIMER Experiment [35, 36] is another example of coastal variability research in this region. The Strait of Gibraltar Acoustic Monitoring experiment [37, 38] took place during the spring of 1996, with the purpose of determining the effects on high-frequency acoustic propagation through high-amplitude (∼100 m) internal waves. Taking place in August of 1992, the Barents Sea Polar Front (BSPT) experiment [39, 40, 41] gathered data to analyze the acoustic propagation and unique water column dynamics within that region. The INternal Tide Investigation by Means of Acoustic Tomography Experiment (INTIMATE) [42, 43, 44, 45, 46, 47] of 1996 took place off the coast of Portugal. Here researchers’ interest was the use of acoustic tomography, among other methods, for the detection and inversion of internal waves and tides. The Shelf Edge Study Acoustic Measurement Experiments (SESAME) [48, 49, 50, 51] (two experiments during the summers of 1995 and 1996) were conducted off the coast of Scottland. In these experiments both moored and towed (similar to TAVEX) hydrographic measurements were made. Finally, the Asian Sea International Acoustics Experiment (ASIAEX) [52, 53, 54, 55, 56], was conducted by the US, South Korea, China, Japan, Taiwan, Singapore, and Russia during 2000 and 2001. Taking place in both the South and East China Seas (south of the TAVEX experimental region), ASIAEX was conducted with primary interests of acoustic boundary interaction and geoacoustic inversion. However, some data from this experiment is useful in research connected with internal waves [57, 58].

Some papers deserve specific mention for their influence on the work presented in this thesis. Multiple papers were used as the basis of the acoustic data analysis
techniques, as well as references for expected internal wave influence. Two papers by Grigorev and Katsnelson [59, 34] (the latter also includes Lynch as a co-author) helped formulate the analytic ray model used in acoustic analysis, and give a useful conceptual explanation of internal wave field effects on acoustic variability. In [59], a ray model is developed to link a specific aperture of rays to acoustic variability caused by a single propagating soliton. In [34], SW06 data is used to validate this model. Moum and Nash [32] discuss hydrostatic pressure measurements made off the coast of New Jersey in the summer of 2006 (part of SW06). They use temporal filtering methods and display the power spectrum of pressure variation in a low-frequency (internal wave spectrum) spectrogram format. Rutenko [60] also uses a spectral analysis of acoustic data. However, this analysis examines internal wave effects through the frequency dependence of energy transmitted by a noise-like source, not temporal variance. Two papers by Badiey et al. [22, 23] use data from the SWARM experiment, and the connection of acoustic variability to an anisotropic internal wave field. The first [22], uses a combination of vertical modes, horizontal rays, and a horizontal 2D PE model to show the horizontal refraction of acoustic propagation when it is nearly parallel to the internal wave front. It also discusses the frequency dependent behavior of propagation through an internal wave field. The second [23], introduces a theory to examine frequency dependence in the presence of propagating internal waves and breaks down the angular dependence of acoustic propagation through an internal wave field. These two papers give some intuition about the expected acoustic effects of the internal wave field. Specifically, knowledge of the angular dependence of acoustic propagation through an internal wave field. This is important when deciding what internal wave measurements from TAVEX should be used in 2D propagation comparison models. The method of creating an experimentally constrained temporally and spatially dependent computational environment for PE simulations found in [22] is also used during the present research to simulate a temporally evolving acoustic field.
Chapter 2

The TAVEX 2008 Field Experiment

2.1 Chapter 2 Overview

Chapter 2 discusses environmental considerations of the experimental region, data collection systems, and data formatting. Data includes towed CTD, moored temperature and acoustic recordings, bathymetry, and acoustical sediment properties. The towed CTD system is a lesser known measurement method for gathering high-resolution 2D information about water column variability and structure. Moored temperature recordings provide stationary point measurements of vertical water column dynamics and are useful in determining internal wave amplitude and propagation direction. Unfortunately, the TAVEX moored temperature recordings were made too low in the water column to provide useful internal wave signatures. For this reason, towed CTD data will be used in the following chapter to construct an analytic internal wave model used to estimate internal wave propagation directions. In later chapters internal wave measurements and directionality along with acoustic data will be analyzed and compared with acoustic computational models to search for internal wave signatures within acoustic recordings. Bathymetric and sediment data is used in computational acoustic models and was found using the CTD tow vessel’s depth sounder and acoustic signal inversion respectively.
2.2 Description of the Experiment

TAVEX 2008 took place off of the southern coast of South Korea in the northern East China Sea approximately 100 km southwest of Jeju Island at roughly 32°N 40′ latitude and 125° 26′ longitude. The location of this experiment is shown in Figure 2.1 from Google Earth, denoted by a red circle. Figure 2.2 shows an enlargement of the CTD measurement area based upon Global Positioning System (GPS) data using M_Map [61], a MATLAB tool kit for mapping global data. The border of the towed CTD measurement area is shown with a black rectangle and the tow path is marked by a blue track. HLA and source locations are denoted by black squares, and the acoustic axes between the 300 and 500 Hz sources and HLA are denoted by dashed black lines. The HLA is oriented approximately parallel to the acoustic wave front and the southeastern border of the tow area. The lengths of the 300 and 500 Hz acoustic axes are 34 and 20 km respectively. CTD tows were made both parallel and perpendicular to the acoustic axes, however, most tows were made perpendicular, crossing the acoustic axes nearly parallel to the acoustic front. Note that no CTD data was collected along either acoustic axis.

![Figure 2.1. Location of the TAVEX 2008 denoted by red circle.](image)

The East China Sea and surrounding area is an extremely complex region from the perspective of oceanography, hydrography, and bathymetry. In turn, the
internal wave activity within this region is also extremely complex. A discussion of contributors to this complexity follows to provide an understanding of the water column character and other important factors of the experimental region.

Within the East China Sea there are many mechanisms for generating internal waves. Mechanisms include tidal forcing, forcing by the Kuroshio, Tsushima, and Yellow Sea Circular currents, and upwelling induced by the intrusion of the Kuroshio across the continental shelf (although this mostly takes place in the southern region near Taiwan) [62, 63]. Bathymetry is also an important factor in internal wave generation and propagation. Internal waves are generally created by tidal or other hydrodynamic forcing against bathymetric features such as the continental shelf or islands [3]. Figure 2.3 shows the general bathymetry of the region with major boundary currents and the location of the tow area (red rectangle) superimposed. This figure was taken from the East China Sea section of Jackson’s An Atlas of Internal Solitary-like Waves and their Properties [9], however the original bathymetry data is from Smith and Sandwell [64]. Schematic movement of
the Kuroshio boundary and branch currents was taken from Qui and Imasato [63]. The bathymetry of the region local to the TAVEX experiment appears relatively flat in this figure. This is consistent with depth sounder data shown in Figure 2.4. This data was taken during the TAVEX CTD measurements using the R/V EARDO’s depth sounder. It shows that the experimental region is relatively flat, varying from about 70 to 85 m within the 10 by 34 km tow area. The variation along the two acoustic paths (denoted by red dashed lines in this figure) is even less, varying only from 72 to 80 m between each source and the HLA. Important bathymetric internal wave generation points include Jeju and smaller nearby islands, small islands in the Southern Yellow Sea and along the Coast of Mainland China, and the continental shelf break at the East China Sea’s eastern edge.

![Figure 2.3. Bathymetry and major currents of the East China Sea region. Depth contours in meters.](image)

Other important environmental parameters for acoustic propagation are seafloor
The seafloor of the experimental region is reportedly known to be dominated by sediment deposits caused by discharge of the Yangtze and other large rivers flowing into the East China and Yellow Seas. The geoacoustic parameters of the seabed near the HLA were estimated using acoustic precursor arrivals of impulses from large light bulb implosions that were recorded during TAVEX. This analysis was presented by Choi et al. at the 2009 24th Underwater Acoustics Symposium in South Korea [65]. They report that the results show strong patchy-scatterers including mixtures of shell, sand, and mud. Using a PE acoustic propagation model, they invert the acoustic data and fit a layered bottom with the seabed properties given in Table 2.1. These parameters are used later in this work.
Kim et al. [66], also from the Korean Symposium, briefly discuss an inversion of ship radiated noise of the same area. Their estimate for sediment attenuation ($\alpha = 0.2$ dB/λ) is used for this research since the inversion of Choi et al. did not account for sediment attenuation in order to emphasize sediment path arrivals. It is important to mention that sea state is not accounted for in this thesis. However, maximum and minimum significant wave height (average of the one-third largest surface waves) recorded at the Korean Ocean Research Station (IEODO) [67] located approximately 65 km southwest of the CTD tow area were 1.3 and 0.7

Figure 2.4. Bathymetry of CTD tow region measured by R/V Eardo’s depth sounder.
Depth Below Seafloor [m] | Sound Speed [m/s] | Density [kg/m³]
---|---|---
0-2.5 | 1500 | 1450
2.5-12 | 1580 | 1850
12-20 | 1615 | 1950
>20 | 1800 | 2200

Table 2.1. Geoacoustic parameters estimated for the tow area by Choi et al.

m respectively. Average significant wave height throughout the entire period was 0.92 m.

### 2.3 Towed CTD Measurements

Towed CTD data was collected by the Applied Research Laboratory (ARL) at Penn State University from 8/28/2008 9:15:16 GMT to 8/29/2008 21:08:26 GMT. This data was collected using the "ADM Towed CTD-Chain" designed by Jurgen Sellschopp [2]. This system measures the conductivity, temperature, and pressure at specific points in the water column using CTD fins arranged along a coated steel cable. This arrangement is towed behind a research vessel with cable and fins held at a relatively vertical angle in the water column by a winged depressor connected to the bottom of the tow cable. Each CTD fin is equipped with temperature, conductivity, and pressure sensors. Towing the chain through the water provides depth and range measurements of the water column in conductivity, temperature, and pressure. The latter is used to approximate the depth of each CTD fin at the time of each sample and create two-dimensional (2D) fields of conductivity and temperature in the water column. Each fin communicates its readings in real-time to an on-ship control unit known as the CTD deck box by means of an inductive connection to the cable, which also supplies power to the fins. This inductive coupling scheme allows maximum flexibility in setting fin sampling depths. The single conductor insulated steel tow cable uses the sea water as the return path for both the communications and powering of the CTD sensors. The deck box is controlled via RS232 connection to an on-ship PC, and the PC controls communication, sampling, and data storage operations. The deck box controls low level communications between the PC and CTD fins [2] [68] [69].

The inductive communications loop consists of the following components; deck
box, coated steel tow cable, load resistance at the bottom of the cable, an electrode connected to the resistor at the bottom in connection with the seawater, and an electrode in connection with the sea water near the ocean’s surface that returns the signal back to the deck box. Note that the inductive communication and powering of this system makes it relatively easy to move the sensor fins to any location along the cable in order to adjust the vertical spacing of water column measurements. It also allows the sensors to spin freely around the cable, decreasing the amount of drag applied to both the sensors and cable during higher tow speeds. Rubber clamps or stops are placed above and below the fins to keep them in place along the cable. A physical representation of this system can be found in Figure 2.5. The construction of CTD fins and the location of sensors within them can be found in Figure 2.6, which is taken from the ADM CTD chain website [68].

![Figure 2.5. Schematic of the CTD chain measurement system used in the TAVEX 2008.](image)

During TAVEX 2008, towed CTD measurements were made using thirty-four fins. The fins were positioned more closely together within the thermocline region to allow better resolution in more dynamic areas of the water column. The distance each CTD fin was placed along the cable (linear cable length, not depth) from the approximate point that the cable entered the water at the stern of the ship is given in Table 2.2. This table displays the length of cable between the water
Figure 2.6. CTD fin enlargement from the ADM CTD Chain website.

surface and each CTD fin, not the actual depth of the fin. Calculation of CTD fin depths from pressure recordings is discussed in Appendix A. The water column was sampled throughout the thermocline upward into the lower regions of the mixed layer and downward into the upper regions of the isothermal layer. The upper and lowermost regions of the water column were not sampled. However, these regions are within the low-variability mixed (semi-isothermal) and deep isothermal layers. Therefore, temperature and conductivity in these regions can be assumed to be constant values from the upper most sampled region to the sea surface and from the lower most sampled region to the seafloor. Using conductivity, temperature, and pressure data, the 1978 practical salinity scale [70] can be used to calculate salinity and the UNESCO algorithm [71] is used to calculate sound speed. The effect of pressure increasing with depth is accounted for where there are no CTD measurements.

CTD chain measurements were made for 36 hours during the TAVEX 2008 experiment. Sample rates were 1 or 2 s, with most of the data collected at a 2 s sample rate. All data presented in this thesis was recorded at a 2 s sample rate. The resulting horizontal resolution of this measurement is approximately 3.6 m based upon an average tow speed of 3.5 knots (1.8 m/s). Vertical resolution depends upon sensor spacing along the cable (Table 2.2) which ranges from 0.6 to 2.4 m, with shorter spacings used within the thermocline region as previously
<table>
<thead>
<tr>
<th>Sensor Address #</th>
<th>Distance [ft]</th>
<th>Distance [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>44</td>
<td>13.41</td>
</tr>
<tr>
<td>50</td>
<td>46</td>
<td>14.02</td>
</tr>
<tr>
<td>33</td>
<td>48</td>
<td>14.63</td>
</tr>
<tr>
<td>16</td>
<td>50</td>
<td>15.24</td>
</tr>
<tr>
<td>41</td>
<td>52</td>
<td>15.85</td>
</tr>
<tr>
<td>31</td>
<td>54</td>
<td>16.46</td>
</tr>
<tr>
<td>17</td>
<td>56</td>
<td>17.07</td>
</tr>
<tr>
<td>43</td>
<td>58</td>
<td>17.68</td>
</tr>
<tr>
<td>47</td>
<td>60</td>
<td>18.29</td>
</tr>
<tr>
<td>13</td>
<td>62</td>
<td>18.90</td>
</tr>
<tr>
<td>46</td>
<td>64</td>
<td>19.51</td>
</tr>
<tr>
<td>40</td>
<td>66</td>
<td>20.12</td>
</tr>
<tr>
<td>21</td>
<td>68</td>
<td>20.73</td>
</tr>
<tr>
<td>24</td>
<td>70</td>
<td>21.34</td>
</tr>
<tr>
<td>30</td>
<td>72</td>
<td>21.95</td>
</tr>
<tr>
<td>44</td>
<td>80</td>
<td>24.38</td>
</tr>
<tr>
<td>18</td>
<td>88</td>
<td>26.82</td>
</tr>
<tr>
<td>12</td>
<td>96</td>
<td>29.26</td>
</tr>
<tr>
<td>39</td>
<td>104</td>
<td>31.70</td>
</tr>
<tr>
<td>35</td>
<td>108</td>
<td>32.92</td>
</tr>
<tr>
<td>7</td>
<td>112</td>
<td>34.14</td>
</tr>
<tr>
<td>48</td>
<td>116</td>
<td>35.36</td>
</tr>
<tr>
<td>28</td>
<td>120</td>
<td>36.58</td>
</tr>
<tr>
<td>19</td>
<td>124</td>
<td>37.80</td>
</tr>
<tr>
<td>14</td>
<td>128</td>
<td>39.01</td>
</tr>
<tr>
<td>27</td>
<td>132</td>
<td>40.23</td>
</tr>
<tr>
<td>49</td>
<td>136</td>
<td>41.45</td>
</tr>
<tr>
<td>29</td>
<td>144</td>
<td>43.89</td>
</tr>
<tr>
<td>9</td>
<td>152</td>
<td>46.33</td>
</tr>
<tr>
<td>26</td>
<td>160</td>
<td>48.77</td>
</tr>
<tr>
<td>8</td>
<td>168</td>
<td>51.21</td>
</tr>
<tr>
<td>34</td>
<td>176</td>
<td>53.65</td>
</tr>
<tr>
<td>37</td>
<td>184</td>
<td>56.08</td>
</tr>
<tr>
<td>45</td>
<td>192</td>
<td>58.52</td>
</tr>
</tbody>
</table>

**Table 2.2.** Linear distance of each CTD fin along the tow cable (not depth when ship is in motion) measured from the point that the cable enters the water at the stern of the research vessel.
mentioned.

The CTD system includes a 2 m diameter drum to store and deploy the cable/fin assembly (Figure 2.7). The cable and fins were deployed through a block at the rear of the ship as shown in Figure 2.8. A standard YSI type 167 V-fin [72] (Figure 2.9) was attached to the lower end of the cable directly below the lower CTD Chain electrode.

Figure 2.7. CTD cable/fin assembly stored on drum prior to deployment.

Two DOS based executable programs, CTDSENS.EXE and CTDCHAIN.EXE, are required to use the towed CTD system [2, 69]. CTDSENS.EXE is used to poll the CTD system to find which CTD fins are working properly, and allows the user to change certain parameters within the CTD fins. These parameters include fin address number (1-255) and what measurements that particular fin should make (temperature, conductivity, pressure; any two, or all three). CTDCHAIN.EXE is the measurement and recording program used to start CTD measurements after all parameters have been set using CTDSENS.EXE. In order to use CTDCHAIN.EXE, ASCII files with extensions of .CAL and .CFG must be in the same directory. These are fin calibration and configuration files, respectively. The calibration file contains
Figure 2.8. CTD cable/fin assembly deployed through block at stern of ship.

Figure 2.9. Vfin (type 167) on rear deck of ship just before deployment.
the coefficients necessary to convert each sensor’s analog-to-digital (A/D) integer values to physical units. The configuration file tells CTDCHAIN.EXE where each fin is physically located on the cable and assigns a color to be used to represent that fin’s measurements on a real-time display. Both the calibration and configuration files that were used to record CTD data discussed in this document can be found in Appendix A.

A Toshiba Satellite 4020CDT laptop (part number PAS402U-T2CW8) with a 300 Mhz processor and 64 MB RAM was used as the control and storage PC. This laptop was used because the processor speed can be lowered so that the DOS executables (CTDCHAIN.EXE and CTDSENS.EXE) needed to operate the CTD chain would run. Choosing the slowest possible CPU clock speed in the BIOS, both executables run correctly and consistently with this PC. Before decreasing the CPU speed, if an attempt was made to run either of the necessary executables on this laptop a “Runtime Error 200” would be encountered. This error is generally caused by executing older DOS programs on a computer with CPU speed greater than 200 MHz. This error occurs because these programs do not adjust their operation speed, instead using the CPU to clock operations directly. A second PC was also taken on the measurement cruise as a backup in case of a failure by the Toshiba. The backup PC was an IBM ThinkPad 390E laptop with a 233 MHz Pentium II processor and 64MB RAM. This PC was operable with the CTD system during pre-experiment system tests, but was found to be more susceptible to problems than the Toshiba Satellite. The IBM’s inconsistent operation may have been due to the CPU speed being slightly faster than 200 MHz. The backup PC was not used during the TAVEX experiment.

During CTD data collection, data was periodically backed up on 3.5 inch floppy disks. This was generally done while the ship was turning in order to avoid missing useful sections of data (motion of the CTD chain is difficult to predict while the ship is turning). After 36 hours of CTD measurements, the entire hard drive was backed up by physically removing the hard drive and using an IDE to USB hard drive adapter to copy all disk contents to a second computer. In this way, there were three separate copies of all CTD data and configuration files.

The towed CTD data collection system stores conductivity, temperature, pressure, and date/time measurements in a binary format file denoted by the extension
MATLAB scripts were written to read these binary files, filter the data to remove erroneous points caused by communication errors and bad sensors (there were many bad pressure sensors), and then extrapolate the data into 2D fields. MATLAB scripts written for this processing can be found in Appendix A. After processing of raw CTD data, salinity and sound speed are calculated using the 1978 practical salinity scale \cite{70} and the UNESCO algorithm \cite{71} respectively. Density is calculated using Phil Morgan’s seawater MATLAB toolbox \cite{73} from CSIRO Marine Research.

2.4 Moored Temperature Measurements

Moored temperature recordings were made at the location of both sources and the HLA using Star-Oddi brand Seamon and Starmon Mini temperature recording devices. These devices have a resolution of $\pm 0.05^\circ$C and a range of -2 to $+40^\circ$C. A combination of both recorders were used on the HLA but only Starmon Mini type were used at source locations.

Recorders were placed along the cables of the HLA. Placed on the sea floor, these temperature measurements varied less than $0.25^\circ$C throughout nine days of deployment. The resolution of these temperature sensors was not appropriate for such small temperature deviations ($< 0.25^\circ$C) and no internal wave signatures were detected in moored temperature data.

Temperature recording devices were also placed at the 300 and 500 Hz source locations along the cable to the floats holding the sources vertical on the moorings. In Figure 2.10, red circles indicate some of the temperature recording devices. Temperature measurements made at each source are shown in Figures 2.11 (300 Hz) and 2.12 (500 Hz). Temperature was sampled at 2 s intervals. The deepest sensor was placed 2 m above the source depth (64 m) and above it sensor spacing was 4 m. Seven recorders were placed on the 300 Hz cable and 8 on the 500 Hz cable, so that the depths of the shallower recorders were 38 and 34 m respectively. The recorders placed nearest the mooring floats (highest in the water column) measured maximum excursions in temperature $\sim 4^\circ$C. It was intended that the upper temperature measurements would be used to verify internal wave propagation direction and acoustic axis cross-time (using event-time triangulation). Unfortunately, tem-
Figure 2.10. 300 and 500 Hz sources and moorings prior to deployment. Several temperature recorders are circled in red to show their placement on the mooring cable. Left, both sources are shown along with a mooring float. Right, a close up of the cable and temperature recorders.

Temperature measurements ended about 10 h before the end of the CTD tow because their memory was full. Measurements were also found to be too low in the water column to detect internal wave passings. Looking at Figures 2.11 (300 Hz) and 2.12 (500 Hz), it is not possible to distinguish individual internal wave events from other temperature fluctuations. As a result, moored temperature measurements were not used in the present work.

2.5 Acoustic Measurements

Acoustic data was collected by the Naval Research Laboratory of Washington, DC. This data includes recordings from a 200 m long, 96-channel, HLA placed approximately perpendicular to the acoustic axes between each source and the receiver array. Placement of both 300 and 500 Hz sources and the HLA are shown in Figure 2.2. Both sources were placed at approximately 64 m depth. Each transmitted alternating CW and LFM pulses (LFM bandwidth was ±10% of CW frequency), each approximately 2 s in duration with approximately 2 s of dead time between each pulse. This resulted in a 8.388 s period center-to-center between CW and LFM pulse arrivals (from one CW to the next or from one LFM to the next). The 300 Hz signal propagated 34 km to the HLA and the 500 Hz signals propagated
Figure 2.11. Temperature measurements made along the 300 Hz source’s mooring cable. Color represents depth shown in legend. Time axis is in hours of August 28\textsuperscript{th} and 29\textsuperscript{th} of 2008. CTD tow starts at 8/28/2008 9:15:16 GMT.

Figure 2.12. Temperature measurements made along the 500 Hz source’s mooring cable. Color represents depth shown in legend. Time axis is in hours of August 28\textsuperscript{th} and 29\textsuperscript{th} of 2008. CTD tow starts at 8/28/2008 9:15:16 GMT.
20 km to the HLA. A diagram of the HLA deployment is shown in Figure 2.13, which is taken from NRL’s pre-experiment mooring diagrams. Figure 2.14 displays the deployment diagram for the 300 Hz source (500 Hz source is identical), also taken from NRL’s pre-experiment mooring diagrams.

**Figure 2.13.** Diagram of HLA mooring. Batteries and recording equipment are housed at the central weighted sled and the 32 and 64 hydrophone arrays were laid as straight as possible away from one another.

The HLA recorded acoustic data from August 22nd until August 31st of 2008. However the present work is focused on data taken during the 36 hour period when the CTD tow was underway (09:00:43 August 28th - 21:25:09 August 29th GMT). Each hydrophone was sampled at 3906.25 Hz and data was stored in a weighted sled located in the center of the HLA (see Figure 2.13) where batteries and recording equipment were located. The HLA was composed of two separate line arrays laid horizontal along the seafloor; one having 32 and one having 64 hydrophones. The 32 hydrophone array collected almost no useful data due to
technical issues (two hydrophones did collect some usable data), so this array is not used in this work. The 64 hydrophone array did function consistently and is the source of all acoustic data used in this work.

Since later analysis in this work deals with acoustic pressure, linear coefficients are used to convert hydrophone recording voltage to pressure. Manufacturer specifications for hydrophone sensitivity are -180 dB re 1V/µPa. A calibration coefficient $C_1$ for conversion of the acoustic data in volts to micropascals can be calculated using the following equations.

\[ -180 = 20\log_{10} \left| \frac{C_1}{1V/\mu Pa} \right| \]  \hspace{1cm} (2.1)

\[ C_1 = 10^{-\frac{180}{20}} \text{ V/µPa} = 10^{-9} \text{ V/µPa} \]  \hspace{1cm} (2.2)
There is also a data acquisition (DAQ) gain (composite gain of all recording system electronics) which is thought to be approximately 38 dB. This value for DAQ gain is taken from an experiment conducted prior to the TAVEX 2008. There is some indication that this value is not correct, the DAQ gain is assumed to have been changed since the previous experiment. In any event, coefficient $C_2$ is used to account for DAQ gain and is calculated as follows.

$$38 = 20\log_{10}|C_2|$$  \hspace{1cm} (2.3)

$$C_2 = 10^{\frac{38}{20}} \approx 79.43$$ \hspace{1cm} (2.4)

Using these two coefficients and a factor of $1/10^6$ to convert $\mu Pa$ to Pa, the recorded signal (RS) can be converted to received pressure with units of pascals as follows

$$p = \frac{1}{10^6} \frac{RS}{C_1 C_2} \approx \frac{RS}{10^6 10^{-9} 79.43} \text{ Pa} \approx 12.59 RS \text{ Pa.}$$ \hspace{1cm} (2.5)

Averaged received pressure is calculated by averaging over the center 1 second of received root mean squared (RMS) pressure for each CW pulse at each frequency. This is done in the frequency domain by taking the power spectral density (PSD) of the acoustic data, using only the 300 and 500 Hz bins, and then converting to RMS pressure by taking the square root of the average PSD multiplied by the effective frequency resolution. Fast Fourier transform (FFT) parameters are a 2048 point transform, Hanning window, and 50% overlap. An example of the output of this operation is shown in Figure 2.15 where a small section (taken from near 12.5 h into the 36 h CTD tow) of the received RMS pressure at hydrophone #96 at a frequency of 300 Hz is shown. Also within this figure is an enlargement of the pressure peak caused by a single CW pulse at the 300 Hz bin displaying the section used to find the RMS pressure (denoted by red markers). Figure 2.16 displays the averaged CW pressure throughout the CTD tow for hydrophone #96, showing typical variation. It is apparent from these plots that there is much variation in received pressure levels. The mean level of CW pulses recorded during the CTD tow is 119 and 110 dB re 1$\mu$Pa (referred to as only dB’s henceforth) for the 300 and 500 Hz signals respectively (hydrophones #’s 33, 53, 75, and 96 were used for this calculation).
Figure 2.15. A portion of the recorded RMS pressure at 300 Hz for hydrophone #96. Average pressure of each CW pulse is obtained by averaging of the center 1 second of each pulse (average value denoted by red markers).

Figure 2.16. Averaged received signal level (RL) for CW pulses for hydrophone #96. Average level is obtained by averaging of the center 1 second of each pulse.
No acoustic recordings were made near either source and source levels (SLs) were not measured prior to deployment, so calibrated source levels are not available. However, SL will be needed in Chapter 5 to estimate the received level (RL) from acoustic models. Therefore, source levels are estimated from RMS pressure and a transmission loss (TL) calculation using average water column parameters and a bathymetry estimated by linearly interpolating between depth sounder measurements along the acoustic axis (depth sounder measurements displayed in Figure 2.4). The range independent sound speed profile and bathymetry used in the acoustic modeling can be found in Figure 2.17 and Table 2.3 respectively. The TL fields were computed using Michael Collin’s Range Dependent Acoustic Model (RAM) [74] for each source and range and are shown in Figures 2.18 and 2.19. Dashed boxes indicate the regions used to find the average TL for each range, which is 70 dB for the 300 Hz and 66 dB for the 500 Hz source. Further explanation of the implementation of RAM used in this work is discussed in Chapter 4 and Appendix B.

![Figure 2.17. Average sound speed profile for entire CTD tow, which was used for range independent propagation models.](image)
**Figure 2.18.** RAM calculated TL for 300 Hz source with an average water column sound speed profile. Placement of source and receiver is shown as well as the region (encompassed by dashed box) used to approximate average TL at the HLA.

**Figure 2.19.** RAM calculated TL for 500 Hz source with an average water column sound speed profile. Placement of source and receiver is shown as well as the region (encompassed by dashed box) used to approximate average TL at the HLA.
Table 2.3. Bathymetry values used in RAM PE 2D computational models for each source’s acoustic path to the HLA (RAM linearly interpolates between these points). Notice that in general the bathymetry is flat, but is included for model accuracy.

Using the average received level of the CW pulses, along with average TL from the PE code, source levels are calculated to be:

\[
SL_{300\text{Hz}} = 119 \text{ dB} + 70 \text{ dB} = 189 \text{ dB}
\]

\[
SL_{500\text{Hz}} = 110 \text{ dB} + 66 \text{ dB} = 176 \text{ dB}.
\]

Both source levels were expected to be below 183 dB, the maximum source level of these sources at the time of purchase. Since then, they have been used in many field experiments and it is known that the output power was lowered for a previous experiment. However, exactly how much it was lowered is not known. The 500 Hz source level is within an acceptable range, but the 300 Hz source is 6 dB greater than the original maximum SL. This is of course a result of a lack of knowledge of the actual DAQ gain and environmental parameters. None the less, source levels of 189 and 176 dB for the 300 and 500 Hz sources, respectively, will be used in Chapter 5. As discussed in later chapters, acoustic analysis of internal wave effects...
is only concerned with relative pressure and is not affected by SL errors.
Chapter 3

Internal Waves

3.1 Chapter 3 Overview

This chapter discusses the measurements of internal waves recorded during the TAVEX 2008 experiment. Since moored temperature measurements were found not to contain internal wave detections, CTD measurements and an internal wave model are used to estimate internal wave propagation direction for each measurement event. Propagation direction is important because the direction of wave travel relative to the acoustic axis greatly affects the impact of the internal wave on the acoustic field [23, 75]. If the travel direction is not known, it is not possible to correctly model internal wave effects on the acoustic field, as discussed in the following chapter.

3.2 Internal Wave Measurements

As discussed in the previous chapter, internal wave activity within the TAVEX measurement area can be very complicated. To illustrate this point three synthetic aperture radar (SAR) images of the measurement area are used, the locations of which are presented in Figure 3.1. The RADARSAT-2 SAR image was taken approximately 8 hours prior to the start of towed CTD measurements, while Envisat-1 ASAR images were taken about 35 hours prior. This figure was taken from Observations of Internal Tides and Internal Waves near Ieodo Ocean Research Station in the East China Sea by Kim et al. [76]. Figure 3.2 displays the
three SAR images from Figure 3.1 superimposed and oriented northward. Some of the more visible internal wave fronts are highlighted in red, and the CTD tow area is denoted by a green rectangle. Highlighted internal waves include visibly apparent waves from all three SAR captures. It is apparent that there is an abundance of internal wave activity near the tow area, and that internal wave propagation through the CTD tow area could be in any direction with the possible exception of east to northeast.

**Figure 3.1.** Times and locations of SAR images shown in Figure 3.2. The single RADARSAT-2 SAR image was taken about 8 hours prior to CTD deployment, and the two Envisat-1 ASAR images were taken about 35 hours prior. Depth contours in meters.
During the 36 hours of towed CTD data collection, there were six measurement events in which significant depressions in the thermocline attributed to propagating internal waves were detected. The location of each of these events is shown in Figure 3.3 with each internal wave event labeled chronologically (IW1-IW6). Three events (IW1, IW5, and IW6) were encountered close to the acoustic axes between the 300 and 500 Hz sources and the receiver array. The legend in Figure 3.3 shows the approximate measurement time (based on the center of each event) for each event with respect to the 36 hour period (8/28/2008 9:15:16 GMT - 8/29/2008 21:08:26 GMT) of the CTD tow. The start and end placement of the ship during the CTD tow is annotated in this figure by "Tow Start" and "Tow End" respectively, and the path of ship travel is shown by black arrows along the ship track. Notice the ship track region annotated with "A". This denotes a region where the ship track is easily confused. At this point, the ship first turns southeast.
towards the HLA just after IW1 measurements. After that, it visits the HLA site before turning around and proceeding to the northwest and IW2’s measurements. The ship then returns to point A after IW6 measurements and turns here again toward the northwest just before the end of the CTD tow.

Figure 3.3. Track of CTD tow denoted by blue track. Location of internal wave events denoted by red segments. Internal wave events are labeled chronologically.

Figures 3.4 through 3.9 display the 2D sound speed fields calculated from CTD measurements for each internal wave event indicated in Figure 3.3. It is important to remember that these measurements were made by a moving research vessel, resulting in the apparent period ($T_{ap}$) and frequency ($f_{ap} = \frac{1}{T_{ap}}$) of the internal waves. The internal wave propagation speed and direction, as well as the ship’s speed and direction, must be taken into account in order to calculate the true
period and frequency. Every effort was made to keep the ship’s speed and heading constant between turns at the CTD region’s boundary, so that calculation of the true internal wave properties is a simple scaling function. The average $T_{ap}$ of each internal wave event is indicated in Figures 3.4-3.9, as well as the average maximum displacement of each internal wave $2\eta_0$. $T_{ap}$ is the average period of time between adjacent significant depressions or solitons of a wave event, and $2\eta_0$ is the average maximum displacement of all solitons within an event measurement. In this work, any significant single thermocline depression will be referred to as a soliton as is common in the literature. Technically, however, the term soliton should only be used for single, propagating depressions that retain their shape over range.

The measurements of IW1 in Figure 3.4 show at least four individual soliton depressions (there is of course some amount of subjectiveness in interpreting these data). This wave event was not completely measured since it is cut off at the right side of the plot due to the ship turning. Notice that the sound speed field does not return to a stable thermocline (as in the left portion of Figure 3.4). This makes it evident that not all of this event was captured. Figures 3.5 and 3.6 both show three individual soliton events, however, these events do not appear as simple depressions. These disturbances seem to display an upward displacement in the upper thermocline and a downward displacement in the lower thermocline. These are the characteristics of so-called mode 2 internal waves. Mode 1 and mode 2 internal waves differ in their displacement pattern within the water column. Each soliton of a mode 1 internal wave is composed of only an elevation or depression displacement, while mode 2 internal waves are composed of vertically opposing elevation and depression displacements (note that all TAVEX mode 1 measurements are depressive waves). Because IW2 and IW3 are mode 2 internal waves, the apparent period is measured across all three thermocline disturbances effectively doubling the apparent period (see Figures 3.5 and 3.6). This is done so the apparent period will be comparable to the mode 1 model discussed in the next section (assuming that the mode 1 period is effectively twice the mode 2 period). In Figure 3.7 only a single depression is measured. This event is truly a propagating soliton or solitary internal wave. Because of this, there is no measurable apparent period, so the period of this wave is approximated using the mathematical relationship found in Equation 3.6 relating the width of the soliton to the expected distance between
each depression as if it were a multi-soliton wave packet. This will be discussed further in the following section. IW5 and IW6 events are shown in Figures 3.8 and 3.9. They seem to display similar soliton characteristics, such as the shape of the onset of the packet and displacement amplitude. However, measurements of IW5 are cut off at the edge of the tow area much like the measurements of IW1. Looking at the depressions in the lower water column (since within this wave event it is difficult to distinguish specific solitons) IW5 seems to have only two significant solitons while IW6 has three. None the less, the IW5 and IW6 events are quite similar in appearance.

The internal wave measurements shown in Figures 3.4-3.9 are, by themselves, insufficient to estimate the true wavelength and propagation speed of the internal waves. The propagation direction of the internal wave must be known as well. In the following two sections a model for the internal wave field is developed and used to estimate wave direction. In Chapter 5 the model is used in acoustic computational comparisons.

### 3.3 Internal Wave Modeling

To determine the effects of internal wave properties on acoustic propagation, it is necessary to know the direction of propagation of the internal wave relative to the acoustic propagation path [23, 75]. It is also helpful to have a relatively simplistic analytic model of an internal wave with which to develop theory and run computational models for acoustic analysis. In this section, internal wave and environmental characteristics measured during TAVEX CTD tows are used to create an analytic model of each internal wave event. To accomplish this, each internal wave soliton perturbation will be modeled using the classic Korteweg-de Vries (KdV) solution [9, 18] for a 2-layer fluid. Then the closely linked Cnoidal model’s [18] relationship between characteristic width $2\Delta$ and wavelength $\lambda_{iw}$ (the distance between maximum depression points of individual soliton depressions within a wave packet) will be used to construct propagating wave packets from single solitons. Examining the internal wave literature, the classic KdV solution is the most commonly used soliton model. This is both because of its applicability and simplicity. Although it uses only a hyperbolic secant squared solution (this will be shown
Figure 3.4. Sound speed field of IW1 calculated from towed CTD measurements. Color scale in [m/s].

Figure 3.5. Sound speed field of IW2 calculated from towed CTD measurements. Color scale in [m/s].
Figure 3.6. Sound speed field of IW3 calculated from towed CTD measurements. Color scale in [m/s].

Figure 3.7. Sound speed field of IW4 calculated from towed CTD measurements. Color scale in [m/s].
Figure 3.8. Sound speed field of IW5 calculated from towed CTD measurements. Color scale in [m/s].

Figure 3.9. Sound speed field of IW6 calculated from towed CTD measurements. Color scale in [m/s].
shortly), this model captures the general variation of single soliton perturbations in most ocean scenarios and has been verified experimentally [18]. The Cnoidal model is a generalization of the KdV model in which Jacobian elliptic functions are used to create a wave packet rather than a single propagating soliton. This model has a dimensionless parameter $s$ which varies from 1 at the front of the packet to 0 at the trailing end of the packet. When $s = 1$ the Cnoidal model is identical to the KdV hyperbolic secant squared solution; when $s = 0$ the Cnoidal model is equal to a cosine squared function [18]. This creates a generally realistic wave packet with nonlinear solitons at the front of the packet and linear or sinusoidal solitons at the rear of the traveling packet. More detailed discussion and derivation of these models and parameters can be found in a paper by Apel published in 2003 [18], an electronic publication by Jackson [9], and Apel’s *Principles of Ocean Physics* text [3].

The single soliton KdV model is constructed using the average density measured above and below the thermocline to approximate a 2-layer fluid model. Figure 3.10 displays the regions of the water column (in dashed red boxes) that were averaged from before and after each internal wave measurement. This figure also shows the average density profile measured near the region where each internal wave was detected. Each profile was obtained by averaging CTD tow data horizontally for approximately 1.5 hours before and after each internal wave event. The upper (mixed layer) region of the water column is averaged to find $\rho_1$ (upper layer fluid density), the lower (isothermal) region of the water column is averaged to find $\rho_2$ (lower layer fluid density), and the thermocline region is averaged to find both $\bar{\rho}$ (average density at 2-layer interface) and $h_1$ the depth at which the density = $\bar{\rho}$ (upper layer thickness). This depth is the interface depth in the 2-layer fluid model. Using an average total water depth of $H = h_1 + h_2 = 75$ m for the tow region (approximated from bathymetric depth sounder data), gives the distance from the interface depth to the seafloor ($h_2$). Schematic representation of these quantities can be found in Figures 3.10 and 3.11, and parameter values for each internal wave event are given in Table 3.1.

Using the 2-layer KdV model and the parameters from Table 3.1, the linear
Figure 3.10. Average density profiles of water column near internal wave measurements and 2-layer model averaging regions.

Table 3.1. Parameters of 2-layer model for the environment surrounding each internal wave measured during CTD tow.

<table>
<thead>
<tr>
<th>internal wave</th>
<th>$\bar{\rho}$ [kg/m$^3$]</th>
<th>$\rho_1$ [kg/m$^3$]</th>
<th>$\rho_2$ [kg/m$^3$]</th>
<th>$h_1$ [m]</th>
<th>$h_2$ [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>IW1</td>
<td>1023.6</td>
<td>1020.0</td>
<td>1025.5</td>
<td>27</td>
<td>48</td>
</tr>
<tr>
<td>IW2</td>
<td>1023.6</td>
<td>1020.5</td>
<td>1025.5</td>
<td>27</td>
<td>48</td>
</tr>
<tr>
<td>IW3</td>
<td>1023.4</td>
<td>1020.5</td>
<td>1025.4</td>
<td>23</td>
<td>52</td>
</tr>
<tr>
<td>IW4</td>
<td>1023.2</td>
<td>1020.1</td>
<td>1025.4</td>
<td>23</td>
<td>52</td>
</tr>
<tr>
<td>IW5</td>
<td>1023.3</td>
<td>1020.0</td>
<td>1025.4</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>IW6</td>
<td>1023.3</td>
<td>1020.2</td>
<td>1025.4</td>
<td>24</td>
<td>51</td>
</tr>
</tbody>
</table>

phase speed $c_l$ (explained below) is found from

$$c_l \approx \sqrt{\frac{g(\rho_2 - \rho_1)h_1h_2}{\bar{\rho}H}}, \quad (3.1)$$
and the nonlinear phase $c_n$ speed (again, explained below) is found from

$$c_n = c_l \left( 1 + \frac{h_1 - h_2}{h_1 h_2 \eta_0} \right),$$  \hspace{1cm} (3.2)$$

where $g$ is the acceleration of gravity, $h_1$ and $h_2$ have been previously defined, and $2\eta_0$ is the maximum displacement amplitude of the soliton as described in the previous section. The linear phase speed of an internal wave packet is the propagation speed of the reasonably linear (sinusoidal) trailing soliton perturbations. The nonlinear phase speed is the propagation speed of nonlinear leading soliton perturbations within the wave packet. In this research, the traveling speed $c_{iw}$ of the entire wave packet is taken as the average of these two speeds

$$c_{iw} = \frac{c_l + c_n}{2}. $$  \hspace{1cm} (3.3)$$

This assumes that the average of the leading and trailing soliton speeds will be
approximately equal to the group speed of the wave packet. The KdV solution for the displacement \( A(x, t) \) of a soliton is

\[
A(x, t) = 2\eta_0 \text{sech}^2 \left( \frac{x - c_{iw} t}{\Delta} \right),
\]

which describes a single soliton moving with constant speed \( c_{iw} \). In this equation \( 2\eta_0 \) is taken from CTD data (Figures 3.4-3.9), \( 2\Delta \) is the characteristic width of the soliton, \( x \) is the location in range of the soliton depression, and \( t \) is time. The characteristic width is calculated using

\[
2\Delta = 2\sqrt{\frac{2(h_1 h_2)^2}{3(h_1 - h_2)\eta_0}}.
\]

In order to create a multi-soliton wave packet from the single soliton KdV model presented above, the Cnoidal model’s relation between the characteristic width of a single soliton \( 2\Delta \) and packet wavelength \( \lambda_{iw} \)

\[
\lambda_{iw} = 2\Delta K(s)
\]

is used, where \( K(s) \) represents the complete elliptic integral of the first kind and \( s \) is a measure of the nonlinearity of the wave (as described previously). A value of \( s = 0.5 \) is chosen to approximate the average nonlinearity throughout the wave packet. This provides a wave packet with identical solitons spaced evenly with a spacing of \( \lambda_{iw} \).

Finally, the depth-dependence of soliton amplitude decreasing away from the region of the highest density gradient in the water column must be accounted for. To do this, \( A(x, t) \) is scaled by the function \( W(z) \) of the Taylor-Goldstein (TG) equation [18]. The TG equation must be solved numerically, and will be a different function for each internal wave mode. But because only mode 1 internal waves are used in computational models in this work, only a mode 1 amplitude function is needed. The 2D soliton displacement is thus

\[
\eta(x, z, t) = W(z)A(x, t).
\]
The mode 1 $W(z)$ function that will be used for acoustic computational models in Chapter 5 is displayed in Figure 3.12. This function was computed by Peter Mignerey of NRL using the program iw-eigen, a FORTRAN code written by T.H. Bell at the NRL [77]. Using an approximation of a linear internal wave field and average parameters of the tow area, this program approximates the dispersion relation of the internal wave field from the average buoyancy frequency profile to numerically solve the TG equation. The dispersion relation relates the propagation speed of the internal wave to wavenumber and mode. An example of a 2D internal wave field created using Equation 3.7 is found in Figure 3.13. This figure displays a range-independent sound speed field (from average CTD tow measurements) perturbed by five individual soliton depressions (range and depth are arbitrary). This represents a typical internal wave field model. Internal wave model parameters, both measured and calculated, for each internal wave event measured during the CTD tow can be found in Table 3.2. The maximum wave amplitude and apparent frequency are taken from CTD measurements ($f_{ap} = \frac{1}{T_{ap}}$). The wave speed (Equation 3.3), wavelength (Equation 3.6), and characteristic width (Equation 3.5) are all modeled quantities. The parameters in Table 3.2 are now used to estimate internal wave propagation direction.

### 3.4 Estimating Internal Wave Propagation Direction

Since CTD measurements were not made directly between each source and the receiver array, it is important to estimate each internal wave’s speed and direction of travel. From these values the wavelength, characteristic width, and travel speed of the internal wave apparent to the acoustic track can be inferred. Using the internal wave parameters in Table 3.2, the propagation direction of each internal wave can be approximated with respect to ship travel using the Doppler shift induced by ship speed and the geometry of an infinite horizontal internal wave front with respect to the ship course found in Figure 3.14. The vector relations displayed in Figures 3.14 and 3.15 relate the travel speed $c_{iw}$ and direction of the internal wave packet and average ship speed $v_{sh}$ to the apparent internal wave
Figure 3.12. Depth-dependent internal wave amplitude function used to give internal wave models realistic amplitude over depth.

Figure 3.13. Example of an average sound speed field perturbed by modeled internal waves (arbitrary units of depth and range). Color scale in [m/s].
<table>
<thead>
<tr>
<th>Internal Wave</th>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IW1:</td>
<td>wave amplitude</td>
<td>$2\eta_0$</td>
<td>5.3 [m]</td>
</tr>
<tr>
<td></td>
<td>apparent frequency</td>
<td>$f_{ap}$</td>
<td>2.3 [mHz]</td>
</tr>
<tr>
<td></td>
<td>modeled wave speed</td>
<td>$c_{iw}$</td>
<td>0.97 [m/s]</td>
</tr>
<tr>
<td></td>
<td>modeled wavelength</td>
<td>$\lambda_{iw}$</td>
<td>526 [m]</td>
</tr>
<tr>
<td></td>
<td>modeled characteristic width</td>
<td>$2\Delta_{iw}$</td>
<td>284 [m]</td>
</tr>
<tr>
<td></td>
<td>average ship speed</td>
<td>$v_{sh}$</td>
<td>1.6 [m/s]</td>
</tr>
<tr>
<td>IW2:</td>
<td>wave amplitude</td>
<td>$2\eta_0$</td>
<td>3.1 [m]</td>
</tr>
<tr>
<td></td>
<td>apparent frequency</td>
<td>$f_{ap}$</td>
<td>2.0 [mHz]</td>
</tr>
<tr>
<td></td>
<td>modeled wave speed</td>
<td>$c_{iw}$</td>
<td>0.92 [m/s]</td>
</tr>
<tr>
<td></td>
<td>modeled wavelength</td>
<td>$\lambda_{iw}$</td>
<td>688 [m]</td>
</tr>
<tr>
<td></td>
<td>modeled characteristic width</td>
<td>$2\Delta_{iw}$</td>
<td>371 [m]</td>
</tr>
<tr>
<td></td>
<td>average ship speed</td>
<td>$v_{sh}$</td>
<td>1.7 [m/s]</td>
</tr>
<tr>
<td>IW3:</td>
<td>wave amplitude</td>
<td>$2\eta_0$</td>
<td>3.2 [m]</td>
</tr>
<tr>
<td></td>
<td>apparent frequency</td>
<td>$f_{ap}$</td>
<td>4.1 [mHz]</td>
</tr>
<tr>
<td></td>
<td>modeled wave speed</td>
<td>$c_{iw}$</td>
<td>0.88 [m/s]</td>
</tr>
<tr>
<td></td>
<td>modeled wavelength</td>
<td>$\lambda_{iw}$</td>
<td>532 [m]</td>
</tr>
<tr>
<td></td>
<td>modeled characteristic width</td>
<td>$2\Delta_{iw}$</td>
<td>287 [m]</td>
</tr>
<tr>
<td></td>
<td>average ship speed</td>
<td>$v_{sh}$</td>
<td>2.0 [m/s]</td>
</tr>
<tr>
<td>IW4:</td>
<td>wave amplitude</td>
<td>$2\eta_0$</td>
<td>3.3 [m]</td>
</tr>
<tr>
<td></td>
<td>apparent frequency</td>
<td>$f_{ap}$</td>
<td>2.6 [mHz]</td>
</tr>
<tr>
<td></td>
<td>modeled wave speed</td>
<td>$c_{iw}$</td>
<td>0.92 [m/s]</td>
</tr>
<tr>
<td></td>
<td>modeled wavelength</td>
<td>$\lambda_{iw}$</td>
<td>523 [m]</td>
</tr>
<tr>
<td></td>
<td>modeled characteristic width</td>
<td>$2\Delta_{iw}$</td>
<td>282 [m]</td>
</tr>
<tr>
<td></td>
<td>average ship speed</td>
<td>$v_{sh}$</td>
<td>2.1 [m/s]</td>
</tr>
<tr>
<td>IW5:</td>
<td>wave amplitude</td>
<td>$2\eta_0$</td>
<td>4.4 [m]</td>
</tr>
<tr>
<td></td>
<td>apparent frequency</td>
<td>$f_{ap}$</td>
<td>0.4 [mHz]</td>
</tr>
<tr>
<td></td>
<td>modeled wave speed</td>
<td>$c_{iw}$</td>
<td>0.95 [m/s]</td>
</tr>
<tr>
<td></td>
<td>modeled wavelength</td>
<td>$\lambda_{iw}$</td>
<td>510 [m]</td>
</tr>
<tr>
<td></td>
<td>modeled characteristic width</td>
<td>$2\Delta_{iw}$</td>
<td>275 [m]</td>
</tr>
<tr>
<td></td>
<td>average ship speed</td>
<td>$v_{sh}$</td>
<td>1.8 [m/s]</td>
</tr>
<tr>
<td>IW6:</td>
<td>wave amplitude</td>
<td>$2\eta_0$</td>
<td>4.6 [m]</td>
</tr>
<tr>
<td></td>
<td>apparent frequency</td>
<td>$f_{ap}$</td>
<td>1.5 [mHz]</td>
</tr>
<tr>
<td></td>
<td>modeled wave speed</td>
<td>$c_{iw}$</td>
<td>0.92 [m/s]</td>
</tr>
<tr>
<td></td>
<td>modeled wavelength</td>
<td>$\lambda_{iw}$</td>
<td>470 [m]</td>
</tr>
<tr>
<td></td>
<td>modeled characteristic width</td>
<td>$2\Delta_{iw}$</td>
<td>254 [m]</td>
</tr>
<tr>
<td></td>
<td>average ship speed</td>
<td>$v_{sh}$</td>
<td>1.6 [m/s]</td>
</tr>
</tbody>
</table>

Table 3.2. Internal wave event parameters both measured and calculated.
propagation speed $c_{ap}$. $c_{ap}$ is a conceptual parameter representing the speed an internal wave would have to be traveling to cause an equal $T_{ap}$ to be observed in moored CTD data (rather than a towed CTD chain). This parameter takes into account both internal wave speed and direction as well as ship speed and direction, two vector quantities. This parameter can be viewed mathematically as the difference (because of the geometry defined in Figure 3.14) between $c_{iw}$ and

Figure 3.14. Basic geometry used to find internal wave propagation direction.

Figure 3.15. Vector connection between ship speed, internal wave speed, and apparent internal wave propagation speed.
the projection of $v_{sh}$ onto $c_{iw}$ (since the internal wave propagation direction is the reference direction)

$$c_{ap} = c_{iw} - v_{sh} \cos \theta.$$  \hspace{1cm} (3.8)

Using the Doppler shift equation (Equation 15.9.3 from Kinsler et al. [78]) and the relation $\lambda = \frac{c}{f}$

$$f_{ap} = \frac{f_{iw} c_{ap}}{c_{iw}} = \frac{c_{ap}}{\lambda_{iw}},$$  \hspace{1cm} (3.9)

we can relate the Doppler shifted apparent frequency $f_{ap}$ (which is found from the inverse of the apparent period $T_{ap}$ of each wave event in CTD measurements) to the modeled internal wave wavelength $\lambda_{iw}$ and $c_{ap}$. The angle between the ship track and internal wave propagation direction $\theta$ can then be found by inserting Equation 3.8 into Equation 3.9 and solving for theta

$$f_{ap} = \frac{(c_{iw} - v_{sh} \cos \theta)}{\lambda_{iw}}$$  \hspace{1cm} (3.10)

$$f_{ap} \lambda_{iw} = c_{iw} - v_{sh} \cos \theta$$  \hspace{1cm} (3.11)

$$\cos \theta = \frac{1}{v_{sh}} (c_{iw} - f_{ap} \lambda_{iw}).$$  \hspace{1cm} (3.12)

The Doppler shift relation in Equation 3.9 is explained in Fundamentals of Acoustics by Kinsler et al. [78] in terms of acoustic waves, which can be considered analogous to horizontally propagating internal waves. Using Equation 3.12, the parameters displayed in Table 3.2 and the ship’s navigational track (Figure 2.2), internal wave propagation direction $\theta$ can be determined. It is necessary to use the ship’s navigational track to rule out the angular ambiguity found in Equation 3.12 caused by positive and negative angles of arccosine and the similar travel speeds of the ship and propagating internal waves. A graphical representation of this ambiguity is shown in Figure 3.16. In this figure there are four individual diagrams, each representing two angles of $\theta$ that will cause identical measurements of $T_{ap}$ and $f_{ap}$ because of the interaction between internal wave and ship speed and direction. Although this figure displays velocity vectors, these diagrams can be thought of in the spatial xy-plane as well. On the left side of Figure 3.16, it is quite intuitive to imagine that the ship moving through the internal wave front at an angle of $\pm \theta$ will produce equal measurements of $T_{ap}$ ($T_{ap} = \frac{\lambda_{iw}}{c_{ap}}$). However, on
the right of Figure 3.16, it is not so intuitive to imagine that the ship traveling at all of these angles will also produce equal measurements of $T_{ap}$. This is caused by the propagation speed of the internal wave being slightly less than the ship speed. It is not possible to determine from measurements whether the $c_{ap}$ of Equation 3.8 is positive or negative (these quantities are labeled as ”equal distances” in Figure 3.16). Looking at Figure 3.15 it is apparent that $c_{ap}$ can be either positive or negative when the ship speed vector moves into the 1st and 4th quadrants (since the magnitude of $v_{sh}$ is only slightly more than $c_{iw}$). Mathematically speaking, if

$$|c_{ap}| = |c_{iw} - v_{sh} \cos \theta| < |c_{iw} - v_{sh}|,$$  \hspace{1cm} (3.13)
then there are four different (but related) possible angles of travel for the internal wave. Two of these angles are found using Equation 3.12 and the remaining two are found from

\[ \theta_2 = \pm \arccos \left( \frac{2 c_{iw}}{v_{sh}} - \cos \theta \right), \]

where \( \theta \) is equal to the theta from Equation 3.12. The only TAVEX event that fell into this four angles group was IW5. In the current work, using the ship’s navigational track, it is possible to rule out most of these angles by eliminating the region that the ship just transversed if it had no event measurements. In Figure 3.17, the approximate angular region where there is 2 possible angles (light blue region) and four possible angles (light brown region) is shown. This figure displays much of the same information as Figure 3.16 but using a single xy-plane and average parameters for \( c_{iw} \) and \( v_{sh} \) for all events. This figure displays the general relationship between ship direction through the internal wave front and the number of possible \( \theta \)s (using average values for TAVEX).

**Figure 3.17.** Plot demonstrating angular ambiguity found in Equation 3.12 when wave travel speed is comparable to ship speed.
This analysis seems appropriate for IW1, IW5, and IW6 (Figures 3.4, 3.8, and 3.9 respectively) where there are multiple mode 1 solitons within the internal wave packet. However, looking at IW2 and IW3 (Figures 3.5 and 3.6), these are mode 2 internal waves as was discussed in the first section of this chapter, and performing this analysis on these waves initially resulted in unphysical imaginary numbers because the KdV model is based on mode 1 wave activity (mode 1 $\lambda_{iw}$ and $T_{ap}$).

For this scenario, it is assumed that the period of these mode 2 waves will be approximately equal to half of the mode 1 (current model) waves. Looking at IW4 (Figure 3.7), only a single propagating soliton is observed as opposed to a packet. Therefore, the relationship between the soliton’s wavelength and characteristic width (Equation 3.6) is used to estimate wavelength for propagation direction analysis. Since this problem is addressed in the time domain, multiplication of the temporal characteristic width of this single soliton by the complete elliptic integral of the first kind results in the period of the internal wave.

Using the parameters from Table 3.2 and Equation 3.12, $\theta$ is found for IW1 from

$$\theta = \arccos \left( \frac{1}{v_{sh}} (c_{iw} - f_{ap} \lambda_{iw}) \right)$$  \hspace{1cm} (3.15)

$$\theta = \arccos \left( \frac{1}{1.6 \text{ m/s}} (0.97 \text{ m/s} - (2.3 \text{ mHz})(526 \text{ m})) \right)$$  \hspace{1cm} (3.16)

$$\theta \approx \pm 99^\circ.$$  \hspace{1cm} (3.17)

As shown in Figure 3.18, this means that the angle between the course of the ship and the internal wave propagation direction is $\pm 99^\circ$. Now, because of the ship’s navigational track (the ship turned northwest, then southwest, and then southeast after measuring this wave) $+99^\circ$ can be ruled out as a possibility for $\theta$ since the wave was not measured again. Similar calculations for $\theta$ are carried out for IW2-IW6, and the results are summarized in Table 3.3. The angle between the 300 Hz source axis and and the trajectory of the wave is used for all calculations, the slight difference in angle ($\sim 8.5^\circ$) of the 500 Hz source axis to the 300 Hz axis is not accounted for.

After completing event direction analysis, the directionality of wave measurement events IW2 and IW3 as well as IW5 and IW6 are found to be nearly equal. These event pairs were also measured at nearly the same time during the tow. Close
Table 3.3. Internal wave events and propagation angles. θ is referenced to the ship’s course during event measurement.

<table>
<thead>
<tr>
<th>Internal Wave</th>
<th>θ°</th>
</tr>
</thead>
<tbody>
<tr>
<td>IW1</td>
<td>-99</td>
</tr>
<tr>
<td>IW2</td>
<td>106</td>
</tr>
<tr>
<td>IW3</td>
<td>131</td>
</tr>
<tr>
<td>IW4</td>
<td>-102</td>
</tr>
<tr>
<td>IW5</td>
<td>58</td>
</tr>
<tr>
<td>IW6</td>
<td>-82</td>
</tr>
</tbody>
</table>

examination of the CTD measurements and comparison with the ship’s track, reveal that IW2 and IW3 as well as IW5 and IW6 have very similar measurement characteristics (taking into account Doppler shift induced by the moving measurement vessel). Looking at Figures 3.5 and 3.6, one notices the same basic wave characteristics, including three mode 2 solitons. Looking at Figures 3.8 and 3.9, although part of the measurements of IW5 were cut off due to the ship turning at the edge of the CTD tow area, similar soliton and water column characteristics are observed. This leads to the conclusion that these two sets of wave events are actually single internal waves measured twice at different ship courses. Therefore, from this point forward, the single internal wave causing IW2 and IW3 events will be referred to as IW2&IW3, and the single internal wave causing events IW5 and IW6 will be referred to as IW5&IW6. The result of the analysis is therefore just four different internal waves propagating through the tow region during 36 hours of CTD measurements.

The angle of internal wave propagation corresponding with the acoustic axis can now be found. Ship course is assumed to be either parallel or perpendicular with the acoustic axis, ignoring minute changes in ship course (the only exception to this is calculations for IW2, which is approximately 60° from the acoustic axis). The angle φ of the IW1 wave event referenced to the acoustic axis is then

$$\phi = -99° - 90° = -9°.$$  

Similar methods were used to find φ for each internal wave from values of θ for each event measurement. Accounting for ship direction, and averaging the difference in event directions for IW2&IW3 and IW5&IW6, Table 3.4 displays the resulting
propagation angles ($\phi$) for all four internal waves. This table also displays internal wave cross-times, these will be discussed shortly. The final results of internal wave direction analysis are shown in Figure 3.19 using heavy black arrows through the event measurements for each internal wave. IW5 had two possible angles, 50 and 66°, which were averaged to 58°.

**Figure 3.18.** Plot demonstrating the angular ambiguity found in Equation 3.12 for the IW1 event.

Internal wave event directions are now used to find the approximate times when the effects of these waves should be evident in acoustic data. This is done using the angles from Table 3.4 and assuming each internal wave has an infinitely long planar horizontal front. Using the relations found in Figure 3.20, the length of time that the internal wave should take to cross the acoustic axes can be determined using the modeled wave speed and the angle from the acoustic axis. For example, IW1 was measured 13.3 h after the beginning of the towed CTD measurements roughly six-tenths of the way between the source and receiver (see Figure 3.3). Using the propagation direction angle with reference to the acoustic axis of $\phi = -9^\circ$ (the negative sign simply denotes the direction of propagation along the axis), the relationship between internal wave speed and the speed projected onto the acoustic
Figure 3.19. Enlargement of Figure 3.3 displaying the approximate direction of each internal wave measurement during CTD tows.

axis $c_{proj}$ from Figure 3.20, $c_{proj}$ is found

$$c_{proj} = \frac{c_{iw}}{\cos \phi} = 0.98 \text{ m/s.} \quad (3.19)$$

Since the 300 Hz acoustic track is 34 km, the internal wave must have taken about 9.5 hours to cross the acoustic track. The IW1 cross-time start and stop limits can then be found by

$$\text{start} = 13.3 \text{ h} - \frac{6}{10} \cdot 9.5 \text{ h} = 7.6 \text{ h} \quad (3.20)$$

and

$$\text{stop} = 13.3 \text{ h} + \frac{4}{10} \cdot 9.5 \text{ h} = 17.1 \text{ h.} \quad (3.21)$$

Using this method, the times that each internal wave should begin propagating through the acoustically probed region and leave the probed region are calculated and shown in Table 3.4 rounded to the nearest half hour. About an hour has been added to each end of these times to allow each event to fall between plot annotations in Chapter 5. Also, since IW2 and IW3 as well as IW5 and IW6 are measurements of the same internal waves, IW2&IW3 and IW5&IW6 propagation angles are found by averaging the difference between the two angle calculations for each measurement event in Table 3.3.
\[ k_{ny} = k_{iw} \cos \phi \]
\[ c_{proj} = \frac{c_{iw}}{\cos \phi} \]

**Figure 3.20.** Geometric relationship of linear wave fronts.

<table>
<thead>
<tr>
<th>Source</th>
<th>Internal Wave</th>
<th>( \phi^\circ )</th>
<th>Start [h]</th>
<th>Stop [h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>300 Hz:</td>
<td>IW1</td>
<td>-9</td>
<td>6.5</td>
<td>18.0</td>
</tr>
<tr>
<td></td>
<td>IW2&amp;IW3</td>
<td>44</td>
<td>24.5</td>
<td>33.5</td>
</tr>
<tr>
<td></td>
<td>IW4</td>
<td>-102</td>
<td>31.0</td>
<td>35.0</td>
</tr>
<tr>
<td></td>
<td>IW5&amp;IW6</td>
<td>-12</td>
<td>28.0</td>
<td>40.0</td>
</tr>
<tr>
<td>500 Hz:</td>
<td>IW1</td>
<td>-9</td>
<td>11.0</td>
<td>18.0</td>
</tr>
<tr>
<td></td>
<td>IW2&amp;IW3</td>
<td>44</td>
<td>28.0</td>
<td>33.5</td>
</tr>
<tr>
<td></td>
<td>IW4</td>
<td>-102</td>
<td>31.0</td>
<td>34.0</td>
</tr>
<tr>
<td></td>
<td>IW5&amp;IW6</td>
<td>-12</td>
<td>32.0</td>
<td>40.0</td>
</tr>
</tbody>
</table>

**Table 3.4.** List of four internal waves and their expected acoustic axis cross-times.
Chapter 4

Acoustic Modeling

4.1 Chapter 4 Overview

In this chapter, acoustic analysis and modeling methods are developed in order to analyze the effects of internal waves on acoustic propagation. Spectral analysis of the averaged CW pulse pressure is discussed. A ray model is used to develop an understanding of the physics associated with internal wave induced acoustic variation, and a computational PE model is introduced which will be used to model received acoustic pressure in the following chapter. The techniques developed in this chapter will be applied in Chapter 5 to search for internal wave signatures within the acoustic data taken during TAVEX.

4.2 Harmonic Variation Analysis

In order to look for the effects of propagating internal waves on the acoustic data, spectral analysis is applied to the averaged CW pressure. This method is used because of variations of the received pressure in hydrophone recordings, and the periodic nature of internal waves. The periodic nature of the internal wave model implies that an acoustic signal transmitted through an internal wave field will be modulated in some manner related to the speed and temporal characteristics of the internal wave. Since internal wave perturbations are nonsinusoidal, it is expected that this modulation will also be nonsinusoidal, causing the spectral analysis to be composed of a fundamental frequency and harmonics of the fundamental frequency.
Complex temporal behavior can be explained using Fourier theory, which states that any signal can be decomposed into a fundamental frequency \( f_0 \) and integer harmonics of this frequency (\( 2f_0, 3f_0, 4f_0, \ldots \)). The modulation of acoustic pressure can be quantitatively analyzed using Fourier analysis.

Modulation of the acoustic pressure will be analyzed for individual hydrophones of the HLA. This method is similar to that used by Moum and Nash [32], Rutenko [60], Katsnelson and Grigorev [34], and Grigorev and Katsnelson [59], but modified to evaluate the variation at a single frequency over time. The well known Discrete Fourier Transform (DFT)

\[
X(m\Delta f) = \sum_{n=0}^{N-1} x(n\Delta t)e^{-j\frac{2\pi}{N}mn\Delta t} \quad m = 0, 1, 2, \ldots, N-1 \tag{4.1}
\]

is used to calculate the frequency spectrum of the acoustic pressure of CW pulses during the CTD tow time. In this equation \( x \) represents a discrete signal, \( X \) is the corresponding discrete frequency spectrum, \( N \) is the length of the transform, \( \Delta t \) is the sampling period (\( \Delta t = 1/f_s \)), \( m \) is the frequency bin number, and \( n \) is the discrete time signal index. The time signal is the averaged RMS pressure of CW pulses (referred to as simply acoustic pressure or signal henceforth) over time. The time series is stepped through to create a spectrogram showing evolving spectral content over the time of the CTD tow. This process will be referred to as harmonic variation analysis (HVA) in this work, and the output intensity (represented by a color scale) will be referred to as harmonic variation intensity (HVI). This process is represented mathematically by

\[
HVI_{p_{cw}} = \frac{2}{T}|DFT\{p_{cw}\}|^2 \tag{4.2}
\]

\[
HVI_{p_{cw}}(m\Delta f) = \frac{2}{T}|X(m\Delta f)|^2, \tag{4.3}
\]

where \( T \) is equal to the temporal length of the transform (\( T = (N-1)\Delta t \)). HVI is expressed in dBs as

\[
HVI_{dB} = 10\log_{10}\left(\frac{HVI_{p_{cw}}}{HVI_{ref}}\right), \tag{4.4}
\]
where $HVI_{ref} = 1 \text{Pa}^2/\text{Hz}$. All HVA in this document uses a 40 s sampling rate, 256 element transform window, and 80% overlapping with no windowing. Figures 5.21 and 5.22, which will be discussed in Chapter 5, show the HVI for the 36 hour CTD tow period (the zeroth hour on all HVI plots will correspond to the start of towed CTD measurements). It is expected that there should be increases in HVI within the frequency spectrum when internal waves are crossing the acoustic axes. The ellipses and other annotations that delineate internal wave crossings and HVI contribution will be discussed in Chapter 5, for now attention is turned to examining the mechanisms whereby the passage of internal waves modulate the received signal.

4.3 Ray Model

In order to understand the mechanics behind an internal wave's effect on the acoustic field, we follow the methods in [59] and [34] and use ray theory to model how an internal wave might influence acoustic propagation and approximate the fundamental frequency at which the received signal will fluctuate. Although ray theory is inherently a high-frequency acoustic approximation, and not necessarily appropriate for use at frequencies of 300 and 500 Hz, it will provide a valuable conceptual look at how the internal wave field can influence the ocean waveguide. It will also give a general idea of the frequency range necessary for spectral analysis of computational models and acoustic recordings.

While acoustic pressure recorded by a single receiver is the sum of many rays, it is assumed that only rays that travel through the thermocline region are affected by the perturbations caused by propagating internal waves. This is because internal wave amplitude is greatest within the thermocline [3], so rays that spend more time traveling through the thermocline region (i.e. have downward turning points near the upper thermocline) will be more affected by the internal wave and play a dominant role in causing acoustic fluctuations [59]. Continuing the terminology of [59, 34], the ray that reaches the uppermost part of the thermocline and then downwardly refracts is referred to as the critical ray $R_c$ [59, 34]. It is also important to note that although the critical ray may reflect from the ocean floor many times, and thus is subject to bottom losses, it does not reflect from the sea
surface where scattering would further reduce the received signal. Figure 4.1 is a drawing depicting the critical ray and another ray that turns below the thermocline. This figure also shows the iso-sound speed lines from an internal wave soliton measurement taken during TAVEX (IW4). $R_c$ denotes the critical ray, which will dominantly affect the received pressure fluctuations. This figure displays how ray height and cycle distance can be affected when $R_c$ encounters the wave depression. $R_1$ denotes a ray that does not encounter the thermocline, and thus is not affected by the wave depression. It is important to note that although a single soliton is shown here for simplicity, the following methods are equally appropriate for internal wave packets. It is suspected that multi-soliton wave packets may have differing intensities of harmonics compared to that of a single soliton model with similar wave characteristics, but the fundamental frequency should remain roughly equal.

![Figure 4.1](image)

**Figure 4.1.** Drawing (not a ray trace) showing how an internal wave can affect the critical ray $R_c$ while having little impact on other rays that do not encounter the thermocline. $D_{R_c}$ is the critical ray cycle distance.

Since the $R_c$ will only encounter the internal wave when the wave propagates into a region where the critical ray is in the thermocline, the fundamental frequency $f_0$ of the acoustic intensity fluctuations can be found using the travel speed of the
soliton and the critical ray cycle distance $D_{rc}$ [59, 34]. Thus, the fundamental frequency is the ratio of the projection of $c_{iw}$ onto the acoustic path $\frac{c_{iw}}{\cos \phi}$ to $D_{rc}$

$$f_0 = \frac{c_{iw}}{D_{rc} \cos \phi},$$

(4.5)

where $\phi$ is the angle of the internal wave direction to the acoustic track as defined in Chapter 3 (see Figure 3.20). Equation 4.5 can also be used to connect the expected $f_0$ to the internal wave frequency $f_{iw}$ using the relation $c_{iw} = f_{iw} \lambda_{iw}$. This shows that $f_0$ is simply $f_{iw}$ scaled by the ratio of the apparent acoustic wavelength of the internal wave to the critical ray cycle distance

$$f_0 = f_{iw} \frac{\lambda_{iw}}{D_{rc} \cos \phi}.$$

(4.6)

Equations 4.5 and 4.6 display how internal wave propagation direction can affect the acoustic field for internal wave directions of roughly $\phi < 45^\circ$ [23, 75]. As discussed in Badiey et al. [23], it can be assumed that acoustic mode coupling, induced by propagating internal wave perturbations, is the dominant mechanism of acoustic fluctuation within this angular region of $\phi$. Another method of looking at the acoustic effects in this angular region of $\phi$ is a specific ray (a single acoustic mode) that is most affected by the structure of the internal wave. The angle of this ray (or mode) is transferred to differing angles (modes) by the wave perturbations. This means that mode coupling is another way of looking at the critical ray model. This mechanism of internal wave induced acoustic variation will be referred to as vertical coupling in this work.

At angles above $45^\circ$ other mechanisms become key in the acoustic field influence, including an adiabatic region where the acoustic field is affected little by the wave field, and a horizontal refraction region where acoustic energy begins to travel along the troughs of a wave packet or soliton [23]. In this work, only circumstances where internal wave directions are well within this vertical coupling region ($\ll 45^\circ$) are considered so the influences of horizontal refraction can be neglected [23, 59]. The purpose for this is that the 2D computational model discussed in the following section is capable of properly modeling the acoustic field within this angular region of $\phi$. However, it is not able to properly the model
horizontal refraction (no horizontal acoustic coupling) caused by internal waves at higher angles. Two of the four waves (IW1 and IW5&IW6) measured during the CTD tow (see Table 3.4) fall in to this $\phi << 45^\circ$ region, and allow acoustic data analysis and model comparisons using this 2D computational model.

In order to find $f_0$, $D_{Re}$ must be found. To do this analytically, Snell’s Law is used [78]

$$\frac{\cos \theta_1}{c_1} = \frac{\cos \theta_2}{c_2} = a,$$  \hspace{1cm} (4.7)

as well as a linear approximation of the average sound speed profile for the experimental region (Figure 4.2). In this equation, $\theta_1$ and $\theta_2$ are the initial and final ray angles, $c_1$ and $c_2$ are the initial and final sound speed of the ray’s placement, and $a$ is the Snell’s Law constant. Next the launch angle of the critical ray that will reach the upper most point of the thermocline but still downwardly refract toward the seafloor without entering the mixed layer is determined. The linear approximation of the water column assumes that the isothermal layer sound speed $c_{iso}$, and the mixed layer sound speed $c_{mix}$ does not change over depth. These values were found by averaging the profile over these regions of the water column, and are shown in Figure 4.2. This means that the critical ray angle $\theta_{Rc}$ will remain constant throughout isothermal layer, but upon entering the thermocline will downwardly refract until $\theta_{Rc} = \theta_{Rc2} = 0^\circ$ upon reaching the top of the thermocline and the edge of the mixed layer. Using Snell’s Law, and $\theta_{Rc1}$ and $\theta_{Rc2}$ to denote the angle of $R_c$ at the seafloor and thermocline/mixed layer interface respectively as shown in Figure 4.3

$$\cos \theta_{Rc1} = \frac{c_{iso}}{c_{mix}} \cos \theta_{Rc2}; \quad \theta_{Rc2} = 0^\circ$$ \hspace{1cm} (4.8)

$$\theta_{Rc1} = \arccos \left( \frac{c_{iso}}{c_{mix}} \right).$$ \hspace{1cm} (4.9)

Using values of $c_{mix} = 1533$ and $c_{iso} = 1499.5$ m/s, as shown in Figure 4.2, results in an critical ray at the seafloor (and entering the thermocline) of $\theta_{Rc1} = 12^\circ$. Knowing this angle, simple geometry is used to find the range traveled by the ray from the seafloor to the bottom of the thermocline as well as the range traveled by the ray reentering the isothermal layer back to the seafloor (equal distances of $d_1$). Using the thickness of the isothermal layer $h_{iso}$, the total distance of this range $2d_1$
Figure 4.2. Average water column sound speed profile for all towed CTD data with linear approximation used for analytic ray calculations.

will be

\[
2d_1 = \frac{2h_{iso}}{\tan \theta_{Rc1}}.
\]  

(4.10)

The distance in range that \( R_c \) travels while in the thermocline can also be broken into two equal sections, each of length \( d_2 \). These sections are the range from where \( R_c \) enters the thermocline to the upper turning point where \( \theta_{Rc} = \theta_{Rc2} = 0^\circ \), and from the upper turning point back to the thermocline/isothermal layer interface where \( \theta_{Rc} = \theta_{Rc1} = 12^\circ \). To determine the total distance that \( R_c \) travels in the thermocline \( 2d_2 \), the integral method in *Acoustical Oceanography* by Clay and Medwin [6] is used. Starting with Clay and Medwin’s Equation 3.2.8

\[
2d_2 = 2 \int_{z_i}^{z_f} \frac{ac(z)}{\sqrt{1 - a^2c^2(z)}} \, dz
\]  

(4.11)

where \( a \) is equal to the Snell’s Law constant found in Equation 4.7, \( z_i \) is equal to the initial depth, and \( z_f \) the final depth. Changing variables to simplify the math, and using the gradient \( g \) of the sound speed with respect to depth of the linearly
fit thermocline in Figure 4.2

$$\frac{dc}{dz} = g ; \; \frac{dz}{g}, \; \frac{dc}{dz} = g,$$

(4.12)

the integral over the initial and final sound speeds $c_{iso}$ and $c_{mix}$ is

$$2d_2 = 2 \int_{c_{iso}}^{c_{mix}} \frac{ac(z)}{\sqrt{1 - a^2 c^2(z)}} \frac{dc}{g} = 2 \frac{a}{g} \int_{c_{iso}}^{c_{mix}} \frac{c(z)}{\sqrt{1 - a^2 c^2(z)}} \; dc. \; (4.13)$$

Then using the integral relation

$$\int \frac{x}{\sqrt{b + cx^2}} \; dx = \frac{\sqrt{b + cx^2}}{e} \; (4.14)$$

from Appendix A (Equation A1.5.3) of Clay and Medwin [6] results in

$$2d_2 = 2 \frac{a}{g} \int_{c_{iso}}^{c_{mix}} \frac{c(z)}{\sqrt{1 - a^2 c^2(z)}} \; dc = 2 \frac{1}{ag} \left( \frac{\sqrt{1 - a^2 c^2(z)}}{c_{iso}} \right). \; (4.15)$$

Again using Snell’s Law to simplify

$$2d_2 = 2 \frac{1}{ag} (\sin \theta_{Rc1} - \sin \theta_{Rc2}) ; \; \theta_{Rc2} = 0^\circ \; (4.16)$$

$$2d_2 = 2 \frac{1}{ag} \sin \theta_{Rc1}. \; (4.17)$$

Combining Equations 4.10 and 4.17, the total $D_{Rc}$ for the critical ray using $h_{iso} = 40$ m, $g = 1.19$ 1/s, and an initial $\theta_{Rc1} = 12^\circ$ is

$$D_{Rc} = 2d_1 + 2d_2 = 2 \frac{h_{iso}}{\tan \theta_{Rc1}} + 2 \frac{1}{ag} \sin \theta_{Rc1}, \; (4.18)$$

$$D_{Rc} = 2 \left( \frac{h_{iso}}{\tan \theta_{Rc1}} + \frac{c_{iso}}{g} \tan \theta_{Rc1} \right) = 912$$m. \; (4.19)

Using Equation 4.5 and an average wave speed of $c_{iw} = 0.93$ m/s, yields $f_0 = 1.019$ mHz. This is within the range that Moum and Nash [32] refer to as the nonlinear internal wave band (0.3-50 mHz). A schematic representation of each of these model parameters is shown in Figure 4.3.

In order to check the accuracy of this linear approximation, a range-independent
computational ray model is used. The Acoustic Toolbox User interface and Post Processor (AcTUP) [79] software package from the Centre for Marine Science and Technology (CMST) was used to run Michael Porter’s BELLHOP [80] ray trace code with the average CTD water column profile shown in Figure 4.2 (not the linear approximation). A set of rays between 7 and 17° were plotted with 0.1° resolution and the upper most turning ray was found to be equal to 12.0°, just as the linear approximation. For angular resolution of 0.1°, the linearized sound speed model appears quite accurate. During simulation it was noted that this ray did not fall near the receiver at 20 km range, but a ray with a launch angle of 11.8° did fall directly on the receiver so is used for further calculations. The output of Bellhop for these two launch angles is shown in Figure 4.4. \( D_{Rc} \) is then calculated from the average distance between each critical ray interaction with the seafloor from the source to the receiver for the 500 Hz scenario shown in this figure. Using BELLHOP, the critical ray angle is \( \theta_{Rc} = 11.8° \) and cycle distance is \( D_{Rc} = 913 \) m, both close to the analytic model. Using Equation 4.5 and an average wave speed \( c_{iw} = 0.93 \) m/s, a fundamental frequency of 1.018 mHz is calculated. From these models, it is expected that the fundamental frequency of acoustic pressure variations caused by internal waves propagating roughly parallel
to acoustic propagation paths should be near 1 mHz. Also, since these variations will be nonsinusoidal, harmonics of this frequency are expected. Also interesting to note from BELLHOP modeling is that the arrival times for rays at 0 and 12° (0° ray does not touch sea surface or seafloor) are 13.36 and 13.39 s respectively. This gives a travel time difference for these two rays of 30 mS. Interference between separate arrivals is accounted for by averaging over the center 1 s of the CW pulse recordings (as discussed in Chapter 2).

Although one single soliton was measured during the CTD tow (IW4), the other three wave measurements are multi-soliton wave packets (and neither IW1 or IW5&IW6, which have propagation direction angles with the acoustic axis \(<\sim 45°\), are single solitons). Multiple solitons could affect the accuracy of this simple model depending on the distance between the solitons and the critical ray cycle distance. However, assuming the solitons within a wave packet are traveling at similar speeds, the fundamental frequency developed here should remain relatively unchanged (due to \(f_0\)’s strong dependence upon \(c_{iw}\) and \(D_{Rc}\)). It is expected that multiple solitons will affect the intensity of specific harmonics of HVA however. To examine this and create a more realistic model of the TAVEX scenario, the next section discusses a more complete model that accounts for multiple solitons and is valid at low frequencies. This acoustic model will display the expected harmonic nature of multi-soliton wave packets measured during TAVEX CTD tows.

4.4 Parabolic Equation Model

The 2D PE model discussed in Chapter 2 and Appendix B is used in Chapter 5 to further examine the harmonic nature of internal wave effects on acoustic data, and compare the HVA of acoustic recordings to the HVA of PE models perturbed by internal wave models. A PE model is used to simulate the acoustic effects of propagating internal waves using wave models with similar characteristics to the waves measured during the CTD tow. With this PE model, the frequency analysis of the previous section is made more complete, and the harmonics of the acoustic pressure variation are approximated along with the fundamental frequency. The PE model used in this work is Michael Collin’s Range-dependent Acoustic Model (RAM) with a self-starting acoustic field [74], a FORTRAN code available by
anonymous ftp from the Naval Research Laboratory’s Acoustics Division website [81]. MATLAB was used to configure all environment files and parameters, and the compiled FORTRAN code was then run as an executable file. GNU FORTRAN 95/2003 (GFortran) compiler along with the GNU Emacs text editor was used to edit and compile the RAM source code (ram1.5.f from the NRL website). The only modifications to this code were to change it slightly to output complex pressure rather than only TL. The modified FORTRAN code is found in Appendix B along with the MATLAB code used to run computational PE models.

Computational environments are constructed for IW1 and IW5&IW6 for both 300 and 500 Hz propagation tracks. These waves are used because of their small propagation direction angle $\phi$ ($\sim 8.5^\circ$) from the acoustic axes (as discussed in the previous section), so they will be most closely represented by a 2D PE model. This is because horizontal refraction can be neglected for these waves [23]. This should provide a reasonable representation of the evolving acoustic field for each of these waves since most of the acoustic effects should be caused by vertical coupling rather than horizontal refraction which would required a three-dimensional (3D) model [23, 75]. The general time window where the effects of these internal waves upon the acoustic field is also known with reasonable certainly without any calculated propagation direction knowledge. This is because these waves were measured close

Figure 4.4. Diagram of critical ray ($\theta_{Rc} = 11.8^\circ$) most influenced by soliton passing for 500 Hz source (Left). Average water column sound speed profile used for computations (right).
to the acoustic axes and towards the center of the tow area.

Each wave environment is modeled using a range-independent average sound speed profile from CTD measurements, taken from near the internal wave event over a linearly approximated bathymetry (Table 2.3) of the propagation path. Sediment properties (Table 2.1) for both 300 and 500 Hz models are also included. The average sound speed profile is then perturbed using the analytic KdV model discussed in Chapter 3, and this wave model is propagated in range from prior to the source until after the wave has passed over the receiver using $c_{iw}$ from Table 3.2 to advance range steps. HVA is performed on the pressure output of these models to find $f_0$ and harmonics in the next chapter. To check the dependence of $f_0$ on internal wave propagation direction shown in Equation 4.5, the HVA of two wave models differing only in angle $\phi$ is first discussed. A comparison of an analytic wave field to a wave field using measured data is also made to show the validity of using analytic KdV wave models.
Chapter 5

Acoustic Analysis

5.1 Chapter 5 Overview

This chapter uses the acoustic analysis and models discussed in Chapter 4 to compare modeled and recorded acoustic data. The first comparison is of two PE models, both perturbed by an analytic internal wave field with the characteristics of IW5&IW6. The IW5&IW6 perturbations differ only in simulated propagation angles in order to check the fundamental frequency calculation developed using ray theory in Chapter 4. The second comparison is also of two PE models, one perturbed by an analytic IW5&IW6 wave field and one by measurements of the sound speed field of IW6 recorded during the TAVEX CTD tow. This comparison is used to validate the use of analytic KdV internal wave fields, and provide a quantitative look at the impact of small scale internal wave irregularities (finestructure) on acoustic data. The final comparison is of PE models perturbed by the KdV wave fields of IW1 and IW5&IW6 to recorded acoustic data during the CTD tow. This comparison is used to search acoustic data recorded during TAVEX for internal wave signatures using harmonic variation analysis (HVA). At the closing of this chapter, acoustic SNR and ship traffic near the HLA is discussed as it pertains to the impact of ship traffic affecting these measurements and analysis techniques.
5.2 Acoustic Analysis of KdV Wave Field at Multiple Propagation Angles

In Chapter 4, the fundamental frequency of acoustic variation caused by a passing internal wave was calculated using ray theory. In this section, the HVA of the pressure output of PE models is used to determine the harmonic content of the acoustic field caused by an internal wave crossing the acoustic path. A comparison between two wave fields at different propagation angles is made to validate the relationship defined in Equation 4.5. Figures 5.1 and 5.2 show a single range step of the IW5&IW6 analytic internal wave field with simulated propagation angles $\phi$ of 0 and 45$^\circ$ respectively. Parameters for the IW5&IW6 wave at 0$^\circ$ are found by averaging the parameters from Table 3.2 for IW5 and IW6 since they are the same internal wave measured at different times and ship courses. The parameters used for the 0$^\circ$ field are $\lambda_{iw} = 490$ m, $c_{iw} = 0.94$ m/s, and $2\Delta_{iw} = 265$ m. The parameters for the IW5&IW6 wave at 45$^\circ$ are based on the 0$^\circ$ parameters and the relations found in Figure 3.20. The parameters of this wave field are $\lambda_{iw} = 693$ m, $c_{iw} = 1.33$ m/s, and $2\Delta_{iw} = 375$ m.

Figures 5.3 and 5.4 show the TL field (300 Hz source) of the environment found in Figures 5.1 and 5.2 respectively. Although the internal wave fields pictured in Figures 5.1 and 5.2 seem to differ greatly to the eye, the TL fields of these two sound speed structures don’t exhibit a corresponding difference. These figures represent only a single range step of the iterative 2D PE model used to simulate fluctuations in acoustic pressure over time due to internal waves. The output of the iterative model results in hundreds of evolving fields (TL or pressure), and the variation at the range and depth of the HLA is found by taking the values from this point from all these fields. The received level (RL) at the range and depth of the HLA for each IW5&IW6 field (0$^\circ$ and 45$^\circ$) can be found in Figure 5.5. RL is calculated using the TL output of the RAM model and the source levels discussed at the end of Chapter 2. The duration of each of these events is of course effected by the speed of the internal wave over the acoustic path, this is the cause of differing event durations in Figure 5.5.

Each time series in Figure 5.5 seems to have similar structure and level. Note also that the RL of these models has similar level and variation structure compared
Figure 5.1. Average sound speed field of IW5&IW6 perturbed by analytic KdV model over 300 Hz propagation path. Propagation angle assumed to be 0° with acoustic axis. Color scale in m/s.

Figure 5.2. Average sound speed field of IW5&IW6 perturbed by analytic KdV model over 300 Hz propagation path. Propagation angle assumed to be 45° with acoustic axis. Color scale in m/s.
Figure 5.3. TL for the single range step of the IW5&IW6 sound speed field shown in Figure 5.1. Color scale in dB.

Figure 5.4. TL for the single range step of the IW5&IW6 sound speed field shown in Figure 5.2. Color scale in dB.
Figure 5.5. Received level at the range and depth of the HLA for IW5&IW6 KdV wave fields with propagation angles of 0 and 45°.

to the hydrophone recordings in Figure 2.16. HVA of both PE models’ acoustic pressure output gives Figures 5.6 and 5.7. These harmonic variation intensity (HVI) plots show the relative changes in intensity of the variation shown in Figure 5.5 with respect to frequency (millihertz) and time (based on the 36 hours of the CTD tow). Constant pressure data has been placed before and after the cross-times of each internal wave (dark blue shaded area) in all HVI plots to keep the time axis in all plots equal and referenced to CTD tow times. Using Equation 4.5, we find the approximate fundamental frequencies of these two models as predicted by the ray theory in Chapter 4

\[
f_{0_0} = \frac{0.94 \text{ m/s}}{913 \text{ m} \cos (0^\circ)} = \frac{0.94 \text{ m/s}}{913 \text{ m}} \approx 1 \text{ mHz}
\]

\[
f_{0_{45^\circ}} = \frac{0.94 \text{ m/s}}{913 \text{ m} \cos (45^\circ)} = \frac{1.33 \text{ m/s}}{913 \text{ m}} \approx 1.5 \text{ mHz}.
\]

\(f_{0_{0^\circ}}\) and \(f_{0_{45^\circ}}\) are denoted in Figures 5.6 and 5.7 by bold, dashed, black lines. Notice that each of these lines intersects the lowest spectral region of increased HVI. This implies that the conceptual ray model presented in Chapter 4 is correct,
at least for passing waves whose propagation angle is roughly $< 45^\circ$ (as discussed in Chapter 4). It is interesting to note in both Figures 5.6 and 5.7 that there is much less energy in the harmonics than $f_0$.

5.3 Acoustic Analysis of KdV and CTD Measurement Wave Fields

In this section, the HVA of PE models is employed to determine the differences between the frequency spectrum of PE models using the analytic KdV wave model to perturbed sound speed field, and PE models using measured CTD wave recordings to perturb the sound speed field. This comparison will determine both the importance of internal wave finestructure in HVA, and provide a check of the validity of the analytic wave field’s capability to model the structure of internal wave perturbations in the sound speed field. In this work, finestructure refers to small deviations in the typical wave field due to unideal real-world circumstances. This term is used to describe both the small irregularities in internal wave events, and the lower-intensity higher-frequency variations caused by these irregularities in HVA. By looking at the differences in HVI of these two models, it can be determined if a real internal wave field, a less smooth function than the KdV solution, will create the same basic spectral content as is seen using analytic solutions.

The sound speed field over the range from the 300 Hz source to receiver for the two wave models is shown in Figures 5.8 and 5.9. The wave field shown in Figure 5.8 is identical to Figure 5.1, with wave parameters equal to those of the $0^\circ$ wave field of the last section. It has been placed here also to provide easy reference between the shapes of the KdV wave field and the data wave field in Figure 5.9. The data measurement internal wave field was created using the CTD measurements of IW6 because the entire internal wave (IW5&IW6) was captured during these measurements (as opposed to IW5 measurements which were cut off). The data based internal wave field was created by replacing a section of the a range-independent sound speed field (average CTD tow profile from around the event measurement) with the actual measurements of the internal wave shown in Figure 3.9, but scaling Doppler shifted CTD measurements to the same $\lambda_{iw}$ as the
Figure 5.6. HVI of IW5&IW6 propagating PE model for 300 Hz analytic KdV model. Propagation angle assumed to be 0° with acoustic axis. Color scale in dB/Hz.

Figure 5.7. HVI of IW5&IW6 propagating PE model for 300 Hz CTD data model. Propagation angle assumed to be 45° with acoustic axis. Color scale in dB/Hz.
\(^0\) KdV wave field. This real wave field is then propagated in range also using the same \(c_{iw}\) as the \(^0\) KdV wave field.

The IW5&IW6 internal wave (IW5 and IW6 event measurements) was chosen for this comparison because of the two internal waves believed to be propagating roughly from source to receiver (IW1 and IW5&IW6), it is the only wave that was captured entirely during measurements (see Figures 3.4, 3.8, and 3.9). The RL of both models is shown in Figure 5.10 there are similar levels and variation structure in both models. These models also show similar levels and variation to some regions of the hydrophone recordings of TAVEX (see Figure 2.16).

The HVI of both analytic and data based acoustic models for IW5&IW6 is found in Figures 5.11 and 5.12 respectively. All HVI plots of PE calculations used in this work are scaled so the color scale is equal to the the scale of the HVI of acoustic data (shown later in Figures 5.21 and 5.22). Notice that Figures 5.11 and 5.12 display very similar frequency structure including a fundamental frequency near 1 mHz as predicted by the conceptual ray model in the previous chapter.

The three black ellipses encompass individual regions of increased variation in acoustic energy in these plots, not necessarily a perfectly harmonic (\(2f_0\), \(3f_0\), \(4f_0\),...) spectral structure. However, there does appear to be a harmonic nature to these analyses. These ellipses encompass the same regions of each plot to aid the readers eye in comparison. The spectral placement and distribution of these regions of increased HVI above the fundamental frequency are also very similar, as well as the temporal placement of large low-frequency variations occurring during the time when the internal wave is propagating over the path between source and receiver. This comparison shows that the analytic KdV wave field does simulate the major acoustic effects of the real internal wave field well, although HVI peaks are not quite as well defined in the data based calculations. However, there does appear to be scattered regions of increased HVI in the upper regions of the spectrum (circled in red) within the data based display that are much less noticeable in the analytic display. This difference is caused by the inclusion of internal wave finestructure in the measured internal wave rather than a smooth analytic function. These low-intensity high-frequency variations would be difficult to match within acoustic recordings, so the majority of the spectral structure from the real data field is described well with the analytic KdV wave model. These analytic wave models
Figure 5.8. Average sound speed field of IW5&IW6 perturbed by analytic KdV model over 300 Hz propagation path. Color scale in m/s.

Figure 5.9. Average sound speed field of IW5&IW6 perturbed by actual CTD measurements over 300 Hz propagation path. Color scale in m/s.
Figure 5.10. Received level at the range and depth of the HLA for both the analytic KdV and CTD measurement wave fields of IW5&IW6.

are also much easier to create and manipulate, so they will be used to compare to acoustic data in the next section.

5.4 Harmonic Variation Analysis of HLA Data and Acoustic Models

The HVI of four PE models perturbed by analytic internal wave fields is compared with the analysis of received acoustic pressure on the HLA. These include both 300 and 500 Hz and their prospective propagation ranges for IW1 and IW5&IW6. Single steps in range of the analytic internal wave fields for each wave are shown in Figures 5.13-5.16. These figures correspond to 300 Hz for IW1, 500 Hz for IW1, 300 Hz for IW5&IW6, and 500 Hz for IW5&IW6 respectively. The parameters of each of these wave fields can be found in Table 3.2, but the parameters of the IW5&IW6 model are the average values of IW5 and IW6 within this table (as mentioned in the previous two sections). Also, because of the slight change in angle from the acoustic axes to the internal wave directions for these waves, no corrections are
**Figure 5.11.** HVI of the IW5&IW6 based propagating PE model at 300 Hz for analytic KdV internal wave. Color scale in dB/Hz.

**Figure 5.12.** HVI of the IW5&IW6 based propagating PE model at 300 Hz for CTD data internal wave. Color scale in dB/Hz.
made to the internal wave model for propagation angle in computational models (all wave models are assume to be at a propagation angle of 0°).

Each simulated water column shows the sound speed field that acoustic energy must navigate through for a single range step of the iterative propagation model. Five individual solitons are included in the IW1 model simply as an estimation since there are three to four visible solitons in IW1 measurements (Figure 3.4) and it appears that these wave measurements were cut off. Three solitons are included in the IW5&IW6 model because of the three main depressions found in IW6 measurements (Figure 3.9). As previously mentioned, bathymetry and seafloor properties can be found in Tables 2.1 and 2.3.

HVI for each of the four internal wave fields can be found in Figures 5.17-5.20. As in the previous section, black ellipses have been placed around the general spectral structure within each of these figures to ease comparison by the reader. These ellipses are placed in the same spectral location in all HVI plots. HVI of the acoustic signal received at four hydrophones (#’s 33, 53, 75, and 96) is averaged to give Figures 5.21 and 5.22. The analysis of each individual hydrophone can be found in Appendix C for reader comparison. Only these four hydrophones were used in analysis in order to reduce computational processing demands, however, care was taken to choose spatially distributed hydrophones. These hydrophones were positioned across the operating section of the HLA (hydrophone #’s 33-96), spaced at roughly 42 m. Ellipses are placed in the same spectral locations as in the previously discussed models (Figures 5.17-5.20). These sets of ellipses are labeled A (IW1) and B (IW5&IW6) to represent the expected spectral structure (from the previously discussed acoustic models) of each internal wave crossing. The temporal locations of each internal wave cross-time is marked at the top of Figures 5.21 and 5.22.

Interesting to note is the difference in the relative intensity of the fundamental frequency and other regions of increased HVI between the IW1 and IW5&IW6 fields. Although these peaks in HVI are not necessarily perfect harmonics (2f₀, 3f₀, 4f₀,...) of the fundamental frequency, these regions of increased HVI above f₀ will be referred to as harmonics from this point forth to ease explanation. The HVI of each wave model has a peak at the fundamental frequency and the relative harmonics. However, while the IW5&IW6 display shows a maximum peak at the
Figure 5.13. Average sound speed field of IW1 perturbed by analytic KdV model over 300 Hz propagation path. Color scale in m/s.

Figure 5.14. Average sound speed field of IW1 perturbed by analytic KdV model over 500 Hz propagation path. Color scale in m/s.
Figure 5.15. Average sound speed field of IW5&IW6 perturbed by analytic KdV model over 300 Hz propagation path (same as Figures 5.1 and 5.8 but placed here for easy reference to other models). Color scale in m/s.

Figure 5.16. Average sound speed field of IW5&IW6 perturbed by analytic KdV model over 500 Hz propagation path. Color scale in m/s.
Figure 5.17. HVI of the IW1 based propagating PE model at 300 Hz for the analytic KdV model. Color scale in dB/Hz.

Figure 5.18. HVI of the IW1 based propagating PE model at 500 Hz for the analytic KdV model. Color scale in dB/Hz.
Figure 5.19. HVI of the IW5&IW6 based propagating PE model at 300 Hz for the analytic KdV model (same as Figure 5.11 but placed here to allow easy comparison with other models). Color scale in dB/Hz.

Figure 5.20. HVI of the IW5&IW6 based propagating PE model at 500 Hz for the analytic KdV model. Color scale in dB/Hz.
fundamental frequency and diminishing intensity levels at HVI peaks thereafter, the 5 soliton IW1 display does not show this characteristic. It seems that the IW1 display shows the second harmonic having far greater intensity than the fundamental frequency. This is an interesting anomaly of this model with two more solitons. Further computational modeling is needed to fully understand the cause of this difference in spectral structure.

In the region of IW1’s cross-time (ellipse set A) in Figures 5.21 and 5.22, HVI peaks are seen throughout the spectral region of IW1’s crossing, especially in the midspectrum region. An harmonic structure corresponding to the harmonic structure of the computational models in Figures 5.17 and 5.18 is also seen, albeit it is much less defined. Looking at the black ellipses denoting the expected harmonic structure of IW5&IW6’s cross-time (ellipse set B), it much more difficult to distinguish a defined structure within this region. There is certainly increased HVI near the first and second harmonic regions, but it is difficult to distinguish whether this is due to internal wave interaction or other low-frequency noise variation. The absence of distinct higher-order harmonics (as in the IW1 analysis) makes it difficult to determine whether this internal wave signature is present within this region. Also inhibiting the distinction of IW5&IW6’s acoustic influence is the strong vertical strip of HVI marked with dotted black lines and denoted by D. Another similar vertical strip is marked by C. The cause of these strips is unknown, however, it is not believed to be the effects of water column dynamics on the acoustic transmissions. This will be discussed further in the following section.

Also important to mention is the region encompassed by a red ellipse marked by E (Figures 5.17 and 5.18). None of the internal wave cross-times correspond to this region although this activity (compared with IW1’s signature) seems similar to an internal wave signature, although it is difficult to make out a specific harmonic structure. There are two possible explanations for this. First, assuming infinite wave fronts, any internal wave propagating through the CTD region should be measured by the towed CTD system at some point. However, if internal waves with non-infinite horizontal fronts are considered, it is possible that this activity is simply the edge effects of a wave that was missed during CTD measurements. The second possibility is one or more of the assumptions made during propagation direction analysis for IW2&IW3 or IW4 is incorrect causing incorrect calculations of
the cross-time for these waves. The calculated cross-times of IW2&IW3 and IW4 are marked at the bottom of Figures 5.21 and 5.22. There is no clear signatures of these waves during their cross-times, this implies the latter of these two possibilities. This region of increased HVI encompassed by E could be the result of one or both of these waves. However, since it is not possible to resolve specific internal wave arrivals from moored temperature measurements (discussed in Chapter 2), this cannot be concluded for certain.

5.5 Correlation of Nearby Ship Traffic with Acoustic Events

As mentioned in the previous section, strong vertical strips of HVI marked with dotted black lines are denoted by C and D in Figures 5.21 and 5.22. These strips are not believed to be the influence of internal waves or other hydrodynamics because of their very regular and intense nature. Although the cause of these distinct vertical strips is not known for certain, ships can produce large amounts of acoustic energy within the same frequency range as the acoustic sources. The process of a ship moving away from or towards the HLA could cause harmonic intensity variations that would be detected using HVA. For this reason, it is important to make certain that the HVA in this work is not simply measuring variations of 300 and 500 Hz acoustic energy caused by vessels in the experimental region. This can be done with the human ear by listening to recorded acoustic data to determine if a ship is in the vicinity of the HLA during each acoustic event within HVI plots. However, since it is known that ships produce broadband energy, this can also be done visually by plotting the power spectral density (PSD) of the full acoustic signal (not of the averaged RMS pressure of received CW pulses) in a spectrogram format and looking for strong broadband signals. By listening to recorded .wav files, it has been determined that ships are the source of short duration broadband energy throughout the spectrum in Figure 5.23. PSD parameters include 2.2 minute sections of acoustic data, sampling frequency of 3906.25 Hz, a Hanning window, and 50% overlap throughout the CTD tow time. This data was then downsampled in the frequency domain by 10X (effectively decreasing bin resolution by 10X) to
decrease the array dimensions to a manageable size for MATLAB plotting. This figure shows that there was significant ship traffic during the experimental period. If ship traffic shows consistently similar times and durations to regions of increased HVI in Figures 5.21 and 5.22, it is possible that this analysis is actually sensing the movement of ships and not the passing of internal waves.

Marked with F and G in Figure 5.23, the spectral region where horizontal bands of higher acoustic energy from both acoustic sources can be distinguished. The strong narrow band located at 300 and 500 Hz are of course caused by CW pulses, and the region of raised energy around these are caused by the LFM pulses. A few things are evident looking at this plot. First, regions marked by C and D in Figures 5.21 and 5.22 have similar high intensity broad band signals in the center of their temporal region. These bands are most likely the result of variation of some man-made source. Second, although ship traffic is rather heavy throughout the CTD tow time, ship traffic does not appear to correlate well with regions of increased HVI attributed to internal wave passings in the previous section. Although not conclusive, it does not appear that ship traffic is the cause of the harmonic structure seen in HVI of the IW1 cross-time and other temporal regions.

Although ship traffic does not appear to correlate with internal wave times in HVA, the level of ship noise can certainly affect the analysis by raising the noise floor of acoustic measurements. Figure 5.24 displays the averaged CW pulses $RL_{cw}$ (blue line), averaged regions between CW pulses $RL_{noise}$ (green line), and the signal-to-noise ratio (SNR) (red line) of hydrophone #96 within the same time window as the spectral analyses previously shown. SNR is calculated by

$$SNR_{dB} = 20 \log_{10} (p_{cw}) - 20 \log_{10} (p_{noise}),$$

(5.3)

where $p_{cw}$ is the averaged RMS pressure of CW pulses and $p_{noise}$ is the averaged RMS pressure of the noise floor between CW pulses. Averaged RMS noise pressure is found in a similar fashion to averaged RMS CW pressure, which was discussed in Chapter 2. A 1 s average is taken during the gap between LFM and CW arrivals just prior to each CW pulse (see Figure 2.15). This allows calculation of the temporal changes in SNR over CTD tow times.

Comparing Figures 5.23 and 5.24 it is apparent that passing ships often cause
SNR to drop very low for short periods of time. Notice that these dramatic drops in SNR are generally associated with a broadband occurrence in Figure 5.23. However, the majority of the testing period has a reasonably high SNR with a mean SNR of hydrophone #96 during this time of 21 dB. The SNR of the other three hydrophones used in this analysis (hydrophone #’s 33, 53, and 75) are similar to #96.

It does not appear that SNR is a concern during most the cross-time for IW1, with SNR generally being far above 6 dB except towards the final portion of the expected cross-time (~17 h). Repeated drops in SNR during the cross-time of IW5&IW6 could be part of what is obscuring the internal wave signature in Figures 5.21 and 5.22 once again however, most of this region is also well above 6 dB. However, the long drop in SNR near 25 h is interesting. This seems to correspond to the region marked with E in Figures 5.21 and 5.22 and a wide region of broadband energy in Figure 5.23. This implies that this region of increased HVI may not be useful for the identification of internal wave signatures due to an SNR repeatedly dropping near 0 dB.

5.6 Result of Analysis

The analysis in this chapter revealed several important acoustic behaviors. First, the relation between internal wave speed, critical ray cycle distance, and propagation direction (Equation 4.5) derived in Chapter 4 was verified computationally. Then a comparison of the HVI of two acoustic models, one with a real data internal wave field and one with an analytical KdV internal wave field, determined that the acoustic impact of the KdV internal wave model has much of the same spectral structure as the real data with the exception of high frequency finescale structure. It was also found that the HVI of recorded acoustic pressure from the HLA for the IW1 cross-time displays similar harmonic structure to IW1 acoustic models using analytically perturbed wave fields. This gives substantiality to using HVI signatures for the identification of crossing internal waves. However, HVI of the IW5&IW6 cross-time was not so fruitful. It is believed that this is due to an excess of noise sources during this temporal region of acoustic data, which is also demonstrated by the low SNR of acoustic data during this period.
Figure 5.21. Average HVI of received RMS acoustic pressure of 300 Hz averaged CW pulses (average of Figures C.1–C.4 in Appendix C). Color scale in dB/Hz.

Figure 5.22. Average HVI of received RMS acoustic pressure of 500 Hz averaged CW pulses (average of Figures C.5–C.8 in Appendix C). Color scale in dB/Hz.
Figure 5.23. Spectrogram of acoustic data looking for ship traffic. Color scale in dB/Hz re 1\mu Pa.

Figure 5.24. RL of CW pulses and noise floor from between CW pulses, as well as the SNR from during CTD measurement times.
Chapter 6

Conclusions and Suggestions for Future Work

6.1 Conclusions

During the TAVEX 2008, data collected includes 2D high-resolution towed CTD data, bathymetry from depth sounder data, acoustical properties of the seafloor, moored vertical thermistor array recordings, and acoustic recordings from a HLA. Frequencies transmitted to the HLA include 300 and 500 Hz over approximately 34 and 20 km respectively. Several internal waves were recorded within the CTD tow area during the experiment, with periods of 4-42 min and amplitudes of 3-6 m. Acoustic recordings during the experiment display harmonic variations in signal level similar in period to internal wave events, and RL variations on the order of 6-10 dB.

The research discussed in this thesis sought to examine the connection between measured variability in received acoustic signal from the HLA, to variability in the sound speed field due to internal wave perturbations measured during CTD tows. A conceptual analytic ray analysis of the average sound speed profile for the experimental region was used to find the expected fundamental frequency of acoustic pressure variations at the HLA. PE acoustic models, using both analytic KdV and real data measurement internal wave fields for internal wave perturbations were also developed. It was shown that analytic internal wave fields do simulate the major acoustic fluctuations due to wave passings well, while higher-frequency
fluctuations caused by wave event finestructure was not reproduced. A spectral analysis was then used to compare acoustic data to computational model output. This spectral method was labeled harmonic variation analysis (HVA), and an analytic ray approximation for fundamental frequency proved to match closely with both the HVA of PE models and acoustic data results. Comparing the HVA of modeled acoustic pressure to recorded acoustic pressure of HLA hydrophones does show similar characteristics in spectral signature for IW1, however, for IW5&IW6 it is more difficult to determine if increased HVI within the lower frequency regions of the plot are due to IW5&IW6 or other sources. It is believed that a combination of local ship activity and other sources of harmonic variation in this region cloud the signature of this internal wave. It is also believed that another temporal region of increased HVI in the HVA analysis (region E), which has no corresponding internal wave even measurement, could be the result of IW2&IW3 and/or IW4 because of incorrect assumptions in the propagation direction analysis of these two wave events. However, a rather low SNR within this region makes it difficult to prove this.

Although the results of this experimental/computational exercise are not conclusive, there is strong evidence that HVA is of practical importance in identifying the signatures of propagating internal waves. Identifying internal wave signatures with a reasonable degree of certainty would allow large-region acoustic monitoring of these gravitational waves for purposes such as adaptive shallow-water acoustic communications schemes. This work helps to further the present understanding of the effect of internal waves propagating through an acoustically probed shallow-water region, and presents a very practical measurement method for the detection of propagating internal waves using the acoustic pressure of single receivers.

6.2 Future Work

There are many avenues one could pursue to advance the goals of this work including further experimental work within the same region. Moored temperature recordings of the entire vertical water column taking place during a towed CTD data collection would provide well defined internal wave cross-times and propagation directions, while still allowing reconstruction of the finestructure of internal
wave events using towed CTD data. This would greatly decrease ambiguity in the analysis. Repeated towing over the acoustic axis with the CTD system would also provide more accurate data for acoustic model comparisons. Also, including the use of a broadband source would allow comparison of this acoustic analysis across many frequencies and determine the limits of its practicality in frequency bands. Inclusion of a vertical line array (VLA) in the experimental region would allow comparison of this analysis to different regions in the vertical water column, and allow a more complete comparison of models.

There are also many avenues that could be taken to exploit the analysis of the TAVEX 2008 data set. The most obvious subsequent steps in understanding the current data set follow. First, further 2D computational work involving more iterative models with differing numbers of solitons and amplitudes in the internal wave field would be an extremely useful addition to the present work to determine the specific effects of soliton number, spacing, and amplitude upon received acoustic pressure. Second, a 3D PE model that includes horizontal coupling of acoustic energy would greatly increase the understanding of variations in acoustic pressure caused by internal waves IW2&IW3 and IW4 since these waves have a propagation direction angle with the acoustic axis greater than or near 45°. Finally, a time-of-arrival (TOA) analysis looking for variations in CW and LFM travel times attributed to the passage of internal waves would also be a valuable addition to this work.
Appendix A

Towed CTD Data Collection and Processing Scripts

A.1 Calibration and Configuration Scripts

CTDCHAIN.EXE is the DOS based executable used to control the CTD deck box and store data. CTDSENS.EXE, also a DOS executable, is used to set CTD fin parameters such as address and measurement types (conductivity, temperature, and pressure). Two ASCII files are also need in order to use the towed CTD system. These files are a calibration (.CAL) and a configuration (.CFG) file, and both must be in the same directory as CTDCHAIN.EXE (this is also the directory that the .DTA data files will be stored). The calibration file contains all of the necessary coefficients to convert each sensor’s analog-to-digital integer values to physical units. The configuration file tells CTDCHAIN.EXE each fin’s physical location on the tow cable, and gives it a color to represent each fin’s measurements in a real-time updating display. Calibration and configuration files used during the collection of CTD data discussed in this thesis are found in the next two sections.

A.2 Calibration Script (FULCHAIN.CAL)

* Edited by Chad 07/21/2008
* This is the full (book values) calibration table for all 50 sensors
* Addr = Sensor address (number between 1 and 255)
* Type = 1 : first value temperature, second value pressure
* 2 : first value temperature, second value conductivity
* 3 : first value temperature, second value pressure, third
* value conductivity
* Raw data x (16bit integer) are converted to physical units
* by y = a2x+a1x+a0,
* units are deg C, dBar, and mmho/cm.
* In comparison to manual (y=a+bx) b=a1 and a=a0.
*
*!!!!! Be CERTAIN you space-bar out the pressure coefficient if using
* "Type 2" to measure only the temp. and cond.
* *
* *
* Temperature >>
* ------------->>
*Addr Type a2 a1 a0 >>
<table>
<thead>
<tr>
<th>Addr</th>
<th>Type</th>
<th>a2</th>
<th>a1</th>
<th>a0</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>3</td>
<td>0.0</td>
<td>6.388121e-04</td>
<td>-5.237200e+00</td>
</tr>
<tr>
<td>002</td>
<td>3</td>
<td>0.0</td>
<td>6.450390e-04</td>
<td>-5.471214e+00</td>
</tr>
<tr>
<td>003</td>
<td>3</td>
<td>0.0</td>
<td>6.403929e-04</td>
<td>-5.594816e+00</td>
</tr>
<tr>
<td>004</td>
<td>3</td>
<td>0.0</td>
<td>6.425199e-04</td>
<td>-5.705105e+00</td>
</tr>
<tr>
<td>005</td>
<td>3</td>
<td>0.0</td>
<td>6.447883e-04</td>
<td>-5.693930e+00</td>
</tr>
<tr>
<td>006</td>
<td>3</td>
<td>0.0</td>
<td>6.414971e-04</td>
<td>-5.615633e+00</td>
</tr>
<tr>
<td>007</td>
<td>3</td>
<td>0.0</td>
<td>6.424688e-04</td>
<td>-5.498062e+00</td>
</tr>
<tr>
<td>008</td>
<td>3</td>
<td>0.0</td>
<td>6.407448e-04</td>
<td>-5.512478e+00</td>
</tr>
<tr>
<td>009</td>
<td>3</td>
<td>0.0</td>
<td>6.445345e-04</td>
<td>-5.396430e+00</td>
</tr>
<tr>
<td>010</td>
<td>3</td>
<td>0.0</td>
<td>6.406290e-04</td>
<td>-5.585835e+00</td>
</tr>
<tr>
<td>011</td>
<td>3</td>
<td>0.0</td>
<td>6.408446e-04</td>
<td>-5.687895e+00</td>
</tr>
<tr>
<td>012</td>
<td>3</td>
<td>0.0</td>
<td>6.433864e-04</td>
<td>-5.561036e+00</td>
</tr>
<tr>
<td>013</td>
<td>3</td>
<td>0.0</td>
<td>6.446212e-04</td>
<td>-5.521620e+00</td>
</tr>
<tr>
<td>014</td>
<td>3</td>
<td>0.0</td>
<td>6.414370e-04</td>
<td>-5.269178e+00</td>
</tr>
<tr>
<td>015</td>
<td>3</td>
<td>0.0</td>
<td>6.416992e-04</td>
<td>-5.407381e+00</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>016</td>
<td>3 0.0 6.477491e-04 -5.729769e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>017</td>
<td>3 0.0 6.488762e-04 -5.760862e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>018</td>
<td>3 0.0 6.479632e-04 -5.770536e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>019</td>
<td>3 0.0 6.494633e-04 -5.537518e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>020</td>
<td>3 0.0 6.451223e-04 -5.284169e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>021</td>
<td>3 0.0 6.015021e-04 -3.868173e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>022</td>
<td>3 0.0 6.427014e-04 -5.375502e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>023</td>
<td>3 0.0 6.437826e-04 -5.568597e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>024</td>
<td>3 0.0 6.497394e-04 -5.491178e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>025</td>
<td>3 0.0 6.042125e-04 -4.178414e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>026</td>
<td>3 0.0 6.468522e-04 -5.594946e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>027</td>
<td>3 0.0 6.494130e-04 -5.610301e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>028</td>
<td>3 0.0 6.466849e-04 -5.541311e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>029</td>
<td>3 0.0 6.436724e-04 -5.039599e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>030</td>
<td>3 0.0 6.441535e-04 -5.433225e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>031</td>
<td>3 0.0 6.453020e-04 -5.514588e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>032</td>
<td>3 0.0 6.925157e-04 -6.776834e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>033</td>
<td>3 0.0 6.445239e-04 -5.304481e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>034</td>
<td>3 0.0 5.991895e-04 -3.853579e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>035</td>
<td>3 0.0 6.450206e-04 -5.475468e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>036</td>
<td>3 0.0 6.460496e-04 -5.365209e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>037</td>
<td>3 0.0 6.434195e-04 -5.376777e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>038</td>
<td>3 0.0 6.459896e-04 -5.501865e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>039</td>
<td>3 0.0 6.486302e-04 -5.514019e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>040</td>
<td>3 0.0 6.445857e-04 -5.371177e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>041</td>
<td>3 0.0 6.513654e-04 -5.569640e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>042</td>
<td>3 0.0 6.459602e-04 -5.095233e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>043</td>
<td>3 0.0 6.480762e-04 -5.539404e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>044</td>
<td>3 0.0 6.399726e-04 -5.369192e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>045</td>
<td>3 0.0 6.432253e-04 -5.350685e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>046</td>
<td>3 0.0 6.454588e-04 -5.426552e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>047</td>
<td>3 0.0 6.442167e-04 -5.435675e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>048</td>
<td>3 0.0 6.428874e-04 -5.208488e+00 &gt;&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>049</td>
<td>3 0.0 6.448178e-04 -5.548393e+00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>--------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>050</td>
<td>3 0.0 6.428522e-04 -5.397288e+00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pressure
-------

a2 a1 a0

<table>
<thead>
<tr>
<th>0.0 1.907395e-02 -627.3467e+00</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0 1.907395e-02 -626.4694e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -626.2023e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -627.1560e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -625.7636e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -626.792e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -626.3740e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -625.5920e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -628.1479e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -626.0879e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -624.9053e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -624.9887e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -624.1415e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -626.8231e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -624.6835e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -624.9887e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -624.1415e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -626.1921e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -625.2384e+00</td>
</tr>
<tr>
<td>0.0 1.908106e-02 -624.3640e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -624.6662e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -624.4775e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -625.4291e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -626.6662e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -626.3829e+00</td>
</tr>
<tr>
<td>0.0 1.907395e-02 -626.6662e+00</td>
</tr>
</tbody>
</table>
Conductivity

\[
\begin{array}{ccc}
0.0 & 1.90730e-02 & -625.6199e+00 \\
0.0 & 1.90730e-02 & -625.8106e+00 \\
0.0 & 1.90730e-02 & -624.0940e+00 \\
0.0 & 1.90730e-02 & -626.0014e+00 \\
0.0 & 1.90730e-02 & -625.4291e+00 \\
0.0 & 1.90730e-02 & -625.7245e+00 \\
0.0 & 1.90730e-02 & -626.7643e+00 \\
0.0 & 1.90730e-02 & -625.4291e+00 \\
0.0 & 1.90730e-02 & -625.7245e+00 \\
0.0 & 1.90730e-02 & -626.7643e+00 \\
\end{array}
\]

\[
\begin{array}{ccc}
0.0 & 1.11523e-03 & -0.6771761e+00 \\
0.0 & 1.094649e-03 & -0.7133874e+00 \\
0.0 & 1.087251e-03 & -0.6613978e+00 \\
0.0 & 1.079726e-03 & -0.9245383e+00 \\
0.0 & 1.102316e-03 & -0.8600624e+00 \\
0.0 & 1.126033e-03 & -0.8005460e+00 \\
\end{array}
\]
0.0 1.056731e-03 -0.7160572e+00
0.0 1.092919e-03 -0.6911948e+00
0.0 1.116287e-03 -0.8842492e+00
0.0 1.079394e-03 -0.7889649e+00
0.0 1.095487e-03 -0.7515649e+00
0.0 1.109514e-03 -0.7880779e+00
0.0 1.107834e-03 -0.9465210e+00
0.0 1.051570e-03 -0.6588534e+00
0.0 1.091724e-03 -0.7715050e+00
0.0 1.082285e-03 -1.0658890e+00
0.0 1.083902e-03 -1.0704890e+00
0.0 1.080594e-03 -0.9193419e+00
0.0 1.149715e-03 -3.6306130e+00
0.0 1.157927e-03 -3.7297410e+00
0.0 1.153304e-03 -3.7714150e+00
0.0 1.135727e-03 -3.1972110e+00
0.0 1.149696e-03 -3.7766260e+00
0.0 1.157872e-03 -3.8231080e+00
0.0 1.138310e-03 -3.3255960e+00
0.0 1.159167e-03 -3.9449080e+00
0.0 1.166192e-03 -3.9178330e+00
0.0 1.155944e-03 -3.5776630e+00
0.0 1.154809e-03 -3.7747760e+00
0.0 1.147320e-03 -3.5397490e+00
0.0 1.144153e-03 -3.5824530e+00
0.0 1.152409e-03 -3.6679100e+00
0.0 1.151777e-03 -3.7244810e+00
0.0 1.149313e-03 -3.5705580e+00
0.0 1.150145e-03 -3.7210710e+00
0.0 1.162584e-03 -3.8640970e+00
0.0 1.150148e-03 -3.7108840e+00
0.0 1.154640e-03 -3.7203370e+00
0.0 1.153204e-03 -3.7010430e+00
* sensor 15 calibration parameters seem to be missing from >> the manual, original coefficient kept

A.3 Configuration Script (CONFIG2.CFG)

* Edited by Chad 8/28/08
* This is the first physical sensor configuration use during
* TAVEX '08 testing off the coast of Korea.

* The CTD chain configuration file is named fulchain.cal which is
* composed of all of the default manual/book values.

* Sequence is from bottom to top on the chain.
* Addr == address of 2008 lab test CTD sensor (must be between
* 1 and 255).
* Dist == distance between sensor and depressor (end of cable)
* The distance is read in meters but is accepted without
* decimal (ex. 146 == 14.6m).
* Color is palette entry for display on screen (options from 1
* to 15).
<table>
<thead>
<tr>
<th>*Addr</th>
<th>Dist</th>
<th>Color</th>
<th>Status Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>045</td>
<td>22</td>
<td>1</td>
<td>*</td>
</tr>
<tr>
<td>037</td>
<td>46</td>
<td>2</td>
<td>*</td>
</tr>
<tr>
<td>034</td>
<td>71</td>
<td>3</td>
<td>*</td>
</tr>
<tr>
<td>008</td>
<td>95</td>
<td>4</td>
<td>*</td>
</tr>
<tr>
<td>026</td>
<td>120</td>
<td>5</td>
<td>*</td>
</tr>
<tr>
<td>009</td>
<td>144</td>
<td>6</td>
<td>*</td>
</tr>
<tr>
<td>029</td>
<td>168</td>
<td>7</td>
<td>*</td>
</tr>
<tr>
<td>049</td>
<td>193</td>
<td>8</td>
<td>*</td>
</tr>
<tr>
<td>027</td>
<td>205</td>
<td>9</td>
<td>*</td>
</tr>
<tr>
<td>014</td>
<td>217</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>019</td>
<td>229</td>
<td>11</td>
<td>*</td>
</tr>
<tr>
<td>028</td>
<td>242</td>
<td>12</td>
<td>*</td>
</tr>
<tr>
<td>048</td>
<td>254</td>
<td>13</td>
<td>*</td>
</tr>
<tr>
<td>007</td>
<td>266</td>
<td>14</td>
<td>*</td>
</tr>
<tr>
<td>035</td>
<td>278</td>
<td>15</td>
<td>*</td>
</tr>
<tr>
<td>039</td>
<td>290</td>
<td>1</td>
<td>*</td>
</tr>
<tr>
<td>012</td>
<td>315</td>
<td>2</td>
<td>*</td>
</tr>
<tr>
<td>018</td>
<td>339</td>
<td>3</td>
<td>*</td>
</tr>
<tr>
<td>044</td>
<td>363</td>
<td>4</td>
<td>*</td>
</tr>
<tr>
<td>030</td>
<td>388</td>
<td>5</td>
<td>*</td>
</tr>
<tr>
<td>024</td>
<td>394</td>
<td>6</td>
<td>*</td>
</tr>
<tr>
<td>021</td>
<td>400</td>
<td>7</td>
<td>* very intermittent</td>
</tr>
<tr>
<td>040</td>
<td>406</td>
<td>8</td>
<td>*</td>
</tr>
<tr>
<td>046</td>
<td>412</td>
<td>9</td>
<td>*</td>
</tr>
<tr>
<td>013</td>
<td>418</td>
<td>10</td>
<td>*</td>
</tr>
<tr>
<td>047</td>
<td>424</td>
<td>11</td>
<td>*</td>
</tr>
<tr>
<td>043</td>
<td>431</td>
<td>12</td>
<td>*</td>
</tr>
<tr>
<td>017</td>
<td>437</td>
<td>13</td>
<td>* intermittent</td>
</tr>
<tr>
<td>031</td>
<td>443</td>
<td>14</td>
<td>* intermittent</td>
</tr>
<tr>
<td>041</td>
<td>449</td>
<td>15</td>
<td>*</td>
</tr>
</tbody>
</table>
A.4 Towed CTD Data Processing Scripts

The ADM towed CTD system measures and stores conductivity, temperature, pressure, and date/time measurements in a binary format denoted by the file extension "*.DTA". A script was written to convert these files to *.mat files. Then the challenging problem of getting rid of false data points (or shotgun scatter as it was chosen to be referred to) caused by parity errors and faulty sensors during measurements had to be tackled. The most difficult challenge when dealing with this particular data set was dealing with faulty pressure sensor readings. During the CTD tow most of the CTD fin’s pressure sensors gave either faulty or miscalibrated data. A polynomial fit of seven reliable pressure sensors verse cable placement was used to approximate the placement of each CTD fin in the water column both vertically and horizontally from the sea surface and research vessel respectively. Finally, all of the data had to be extrapolated in order to fill holes created by filtering, and an appropriate mesh grid had to be set to allow computational propagation experiments. The following MATLAB scripts were used for these functions.

A.5 Converting *.DTA to *.mat Files (load_dta.m)

The following script was used to convert the binary .DTA files to easily loadable .mat files.

```matlab
1 % Last updated by Chad 7/28/08
2 % Data analyzation software for the CTD sensor chain.
3 *
4 % This M-file will prompt the user to pick a specific .DTA file to
5 % convert to a mat file. Data is saved in the variable raw_data.
```
clear all
close all
clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%% initial conditions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cal_filename = 'fulchain.cal'; % name of calibration parameters
cfg_filename = 'config2.cfg'; % name of calibration parameters

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%% Load and Calibrate Data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% load numerical_order % loads all variables from the file
[data_filename, pathname] = uigetfile('*.dta', 'Pick a CTD file');

fid = fopen([pathname cfg_filename], 'r'); % open file for reading
cfg_data = textscan(fid, '%u %f64 %u ', 'CommentStyle', '+');
status = fclose(fid); % closes data file

fid = fopen([pathname cal_filename], 'r'); % open file for reading
cal_data = textscan(fid, ...
    '%u %u %f64 %f64 %f64 %f64 %f64 %f64 %f64 %f64', ...
    'CommentStyle', '+');
status = fclose(fid); % closes data file

fid = fopen([pathname data_filename], 'r');
numofrows = length(cfg_data{1,1})*3+6; % index array size
    % according to number of sensors being used in the config file
raw_data = fread(fid, [numofrows inf], 'uint16'); % read file
status = fclose(fid); % closes data file

A.6 Separate CTD Data (SeparateDataFiles.m)

This MATLAB routine is used to separate the .mat output files of load_dta.m into 30 minute sections to look at all data on equal time scales.
% Chad Smith 10/30/08
% This .m file will segment a data file into sections of data with
% length "samples" and save them into numerically incremented
% names starting with the data file name.

clear all
close all
cclc

[data_filename, pathname] = uigetfile('*mat',...
   'Pick a .mat Data File to Segment');

fid = fopen([pathname data_filename]','r');
numofrows = length(cfg_data{1,1})*3+6;
DataStruct = open ([pathname data_filename]);
data = DataStruct.raw_data;

x = size(DataStruct.raw_data,2);
y = size(DataStruct.raw_data,1);

% # of samples to put into each separate array
samples = 900; % 30 min 2 second samples
% samples = 1800; % 30 min 1 second samples

[file_name, file_extention] = strtok(data_filename, '.');
% separate file name from extention
separations = floor(x/samples);
separations = ceil(x/samples);

for n=1:separations
    starting = 1+samples*(n-1);
    ending = samples+samples*(n-1);
    if (n==separations)
        ending = x;
    end

    data = DataStruct.raw_data(:,starting:ending);

    variable = genvarname(['Z' file_name '_.num2str(n)'], who);
A.7 Calibration and Filtering (cal_filter_cut.m)

The following program takes a 30 minute data section as input. It then parses, calibrates, and filters shotgun scatter from this file using the sub-functions from the following two sections and plots the results. The last line of this program calls the sub-function Gfilter.m. This function allows the user to graphically filter the data by eye if the filtering function filter_bad_pts_strddev.m does not do well. Many of the TAVEX data sections had to be filtered in this tedious manor.

```matlab
% load data, calibration, and config files
[data_filename, pathname] = uigetfile('*.mat', ...
'Pick a CTD data file');
load([ pathname data_filename]); % load data file
%load fulchain_calibration.mat; % load calibration data
```
load include\fulchain_calibration_bookvalues.mat
  % load original book calibration data
load include\sensorcolors.mat;
  % load sensor display colors for plots
load include\config2.mat; % load configuration data
% screensize = get(0,'ScreenSize'); % get screen size vector

% lets make things little easier
numofsensors = size(cfg_data,1);
numberofsamples = size(raw_data,2);
sensorlocation = cfg_data(:,1)';
  % remember sensors are ordered from BOTTOM to TOP

$$$$$$$$$$$$$$ make three raw arrays for temp, cond, and pres
raw_temp_data = zeros(size(cfg_data,1),size(raw_data,2));
raw_pres_data = zeros(size(cfg_data,1),size(raw_data,2));
raw_cond_data = zeros(size(cfg_data,1),size(raw_data,2));

for i = 1:numofsensors
  for j = 1:numberofsamples
    raw_temp_data(i,j) = raw_data(i*3+4,j);
    raw_cond_data(i,j) = raw_data(i*3+5,j);
    raw_pres_data(i,j) = raw_data(i*3+6,j);
  end
end

$$$$$$$$$$ put sensor data in numerical order
temp_data_in_order = zeros(50,numberofsamples);
pres_data_in_order = zeros(50,numberofsamples);
cond_data_in_order = zeros(50,numberofsamples);
for k=1:length(sensorlocation)
  temp_data_in_order(sensorlocation(k),:) = raw_temp_data(k,:);
  pres_data_in_order(sensorlocation(k),:) = raw_pres_data(k,:);
  cond_data_in_order(sensorlocation(k),:) = raw_cond_data(k,:);
end

%%%%% make three arrays of cal coefficients
temp_cal_coef = [cal_data(:,3) cal_data(:,4) cal_data(:,5)];
pres_cal_coef = [cal_data(:,6) cal_data(:,7) cal_data(:,8)];
cond_cal_coef = [cal_data(:,9) cal_data(:,10) cal_data(:,11)];
%%%%%% do calibration calculations (y = a2*x^2+a1*x+a0)
\n```
temp.data = zeros(numofsensors,numberofsamples);
pres.data = zeros(numofsensors,numberofsamples);
cond.data = zeros(numofsensors,numberofsamples);
```

```
for jj = 1:50
    temp.data(jj,:) = (temp.data_in_order(jj,:).^2).*...
                      temp_cal_coef(jj,1)+temp.data_in_order(jj,:).*...
                      temp_cal_coef(jj,2)+temp.data_in_order(jj,:);  
pres.data(jj,:) = (pres.data_in_order(jj,:).^2).*...
                      pres_cal_coef(jj,1)+pres.data_in_order(jj,:).*...
                      pres_cal_coef(jj,2)+pres.data_in_order(jj,:);  
cond.data(jj,:) = (cond.data_in_order(jj,:).^2).*...
                      cond_cal_coef(jj,1)+cond.data_in_order(jj,:).*...
                      cond_cal_coef(jj,2)+cond.data_in_order(jj,:);
end
```

%%%%%% insert NaNs for unused sensors
```
for jj = 1:50
    if (jj \ne sensorlocation)
        temp.data(jj,:) = NaN;
        pres.data(jj,:) = NaN;
        cond.data(jj,:) = NaN;
    end
end
```

%%%%%% pull out time and correct
```
datetime.raw = raw.data(1:6,:);
DateTimeNum = zeros(1,size(datetime.raw,2));
DateTimeStrings = cell(2,size(datetime.raw,2));
for jj=1:size(datetime.raw,2)
    % adjust for y2k and convert to GMT time
    DateTimeNum(jj) = (36524.16666666666666)*...
        datenum(datetime.raw(1,jj),datetime.raw(2,jj),...
        datetime.raw(3,jj), datetime.raw(4,jj),...
        datetime.raw(5,jj), datetime.raw(6,jj));
    DateTimeStrings{1,jj} = datestr(DateTimeNum(jj), 0);
    DateTimeStrings{2,jj} = DateTimeStrings{1,jj}(13:20);
end
```
DateTimeStr = datestr(DateTimeNum(1), 0);

% bulk filter data
[temp_data, cond_data, pres_data, SensorStatusArray]=
filter_bad_pts_strddev(temp_data, cond_data,
pres_data, sensorlocation);

% % 
% [temp_data, cond_data, pres_data, SensorStatusArray]=
% filter_bad_pts_exrema(temp_data, cond_data,
% pres_data, sensorlocation);

% guesstimate depth from pressure here
load include/pres34.mat
% just a guesstimation of the z values to make the surf plot
depth = zeros(size(sensorlocation,2),size(raw_data,2));
for n=1:size(raw_data,2)
    depth(:,n)=z;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% plot data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set up axis to keep all plot relative in time
% if numberofsamples\leq900
%    interval = 30/899;
% else
%    interval = 30/1799;
% end
% mins = 0:interval:30;
mins = 0:2/60:length(temp_data)*(2/60);

% temperature plot
% temperature vs time
figure('Position',[1 1 screensize(3) screensize(4)-70])
tempfig = figure(1); hold on; setwindowstate(tempfig,'maximize');
for i=1:size(sensorlocation,2)
    if 1
        if SensorStatusArray(sensorlocation(i),1)>0 &&
            ~isnan(SensorStatusArray(sensorlocation(i),1))
handle_temp(i) = plot(mins(1:1:length(temp_data)),...
    temp_data(sensorlocation(i),:),'.','MarkerSize',5,'DisplayName',['#' num2str(sensorlocation(i))],...
    'Visible','on','color',[
        sensorcolors(sensorlocation(i),1)
        sensorcolors(sensorlocation(i),2)
        sensorcolors(sensorlocation(i),3)]);
end
end
grid on;
xlabel('Time [minutes]','fontsize',12,'fontweight','b');
ylabel('Temperature [\textdegree{}C]','fontsize',12,'fontweight','b');
title(["Temperature vs Time ('data_filename')"],'fontsize',18,'fontweight','b');
legend off;
xlim([0 mins(end)]);
ylim([12 28]);

%%%%%%%%%%%%%%%%% conductivity plot
%%%%%%%%%%%%%%%%% conductivity vs time
% figure('Position',[1 1 screensize(3) screensize(4)-70])
condfig = figure(2); hold on; setwindowstate(condfig,'maximize');
for i=1:size(sensorlocation,2)
    if 1
        % if SensorStatusArray(sensorlocation(i),2)>0 &&...
        % ~isnan(SensorStatusArray(sensorlocation(i),2))
        handle_cond(i) = plot(mins(1:1:length(cond_data)),...
            cond_data(sensorlocation(i),:),'.','MarkerSize',5,'DisplayName',['#' num2str(sensorlocation(i))],...
            'Visible','on','color',[
                sensorcolors(sensorlocation(i),1)
                sensorcolors(sensorlocation(i),2)
                sensorcolors(sensorlocation(i),3)]);
    end
end
grid on;
xlabel('Time [minutes]','fontsize',12,'fontweight','b');
ylabel('Conductivity [mmho/cm]','fontsize',12,'fontweight','b');
title(["Conductivity vs Time ('data_filename')"],'fontsize',...
18, 'fontweight', 'b');
legend off;
xlim([0 mins(end)]);
ylim([38 50]);

%%%%%%%%%%%%%%%% pressure plot
%%%%%%%%%%%%%%%% pressure vs time
% figure('Position',[1 1 screensize(3) screensize(4)-70])
presfig = figure(3); hold on; setwindowstate(presfig,'maximize');
for i=1:size(sensorlocation,2)
  if 1
    if SensorStatusArray(sensorlocation(i),3)>0 &...  
      ~isnan(SensorStatusArray(sensorlocation(i),3))
      handle_pres(i) = plot(mins(1:1:length(pres_data)),...  
       pres_data(sensorlocation(i),:),'.','MarkerSize',5,...
       'DisplayName',['#' num2str(sensorlocation(i))],...
       'Visible','on','color',...
       [sensorcolors(sensorlocation(i),1)...  
       sensorcolors(sensorlocation(i),2)...  
       sensorcolors(sensorlocation(i),3))];
  end
end
grid on;
xlabel('Time [minutes]', 'fontsize', 12, 'fontweight', 'b');
ylabel('Pressure [dbar]', 'fontsize', 12, 'fontweight', 'b');
% 1 dbar = ~1 meter
title([['Pressure vs Time (' data_filename ')'],'fontsize',...  
18,'fontweight','b']);
legend off;
xlim([0 mins(end)]);
ylim([2 62]);

%%%%%%%%%%%%%%%% temperature surf plot
%%%%%%%%%%%%%%%% temperature vs pressure vs time
% figure('Position',[1 1 screensize(3) screensize(4)-70])
tempsurf = figure(4); hold on; setwindowstate(tempsurf,'maximize');
Y = zeros(size(sensorlocation,2),size(temp_data,2));
surf(mins(1:1:length(temp_data)),Y,depth,...
  temp_data(sensorlocation,:));
shading flat; % shows individual data points
%shading interp; % interpolates between data samples to smooth plot
colormap('jet');
xlabel('Time [minutes]', 'fontsize', 12, 'fontweight', 'b');
zlabel('Pressure [dBar]', 'fontsize', 12, 'fontweight', 'b');
title(['Temperature vs Depth vs Time (' data_filename ')'],...
    'fontsize', 18, 'fontweight', 'b');
view(0, 180); % make z positive
colorbar(); % create colorbar
caxis([12.0 27.0]);
xlim([0 mins(end)]);
zlim([15 57]);
annotation(figure(4), 'textbox', 'String',...
    {'Temperature [{\circ}C]'},...
    'FontWeight', 'bold',...
    'FontSize', 12, ...
    'FitBoxToText', 'off',...
    'LineStyle', 'none',...
    'Position', [0.82 0.928 0.1222 0.03364]);

% create textbox to label colorbar

%%%%%%%%%%%%%%%% conductivity surf plot
%%%%%% conductivity vs pressure vs time
% figure('Position', [1 1 screensize(3) screensize(4)-70]);
% (with position vector)
condsurf = figure(5); hold on; setwindowstate(condsurf, 'maximize');
Y = zeros(size(sensorlocation, 2), size(cond_data, 2));
surf(mins(1:1:length(cond_data)), Y, depth, ...
    cond_data(sensorlocation, :));
shading flat; % shows individual data points
%shading interp; % interpolates between data samples to smooth plot
colormap('jet');
xlabel('Time [minutes]', 'fontsize', 12, 'fontweight', 'b');
zlabel('Pressure [dBar]', 'fontsize', 12, 'fontweight', 'b');
title(['Conductivity vs Depth vs Time (' data_filename ')'],...
    'fontsize', 18, 'fontweight', 'b');
view(0, 180); % make z positive
colorbar(); % create colorbar
caxis([37.0 50.0]);
xlim([0 mins(end)]);
zlim([15 57]);
% Create textbox to label colorbar
annotation(figure(5), 'textbox', 'String', ... 
    {'Conductivity [mmho/cm]'}, ... 
    'FontSize', 12, ... 
    'FitBoxToText', 'off', ... 
    'LineStyle', 'none', ... 
    'Position', [0.7867 0.928 0.1222 0.03364]);

% Create textbox to label colorbar
for n=1:5
    % Print start time
    annotation(figure(n), 'textbox', ... 
        'String', ... 
        {'Start Date/Time: ' datestr(datestr(DateTimeNum(1), 0), 0)}, ... 
        'FontSize', 12, ... 
        'FitHeightToText', 'off', ... 
        'LineStyle', 'none', ... 
        'Position', [0.1261 0.03086 0.187 0.03364]);

    % Print stop time
    annotation(figure(n), 'textbox', ... 
        'String', ... 
        {'Stop Date/Time: ' datestr(datestr(DateTimeNum(length(temp_data)), 0))}, ... 
        'HorizontalAlignment', 'right', ... 
        'FontSize', 12, ... 
        'FitHeightToText', 'off', ... 
        'LineStyle', 'none', ... 
        'Position', [0.7304 0.03295 0.1826 0.03364]);
end

Gfilter(sensorlocation, temp_data, cond_data, pres_data,... 
    handle_temp, handle_cond, handle_pres);

% Call routine to graphically snip bad points
A.8  Auto-filter (filter_bad_pts_strddev.m)

This function is used to automatically cut some of the shotgun scatter from CTD data. Filtering shotgun scatter is a difficult thing to automate, and this algorithm does not always work well depending on the particular data set, so another function called Gfilter.m is also used for filtering.

```matlab
function [temp_data, cond_data, pres_data, SensorStatusArray]=...
    filter_bad_pts_strddev(temp_data, cond_data, pres_data,...
    sensorlocation)

% declare sensors to not use at all
temp_bad = [16 31 17 21 12]; % completely bad temperature sensors
cond_bad = [16 31 17 21 18 49 12 43 13];
    % completely bad conductivity sensors
pres_bad = [16 31 17 21 13 24 12 14 9 8 33 43 7 19 26 27 28];
    % bad pressure sensors

SensorStatusArray = NaN(50,3);
    % array to keep track of percentage of good data

for n=1:size(sensorlocation,2)
```

%%%%%%% Writer(s): Chad Smith
%%%%%%% Last Updated: 07/08/2009.
%%%%%%% Project: TAVEX 2008

%%%%%%% Program Details: This function filters the extrema points (usually parity errors) and other erratic data out of the temperature, conductivity, and pressure arrays. Erratic data is simply replaced by a NaN value in this function.

%%%%%%% This function uses the standard deviation of each sensor's reading's to set max. and min. extrema values a cut data above and below those points respectively.

```matlab
% declare sensors to not use at all
```
Data = temp_data(sensorlocation(n),:); % grab specific sensor

if all(sensorlocation(n) ≠ temp_bad)
    % don’t even process sensors we already know as bad
    % chop extrema 1
    Data_extremachopped = zeros(1,size(Data,2));
    for m=1:size(Data,2) % columns
        if Data(m) < 12 || Data(m) > 27
            % cut above and below all sensors realistic
            % area (53.6 - 80.6 degrees F)
            Data_extremachopped(m) = NaN;
        else
            Data_extremachopped(m) = Data(m);
        end
    end

    % get individual sensor extreme values and ∆
    DataWithoutExtremes = ...
    Data_extremachopped(find(~isnan(Data_extremachopped)));
    % condense data without extremes
    StandardDeviation = std(DataWithoutExtremes);
    % standard deviation
    Mean = mean(DataWithoutExtremes);
    DataMax = 5.0*StandardDeviation + Mean;
    DataMin = -5.0*StandardDeviation + Mean;

    % chop data max and min
    Data_DataMaxMinChopped = zeros(1,size(Data,2));
    for m=1:size(Data,2) % columns
        if Data(m) < DataMin || Data(m) > DataMax
            % cut above and below all sensors 99.9%
            % good data area using standard deviation
            Data_DataMaxMinChopped(m) = NaN;
        else
            Data_DataMaxMinChopped(m) = Data(m);
        end
    end

    % condense data to filter
    IndexOfData = find(~isnan(Data_DataMaxMinChopped));
DataPointCheck = size(IndexOfData,2);
    % how many data points do we have left?

DataCondensed_extremachopped = ...
    Data_extremachopped(IndexOfData);
    % condense data without extremes

    % filter shotgun pattern
    if DataPointCheck>50    % only filter if there is more
        % than 50 good data samples
            % averaging / 0 phase shift FIR filter
            windowSize = 15;
            DataCondensed_averaged=...
                filtfilt(ones(1,windowSize)/windowSize,...
                        1,DataCondensed_extremachopped);

            % compare smoothed out data to real data and
            % cut extreme deviations
            DataCondensed_gooddata = zeros(1,...
                size(DataCondensed_extremachopped,2));
            for m=1:size(DataCondensed_extremachopped,2)
                if isnan(DataCondensed_extremachopped(m))
                    DataCondensed_gooddata(m)=NaN;
                else
                    % cut out deviations greater ∆
                    ∆ = abs(DataCondensed_extremachopped(m)...
                            -DataCondensed_averaged(m));
                    if ∆>1.5    % this number can be a range
                        % between 0.25 and 3 degrees depending on
                        % how dynamic the data is
                            DataCondensed_gooddata(m)=NaN;
                    else
                        DataCondensed_gooddata(m)=...
                            DataCondensed_extremachopped(m);
                    end
                end
            end
        GoodSamples = size(find(~isnan...
           (DataCondensed_gooddata)),2);
        % how many good samples?
% expand data back into correct time domain
GoodData = nan(1,size(Data_extremachopped,2));
GoodData(IndexOfData) = DataCondensed_gooddata;
temp_data(sensorlocation(n),:) = GoodData;
    % assign good data back to data array
SensorStatusArray(sensorlocation(n),1) = ...
    (GoodSamples/size(Data,2))*100;
    % of samples used for this sensor in this run
else
    temp_data(sensorlocation(n),:) = NaN;
    SensorStatusArray(sensorlocation(n),1) = 0;
    % virtually no good samples
end
else
    temp_data(sensorlocation(n),:) = NaN;
    SensorStatusArray(sensorlocation(n),1) = 0;
    % virtually no good samples
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% filter conductivity
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for n=1:size(sensorlocation,2)
    Data = cond_data(sensorlocation(n),:);
    % grab specific sensor data
    if all(sensorlocation(n)\cond_bad) % don't even process
        %sensor we already know as bad
        % chop extrema 1
        Data_extremachopped = zeros(1,size(Data,2));
        for m=1:size(Data,2) %columns
            if Data(m)<38 || Data(m)>50 % cut above and
                %below all sensors realistic area
                Data_extremachopped(m)=NaN;
            else
                Data_extremachopped(m)=Data(m);
            end
        end
    end
end
% get individual sensor extreme values and ∆
DataWithoutExtremes = Data_extremachopped(find...
    (~isnan(Data_extremachopped)));
    % condense data without extremes
StandardDeviation = std(DataWithoutExtremes);
    % standard deviation
Mean = mean(DataWithoutExtremes);
DataMax = 5*StandardDeviation+Mean;
DataMin = -5*StandardDeviation+Mean;

% chop data max and min
Data_DataMaxMinChopped = zeros(1,size(Data,2));
for m=1:size(Data,2)    %columns
    if Data(m)<DataMin || Data(m)>DataMax
        % cut above and below all sensors 99.9% good
        % data area using standard deviation
        Data_DataMaxMinChopped(m)=NaN;
    else
        Data_DataMaxMinChopped(m)=Data(m);
    end
end

% condense data to filter
IndexOfData = find(~isnan(Data_DataMaxMinChopped));
DataPointCheck = size(IndexOfData,2);
% how many data points do we have left?
DataCondensed_extremachopped = Data_extremachopped...
    (IndexOfData); % condense data without extremes

% filter shotgun pattern
if DataPointCheck>50    % only filter if there is more than
    %50 good data samples
    % averaging / 0 phase shift FIR filter
    windowSize = 15;
    DataCondensed_averaged=filtfilt(ones(1,...
        windowSize)/windowSize,1,...
    DataCondensed_extremachopped);
    % compare smoothed out data to real data
% and cut extreme deviations
DataCondensed_goddata = zeros(1,size...
(DataCondensed_extremachopped,2));

for m=1:size(DataCondensed_extremachopped,2)
    if isnan(DataCondensed_extremachopped(m))
        DataCondensed_goddata(m)=NaN;
    else
        % cut out deviations greater \( \Delta \)
        \( \Delta =... \)
        abs(DataCondensed_extremachopped(m)... -DataCondensed_averaged(m));
        if \( \Delta > 1.5 \)
            DataCondensed_goddata(m)=NaN;
        else
            DataCondensed_goddata(m)=...
            DataCondensed_extremachopped(m);
        end
    end
end

GoodSamples = size(find(~isnan...
(DataCondensed_goddata)),2);
% how many good samples?

% expand data back into correct time domain
GoodData = nan(1,size(Data_extremachopped,2));
GoodData(IndexOfData) = DataCondensed_goddata;

cond_data(sensorlocation(n),:) = GoodData;
% assign good data back to data array
SensorStatusArray(sensorlocation(n),2) =...
(GoodSamples/size(Data,2))*100;
% of samples used for this sensor in this run
else
    cond_data(sensorlocation(n),:) = NaN;
    SensorStatusArray(sensorlocation(n),2) = 0;
    % virtually no good samples
end
else
    cond_data(sensorlocation(n),:) = NaN;
    SensorStatusArray(sensorlocation(n),2) = 0;
% virtually no good samples

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%% filter pressure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for n=1:size(sensorlocation,2)
    Data = pres_data(sensorlocation(n),:);
    % grab specific sensor data

    if all(sensorlocation(n) \= pres_bad)
        % don't even process sensor we already know as bad
        % chop extrema 1
        Data_extremachopped = zeros(1,size(Data,2));
        for m=1:size(Data,2) % columns
            if Data(m)<0 || Data(m)>60
                % cut above and below all sensors
                % realistic area
                Data_extremachopped(m)=NaN;
            else
                Data_extremachopped(m)=Data(m);
            end
        end

        % get individual sensor extreme values and \Delta
        DataWithoutExtremes = Data_extremachopped...
            (find(~isnan(Data_extremachopped)));
        % condense data without extremes
        StandardDeviation = std(DataWithoutExtremes);
        % standard deviation
        Mean = mean(DataWithoutExtremes);
        DataMax = 3.2*StandardDeviation+Mean;
        DataMin = -3.2*StandardDeviation+Mean;

        % chop data max and min
        Data(DataMaxMinChopped = zeros(1,size(Data,2));
            for m=1:size(Data,2) % columns
                if Data(m)<DataMin || Data(m)>DataMax
                    % cut above and below all sensors 99.9%
                end
            end
        end
    end
end
% good data area using standard deviation
    Data_DataMaxMinChopped(m) = NaN;
else
    Data_DataMaxMinChopped(m) = Data(m);
end

% condense data to filter
IndexOfData = find(~isnan(Data_DataMaxMinChopped));
DataPointCheck = size(IndexOfData, 2);
% how many data points do we have left?
DataCondensed_extremachopped = ...
    Data_extremachopped(IndexOfData);
% condense data without extremes

% filter shotgun pattern
if DataPointCheck > 100 % only filter if there is more
    % than 50 good data samples
    % averaging / 0 phase shift FIR filter
    windowSize = 30;
    DataCondensed_averaged = filtfilt(ones(1, windowSize)/windowSize, 1,...
        DataCondensed_extremachopped);

% compare smoothed out data to real data
% and cut extreme deviations
DataCondensed_gooddata = zeros(1, size...
    (DataCondensed_extremachopped, 2));
for m = 1:size(DataCondensed_extremachopped, 2)
    if isnan(DataCondensed_extremachopped(m))
        DataCondensed_gooddata(m) = NaN;
    else
        % cut out deviations greater ∆
        ∆ = abs( DataCondensed_extremachopped(m) -...
            DataCondensed_averaged(m) );
        if ∆ > 3
            DataCondensed_gooddata(m) = NaN;
        else
            DataCondensed_gooddata(m) = ...
DataCondensed_extremachopped(m);

end
end

GoodSamples = size(find...
(~isnan(DataCondensed_gooddata)),2);
% how many good samples?

% expand data back into correct time domain
GoodData = nan(1,size(Data_extremachopped,2));
GoodData(IndexOfData) = DataCondensed_gooddata;
pres_data(sensorlocation(n),:) = GoodData;
% assign good data back to data array
SensorStatusArray(sensorlocation(n),3) =...
(GoodSamples/size(Data,2))*100;
% of samples used for this sensor in this run

else
pres_data(sensorlocation(n),:) = NaN;
SensorStatusArray(sensorlocation(n),3) = 0;
% virtually no good samples
end
else
pres_data(sensorlocation(n),:) = NaN;
SensorStatusArray(sensorlocation(n),3) = 0;
% virtually no good samples
end

A.9  Graphically Filter (Gfilter.m)

Filtering the shotgun scatter caused in CTD data by communication errors and faulty sensors is difficult to do with an auto filtering method such as shown in the previous section. In order to make certain that data is appropriately filtered, this GUI allows the user select points to be replaced by NaN values within the data set. Although time consuming, this method lets a user be certain that faulty data has been cut without cutting good data points.
function Gfilter(sensorlocation, temp_data, cond_data, ...
    pres_data, handle_temp, handle_cond, handle_pres)

    sensors = {'#36'; '#50'; '#33'; '#{16'; '#31'; '#17'; '#43'; ...
        '#47'; '#13'; '#{66'; '#{40'; '#{21'; '#{24'; '#{30'; '#{44'; '#{18'; ...
        '#{12'; '#{35'; '#{7'; '#{48'; '#{21'; '#{24'; '#{30'; '#{44'; '#{18'; ...
        '#{12'; '#{35'; '#{7'; '#{48'; '#{21'; '#{24'; '#{30'; '#{44'; '#{18'; ...
        '#{49'; '#{29'; '#{9'; '#{26'; '#{8'; '#{34'; '#{37'; '#{45';
    }

    % create and hide GUI as it is being constructed
    f = figure('Visible','off','Position',[300,400,250,150]);

    % construct components
    temp = uicontrol('Style','radiobutton','String','...
        'Temperature','Position',[0,110,80,25],...
        'Callback',{@tempbutton_Callback});
    cond = uicontrol('Style','radiobutton','String','...
        'Conductivity','Position',[0,90,80,25],...
        'Callback',{@condbutton_Callback});
    pres = uicontrol('Style','radiobutton','String','...
        'Pressure','Position',[0,70,80,25],...
        'Callback',{@presbutton_Callback});
    showall = uicontrol('Style','pushbutton','String','...
        'Show All','FontWeight','bold','...
        'FontSize',10,..
        'Position',[150,50,100,40],...
        'Callback',{@showallbutton_Callback});
    go = uicontrol('Style','pushbutton','String','GO',...
        'FontWeight','bold','...
        'FontSize',14,..
        'Position',[0,0,100,40],...
        'Callback',{@gobutton_Callback});
    done = uicontrol('Style','pushbutton','String','Done',...
'FontWeight', 'bold', ...
'FontSize', 14, ...
'ForegroundColor', 'r', ...
'Position', [150, 0, 100, 40], ...
'Callback', {@donebutton_Callback};

list = uicontrol('Style', 'popupmenu', ...
'String', sensors, ...
'FontWeight', 'bold', ...
'FontSize', 10, ...
'Position', [150, 100, 100, 40], ...
'Callback', {@list_menu_Callback});

% initialize
% assign the GUI a name to appear in the window title
set(f, 'Name', 'Graphically Filter Data')
% move the GUI to the center of the screen
movegui(f, 'northeast')
% make GUI visible
set(f, 'Visible', 'on');
guidata(f, {temp_data, cond_data, pres_data})
% save data to GUI data
state = 0;
sensor_index = 34;
chopall = 0;

% pushbutton callbacks
function tempbutton_Callback(source, eventdata)
    % cut temp data
    state = 1;
    set(cond, 'Value', 0)
    set(pres, 'Value', 0)
end

function condbutton_Callback(source, eventdata)
    % cut cond data
    state = 2;
    set(temp, 'Value', 0)
    set(pres, 'Value', 0)
end
function presbutton_Callback(source, eventdata)
    % cut pres data
    state = 3;
    set(cond,'Value',0)
    set(temp,'Value',0)
end

function showallbutton_Callback(source, eventdata)
    % cut pres data
    set(handle_temp,'visible','on')
    set(handle_cond,'visible','on')
    set(handle_pres,'visible','on')
    % chopall = 1; % uncomment this line to chop lots of sensors
end

function list_menu_Callback(source, eventdata)
    % determine the selected sensor
    str = get(source, 'String');
    val = get(source,'Value');
    % select current sensor
    switch str{val};
    case sensors(1) % sensor #36 selected
        current_sensor = 36;
    case sensors(2) % sensor #50 selected
        current_sensor = 50;
    case sensors(3) % sensor #33 selected
        current_sensor = 33;
    end
    sensor_index = int8(34-(val-1)); % sensors are turned around
    chopall = 0;
end

function gobutton_Callback(source, eventdata)
    data = guidata(f); % grab data
    sensorlocationflipped = fliplr(sensorlocation); % inversion
    index = int8(1:1:34);
    index(sensor_index) = []; % exclude selected sensor
    switch state

case 1
    figure(1) % temp data
    handles = handle_temp(index);

    if chopall
        set(handle_temp,'visible','on')
        [pointslist,xselect,yselect] = selectdata();
        % get bad points
        for nn=1:length(pointslist)
            data{1}(sensorlocationflipped(nn),...
                pointslist{nn}) = NaN;
        end
    else
        set(handle_temp,'visible','off')
        set(handle_temp(sensor_index),'visible','on')
        % display selected
    end

    case 2
    figure(2) % cond data
    handles = handle_cond(index);

    if chopall
        set(handle_cond,'visible','on')
        [pointslist,xselect,yselect] = selectdata();
        % get bad points
        for nn=1:length(pointslist)
            data{2}(sensorlocationflipped(nn),...
                pointslist{nn}) = NaN;
        end
    else
        set(handle_cond,'visible','off')
        set(handle_cond(sensor_index),'visible','on')
        % display selected
    end
selectdata('Ignore',handles); % get bad points
data{2}(sensorlocation(sensor_index),...
    pointslist) = NaN;
end

% get bad points

figure(3) % pres data
handles = handle_pres(index);

if chopall
    set(handle_pres,'visible','on')
    [pointslist,xselect,yselect] = selectdata();
    % get bad points
    for nn=1:length(pointslist)
        data{3}(sensorlocationflipped(nn),...
            pointslist{nn}) = NaN;
    end
else
    set(handle_pres,'visible','off')
    set(handle_pres(sensor_index),'visible',...
        'on') % display selected
    [pointslist,xselect,yselect] = ...
        selectdata('Ignore',handles);
    % get bad points
    data{3}(sensorlocation(sensor_index),...
        pointslist) = NaN;
end

otherwise
    state = 0;
    warning('Something done happened!!!') % opps
end

guidata(f,data) % save function data to GUI data
figure(f) % bring GUI window back

function donebutton_Callback(source,eventdata)
data = guidata(f);  % uisave('data','data.mat') % save data as a .mat
A.10 Extrapolate (extrapolate.m)

The program extrapolate.m is used to extrapolate CTD data and fill gaps caused by the filtering process. Temperature and conductivity are first extrapolated horizontally, and then vertically to fill gaps from filtering using the functions fillnans.m, int4nan.m, and int4nan_hor.m. The algorithm used to fill NaN values is from J. Sellschopp's ctdchain tool box, modified slightly for the TAVEX data circumstances. Pressure data, however, had to be fit to a few trusted sensors because many of the pressure sensors were faulty. This pressure fit is done using the function fit_pres.m. Extrapolate.m and its sub-functions are found below.
% load raw data, calibration, and config files
[data_filename, pathname] = uigetfile('temp_filtered*.mat', 'Pick a CTD data file');
load([pathname 'temp_filtered.mat']);
load([pathname 'cond_filtered.mat']);
load([pathname 'pres_filtered.mat']);
load([pathname 'DateTimeNum.mat']);
load include\sensorlocation.mat
load include\sensorcolors.mat % load sensor colors
bkslshs = strfind(pathname, '\');
sectionname = pathname(bkslshs(length(bkslshs)-2)+1:bkslshs(length(bkslshs)-1)-1);
sectionnumber = str2double(sectionname(4:end));
load include\config2.mat
z0 = cfg.data(:,2)/10; % spacing of sensors from depressor

% extrapolate temp
temp.data.unextrapolated = temp.data;
data.inspacialorder = temp.data(sensorlocation,:);
data.filled = fillnans(z0, data.inspacialorder);
temp.data(sensorlocation,:) = data.filled;

% extrapolate cond
cond.data.unextrapolated = cond.data;
data.inspacialorder = cond.data(sensorlocation,:);
data.filled = fillnans(z0, data.inspacialorder);
cond.data(sensorlocation,:) = data.filled;

% extrapolate pressure
pres.data.unextrapolated = pres.data;
data.inspacialorder = pres.data(sensorlocation,:);
[pres.depth xlag] = fit_pres(z0, data.inspacialorder,...
    sensorlocation);
pres.data(sensorlocation,:) = pressure;
z = nan(size(pres.data,1),size(pres.data,2));
x = nan(size(pres.data,1),size(pres.data,2));
z(sensorlocation,:) = depth;
x(sensorlocation,:) = xlag;
function fld = fillnans(z0, fld0)

if min(size(z0)) > 1
    z0 = z0(:,2)/10;   % Sensor distance from tail (depressor)
end

% Rows without numbers are excluded from special processing.
rows = find(sum(~isnan(fld0)));
tm1 = fld0(rows,:);

tm1=int4nan_hor(tm1);       % horizontal interpolation
fld0(rows,:) = tm1;

fld=int4nan(z0,fld0);       % vertical interpolation

function y1 = int4nan(x, y)
% INT4NAN replaces missing values by interpolation
% and extrapolation
% Y1=INT4NAN(X, Y) is a copy of the matrix Y with gaps
% (imbedded NaN) filled by linear interpolation according
% to abscissa values X. The length of the vector X must
% match the number of rows or columns in Y. If Y is a
% square matrix, interpolation is in columns.
if min(size(x)) == 1
    x = x(:); % column vector
else
    error('First parameter X must be a vector')
end
n = length(x);

if size(y,1) == n
    y1 = y;
elseif size(y,2) == n
    y1 = y';
else
    error(['Length of vector X does not match either'
           'dimension of matrix Y'])
end

% [n, m] = size(y1);
gaps = sum(isnan(y1));

for i = find(gaps > 0 & gaps < n-1)% at least 2 values required
    avail=find(~isnan(y1(:,i)));
    unavail=find(isnan(y1(:,i)));
    y1(unavail,i) = interp1q(x(avail), y1(avail,i), x(unavail));
    extra = find(unavail < avail(1)) % leading NaN, extrapolate
    if ~isempty(extra)
        y1(unavail(extra),i) = y1(avail(1),i) + ...
            (y1(avail(2),i) - y1(avail(1),i))/(x(avail(2)) -...
            x(avail(1))) * ...
            (x(unavail(extra)) - x(avail(1)));
    end
    extra = find(unavail > avail(end)) % trailing NaN, extrapolate
    if ~isempty(extra)
        y1(unavail(extra),i) = y1(avail(end),i) + ...
            (y1(avail(end-1),i) - y1(avail(end),i))/(x(avail(end-1)) -...
            x(avail(end))) * ...
            (x(unavail(extra)) - x(avail(end)));
    end
function y1 = int4nan_hor(y)

n = size(y,2);
x = [1:1:n]';
y1 = y;
gaps = sum(isnan(y1'));

for i = find(gaps > 0 & gaps < n-50) % at least 50 values
    avail=find(~isnan(y1(i,:)));
    unavail=find(isnan(y1(i,:)));
    y1(i,unavail) = interp1(x(avail), y1(i,avail)',... x(unavail),'cubic');
    extra = find(unavail < avail(1)); % leading NaN, extrapolate
    if ~isempty(extra)
        y1(i,unavail(extra)) = y1(i,avail(1)) + ...
            (y1(i,avail(2)) - y1(i,avail(1)))/...
            (x(avail(2)) - x(avail(1)))... *
            (x(unavail(extra)) - x(avail(1)));
    end
    extra = find(unavail > avail(end)); % trailing NaN, extrapolate
    if ~isempty(extra)
        y1(i,unavail(extra)) = y1(i,avail(end)) + ...
```
A.11 Calculate Sound Speed and other Fields
(calculate_SDc.m)

This program and its sub functions are used to calculate sound speed, salinity, density, potential density, potential temperature, and buoyancy frequency from extrapolated fields of conductivity, temperature, and pressure. The sub-functions for sound speed (unesco) and salinity (1978 practical salinity scale) are included below. Phil Morgan’s seawater MATLAB toolbox [73] is the source of all other calculation functions used.
pick data
[filename, pathname] = uigetfile('*temp_extrapolated.mat', ...
    'Pick a CTD data file');
    % returns the name and path of the file selected

load([pathname 'temp_extrapolated.mat'])
load([pathname 'cond_extrapolated.mat'])
load([pathname 'pres_extrapolated.mat'])
load(include\sensorlocation.mat
bkshs = strfind(pathname, '\');
secname = pathname(bkshs(length(bkshs)-2)+...1:bkshs(length(bkshs)-1)-1);

% calculations (using practical salinity scale, UNESCO, 
% and SEAWATER)
reference_pressure = 0; % ocean surface
lat = 32+38/60; % approx. latitude of center of tow area
salinity = salinity(cond_data,temp_data,pres_data); % salinity[psu]
density = sw_dens(salinity,temp_data,pres_data); % density[kg/m^3]
pdensity = sw_pden(salinity,temp_data,pres_data,...
    reference_pressure); % potential density [kg/m^3]
ptemp = sw_ptmp(salinity,temp_data,pres_data,reference_pressure);

soundspeed = unesco(salinity,temp_data,pres_data);
    % sound speed [m/s]
[bfreq_squared,pvort,p_ave] = sw_bfrq(salinity,temp_data,...
    pres_data,lat); % brunt-vaisala frequency squared
bfreq = sqrt(bfreq_squared); % brunt-vaisala frequency [Hz]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%% PLOTTING %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% set up axis to keep all plot relative in time
% numberofsamples = size(temp_data,2);
% if numberofsamples<900
%    interval = 30/899;
% else
%    interval = 30/1799;
% end
% mins = 0:interval:30;
mins = 0:2/60:length(temp_data)*(2/60);

load([pathname 'z.mat'])
depth = z((sensorlocation),:);

%%%%%%%%%%%%%%%% salinity surf plot
Scontour = figure(1); hold on; setwindowstate(Scontour,'maximize');
X = ones(34,1)*mins(1:1:length(salinity));
Slines = [30:1:Sinc:35]; % set isopycnals to plot
contourf(X,depth,salinity(sensorlocation,:),Slines)
xlabel('Time [minutes]','fontsize',12,'fontweight','b');
ylabel('Depth [meters]','fontsize',12,'fontweight','b');
title(['Salinity vs Depth vs Time (' secname '.mat)'],
     'fontsize',18,'fontweight','b');
colorbar(); % create colorbar
% caxis([12.0 27.0]);
xlim([0 mins(end)]);
ylim([5 60]);
axis ij
grid on
annotation(figure(1),'textbox','String',
{"[Contours @ ' num2str(Sinc) ' psu increments']"},...
'FontWeight','bold',...
'FontSize',12,...
'FitBoxToText','off',...
'LineStyle','none',...
'Position',[0.845 0.928 0.1222 0.076]);
annotation(figure(1),'textbox','String',
{"Salinity [psu]"},...
'FontWeight','bold',...
'FontSize',12,...
'FitBoxToText','off',...
'LineStyle','none',...
'Position',[0.845 0.928 0.1222 0.03]);

%%%%%%%%%%%%%%%% density contour plot
Dcountour = figure(2); hold on; setwindowstate(Dcountour,'maximize');
X = ones(34,1)*mins(1:1:length(depth));
densitylines = [1010:Dinc:1035]; % set isopycnals to plot
contourf(X,depth,density(sensorlocation,:),densitylines);
contourf([1450 1450];
ylabel('Depth [meters]', 'fontsize', 12, 'fontweight', 'b');
title(['Density vs Depth vs Time (' secname '.mat)']);
colorbar(); % create colorbar
caxis([12.0 27.0]);
xlim([0 mins(end]));
ylim([5 60]);
axis ij
grid on

% create textbox to label colorbar
annotation(figure(2), 'textbox', 'String',
{{['Contours @ ' num2str(Dinc) ' kg/m^3 increments']},
'FontWeight','bold',... % create textbox to label colorbar
'FontSize',12,...
'FitBoxToText','off',...
'LineStyle','none',...
'Position',[0.845 0.928 0.1222 0.076]);

% create textbox to label colorbar
annotation(figure(2), 'textbox', 'String',
{{'Density [kg/m^3]'}},
'FontWeight','bold',... % create textbox to label colorbar
'FontSize',12,...
'FitBoxToText','off',...
'LineStyle','none',...
'Position',[0.845 0.928 0.1222 0.03]);

% soundspeed contour plot
ccountour = figure(3); hold on; setwindowstate(ccountour,...
'maximize');
X = ones(34,1)*mins(1:1:length(soundspeed));
clines = [1500:cinc:1545]; %
contourf(X,depth,soundspeed(sensorlocation,:),clines);
xlabel('Time [minutes]', 'fontsize', 12, 'fontweight', 'b');
ylabel('Depth [meters]', 'fontsize', 12, 'fontweight', 'b');
title(['Soundspeed vs Depth vs Time (' secname '.mat)']);
colorbar(); % create colorbar
caxis([1500 1540]);
% % Writer(s): Michelle Kingsland
% % Last Updated: 10/07/2009
% % Project: TAVEX 2008
% %
% % USAGE:  [c] = unesco(S,T,P)
% %  S  -  salinity  [psu]
% %  T  -  temperature  [degrees C]
% %  P  -  pressure  [dbar]
% %
% % DESCRIPTION:
% % A function to calculate sound speed using the UNESCO,
% % Chen and Millero (1977) formula.
% %
% % USAGE:  [c] = unesco(S,T,P)
% %  S  -  salinity  [psu]
% %  T  -  temperature  [degrees C]
% %  P  -  pressure  [dbar]

function y = unesco(S,T,P)
P=P./10; %convert dbar to kPa

C00 = 1402.388;
A02 = 7.166E-5;
C01 = 5.03830;
A03 = 2.008E-6;
C02 = -5.81090E-2;
A04 = -3.21E-8;
C03 = 3.3432E-4;
A10 = 9.4742E-5;
C04 = -1.47797E-6;
A11 = -1.2583E-5;
C05 = 3.1419E-9;
A12 = -6.4928E-8;
C10 = 0.153563;
A13 = 1.0515E-8;
C11 = 6.8999E-4;
A14 = -2.0142E-10;
C12 = -8.1829E-6;
A20 = -3.9064E-7;
C13 = 1.3632E-7;
A21 = 9.1061E-9;
C14 = -6.1260E-10;
A22 = -1.6009E-10;
C20 = 3.1260E-5;
A23 = 7.994E-12;
C21 = -1.7111E-6;
A30 = 1.100E-10;
C22 = 2.5986E-8;
A31 = 6.651E-12;
C23 = -2.5353E-10;
A32 = -3.391E-13;
C24 = 1.0415E-12;
B00 = -1.922E-2;
C30 = -9.7729E-9;
B01 = -4.42E-5;
C31 = 3.8513E-10;
B10 = 7.3637E-5;
C32 = -2.3654E-12;
B11 = 1.7950E-7;
A00 = 1.389;
D00 = 1.727E-3;
A01 = -1.262E-2;
D10 = -7.9836E-6;

\[
C_w = (C_{00} + C_{01}T + C_{02}T^2 + C_{03}T^3 + C_{04}T^4 + C_{05}T^5) + \ldots
\]
\[
(C_{10} + C_{11}T + C_{12}T^2 + C_{13}T^3 + C_{14}T^4) \cdot P + \ldots
\]
\[
(C_{20} + C_{21}T + C_{22}T^2 + C_{23}T^3 + C_{24}T^4) \cdot P^2 + \ldots
\]
\[
(C_{30} + C_{31}T + C_{32}T^2) \cdot P^3;
\]

A = (A_{00} + A_{01}T + A_{02}T^2 + A_{03}T^3 + A_{04}T^4) + \ldots
\[
(A_{10} + A_{11}T + A_{12}T^2 + A_{13}T^3 + A_{14}T^4) \cdot P + \ldots
\]
\[
(A_{20} + A_{21}T + A_{22}T^2 + A_{23}T^3) \cdot P^2 + \ldots
\]
\[
(A_{30} + A_{31}T + A_{32}T^2) \cdot P^3;
\]

B = B_{00} + B_{01}T + (B_{10} + B_{11}T) \cdot P;

d = D_{00} + D_{10} \cdot P;

y = C_w + A \cdot S + B \cdot S^2 \cdot (3/2) + d \cdot S^2;

function S = salinity(C,T,P)

% function computes the salinity from conductivity measurements
% in the 1978 Practical Salinity Scale algorithm [Perkin and
% Lewis].
% units: Conductivity [mmho/cm], Temperature [C], Pressure [dbar]
% units: Salinity [parts per thousand]

R = C/42.914;
A1 = 2.070e-5;
\[ A2 = -6.370 \times 10^{-10}; \]
\[ A3 = 3.989 \times 10^{-15}; \]
\[ B1 = 3.426 \times 10^{-2}; \]
\[ B2 = 4.464 \times 10^{-4}; \]
\[ B3 = 4.215 \times 10^{-1}; \]
\[ B4 = -3.107 \times 10^{-3}; \]
\[ RP = 1 + \left( P \cdot (A1 + A2 \cdot P + A3 \cdot P^2) \right) / \left( 1 + B1 \cdot T + B2 \cdot T^2 + \ldots + B3 \cdot R + B4 \cdot R \cdot T \right); \]
\[ c0 = 6.776097 \times 10^{-1}; \]
\[ c1 = 2.005640 \times 10^{-2}; \]
\[ c2 = 1.104259 \times 10^{-4}; \]
\[ c3 = -6.969800 \times 10^{-7}; \]
\[ c4 = 1.003100 \times 10^{-9}; \]
\[ rT = c0 + c1 \cdot T + c2 \cdot T^2 + c3 \cdot T^3 + c4 \cdot T^4; \]
\[ RT = R / (RP \cdot rT); \]
\[ a = [0.0080 -0.1692 25.3851 14.0941 -7.0261 2.7081]; \]
\[ b = [0.0005 -0.0056 -0.0066 -0.0375 0.0636 -0.0144]; \]
\[ S1 = 0; \]
\[ S2 = 0; \]
\[ k = 0.0162; \]
\[ \text{for } j = 1:6 \]
\[ S1 = S1 + a(j) \cdot RT^{((j-1)/2)}; \]
\[ S2 = S2 + b(j) \cdot RT^{((j-1)/2)}; \]
\[ \text{end} \]
\[ S = S1 + (T-15) / (1 + k \cdot (T-15)) \cdot S2; \]
Appendix B

Range-dependent Acoustic Model (RAM)

B.1 Modified FORTRAN Code

The PE model used in this work is Michael Collin’s Range-dependent Acoustic Model (RAM) [74]. The original FORTRAN code is available via anonymous ftp from the Naval Research Laboratory’s Acoustics Division website [81]. The code found below is the source code used to build the RAM executable file for the current work. The original RAM source (ram1.5.f) has been modified to output complex pressure rather than only TL. This file was compiled using the GNU FORTRAN 95/2003 (GFortran) compiler, along with the GNU Emacs text editor.

```fortran
program ram
  
  c
  c  ******************************************************************
  c  ***** Range-dependent Acoustic Model, Version 1.5, 13-Sep-00 *****
  c  ******************************************************************
  
  c This code was developed by Michael D. Collins at the Naval
  c Research Laboratory in Washington, DC. It solves range-dependent
  c ocean acoustics problems with the split-step Pade algorithm
  c user’s guide and updates of the code are available via anonymous
```
ftp from ram.nrl.navy.mil.

Version 1.5 contains a correction to a bug in the dimension of quantities passed to subroutines fndrt and guerre that Laurie Fialkowski noticed.

Version 1.4 contains a correction to a minor bug in subroutine guerre that Dave King noticed (amp1 and amp2 were declared twice) and a few other minor improvements.

Version 1.3 contains a new root-finding subroutine.

Version 1.2 contains a minor modification. The output to tl.grid is no longer zeroed out along the ocean bottom. This was done in previous versions so that the ocean bottom would be highlighted in graphical displays. The graphics codes ramclr, ramctr, and ramcc read in the bathymetry from ram.in and plot the ocean bottom directly.

Version 1.1 contains two improvements:

(1) An improved self starter. Stability is improved by using the factor (1-X)**2 instead of (1+X)**2 to smooth the delta function. The factor (1+X)**2 is nearly singular for some problems involving deep water and/or weak attenuation. Numerical problems associated with this singularity were detected by Eddie Scheer of Woods Hole Oceanographic Institute.

(2) Elimination of underflow problems. A very small number is added to the solution in subroutine solve to prevent underflow, which can adversely affect run time on some computers. This improvement was suggested by Ed McDonald of the SACLANT Undersea Research Centre.

complex ci,ksq,ksqb,u,v,r1,r2,r3,s1,s2,s3,pd1,pd2
real k0, ksqw

mr=bathymetry points, mz=depth grid, mp=pade terms.

parameter (mr=100, mz=8000, mp=10)
dimension rb(mr), zb(mr), cw(mz), cb(mz), rhob(mz), attn(mz), alpw(mz),
  alpb(mz), f1(mz), f2(mz), f3(mz), ksq(mz), ksqw(mz), ksqb(mz), u(mz),
  v(mz), tlg(mz), r1(mz, mp), r2(mz, mp), r3(mz, mp), s1(mz, mp),
  s2(mz, mp), s3(mz, mp), pd1(mp), pd2(mp)

open(unit=1, status='old', file='ram.in')
open(unit=2, status='unknown', file='tl.line')
open(unit=3, status='unknown', file='cmpxP.grid', form='unformatted')

call setup(mr, mz, nz, mp, np, ns, mdr, ndr, ndz, iz, nzplt, lz, ib, ir, dir, dr,
  dz, pi, eta, eps, omega, rmax, c0, k0, ci, r, rp, rs, rb, zb, cw, cb, rhob,
  attn, alpw, alpb, ksq, ksqw, ksqb, f1, f2, f3, u, v, r1, r2, r3, s1, s2, s3,
  pd1, pd2, tlg)

c
March the acoustic field out in range.

c
1 call updat(mr, mz, nz, mp, np, iz, ib, dr, dz, eta, eps, omega, rmax, c0, k0, ci, r,
  rp, rs, rb, zb, cw, cb, rhob, attn, alpw, alpb, ksq, ksqw, ksqb, f1, f2, f3,
  r1, r2, r3, s1, s2, s3, pd1, pd2)
call solve(mz, nz, mp, np, iz, u, v, r1, r2, r3, s1, s2, s3)
  r=r+dr
call outpt(mz, mdr, ndr, ndz, iz, nzplt, lz, ir, dir, eps, r, f3, u, tlg)
if(r.lt.rmax) go to 1

c
  close(1)
close(2)
close(3)

c
stop
end
Initialize the parameters, acoustic field, and matrices.

subroutine setup(mr,mz,nz,mp,np,ns,mdr,ndr,ndz,iz,nzplt,lz,ib,ir,
  dir,dr,dz,pi,eta,eps,omega,rmax,c0,k0,ci,r,rp,rs,rb,zb,cw,cb,
  rhob,attn,alpw,alpb,ksq,ksqw,ksqb,f1,f2,f3,u,v,r1,r2,r3,s1,s2,
  s3,pd1,pd2,tlg)
  complex ci,u(mz),v(mz),ksq(mz),ksqb(mz),r1(mz,mp),r2(mz,mp),
  r3(mz,mp),s1(mz,mp),s2(mz,mp),s3(mz,mp),pd1(mp),pd2(mp)
  real k0,rb(mr),zb(mr),cw(mz),cb(mz),rhob(mz),attn(mz),alpw(mz),
  alpb(mz),f1(mz),f2(mz),f3(mz),ksqw(mz),tlg(mz)

read(1,*)
read(1,*)freq,zs,zr
read(1,*)rmax,dr,ndr
read(1,*)zmax,dz,ndz,zmplt
read(1,*)c0,np,ns,rs

i=1
1 read(1,*)rb(i),zb(i)
  if(rb(i).lt.0.0)go to 2
  i=i+1
  go to 1
2 rb(i)=2.0*rmax
  zb(i)=zb(i-1)

pi=4.0*atan(1.0)
ci=cmplx(0.0,1.0)
eta=1.0/(40.0*pi*alog10(exp(1.0)))
eps=1.0e-20
ib=1
mdr=0
r=dr
omega=2.0*pi*freq
ri=1.0+zr/dz
ir=ifix(ri)
dir=ri-float(ir)
k0=omega/c0
nz=zmax/dz-0.5
nzplt=zmplt/dz-0.5
z=zb(1)
iz=1.0+z/dz
iz=max(2,iz)
iz=min(nz,iz)
if(rs.lt.dr)rs=2.0*rmax

if(nz+2.gt.mz)then
  write(*,*)’ Need to increase parameter mz to ’,nz+2
  stop
end if
if(np.gt.mp)then
  write(*,*)’ Need to increase parameter mp to ’,np
  stop
end if
if(i.gt.mr)then
  write(*,*)’ Need to increase parameter mr to ’,i
  stop
end if

d3 j=1,mp
r3(1,j)=0.0
r1(nz+2,j)=0.0
3 continue
do 4 i=1,nz+2
  u(i)=0.0
  v(i)=0.0
4 continue
lz=0
do 5 i=ndz,nzplt,ndz
  lz=lz+1
5 continue
  write(3)lz

  The initial profiles and starting field.
  
call prof(mz,nz,ci,dz,omega,eta,omega,rmax,c0,k0,rp,cw,cb,rhob,attn,
>   alpw,alpb,ksqw,ksqb)
call selfs(mz,nz,mp,np,ns,iz,zs,dr,dz,pi,c0,k0,rhob,alpw,alpb,ksq,
>    ksqw,ksqb,f1,f2,f3,u,v,r1,r2,r3,s1,s2,s3,pd1,pd2)
call outpt(mz,mnr,ndr,ndz,iz,nzplt,lz,ir,dir,eps,r,f3,u,tlg)

  The propagation matrices.
  
call epade(mp,np,ns,1,k0,c0,dr,rd1,rd2)
call matrc(mz,nz,mp,np,iz,iz,dz,k0,rhob,alpw,alpb,ksq,ksqw,ksqb,
>   f1,f2,f3,r1,r2,r3,s1,s2,s3,dp1,dp2)

  return
end

  Set up the profiles.
  
subroutine prof(mz,nz,ci,dz,omega,eta,omega,rmax,c0,k0,rp,cw,cb,rhob,
>   attn,alpw,alpb,ksqw,ksqb)
complex ci,ksqb(mz)
real k0,cw(mz),cb(mz),rhob(mz),attn(mz),alpw(mz),alpb(mz),ksqw(mz)

  call zread(mz,nz,dz,cw)
call zread(mz,nz,dz,cb)
call zread(mz,nz,dz,rhob)
call zread(mz,nz,dz,attn)
rp=2.0*rmax
read(1,*,end=1)rp

  do 2 i=1,nz+2
ksqw(i)=(omega/cw(i))**2-k0**2
ksqb(i)=((omega/cb(i))*(1.0+ci*eta*attn(i)))**2-k0**2
alpw(i)=sqrt(cw(i)/c0)
alpb(i)=sqrt(rhob(i)*cb(i)/c0)
2 continue

c
return
end

c Profile reader and interpolator.
c
subroutine zread(mz,nz,dz,prof)
real prof(mz)
c
do 1 i=1,nz+2
prof(i)=-1.0
1 continue
read(1,*)zi,profi
prof(1)=profi
i=1.5+zi/dz
prof(i)=profi
iold=i
2 read(1,*)zi,profi
if(zi.lt.0.0)go to 3
i=1.5+zi/dz
if(i.eq.iold)i=i+1
prof(i)=profi
iold=i
go to 2
3 prof(nz+2)=prof(i)
i=1
j=1
4 i=i+1
if(prof(i).lt.0.0)go to 4
if(i-j.eq.1)go to 6
do 5 k=j+1,i-1
prof(k)=prof(j)+float(k-j)*(prof(i)-prof(j))/float(i-j)
5 continue
6 j=i
   if(j.lt.nz+2)go to 4
return
end

The tridiagonal matrices.

subroutine matrc(mz,nz,mp,np,iz,jz,dz,k0,rhob,alpw,alpb,ksq,ksqw,
>   ksqb,f1,f2,f3,r1,r2,r3,s1,s2,s3,pd1,pd2)
   complex d1,d2,d3,rfact,ksq(mz),ksqb(mz),r1(mz,mp),r2(mz,mp),
   >   r3(mz,mp),s1(mz,mp),s2(mz,mp),s3(mz,mp),pd1(mp),pd2(mp)
   real k0,rhob(mz),f1(mz),f2(mz),f3(mz),alpw(mz),alpb(mz),ksqw(mz)
   c
   a1=k0**2/6.0
   a2=2.0*k0**2/3.0
   a3=k0**2/6.0
   cfact=0.5/dz**2
   dfact=1.0/12.0
   c
   New matrices when iz.eq.jz.
   c
   if(iz.eq.jz)then
      i1=2
      i2=nz+1
      do 1 i=1,iz
         f1(i)=1.0/alpw(i)
         f2(i)=1.0
         f3(i)=alpw(i)
         ksq(i)=ksqw(i)
      1 continue
      do 2 i=iz+1,nz+2
\begin{verbatim}
f1(i)=rhob(i)/alpb(i)
f2(i)=1.0/rhob(i)
f3(i)=alpb(i)
ksq(i)=ksqb(i)

2 continue
end if

c
Updated matrices when iz.ne.jz.
c
if(iz.gt.jz)then
i1=jz
i2=iz+1
do 3 i=jz+1,iz
f1(i)=1.0/alpw(i)
f2(i)=1.0
f3(i)=alpw(i)
ksq(i)=ksqw(i)
3 continue
end if

c
if(iz.lt.jz)then
i1=iz
i2=jz+1
do 4 i=iz+1,jz
f1(i)=rhob(i)/alpb(i)
f2(i)=1.0/rhob(i)
f3(i)=alpb(i)
ksq(i)=ksqb(i)
4 continue
end if

c
do 6 i=i1,i2

c
Discretization by Galerkin’s method.
c
\end{verbatim}
\begin{verbatim}
c1 = cfact * f1(i) * (f2(i-1) + f2(i)) * f3(i-1)
c2 = -cfact * f1(i) * (f2(i-1) + 2.0 * f2(i) + f2(i+1)) * f3(i)
c3 = cfact * f1(i) * (f2(i) + f2(i+1)) * f3(i+1)
d1 = c1 + dfact * (ksq(i-1) + ksq(i))
d2 = c2 + dfact * (ksq(i-1) + 6.0 * ksq(i) + ksq(i+1))
d3 = c3 + dfact * (ksq(i) + ksq(i+1))

do 5 j=1,np
  r1(i,j) = a1 + pd2(j) * d1
  r2(i,j) = a2 + pd2(j) * d2
  r3(i,j) = a3 + pd2(j) * d3
  s1(i,j) = a1 + pd1(j) * d1
  s2(i,j) = a2 + pd1(j) * d2
  s3(i,j) = a3 + pd1(j) * d3
5 continue
6 continue

c The matrix decomposition.
c
do 9 j=1,np
do 7 i=i1,iz
  rfact = 1.0 / (r2(i,j) - r1(i,j) * r3(i-1,j))
  r1(i,j) = r1(i,j) * rfact
  r3(i,j) = r3(i,j) * rfact
  s1(i,j) = s1(i,j) * rfact
  s2(i,j) = s2(i,j) * rfact
  s3(i,j) = s3(i,j) * rfact
7 continue

c
do 8 i=i2,iz+2,-1
  rfact = 1.0 / (r2(i,j) - r3(i,j) * r1(i+1,j))
  r1(i,j) = r1(i,j) * rfact
  r3(i,j) = r3(i,j) * rfact
  s1(i,j) = s1(i,j) * rfact
  s2(i,j) = s2(i,j) * rfact
end do
\end{verbatim}
s3(i,j)=s3(i,j)*rfact

8 continue

c
r2(iz+1,j)=r2(iz+1,j)-r1(iz+1,j)*r3(iz,j)
r2(iz+1,j)=r2(iz+1,j)-r3(iz+1,j)*r1(iz+2,j)
r2(iz+1,j)=1.0/r2(iz+1,j)

c
9 continue

c
return
end

c
The tridiagonal solver.

c
subroutine solve(mz,nz,mp,np,iz,u,v,r1,r2,r3,s1,s2,s3)
complex u(mz),v(mz),r1(mz,mp),r2(mz,mp),r3(mz,mp),s1(mz,mp),
> s2(mz,mp),s3(mz,mp)
eps=1.0e-30

do 6 j=1,np

c
The right side.

c
do 1 i=2,nz+1
v(i)=s1(i,j)*u(i-1)+s2(i,j)*u(i)+s3(i,j)*u(i+1)+eps
1 continue

c
The elimination steps.

c
do 2 i=3,iz
v(i)=v(i)-r1(i,j)*v(i-1)+eps
2 continue

do 3 i=nz,iz+2,-1
v(i)=v(i)-r3(i,j)*v(i+1)+eps
3 continue
u(iz+1) = (v(iz+1) - r1(iz+1,j)*v(iz) - r3(iz+1,j)*v(iz+2)) * r2(iz+1,j) + eps

The back substitution steps.

do 4 i=iz,2,-1
   u(i) = v(i) - r3(i,j)*u(i+1) + eps
4 continue

do 5 i=iz+2,nz+1
   u(i) = v(i) - r1(i,j)*u(i-1) + eps
5 continue
6 continue

return
end

Matrix updates.

subroutine updat(mr,mz,nz,mp,np,iz,ib,dr,dz,eta,omega,rmax,c0,k0,
   > ci,r,rp,rs,rb,zb,cw,cb,rhob,attn,alpw,alpb,ksq,ksqw,ksqb,f1,f2,
   > f3,r1,r2,r3,s1,s2,s3,pd1,pd2)
complex ci,ksq(mz),ksqb(mz),r1(mz,mp),r2(mz,mp),r3(mz,mp),
   > s1(mz,mp),s2(mz,mp),s3(mz,mp),pd1(mp),pd2(mp)
real k0,rb(mr),zb(mr),attn(mz),cb(mz),rhob(mz),cw(mz),ksqw(mz),
   > f1(mz),f2(mz),f3(mz),alpw(mz),alpb(mz)

Varying bathymetry.

if(r.ge.rb(ib+1)) ib = ib + 1
   jz = iz
   z = zb(ib) + (r + 0.5*dr - rb(ib)) * (zb(ib+1) - zb(ib)) / (rb(ib+1) - rb(ib))
   iz = 1.0 + z/dz
   iz = max(2,iz)
   iz = min(nz,iz)
if(iz.ne.jz) call matrc(mz,nz,mp,np,iz,jz,dz,k0,rhob,alpw,alpb,ksq,
>             ksqw,ksqb,f1,f2,f3,r1,r2,r3,s1,s2,s3,pd1,pd2)

c
Varying profiles.
c
if(r.ge.rp) then
    call prof1(mz,nz,ci,dz,eta,omega,rmax,c0,k0,rp,cw,cb,rhob,attn,
>              alpw,alpb,ksqw,ksqb)
    call matrc(mz,nz,mp,np,iz,iz,dz,k0,rhob,alpw,alpb,ksq,ksqw,ksqb,
>              f1,f2,f3,r1,r2,r3,s1,s2,s3,pd1,pd2)
end if

c  Turn off the stability constraints.
c
if(r.ge.rs) then
    ns=0
    rs=2.0*rmax
    call epade(mp,np,ns,1,k0,c0,dr,pd1,pd2)
    call matrc(mz,nz,mp,np,iz,iz,dz,k0,rhob,alpw,alpb,ksq,ksqw,ksqb,
>              f1,f2,f3,r1,r2,r3,s1,s2,s3,pd1,pd2)
end if

c  return
c  end

c  The self-starter.
c
subroutine selfs(mz,nz,mp,np,ns,iz,zs,dr,dz,pi,c0,k0,rhob,alpw,
>              alpb,ksq,ksqw,ksqb,f1,f2,f3,u,v,r1,r2,r3,s1,s2,s3,pd1,pd2)
complex u(mz),v(mz),ksq(mz),ksqb(mz),r1(mz,mp),r2(mz,mp),
>             r3(mz,mp),s1(mz,mp),s2(mz,mp),s3(mz,mp),pd1(mp),pd2(mp)
real k0,rhob(mz),alpw(mz),alpb(mz),f1(mz),f2(mz),f3(mz),ksqw(mz)

c  Conditions for the delta function.
si = 1.0 + zs/dz
is = ifix(si)
dis = si - float(is)
u(is) = (1.0 - dis) * sqrt(2.0 * pi / k0) / (dz * alpw(is))
u(is + 1) = dis * sqrt(2.0 * pi / k0) / (dz * alpw(is))

c

Divide the delta function by (1-X)**2 to get a smooth rhs.
c
pd1(1) = 0.0
pd2(1) = -1.0

call matrc(mz, nz, mp, 1, iz, iz, dz, k0, rhob, alpw, alpb, ksq, ksqw, ksqb, > f1, f2, f3, r1, r2, r3, s1, s2, s3, pd1, pd2)
call solve(mz, nz, mp, 1, iz, u, v, r1, r2, r3, s1, s2, s3)
call solve(mz, nz, mp, 1, iz, u, v, r1, r2, r3, s1, s2, s3)

c
Apply the operator (1-X)**2*(1+X)**(-1/4)*exp(ci*k0*r*sqrt(1+X)).
c
call epade(mp, np, ns, 2, k0, c0, dr, pd1, pd2)
call matrc(mz, nz, mp, np, iz, iz, dz, k0, rhob, alpw, alpb, ksq, ksqw, ksqb, > f1, f2, f3, r1, r2, r3, s1, s2, s3, pd1, pd2)
call solve(mz, nz, mp, np, iz, u, v, r1, r2, r3, s1, s2, s3)

c
return
c
c
Output transmission loss. (modified by Chad to output complex pressure)
c
subroutine outpt(mz, mdr, ndr, ndz, iz, nzplt, lz, ir, dir, eps, r, f3, u, tlg)
complex ur, u(mz), tlg(mz)
real f3(mz)

c
c  ur = (1.0 - dir) * f3(ir) * u(ir) + dir * f3(ir + 1) * u(ir + 1)
c  tl = -20.0 * alog10(cabs(ur) + eps) + 10.0 * alog10(r + eps)
c  write(2, *) r, tl
```fortran
                 mdr=mdr+1
                if(mdr.eq.ndr)then
                   mdr=0
                c
                j=0
                do 1 i=ndz,nzplt,ndz
                    ur=u(i)*f3(i)
                    j=j+1
                c
                tlg(j)=-20.0*alog10(cabs(ur)+eps)+10.0*alog10(r+eps)
                tlg(j)=ur
                1 continue
                write(3)(REAL(tlg(j)),j=1,lz)
                write(3)(IMAG(tlg(j)),j=1,lz)
                end if
                c
                return
            end
            c
            c The coefficients of the rational approximation.
            c
            subroutine epade(mp,np,ns,ip,k0,c0,dr,pd1,pd2)
            c
            implicit real*8 (a-h,o-z)
            complex*16 ci,z1,z2,g,dg,dh1,dh2,dh3,a,b
            complex*8 pd1(mp),pd2(mp)
            real*8 nu
            real*4 k0,c0,dr
            parameter (m=40)
            dimension bin(m,m),a(m,m),b(m),dg(m),dh1(m),dh2(m),dh3(m),fact(m)
            pi=4.0d0*datan(1.0d0)
            ci=dcmplx(0.0d0,1.0d0)
            sig=k0*dr
            n=2*np
            c
```
if(ip.eq.1)then
  nu=0.0d0
  alp=0.0d0
else
  nu=1.0d0
  alp=-0.25d0
end if

  The factorials.
  fact(1)=1.0d0
  do 1 i=2,n
    fact(i)=dfloat(i)*fact(i-1)
  1 continue

  The binomial coefficients.
  do 2 i=1,n+1
    bin(i,1)=1.0d0
    bin(i,i)=1.0d0
  2 continue
  do 4 i=3,n+1
    do 3 j=2,i-1
      bin(i,j)=bin(i-1,j-1)+bin(i-1,j)
    3 continue
  4 continue
  do 6 i=1,n
    do 5 j=1,n
      a(i,j)=0.0d0
    5 continue
  6 continue

  The accuracy constraints.
call deriv(m,n,sig,alp,dg,dh1,dh2,dh3,bin,nu)

c
do 7 i=1,n
b(i)=dg(i+1)
7 continue
do 9 i=1,n
if(2*i-1.le.n)a(i,2*i-1)=fact(i)
do 8 j=1,i
if(2*j.le.n)a(i,2*j)=-bin(i+1,j+1)*fact(j)*dg(i-j+1)
8 continue
9 continue

c
The stability constraints.
c
if(ns.ge.1)then
z1=-3.0d0
b(n)=-1.0d0
do 10 j=1,np
a(n,2*j-1)=z1**j
a(n,2*j)=0.0d0
10 continue
end if
c
if(ns.ge.2)then
z1=-1.5d0
b(n-1)=-1.0d0
do 11 j=1,np
a(n-1,2*j-1)=z1**j
a(n-1,2*j)=0.0d0
11 continue
end if
c
call gauss(m,n,a,b)
c
dh1(1)=1.0d0
do 12 j=1,np
  dh1(j+1)=b(2*j-1)
12 continue
  call fndrt(dh1,np,dh2,m)
do 13 j=1,np
  pd1(j)=-1.0d0/dh2(j)
13 continue

c
  dh1(1)=1.0d0
do 14 j=1,np
  dh1(j+1)=b(2*j)
14 continue
  call fndrt(dh1,np,dh2,m)
do 15 j=1,np
  pd2(j)=-1.0d0/dh2(j)
15 continue

c
  return
end

c  The operator function.
c
  function g(ci,sig,x,alp,nu)
    complex*16 ci,g
    real*8 alp,sig,x,nu
  g=(1.0d0-nu*x)**2*cdexp(alp*dlog(1.0d0+x)+
    ci*sig*(-1.0d0+dsqrt(1.0d0+x)))
  return
end

c  The derivatives of the operator function at x=0.
c
  subroutine deriv(m,n,sig,alp,dg,dh1,dh2,dh3,bin,nu)
    implicit real*8 (a-h,o-z)
    complex*16 ci,dg(m),dh1(m),dh2(m),dh3(m)
real*8 bin(m,m),nu


c
dh1(1)=0.5d0*ci*sig
exp1=-0.5d0
dh2(1)=alp
exp2=-1.0d0
dh3(1)=-2.0d0*nu
exp3=-1.0d0
do 1 i=2,n
dh1(i)=dh1(i-1)*exp1
exp1=exp1-1.0d0
dh2(i)=dh2(i-1)*exp2
exp2=exp2-1.0d0
dh3(i)=-nu*dh3(i-1)*exp3
exp3=exp3-1.0d0
1 continue
c
dg(1)=1.0d0
dg(2)=dh1(1)+dh2(1)+dh3(1)
do 3 i=2,n
dg(i+1)=dh1(i)+dh2(i)+dh3(i)
do 2 j=1,i-1
dg(i+1)=dg(i+1)+bin(i,j)*(dh1(j)+dh2(j)+dh3(j))*dg(i-j+1)
2 continue
3 continue
c
return
c
c
end
c
c
Gaussian elimination.
c
c
subroutine gauss(m,n,a,b)
implicit real*8 (a-h,o-z)
complex*16 a(m,m),b(m)
Downward elimination.

```
do 4 i=1,n
   if(i.lt.n) call pivot(m,n,i,a,b)
   a(i,i) = 1.0d0 / a(i,i)
   b(i) = b(i) * a(i,i)
   if(i.lt.n) then
      do 1 j=i+1,n
         a(i,j) = a(i,j) * a(i,i)
      1 continue
      do 3 k=i+1,n
         b(k) = b(k) - a(k,i) * b(i)
      2 continue
      3 continue
   end if
4 continue

Back substitution.

```
```
do 6 i=n-1,1,-1
   do 5 j=i+1,n
      b(i) = b(i) - a(i,j) * b(j)
   5 continue
6 continue

return
end

Rows are interchanged for stability.

subroutine pivot(m,n,i,a,b)
implicit real*8 (a-h,o-z)
complex*16 temp,a(m,m),b(m)

c
i0=i
amp0=cdabs(a(i,i))
do 1 j=i+1,n
amp=cdabs(a(j,i))
if(amp.gt.amp0)then
  i0=j
  amp0=amp
end if
1 continue
if(i0.eq.i)return
c
temp=b(i)
b(i)=b(i0)
b(i0)=temp
do 2 j=i,n
  temp=a(i,j)
  a(i,j)=a(i0,j)
  a(i0,j)=temp
2 continue
c
return
c
The root-finding subroutine.
c
subroutine fndrt(a,n,z,m)
  complex*16 a(m),z(m),root
  real*8 err
c
  if(n.eq.1)then
    z(1)=-a(1)/a(2)
  return
  end if
if(n.eq.2)go to 4

do 3 k=n,3,-1

Obtain an approximate root.

root=0.0d0
err=1.0d-12
call guerre(a,k,m,root,err,1000)

Refine the root by iterating five more times.

err=0.0d0
call guerre(a,k,m,root,err,5)
z(k)=root

Divide out the factor (z-root).

do 1 i=k,1,-1
a(i)=a(i)+root*a(i+1)
1 continue
do 2 i=1,k
a(i)=a(i+1)
2 continue

3 continue

Solve the quadratic equation.

4 z(2)=0.5*(-a(2)+sqrt(a(2)**2-4.0*a(1)*a(3)))/a(3)
z(1)=0.5*(-a(2)-sqrt(a(2)**2-4.0*a(1)*a(3)))/a(3)

return
end
This subroutine finds a root of a polynomial of degree \( n > 2 \) by Laguerre's method.

```fortran
subroutine guerre(a,n,m,z,err,nter)
complex*16 a(m),az(50),azz(50),z,dz,p,pz,pzz,f,g,h,c
real*8 amp1,amp2,rn,eps,err
ci=cmlx(0.0d0,1.0d0)
eps=1.0d-20
rn=real(n)

c
The coefficients of \( p'(z) \) and \( p''(z) \).

do 1 i=1,n
   az(i)=float(i)*a(i+1)
1 continue
do 2 i=1,n-1
   azz(i)=float(i)*az(i+1)
2 continue

c
iter=0
3 p=a(n)+a(n+1)*z
   do 4 i=n-1,1,-1
      p=a(i)+z*p
4 continue
   if(abs(p).lt.eps)return

c
   pz=az(n-1)+az(n)*z
   do 5 i=n-2,1,-1
      pz=az(i)+z*pz
5 continue

c
   pzz=azz(n-2)+azz(n-1)*z
   do 6 i=n-3,1,-1
      pzz=azz(i)+z*pzz
6 continue
```
The Laguerre perturbation.

\[ f = \frac{pz}{p} \]
\[ g = f^{**2} - \frac{pzz}{p} \]
\[ h = \sqrt{\left(\frac{rn - 1.0d0}{rn}\right) \left(\frac{rn * g - f^{**2}}{}\right)} \]
\[ \text{amp1} = \text{abs}(f + h) \]
\[ \text{amp2} = \text{abs}(f - h) \]
\[ \text{if}(\text{amp1} > \text{amp2}) \text{then} \]
\[ dz = -\frac{rn}{f + h} \]
\[ \text{else} \]
\[ dz = -\frac{rn}{f - h} \]
\[ \text{end if} \]

\[ \text{iter} = \text{iter} + 1 \]

Rotate by 90 degrees to avoid limit cycles.

\[ jter = jter + 1 \]
\[ \text{if}(jter \geq 10) \text{then} \]
\[ jter = 1 \]
\[ dz = dz * ci \]
\[ \text{end if} \]
\[ z = z + dz \]

\[ \text{if}(\text{iter} \geq 100) \text{then} \]
\[ \text{write}(\text{*},\text{*})' \]
\[ \text{write}(\text{*},\text{*})' \quad \text{Laguerre method not converging.'} \]
\[ \text{write}(\text{*},\text{*})' \quad \text{Try a different combination of DR and NP.'} \]
\[ \text{write}(\text{*},\text{*})' \]'
\[ \text{stop} \]
\[ \text{end if} \]

\[ \text{if}((\text{abs}(dz) \gt \text{err}) \text{.and.}(\text{iter} \lt \text{nter})) \text{go to 3} \]
return
end

\section*{B.2 Iterative 2D PE Model for MATLAB}

Since all CTD data was converted to MATLAB format data files, and because of the authors greater familiarity with the MATLAB computational language, MATLAB was used to interface with the RAM executable produced by compiling the FORTRAN code found in the prior section. An example of a 300 Hz test run can be found below. MATLAB was used to configure all environment files and parameters using make_ramin.m, and then the compiled FORTRAN code was ran as an executable file. After the RAM executable had finished running, read_tlgrid.m was used to read the complex pressure field from the RAM output *.grid file (the name of read_tlgrid.m was not changed after the FORTRAN code was modified to output complex pressure). To conserve hard drive space (the 2D output files can quickly eat up hard drive space if all are saved), only the complex pressure at the depth and range of the HLA are saved (with the exception of every fiftieth output file to check that the models are running correctly). The function IWfield.m is used to induce internal wave perturbations to the 2D water column. The version of IWfield.m found below is using the KdV analog internal wave model discussed in Chapter 3, but this function can be modified to perturb the water column using any method (such as replacing part of the water column with real data as discussed in Chapter 5).
passage of an internal wave field you must modify the
.m script "IWfield.m" to provide the desired changes to the
average 2D water column environment.

$close all
clear all
clc
disp('*** Memory Cleared ***')
disp(['*** Simulation started: ' datestr(clock) ' ***'])

load C:\RAMIcms\watercolumns\enviro300Hz_wav5and6_aveS.mat
depths=enviro.depths;
range=enviro.range;
dr=enviro.dr; % [m]
dz=enviro.dz; % [m]
clear enviro
testname='test30'; % name of this specific test run
start=1;
stop=1092;
index=33928/dr+1; % range of receiver ~33928[m]

load C:\RAMIcms\include\IwEigen.mat
IWmodes=interp1(Dpth,EigFnc(:,1:3),depths);
IWmodes(find(isnan(IWmodes)))=0.0;
eigfunc=IWmodes(:,1); % using only 1st mode
clear Bfrq2 Dpth EigFnc model

% loop to advance IWs (change these steps to change range)
for m=start:1:stop
    % get general environment
    load C:\RAMIcms\watercolumns\enviro300Hz_wav5and6_aveS.mat
soundspeed_matrix=IWfield(enviro.soundspeed_matrix,range,...
    depths,eigfunc,m);
clear enviro
soundspeed_matrix=single(soundspeed_matrix);

% set up watercolumn matrix to feed to RAM
WaterSoundSpeed=zeros(size(soundspeed_matrix,1),...
                      size(soundspeed_matrix,2)*2);
for n=1:length(range)
    WaterSoundSpeed(:,n*2-1)=depths;
    WaterSoundSpeed(:,n*2)=soundspeed_matrix(:,n);
end

% save a copy
if any(m==sel)
    save(['E:\' testname '\watercolumn' num2str(m) '.mat' ],...
         'soundspeed_matrix') % save watercolumn
end

clear soundspeed_matrix n enviro

disp('*** Environment Loaded ***)

%% create environment
runtitle='RAM Sim, 300Hz Average Water Column with IW Field';
freq=300.0; % [Hz]
SourceDepth=64.0; % [m]
RecieverDepth=72; % (for TL line but not really important) [m]
MaxRange=range(end); % [m]
% dr=0.5; % [m]
% dz=0.5; % [m]
MaxDepth=100; % [m](max depth of output including
% sediment or whatever)
c0=interp1(WaterSoundSpeed(:,1),WaterSoundSpeed(:,2),...
SourceDepth); % [m/s]
NumPadeTerms=5;
ns=1;
rs=0.0;
RangeDecimationFactor=1;
DepthDecimationFactor=1;
MaxDepthCompDepth=MaxDepth*3;%(4/3);

% computational domain bottom

bathymetry=[0 79;
    4503 72;
    7989 79;
    13484 72;
    16517 75;
    19321 74;
    26741 77;
    33928 73;
    34000 73]; % [meters]

% BottomSoundSpeed= [0 1500.0;
    20.1 1500.0]; %[m/s]
BottomSoundSpeed=[0 1500.0;
    2.4 1500.0;
    2.6 1580.0;
    11.9 1580.0;
    12.1 1615.0;
    20.0 1615.0;
    20.1 1800.0]; %[m/s]

% SedDensity= [0 1.0;
    20.1 1.0]; %[g/cc]
SedDensity= [0 1.45;
    2.4 1.45;
    2.6 1.85;
    11.9 1.85;
    12.1 1.95;
    20.0 1.95;
    20.1 2.2]; %[g/cc]
% bottom sound speed and density from "Preliminary results
% of short range propagation measurements using bulb sound
% sources" by Jee et al.
SedAttenuation=[ 0 0.2;
20.1 0.2;
MaxDepthCompDepth*.6 0.02;
MaxDepthCompDepth 20.0]; % absorption layer

% absorption coef. from "Geo-acoustic inversion using ship
% radiated noise in the TAVEX" by Seongil et al.

disp('*** Environment configured, writing RAM.in ***')

%% run RAM
make_ramin(runTitle,freq,SourceDepth,ReceiverDepth,...
MaxRange,dr,RangeDecimationFactor,MaxDepthCompDepth,...
dz,DepthDecimationFactor,MaxDepth,c0,NumPadeTerms,ns,...
rs,bathymetry,WaterSoundSpeed,BottomSoundSpeed,...
SedDensity,SedAttenuation);
% create ram.in file

disp('*** RAM.in finished, running RAM1_5.exe ***')

system('ram1_5.exe');
% run RAM executable to operate on ram.in

disp('*** RAM1_5.exe has finished, reading output ***')

[complexP errid]=read_tllgrid(MaxRange,MaxDepth,dr,dz);  
% read RAM output

%% save

depthCP=complexP(:,r_index);
save(['E:\' testname '\depthCP_step' num2str(m) '.mat'],...
'depthCP')
save(['E:\' testname '\errid_step' num2str(m) '.mat'],'errid')

% save a copy of the whole water column and output
% once in a while

if any(m==sel)
    output=struct;
    output.complexP=complexP;
    output.maxrange=MaxRange;
    output.
output.dr = dr;
output.maxdepth = MaxDepth;
output.dz = dz;
output.err = errid;

save(['E:\' / output num2str(m) '.mat'], 'output') % save watercolumn

clear output

clear WaterSoundSpeed

disp(['*** Page ' num2str(m) ' finished @ ' datestr(clock) ' ...
', output saved ***'])
end

disp(['*** Simulation finished: ' datestr(clock) ' ***'])

%%%%%%% Writer(s): Chad Smith
%%%%%%% Last Updated: 08/06/2010.
%%%%%%% Project: TAVEX 2008
%%%%%%% Program Details: This function is where you need to set
%%%%%%% up the IW field that perturbs the average watercolumn
%%%%%%% environment.

function field=IWfield(field,range,depths,eigfunc,n)
% I want a dt of 40 seconds
% 0.94[m/s]*40[s]=37.6[m] range step size
% (21000+1000+490*2+1000)[m]/37.6[m]=638 range steps
eta0=4.5/2; % m
V=0.94; % [m/s]
Delta=265/2; % m
drange=40*0.94; %[s]*[m/s]
lambda=490;
% starting x-positions
\begin{verbatim}
x1=-1000+drange*(n-1);
x2=x1-lambda;
x3=x2-lambda;
x4=x3-lambda;
x5=x4-lambda;
x6=x5-lambda;
depth_projection=single(nan(size(field))); for n=1:size(field,1)

    vert_displacement=... 
    eigfunc(n)*2*eta0*sech((x1-range)/\Delta).^2+...
    eigfunc(n)*2*eta0*sech((x2-range)/\Delta).^2+...
    eigfunc(n)*2*eta0*sech((x3-range)/\Delta).^2; %+
    eigfunc(n)*2*eta0*sech((x4-range)/\Delta).^2; %+
    eigfunc(n)*2*eta0*sech((x5-range)/\Delta).^2; %+
    eigfunc(n)*2*eta0*sech((x6-range)/\Delta).^2;
    depth_projection(n,:)=depths(n)-vert_displacement;
end

clear vert_displacement x1 x2 x3 x4 x5 x6 n model 

  %% interpolate vertical displacements to sound speed displacements
field(:,1:round(end/2))=interp2(range(1:round(end/2)),depths,...
   field(:,1:round(end/2)),ones(size(depth_projection,1),1)*...
   range(1:round(end/2)),depth_projection(:,1:round(end/2)));
field(:,round(end/2)+1:end)=interp2(range(round(end/2)+1:end),...
   depths,field(:,round(end/2)+1:end),...
   ones(size(depth_projection,1),1)*range(round(end/2)+1:end),...
   depth_projection(:,round(end/2)+1:end));
clear depth_projection
\end{verbatim}

%%%%%% Writer(s): Chad Smith
%%%%%% Last Updated: 09/02/2009.
%%%%%% Project: TAVEX 2008
%%%%%% Program Details: This function sets up the ram.in file
%%%%%% for input to the raml_5.exe compiled FORTRAN code by
function make_ramin(title, freq, zs, zr, rmax, dr, ndr, zmax, dz, ndz, zmplt, c0, np, ns, rs, bath, water, bot, sden, attn)

% proper usage:
% make_ramin(title, freq, zs, zr, rmax, dr, ndr, zmax, dz, ndz, zmplt, c0, np, ns, rs, bath, water, bot, sden, attn)

% variables defined as follows (from ram.pdf):
% title=arbitrary string of characters
% freq=source frequency (Hz)
% zs=source depth (m)
% zr=receiver depth for tl.line (m)
% rmax=maximum range (m)
% dr=range step (m)
% ndr=range decimation factor for tl.grid (l=no decimation)
% zmax=maximum depth (m)
% dz=depth grid spacing (m)
% ndz=depth decimation factor for tl.grid (l=no decimation)
% zmplt=maximum depth of output to tl.grid
% c0=reference sound speed (m/s)
% np=number of terms in rational approximation
% ns=number of stability constraints 1 or 2
% rs=maximum range of stability constraints (m)
% rb=range of bathymetry point (m)
% zb=depth of bathymetry point (m)
% z=depth of profile point (m)
% cw=sound speed in water column (m/s)
% cb=sound speed in sediment (m/s)
% rhob=density in sediment (g/cc)
% attn=attenuation in sediment (dB/wavelength)
% rp=range of profile update (m)
% bath=bathymetry matrix
% water=water column sound speed matrix
% bot=sediment sound speed matrix
% sden=sediment density matrix
% attn=sediment attenuation matrix
fid=fopen('ram.in','W'); % cap W does not erase file each time

fprintf(fid,'%s\ntitle\n',title);

fprintf(fid,'%f %f %f',freq,zs,zr);
fprintf(fid,'\tfreq zs zr\n');

fprintf(fid,'%d %d %d',rmax,dr,ndr);
fprintf(fid,'\trmax dr ndr\n');

fprintf(fid,'%f %f %d',rmax,dr,ndr);
fprintf(fid,'\trmax dr ndr\n');

fprintf(fid,'%f %d %f',bath(1,1),bath(1,2));
fprintf(fid,'\tbath zb\n');

for m=2:size(bath,1)
    fprintf(fid,'%f %f\n',bath(m,1),bath(m,2));
end
fprintf(fid,'-1 -1\n'); % end bathymetry

%%% set up range dependent profiles
for b=1:rmax/dr+1
    n=(b-1)*2+1;

    % water column
    fprintf(fid,'%f %f',water(1,n),water(1,n+1));
    fprintf(fid,'\tz cw\n');
    for m=2:size(water,1)
        fprintf(fid,'%f %f\n',water(m,n),water(m,n+1));
    end
    fprintf(fid,'-1 -1\n'); % end water

    % bottom sound speed
    fprintf(fid,'%f %f',bot(1,n),bot(1,n+1));
fprintf(fid, '%f %f', bot(1,1), bot(1,1+1));
fprintf(fid, '\tz cb\n');
for m=2:size(bot,1)
    fprintf(fid, '%f %f\n', bot(m,n), bot(m,n+1));
end
fprintf(fid, '-1 -1\n');

% bottom density
% fprintf(fid, '%f %f', sden(1,n), sden(1,n+1));
fprintf(fid, '%f %f', sden(1,1), sden(1,1+1));
fprintf(fid, '\tz rhob\n');
for m=2:size(sden,1)
    fprintf(fid, '%f %f\n', sden(m,n), sden(m,n+1));
end
fprintf(fid, '-1 -1\n');

% bottom attenuation
% fprintf(fid, '%f %f', attn(1,n), attn(1,n+1));
fprintf(fid, '%f %f', attn(1,1), attn(1,1+1));
fprintf(fid, '\tz attn\n');
for m=2:size(attn,1)
    fprintf(fid, '%f %f\n', attn(m,n), attn(m,n+1));
end
fprintf(fid, '-1 -1\n');

rp=dr*b; % rp=range for next dependent set
% if(rp<=rmax && size(water,2)>2)
if (b>rmax/dr+1)
    fprintf(fid, '%f', rp);
    fprintf(fid, '\trp\n');
end
end
fclose(fid);
Program Details: % This function pulls complex pressure data from the grid file created by running ram1.5.exe (the compiled version of ram1.5.f written by Michael D. Collins at NRL). Slight modifications have been made to this code to output complex pressure instead of TL.

function [P errid]=read_tlgrid(rmax,zmplt,dr,dz)
% proper usage:
% [P err]=read_tlgrid(rmax,zmplt,dr,dz);
% $P=complex pressure$
% $rmax=max range$
% $zmplt=max depth in the output$
% $dr=range step$
% $dz=depth step$

try
fid = fopen('cmpxP.grid','r');
a=fread(fid,1,'int');
znum=fread(fid,1,'int');
b=fread(fid,1,'int');
if a\ne b; error('tlgrid read error 1'); end
P=nan(znum,rmax/dr);
for i = 1:rmax/dr
    c=fread(fid,1,'int');
    Preal=fread(fid,znum,'float');
    d=fread(fid,1,'int');
    e=fread(fid,1,'int');
    Pimag=fread(fid,znum,'float');
    f=fread(fid,1,'int');
    if c\nd; error('tlgrid read error 2'); end
end
if e\ne; error('tlgrid read error 2'); end

P(:,i)=Preal+1j*Pimag;

% if i==66661
%      keyboard
% end

fclose(fid);

errid='no error';
catch err
      errid = regexp(err.identifier, '(?<=)\w+$', 'match');
end
Appendix C

Supporting Analysis

C.1 Harmonic Variation Analysis of Individual Hydrophones

Only the average HVI of hydrophone #'s 33, 53, 75, and 96 were shown in Chapter 4 for the sake of brevity and simplicity in explanation. Requiring the reader to flip through eight different analysis plots compared to two that can easily fit on a single page and display the same information, would hinder the understanding of the material. The HVI of the averaged RMS CW pressure of each hydrophone at both 300 and 500 Hz is shown below in Figures C.1-C.8 for additional examples of the analysis method. Analysis parameters for these plots are equivalent to the parameters given in Chapter 4.
Figure C.1. HVI of acoustic pressure of 300Hz CW pulses at hydrophone #33. Color scale in dB.

Figure C.2. HVI of acoustic pressure of 300Hz CW pulses at hydrophone #53. Color scale in dB.
Figure C.3. HVI of acoustic pressure of 300Hz CW pulses at hydrophone #75. Color scale in dB.

Figure C.4. HVI of acoustic pressure of 300Hz CW pulses at hydrophone #96. Color scale in dB.
Figure C.5. HVI of acoustic pressure of 500Hz CW pulses at hydrophone #33. Color scale in dB.

Figure C.6. HVI of acoustic pressure of 500Hz CW pulses at hydrophone #53. Color scale in dB.
Figure C.7. HVI of acoustic pressure of 500Hz CW pulses at hydrophone #75. Color scale in dB.

Figure C.8. HVI of acoustic pressure of 500Hz CW pulses at hydrophone #96. Color scale in dB.
Bibliography


Vita
Chad M. Smith

Chad Mahlon Smith was born in Elmira, NY on May 19th, 1983 to Vicky Lynn Winnie and Alton Duane Smith. After receiving his high school diploma from Williamson Senior High School of Tioga, PA he went on to receive his Bachelors of Science from Pennsylvania College of Technology in Williamsport, PA. For two years he worked for QorTek Incorporated in Williamsport, PA as an electrical design engineer. Then, moving to State College, PA in July of 2008, he started his graduate work in Acoustics and Physical Oceanography at The Pennsylvania State University.