

# A study of Forward Models in Seismic Inversion

**Maria Nilsen**

Master of Science in Electronics  
Submission date: June 2007  
Supervisor: Åge Kristensen, IET



## Problem Description

A seismic inversion has a number of variables which can be changed and altered to obtain a good result. The forward model will have a big impact on the results of the inversion. Both the time spent on the inversion, and which parameters the inversion will be best suited to estimate will be determined by the choice of forward model.

This project aims at finding the strengths and weaknesses of a number of different forward models. This is done through a study of literature, and implementation of the models in an inversion program. The inversion is performed with synthetic data generated from a simple geoacoustic model.

Assignment given: 22. January 2007  
Supervisor: Åge Kristensen, IET



# Preface

This master's thesis is written to summarize the work done during my final semester at the master's program at the Department of Electronics and Telecommunications at the Norwegian University of Science and Technology (NTNU). A pre-study was performed in the fall of 2006, and some of the work from that project is included in this thesis.

I would like to thank my supervisor Åge Kristensen, and Statoil for computer access at the research center at Rotvoll. There are many people who deserve thanks after 5 years as a student, the department, fellow students, friends and family. The people I wish to give thanks to know who they are.

Thank you!

Trondheim, June 14, 2007

Maria Nilsen



# Abstract

Knowledge about the physical parameters of the seafloor is often important information. This master's thesis looks at seismic inversion to find these parameters. The choice of forward model is highly emphasised.

A seismic inversion has a number of variables which can be changed and altered to obtain a good result. The forward model will have a big impact on the results of the inversion. Both the time spent on the inversion, and which parameters the inversion will be best suited to estimate will be determined by the choice of forward model.

An inversion code written in Matlab by Fredrik Helland is used. It uses genetic algorithms as optimization, and OSIRIS as forward model. This code is expanded to deal with several forward models and seafloor geometries. Testing of the inversion code shows that all the forward models serves different purposes. The ray tracing model is still at a concept level, but should be usable in the future when it runs a bit faster and can deal with more than 3 layers. The dispersion method and the wave number integration method both work well and the results show that using a combination of them might be the best choice if all the geoaoustic parameters of the seafloor is sought.





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Inversion process</b>	<b>3</b>
2.1	Inversion model . . . . .	3
2.2	Observed data . . . . .	5
2.3	Geoacoustic model . . . . .	5
2.4	Optimization . . . . .	8
2.4.1	Genetic algorithms . . . . .	8
<b>3</b>	<b>Forward models</b>	<b>11</b>
3.1	Wave propagation . . . . .	11
3.1.1	Stress and strain . . . . .	11
3.1.2	Compressional and shear waves . . . . .	12
3.1.3	Wave propagation in layered media . . . . .	13
3.2	Wave number integration . . . . .	14
3.3	Ray tracing . . . . .	16
3.3.1	Introduction to ray tracing theory . . . . .	16
3.3.2	The PlaneRay program . . . . .	20

3.4	The Thomson-Haskell method . . . . .	22
3.4.1	Using the Thomson-Haskell method . . . . .	22
<b>4</b>	<b>Challenges with the Thomson-Haskell method</b>	<b>25</b>
4.1	Testing the Thomson-Haskell algorithm . . . . .	25
4.2	Discussion . . . . .	27
4.3	Conclusion . . . . .	28
<b>5</b>	<b>Effects of the forward models</b>	<b>29</b>
5.1	The Thomson-Haskell method . . . . .	29
5.1.1	Code based on Haskell's article . . . . .	29
5.1.2	Code made by Caiti . . . . .	31
5.2	Ray tracing . . . . .	33
5.3	OSIRIS . . . . .	35
5.4	5 layers . . . . .	37
5.4.1	Caiti . . . . .	38
5.4.2	OSIRIS . . . . .	39
<b>6</b>	<b>Conclusion</b>	<b>41</b>
<b>A</b>	<b>Variables</b>	<b>45</b>
A.1	The geological models . . . . .	45
A.2	Source - receiver geometries . . . . .	47
A.3	CASE I geological model in Haskell's article . . . . .	47
<b>B</b>	<b>Deriving the Thomson-Haskell method</b>	<b>49</b>
<b>C</b>	<b>Flow diagrams of the inversion code</b>	<b>55</b>

# Chapter 1

## Introduction

Information about the physical parameters of the seafloor is often important to find. But finding them without spending a lot of money on expensive and time consuming measurements is challenging. Seismic inversion is one method which can solve this challenge by using information from one geoacoustic measurement to find the seafloor's physical parameters.

Adapting the inversion process to the actual problem is the first and most important step of the inversion, otherwise the physical parameters might be impossible to find. Already when planning the measurements it is necessary to place the receivers in positions which favors the important parameters of the seismic inversion. Once the measured data is obtained the next step is to choose a forward model which produces realistic synthetic data which can be compared to the measured data. There are different forward models to choose from, this thesis has looked into a wave number integration method, a dispersion method and a ray tracing method.

Each forward model will have different qualities, some will be fast, and others will be better at estimating the sought parameters. By taking advantage of these qualities one hopes to be able to estimate the wanted parameters better than by just using one randomly picked model.

An overview of what inversion is, and how it works is given in Chapter 2. The theory behind the three forward models is given in Chapter 3. The Thomson-Haskell method has caused some challenges which are discussed in Chapter 4, before the inversion results are presented and discussed in Chapter 5.



# Chapter 2

## Inversion process

Inversion is figuratively speaking to turn a function or a problem upside down. In geoaoustic inversion this translates to estimating the physical parameters in a geoaoustic model from measured data. The physical parameters we wish to find is compressional wave speed, shear wave speed, density and layer thickness. The challenge of seismic inversion is that acoustic wave propagation is nonlinear, which leads to a number of possible solutions to the same problem.

### 2.1 Inversion model

The inversion problem is solved by splitting it into smaller and more manageable pieces (see Figure 2.1). To be able to perform an inversion we need information about source and receiver(s), and data measured at the receiver(s). The measured data is the basis for the inversion, and what is measured is the acoustic field. The data is used to check if a calculated acoustic field is similar to the measured acoustic field. If the calculated and measured field is similar, the parameters used to make the calculated field are assumed to be the same as the physical parameters in the area where the measurements are taken, and the object of the inversion is achieved.

A geoaoustic model is built on a basis of a priori knowledge about the measured area. An example of this is the largest depth we can obtain measured data from, which is given by the frequency of the transmitted signal. It will be a waste of time trying to estimate parameters in layers at a larger depth than this maximum depth. These kinds of limitations and some approximations makes it possible to produce a simple geoaoustic or geological model which can be used in our calculations.

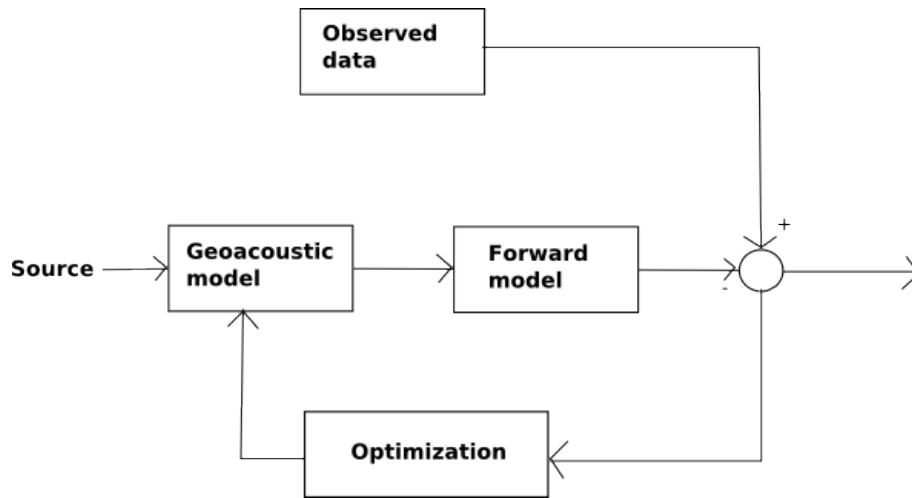


Figure 2.1: Model of the inversion system

The geoacoustic model is used in a forward model which calculates a synthetic acoustic field. There are many different forward models which can be used. In this masters thesis several forward models are looked into to find their strengths and weaknesses. The tested forward models are wave number integration, the program OSIRIS, a dispersion model, the Thomson-Haskell method, and a ray tracing model, PlaneRay.

The synthetic acoustic field generated by the forward model is compared to the observed data. If the observed and calculated data is close enough the acoustic model and the calculated data will be accepted as the right one, but if they are not close enough an optimization is performed before the geological model is changed and a new acoustic field is generated. What "close enough" means must be defined in advance, based on how good the algorithm is assumed to be, and the magnitude of noise on the measured data.

There are many different optimization algorithms to choose from. In this masters thesis a genetic algorithm is used. This algorithm was made by F. Helland in his masters' thesis [1].

The inversion is implemented in a Matlab program where most of the parameters and choices are made in graphical user interfaces. This is done to make the inversion easy to run, and to make sure it is not necessary to understand the complete code to be able to run it. The flow diagrams is given in Appendix C for those interested in the layout of the code.

## 2.2 Observed data

To test if the inversion actually works it is an advantage to have observed data with the least possible amount of noise and disturbance. Real data will never be without noise or uncertainties. Therefore synthetic data is used to insure a good result. The synthetic data is made from the same acoustic model which is used in the forward model.

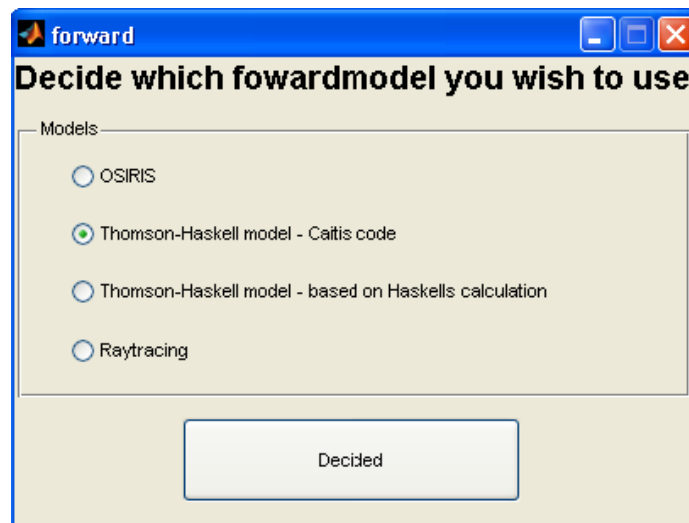


Figure 2.2: The different forward models the inversion program lets you choose from

Since the forward model is used to generate the synthetic data, the decision of which forward model to use has to be made at an early stage. Figure 2.2 shows the choices in the inversion program. Once the true geoacoustic model is chosen, a set of synthetic data to be used throughout the inversion is made.

## 2.3 Geoacoustic model

The same geological model is used in most of the calculations to enable us to compare the results when using different forward models. This geological model is the default model given by the inversion program, as shown in Figure 2.3 and A.1, and Tabular A.1. These values can of course be changed to suit other environments. OSIRIS and PlaneRay also need information about source, receiver and calculation information. To limit the choices, and make the program possible to use without being an expert on the forward models used, and sensible source-receiver geometries this information

is set in advance in the different parts of the program. This added information can be found in Appendix A.2.

Layer	Layer thickness	P-speed	S-speed	Density
Layer 1	50	1500	0	1000
Layer 2	50	1600	500	1600
Layer 3	50	1800	1000	2000
Layer 4	inf	2800	1500	2200
Layer 5	0	0	0	0

Figure 2.3: The default geological model given by the inversion program

Figure 2.3, or Tabular A.1 and Figure A.1, defines the actual geological model, the one we wish to find. Of course this information will not be available to us in advance, but it will be possible to limit the parameters a little by a priori knowledge and assumptions. In the inversion program the limitations can be set by the user. The density is chosen to be known in our calculations

Compressional speed<sup>1</sup>, P-speed, shear speed, S-speed, and layer thickness for each of the three layers are estimated in the inversion process. To lower the calculation time, and make the likelihood of finding the correct value bigger, a few assumptions are

<sup>1</sup>Ordinary notation is velocity, but velocity is a vector, and our value is a scalar, which means speed



limit

**Set the limitations for the parameter search**

Check the boxes of the variables you wish to estimate

Layer thickness     P-speed     S-speed     Density

	Lower limit	Upper limit	Number of bits	Lower limit	Upper limit	Number of bits	Lower limit	Upper limit	Number of bits	Lower limit	Upper limit	Number of bits
Layer 1	20	80	0	1300	1900	0	400	900	0	500	1500	0
Layer 2	20	80	7	1400	2000	9	400	900	9	800	2400	7
Layer 3	20	80	7	1650	2200	8	520	1120	8	1000	3000	7
Layer 4	Inf	Inf	0	2500	3200	8	1110	1710	8	1100	3300	7
Layer 5	0	0	0	0	0	0	0	0	0	0	0	0

Finished

Figure 2.4: Limitations on the geological parameters to make the inversion manageable

made about the magnitude of the variables. Some variables are also more important for a correct result than others. The search area is given a number of bits based on the importance of the parameter. The same number of bits is used in all the forward models to be able to compare the results, even though there is a difference in which parameters the forward models depends most on in there calculations. These parameters will be easier to estimate correct with the given forward model. Figure 2.4, or Tabular A.3, shows the limiting values used in most of the calculations.

OSIRIS has two equations which have to be fulfilled for the program to run.

$$\frac{V_s}{V_p} < \frac{\sqrt{3}}{2} \quad (2.1)$$

$$a_s < a_p \frac{3}{4} \left( \frac{V_p}{V_s} \right)^2 \quad (2.2)$$

The limitations in Equation 2.1 and 2.2 are based on physical limitations, and will be fulfilled in realistic environments, but they are important to take into consideration

when we are setting the limits of our inversion. Equation 2.2 is taken into consideration in the calculation of the limits in Figure 2.4, for the default geoaoustic model. But when changes in the model are made, it is important to make sure the values are still within the limits. Because if the limits we set exclude the actual value, the inversion will not work [2].

## 2.4 Optimization

The optimization algorithm used in this master thesis is a genetic algorithm programmed and tested in Fredrik Helland's master thesis [1]. No changes are made to the genetic algorithm, but it has been used together with other forward models than Helland originally used.

### 2.4.1 Genetic algorithms

A genetic algorithm (GA) is a numeric optimization algorithm which imitates the selection process of natural evolution, namely survival of the fittest [3]. The goal of the algorithm is to find the parameters which give the global minima or maxima of a given cost function.

GA is an algorithm which finds many solutions simultaneously through working with several parameter combinations at once. Or stated by the biological terminology used in GA: You work on a population of individuals simultaneously. The first generation is generated randomly within a given sample space defined by our a priori knowledge about the system. To decide which individuals, or solutions, we want to keep and use in the reproduction, the principle Survival of the Fittest is used. Different methods exist to choose the parents. They are all based on the principle that the individuals with the best cost functions, or fitness, will have the highest probability of being chosen as parents.

After the selection of parents, crossover and mutation is performed to create new individuals. Crossover is the first step on the evolution ladder where information from the two parents is mixed by switching part of the bit string between the two individuals behind a crossover point. The crossover points are found by statistical methods, which mean the physics of the system is not considered. The number of crossover points may vary. Figure 2.5 shows an example of single point crossover.

Mutation is performed after the crossover. The purpose of mutation is to stop the algorithm from locking in a local minima or maxima. Mutation is a change of value

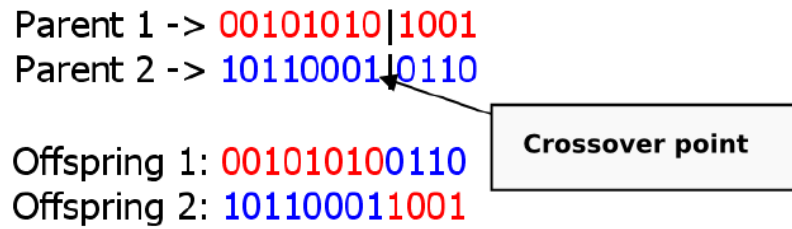


Figure 2.5: Single point crossover

on a randomly picked single bit in the individual. Mutation can in theory happen on several bits in each individual since it is performed uniformly on the entire population, with likelihood  $p_m$  [1]. Mutation can give large changes on the individual, which ensures that the parameter values ends in the global maxima or minima.

The new population which arises after the selection, crossover and mutation can either replace the entire old population, or some of the individuals with the best fitness can be kept. By keeping some of the individuals the total calculation time is shorter since the forward model already has been run for these parameter combinations. In addition the algorithm also becomes more robust when the best individuals from the old population is kept [2].

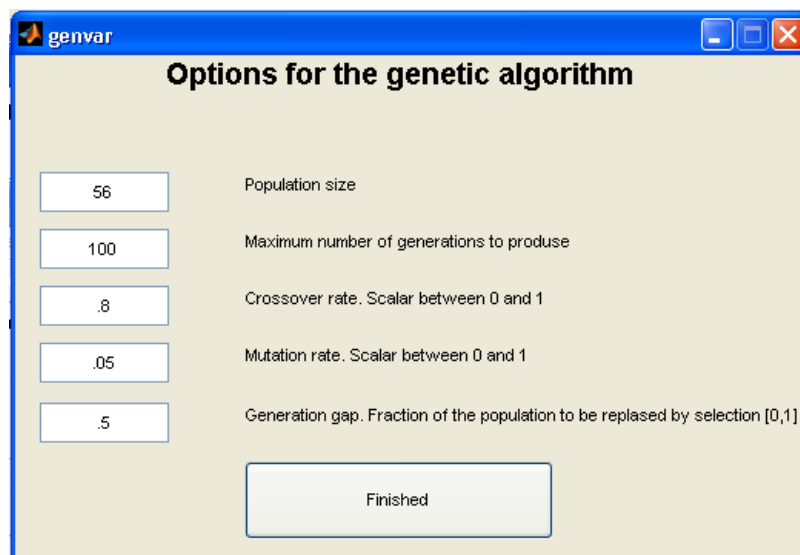


Figure 2.6: The variables which needs to be stated to perform the genetic algorithm

There are two ways of making an inversion stop, to set a threshold value the fitness has to become less than, or to set a maximum number of rounds the inversion is going

to run. The approach selected here is to run the inversion a given number of times, a maximum number of generations. The reason for choosing this approach is the fact that finding a threshold value we know will limit the inversion to the global minima is very difficult, and if we manage to find such a threshold value the inversion will become an infinite loop every time it fails to find the global minima. Choosing the max generations is not easy either, it is important to be sure the global minima is found, and that the code is stable before it is ended. The parameters in Figure 2.6 are chosen based on the recommendations made by Gerstoft in [2], with some small adjustments to fit our situation.

# Chapter 3

## Forward models

Wave propagation is the basis for all the forward models, therefore a small introduction to wave propagation is given before the theory behind the forward models are explained.

### 3.1 Wave propagation

Wave propagation is simply put the theory of how a wave will move in a medium.

#### 3.1.1 Stress and strain

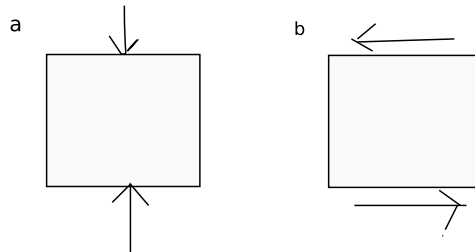


Figure 3.1: a) Normal and b) Shear stress on a particle

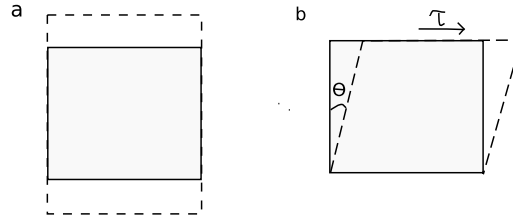


Figure 3.2: a) Normal and b) shear strain on a particle

Stress and strain is wave propagation when the particles in a wave are observed. Stress is force per unit surface, the driving force of change, while strain is a result of the force, a change in shape and/or size of the particle when a force is added.

There are two types of stress and strain, normal and shear stress or strain. In Figure 3.1 and 3.2 these two forces are shown. Longitudinal force is a force working normal on the particle, as in Figure 3.1a, and gives normal strain, as in Figure 3.2a. Shear force works parallel to the object, Figure 3.1b, and gives a shear strain, a rotational deformation of the object, Figure 3.2b.

In most practical situations we will never have a single particle, but a number of particles which are coupled. This coupling will create a chain reaction where the strain on one particle will be translated to stress on other particles. This chain reaction is the wave propagating.

### 3.1.2 Compressional and shear waves

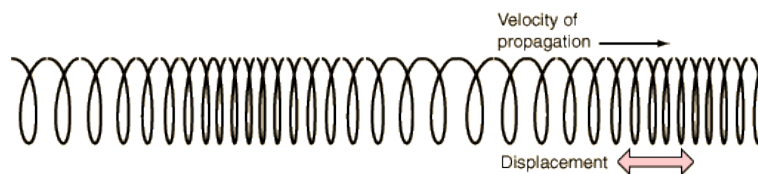


Figure 3.3: Wave propagation of a compressional wave

A wave propagating in a solid media will generate two waves, a compressional wave and a shear wave. The compressional wave, P-wave, is a longitudinal wave where the particles of the media moves in the same direction as the wave, see Figure 3.3. While

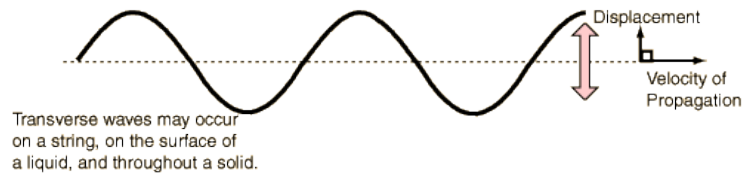


Figure 3.4: Wave propagation of a shear wave

the shear wave is a transversal wave where the particles move perpendicular to the wave direction, see Figure 3.4.

The compressional wave can move in solids, liquids and gasses, while the shear wave only exists in solids. The shear wave will move with a much lower speed than the compressional wave. Information about the amount of energy which is transferred from compressional to shear waves gives a lot of information about the physical properties of the seafloor. This means the shear speed is an important variable to find.

### Surface waves

The S- and P-waves will propagate as two separate waves in homogenous media. But in non-homogenous media, like the transition between sedimentary layers, or a water sediment boundary, there will be a connection between the S- and P-waves [4]. This will appear as a new wave, a surface wave, also called interface wave, with an elliptic movement. The wave will be strongly attenuated with distance to the interface. This means that the wave can be ignored in all other areas than around the interface, hence the name [5].

Surface waves are usually split into three types, Rayleigh waves, when you have an air-solid interface, Stonely waves, solid-solid interfaces, and Scholte waves, water-solid interfaces. Surface waves are the easiest mean of finding information about the shear properties of the seafloor. The Rayleigh wave phase speed is between 87% and 95% of the shear speed of a given layer. This makes the Rayleigh wave speed a good measurement of the shear speed [4].

### 3.1.3 Wave propagation in layered media

Calculation of acoustic waves in the sea bottom is a complex task. To be able to perform such calculations limiting assumptions are made. Variation only with depth,  $z$ , and infinite top and bottom layers are examples of such limitations [6].

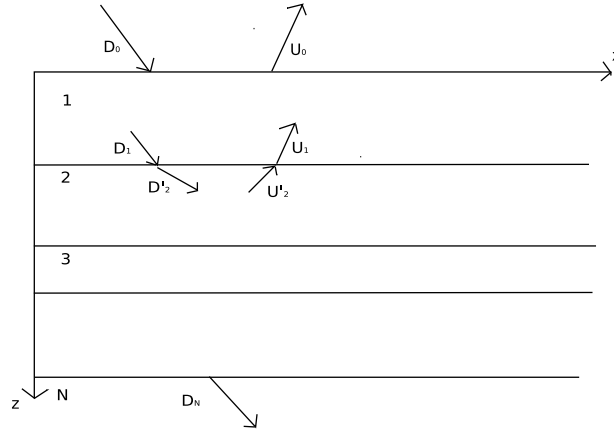


Figure 3.5: Horizontal stratification

The top layer  $D_0$ , in Figure 3.5, contains an incident wave which is reflected and transmitted downwards into the seabed.  $U_0$  is the sum of the reflected wave from the interface between layer 0 and 1, and all the single and multiple reflections in the deeper layers. There will be one downwards,  $D_n$ , and one upwards moving wave,  $U_n$  in all the layers. The exception is the bottom layer where you will have no reflection, and no upwards moving wave. In solids there will be two upwards and two downwards moving waves, namely one for the compressional and one for the shear wave [7].

There are physical limitations which have to be fulfilled at each interface. The displacement in the  $x$  and  $z$  direction has to be continuous, and the tangential and normal pressure has to be continuous [7]. Continuation in the displacement is achieved when the speed of the displacement is continuous [6].

## 3.2 Wave number integration

Hovem shows in [4] that the general definition of an acoustic wave in layered media can be expressed as an integral over horizontal wave numbers. The integrand contains the solution of the Helmholtz equation, also called the depth-separated wave equation. This integral has to be solved numerically, and the wave number integration technique is one of these numerical methods.

The wave number integral  $\phi(r, z)$  is given by



$$\phi(r, z) = \frac{\exp(-i\pi/4)}{\sqrt{2\pi r}} \int_0^\infty \Phi(k, z) \sqrt{k} \exp(ikr) dk, \quad (3.1)$$

when the far-field approximation of the Hankel functions are used.

The numerical solution to this problem is found by discretization of both the horizontal wave number,  $k$ , and the range,  $r$ .

$$k_q = k_{min} + q\Delta k, \quad q = 0, 1 \dots (N - 1), \quad (3.2)$$

$$r_p = r_{min} + p\Delta r, \quad p = 0, 1 \dots (N - 1), \quad (3.3)$$

where  $k_{min}$  and  $r_{min}$  are the minimum wave number and range,  $\Delta k$  and  $\Delta r$  are the sampling intervals and  $N$  is the number of sampling points. The maximum value  $k_{max}$  and  $r_{max}$  are of course given when  $q$  and  $p$  is equal to  $N$ .

The sampling intervals need to satisfy

$$\Delta k \cdot \Delta r = \frac{2\pi}{N}. \quad (3.4)$$

This leads to the final equation

$$\begin{aligned} \phi(r_p, z) \approx & \frac{\Delta k}{\sqrt{2\pi r_p}} \exp(ir_p k_{min}) \exp(-i\pi/4) \\ & \times \sum_{q=0}^{N-1} [\Phi(k_q, z) \exp(iq\Delta k r_{min}) \sqrt{k_q}] \exp(2\pi i \frac{pq}{N}). \end{aligned} \quad (3.5)$$

The summation term in Equation 3.5 is a Discrete Fourier Transform (DFT), which can be computed efficiently by the Fast Fourier Transform (FFT). This is why the method is also called a fast field method [4].

Equation 3.5 is an approximation, and will yield errors. These errors are interpreted as aliasing errors caused by the Fourier transform always being performed by assuming a repeated signal.

### 3.3 Ray tracing

The assumption that sound follows rays normal to surfaces with the same phase is the basis for ray acoustics. A point source in a medium with constant sound speed will form phase front surfaces that are concentric circles, and consequently the sound will follow straight paths out from the source. The complicating factor in ray acoustics is that with a non constant sound speed the sound rays will not be straight, but they will follow curved paths [7].

Ray tracing is a computation technique to calculate the coordinates of the sound rays emanating from the source. The approach can be derived from the wave equation by making some simplifying assumptions. These assumptions make the ray tracing a high-frequency approximation. It is only applicable to frequencies with wavelength considerably smaller than the characteristic distance of variation in sound speed [7].

#### 3.3.1 Introduction to ray tracing theory

One of the big assumptions made which covers many cases of practical interest is that the sound speed is only a function of depth  $z$ . This assumption leads to a differential equation

$$\frac{d\xi}{ds} = -\frac{1}{c^2} \frac{\partial c}{\partial r} = 0, \quad (3.6)$$

where the variable  $\xi$  is proportional to the horizontal wave number components,  $k$ ,  $c$  is the sound speed,  $s$  is the arc length and  $z$  the horizontal direction [7]. In this case  $\xi$  is constant and independent of  $r$ .  $\xi$  is given by Snell's law

$$\xi = \frac{\cos \theta(z)}{c(z)} = \frac{\cos \theta_s}{c_s}, \quad (3.7)$$

where  $\theta$  is the angle of the ray's trajectory at a given point, and  $\theta_s$  and  $c_s$  are the values at a point  $(r_s, z_s)$ .

Equation 3.7 means that  $\xi$  is a constant for all points on a given ray's trajectory. Therefore  $\xi$  is called the ray parameter [7].

The curves propagate along curved paths in accordance with Snell's law. This makes it possible to calculate the radius of the ray paths curvature,  $R$ . After some rewritings the radius at a given depth  $z$  is

$$R(z) = -\frac{1}{\xi g(z)}, \quad (3.8)$$

where  $g(z)$  is the gradient of the speed profile

$$g(z) = \frac{dc(z)}{dz}. \quad (3.9)$$

Equation 3.8 states that at any point in space, the curvature of the ray is given by the ray parameter and the local value of the sound speed gradient  $g(z)$  [7].

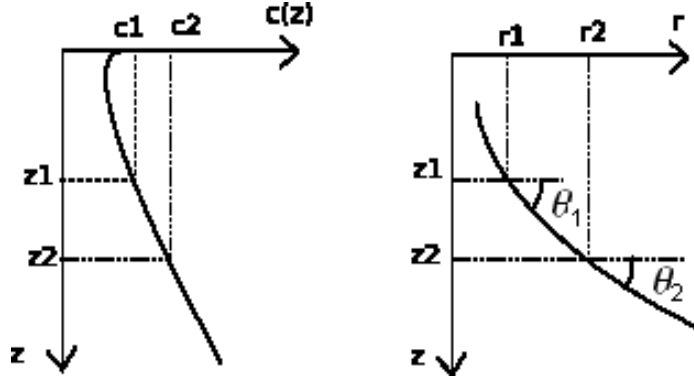


Figure 3.6: A sound speed profile (left) and a portion of a ray (right)

The coordinates and travel time of a ray at two single coordinates,  $(r_1, z_1)$  and  $(r_2, z_2)$  are important parameters to calculate, because they can be used to find a ray's travel time and the coordinates of the ray path [7]

$$r_2 - r_1 = \int_{z_1}^{z_2} \frac{dz}{\tan \theta(z)} = \int_{z_1}^{z_2} \frac{dz}{\sqrt{\left(\frac{c_1}{c(z) \cos \theta_1}\right)^2 - 1}} \quad (3.10)$$

$$\tau_2 - \tau_1 = \int_{z_1}^{z_2} \frac{ds}{c(s)} = \int_{z_1}^{z_2} \frac{dz}{c(z) \sqrt{1 - \xi^2 c^2(z)}}, \quad (3.11)$$

the relationship between  $(r_1, z_1)$ ,  $(r_2, z_2)$  and  $c_1$  are given in Figure 3.6.

### Constant sound speed gradient

The water column is considered divided into many thin layers and the sound speed profile in each layer approximated with straight lines. This is a valid approximation as long as the water column is divided into a significant number of layers. The sound speed profile within a given layer is then approximated by

$$c(z) = c_1 + g(z - z_1), \quad (3.12)$$

where  $c_1$  is the speed at depth  $z_1$ , as in Figure 3.6, and the sound speed gradient  $g$  can be both positive and negative [7].

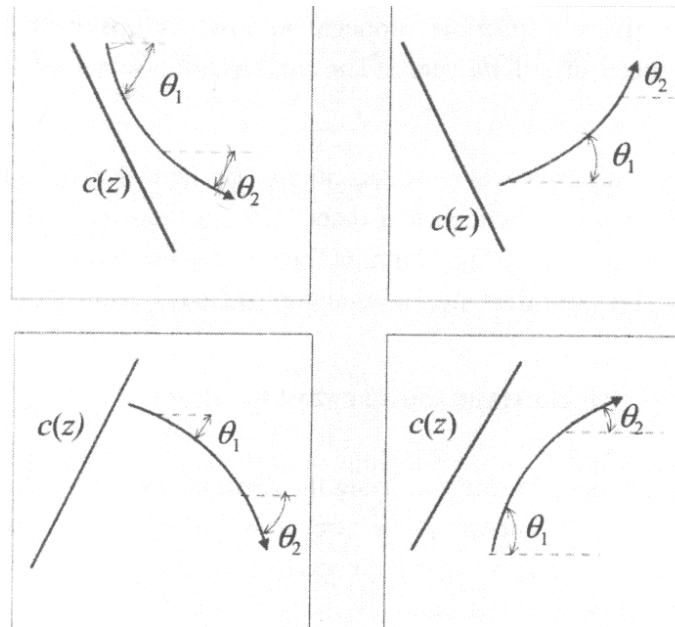


Figure 3.7: The four types of sound path curvature which can occur, depending upon the combination of the ray's positive or negative initial angle with a positive or negative sound speed gradient [7]

The radius of the ray curvature,  $R$ , is treated as positive regardless of the direction of the curvature in most practical situations [7]. This is to ensure we have a constant radius. The direction of the curvature can easily be obtained from the geometry of the speed profile, as Figure 3.7 shows. In the following formulas the sign for the direction is kept. This is done because it will have to be kept in the computer program which uses the ray tracing method.

A ray moving downwards through the layers will get a small change of direction within each layer it passes. As a ray moves through a layer  $i$ , from  $z_i$  to  $z_{i+1}$ , it will also move from  $r_i$  to  $r_{i+1}$ . The range increment formula is found by studying Figure 3.6

$$r_{i+1} - r_i = -R_i(\sin \theta_{i+1} - \sin \theta_i) = \frac{1}{\xi g_i(z)} \left[ \sqrt{1 - \xi^2 x^2(z_{i+1})} - \sqrt{1 - \xi^2 x^2(z_i)} \right]. \quad (3.13)$$

From Equation 3.12 it becomes evident that the local sound speed gradient within a layer can be approximated by

$$g_i = \frac{c(z_{i+1}) - c(z_i)}{z_{i+1} - z_i}. \quad (3.14)$$

The last formula which needs to be stated to be able to calculate the ray path is the travel time. By insertion and integration of Equation 3.11, combined with some re-writings by Snell's law the travel time can be represented by

$$\tau_{i+1} - \tau_i = \frac{1}{|g_i|} \ln \left( \frac{c(z_{i+1})}{c(z_i)} \frac{1 + \sqrt{1 - \xi^2 x^2(z_i)}}{1 - \xi^2 x^2(z_{i+1})} \right). \quad (3.15)$$

Equation 3.13 and 3.15 is used repeatedly in the ray tracing algorithm, for each step  $\Delta z$ , where  $\Delta z$  is given as

$$z_{i+1} = z_i \pm \Delta z. \quad (3.16)$$

The plus sign in Equation 3.16 indicates downward movement, and the minus a ray going upwards. The sign of course changes when the ray hits the bottom, the surface, or a turning point [8].

Acoustic intensity can be calculated using the principle that the power within a space limited by a pair of rays will remain between the two rays, regardless of the rays' paths [7]. This is illustrated in Figure 3.8.

$$I(r) = I_0 \frac{r_0^2 \cos \theta_0}{r \sin \theta} \left| \frac{d\theta_0}{dr} \right| = I_0 \frac{r_0^2 c_0}{r c} \left| \frac{d\theta_0}{dr} \right|, \quad (3.17)$$

where  $I_0$  is the intensity at  $r = r_0$ , the reference distance.

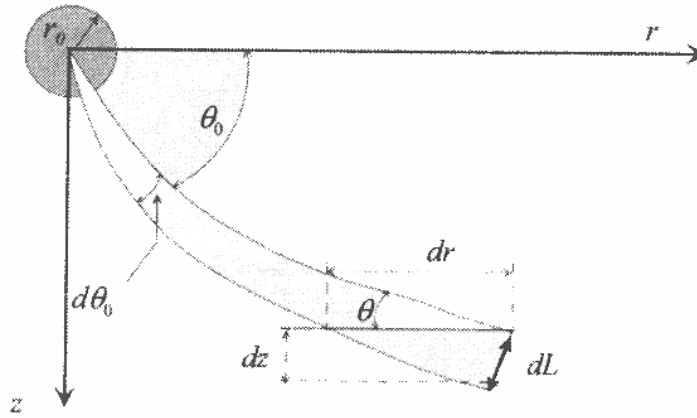


Figure 3.8: The principle of intensity calculations [7]

In the inversion we are to compare results from the forward model, in this case the ray tracing model, with measured data. The variable which has been found best suited for our purpose is the transmission loss. This means that the output from the ray tracing algorithm is the transmission loss, which is found from

$$TL = -10 \log(I/I_0) = 10 \log(r/r_0^2) + 10 \log \left| \frac{d\theta_0}{dr} \right| + 10 \log(c/c_0). \quad (3.18)$$

### 3.3.2 The PlaneRay program

PlaneRay is an acoustic ray tracing program written by Jens M. Hovem [8]. It can treat range dependent environments, and in theory a number of bottom layers can be specified, but the model is still only implemented to deal with one fluid sediment layer and an elastic half space in addition to the water column.

#### Model description

The model can be considered as having three stages. The first is the initial ray tracing using a large number of rays to map the entire sound field. The second step is a unique sorting and interpolation routine to determine the trajectories and the ray history of the eigenrays connecting the source to the receivers. The final third step is to make a synthesis of the acoustic field in frequency domain by coherently adding the

contributions of the eigenrays, and calculation of the full-waveform time responses by Fourier transform [8].

**Initial ray tracing** To start the ray tracing information about the range dependent bathymetry, a sound profile, the source location, and the receiver depth is needed. The ray tracing is started by launching a large number of rays with angles selected to cover the entire space between a fixed source, and out to receiver on a horizontal line at the specified receiver depth. The model computes the ranges and travel times to the locations where the ray intersects the receiver depth, and saves this information for each ray. The location and angle of the bottom and surface reflections are also stored for the following stages [8].

In PlaneRay the rays are only traced in the water column, not in the sediments [8]. By assuming a known water depth it will be possible to change the sediment parameters without solving the initial ray tracing every time. This is a big advantage because the initial ray tracing is the slowest part of the process.

The theory and equations for the ray tracing is given in Section 3.3.1.

**Sorting and interpolation** In this step the eigenrays and their trajectories are determined. This is done by interpolating the results from the initial ray tracing. The interpolation is done on rays with the same type of ray history. Where ray history means the number of reflections [8].

**Synthesis of the sound field** The received sound field is synthesized by coherently adding the contributions from the eigenrays and the bottom reflections. The rays are not traced into the bottom, and a layered bottom is described by the plane ray reflection coefficients. In theory the sediments can be modelled with variable number of layers, but in the current implementation only a fluid sedimentary layer over a homogenous solid half space is implemented. This is a major limitation compared to the other forward models which includes many layers. It should be possible to include a variable number of layers and find the reflection coefficient for the given layers. But because of the time limit this has not been implemented in this master's thesis.

### **Adaptation of the PlaneRay program**

The PlaneRay program had to undergo some cosmetic changes to fit into the inversion program which already existed. The only output from the program which is needed

is the transmission loss. This makes it quite easy to alter the code into a function with the geological parameters as input. Most of the parameters in PlaneRay are set at different places in the code, and it was therefore important to check thoroughly that no parameters were overwritten at later stages in the code by preset parameters.

The last change was a possibility of choosing not to run the initial ray tracing by setting one of the input parameters 1 or 0. This was as mentioned earlier an important step in making the code run faster by skipping the slowest stage.

## 3.4 The Thomson-Haskell method

The Thomson-Haskell method (THM) is a matrix method which can be used to find dispersion of surface waves. In a geoacoustic environment with  $n$  layers, as in Figure 3.5, there will be  $4n - 2$  boundary conditions when the top and bottom layer are assumed infinitely thick. As mentioned in Chapter 3.1.3 the boundary conditions are continuity of the two displacement components, and the two pressure components at each interface, and the pressure component must disappear on the free surface. This leads to  $4n - 2$  equations and unknowns, which needs to be solved. The only possible solution to this problem is if the determinant of the coefficients becomes zero [6]. When the equations are placed in a matrix as Haskell suggests in [6], it is easier to treat an environment with several layers because a systematic calculation method can be used.

The equations used in THM are given in Appendix B. The mathematics are not essential to understanding the method, but they are essential in the programming of THM, and understanding them have been a big part of the work in this masters thesis. For the people with a special interest in the mathematics in Haskell's article reading Appendix B is recommended [6].

### 3.4.1 Using the Thomson-Haskell method

The purpose of the Thomson-Haskell method is to get a dispersion diagram. A plot with speed along the y-axis, and frequency along the x-axis. The unknown variables are speed,  $c$ , and frequency,  $\omega = 2\pi f$ .  $\omega$  is not used by Haskell [6], but the wave number  $k = \omega/c$  is used. To find speed and wave number is equivalent to finding speed and frequency because of the simple transition between these variables.

The unknown variables  $c$  and  $k$  has to be found through a process of trial and error. Haskell gives a few pointers on how to do this in the most efficient way [6]. The article



is written in 1953, which means the pointers are meant to save time when calculating by hand, but most of them are applicable also as limiting factors when we are using numerical methods on a computer.

The calculations become complex when they are hand calculated with a higher number of layers than two. But with the computational power of today's computers, additional layers are a manageable task. The most important factors to limit are  $c$  and  $k$ . According to Haskell the speed is limited by the Rayleigh speed of the first and  $n$ 'th layer [6]. The Rayleigh wave speed is between 87% and 95% of the shear wave speed in a given layer [4]. The wave number can often be limited by knowledge about similar environments. Then the search area is limited to a more manageable size.

The actual implementation of THM has imposed some difficulties. The reasons for this are discussed in chapter 4.



# Chapter 4

## Challenges with the Thomson-Haskell method based on Haskell's article

### 4.1 Testing the Thomson-Haskell algorithm

To be certain that the programming of the Thomson-Haskell algorithm is working, a copy of a Fortran code made by Andrea Caiti has been obtained. The code by Andrea Caiti has been tested and found to be correct by Lars Ivar Leivestad in his master's thesis [9]. By comparing the results from the Caiti code with the results from the program based on Haskell's article it is possible to determine if the program gives a correct result.

An example of this is given in Figure 4.1 where the contour plot shows the entire sample space given by the Thomson-Haskell algorithm. It is from this sample space the phase speed and wave number values are chosen. The red line gives the values found with the Thomson-Haskell algorithm, and the dotted pink line is the values from Caiti's code. The assumed crustal structures are given in Appendix A.3.

From Figure 4.1 it is evident that the two programs give comparable results. At high speeds and low wave numbers the difference between the two are quite small, but the difference grows bigger at lower speeds and higher wave numbers. A discussion of why these differences might appear will be given at a later stage.

By programming the Thomson-Haskell algorithm and testing it on the assumed structures given by Haskell in [6], it is possible to obtain a comparable result to the results

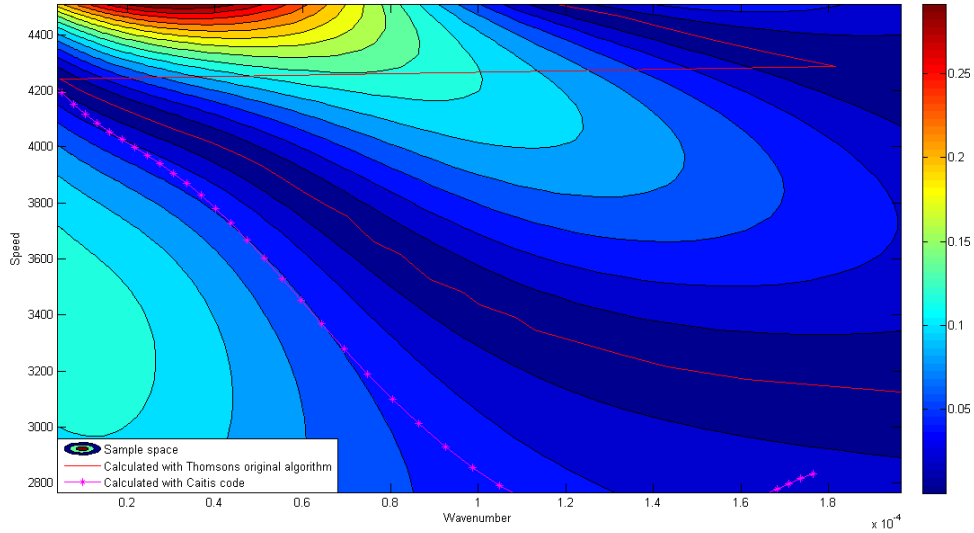


Figure 4.1: Phase speed of Rayleigh waves for assumed crustal structures given by Haskell's article [6]

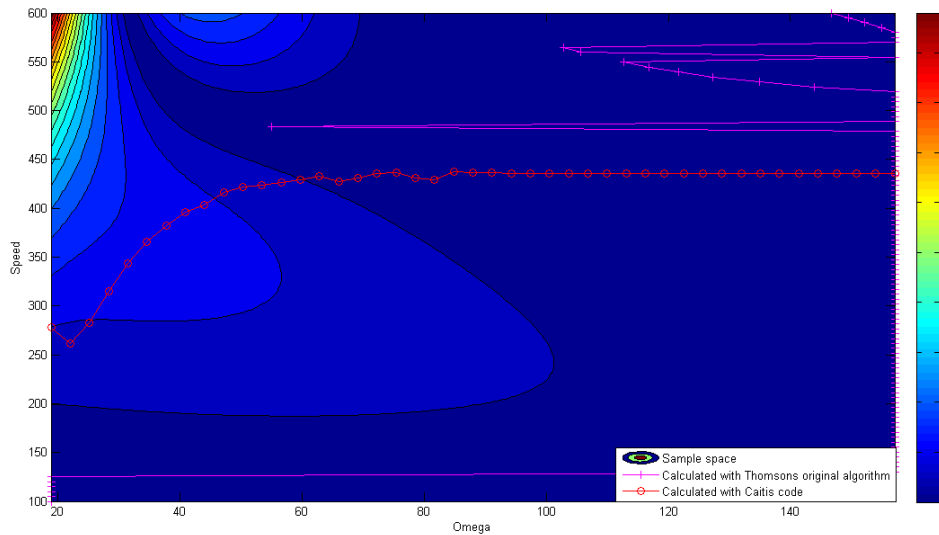


Figure 4.2: The synthetic geological model used in the inversion is used with the algorithm stated in Haskell's article [6]

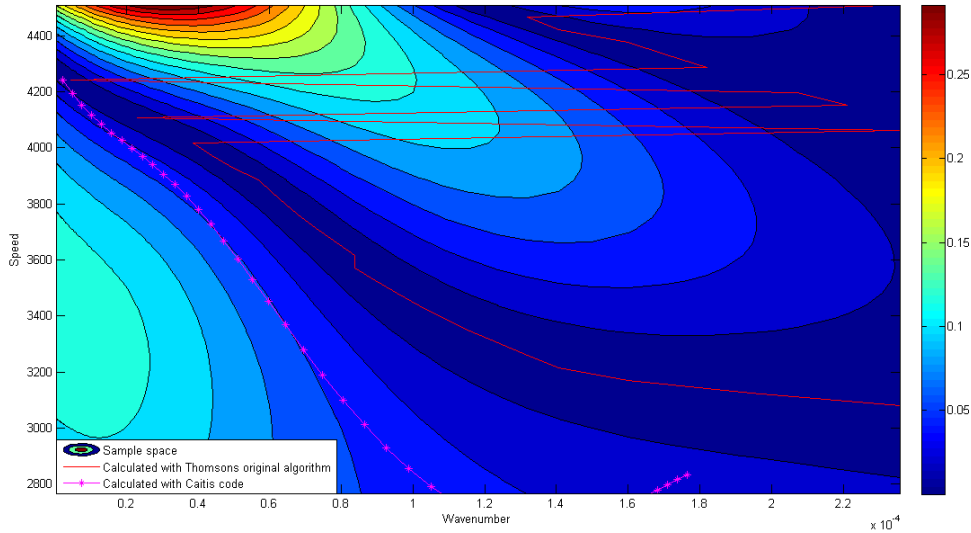


Figure 4.3: An unstable version of the same crustal structure as in Figure 4.1

given in the article [6]. This is shown in Figure 4.1. But it soon becomes obvious that the algorithm fails when dealing with higher frequencies. This is shown in Figure 4.2, where it is clear that finding the same variables as Caiti has found seems impossible with the Thomson-Haskell algorithm. These high frequency problems arise because of large exponential terms which become very large at high frequencies according to Dunkin [10] and Watson [11]. The exponentials will cancel in the final calculations, but they will cause a loss of significant information which can not be retrieved after it is lost.

Caiti claims to use the article by Haskell [6] when he was programming his code [12]. But from studying the code and comparing the results it becomes clear that he has to use some of the improvements authors like Dunkin [10], Watson [11] and others, have suggested to make the code stable also at higher frequencies.

## 4.2 Discussion

The sample space in Figure 4.1 has low magnitudes, which makes it hard to be 100% certain that the value chosen is the best one. A small numerical change can cause a big difference in the calculations. This might be what causes the difference in the values from the Caiti code and my code.

The only difference in calculation between Figure 4.1 and 4.3, is a larger set of wave numbers,  $k$ . The wave numbers has not been limited to the same degree in Figure 4.3, as they have in Figure 4.1.

From these figures it becomes obvious that a small difference in the limitation of the sample space can give a big difference in the result. In Figure 4.3 there are two areas where the equation becomes zero, which means the program has to choose between two different values which both could be correct. A computer will not be able to tell which of the area to choose from, but will usually choose the first one on it's list, or when there are minor differences in value because of numerical limitations, the one with the lowest value is picked, even if it is only  $10^{-10}$  lower. These numerical limitations are the main reason for the jumping between the two zero areas in Figure 4.3.

### 4.3 Conclusion

After several attempts at getting the algorithm presented by Haskell in his article [6] stable without any success for higher frequencies, the conclusion is that the later articles improving the algorithm might have a point [10], [11].

If one wishes to program the Thomson-Haskell method looking into the later articles on the subject is recommended to be able to make a stable code also for higher frequencies. This was discovered to late to be implemented in this thesis, but both the Caiti code and the programmed Thomson-Haskell algorithm is implemented in the inversion. Then there at least is one working dispersion method in the inversion program which is discussed in Chapter 5.

# Chapter 5

## The effect of the forward model on the inversion

The different forward models serve different purposes. When real data is used the forward model which is chosen has to give comparable results to the data collected. All the forward models have different properties, and which forward model you use has to be reflected in the setup of the measurements, and the initial processing of the measured data. The use of synthetic data ensures that the biggest differences between the models are the time consumption and which parameters the inversion is most likely to estimate correctly.

### 5.1 The Thomson-Haskell method

The inversion is implemented with both the Caiti code and the program based on Haskell's article. As we have already stated that the program based on Haskell's article is unstable for the high frequencies we are working with, it should be obvious that the results should not be the best when we use this forward model.

#### 5.1.1 Code based on Haskell's article

Figure 4.2 shows that most of the results are the extreme values of our search area. When we perform an inversion these results from the synthetic generated data will be compared with the data generated in the inversion, which also will get the extreme

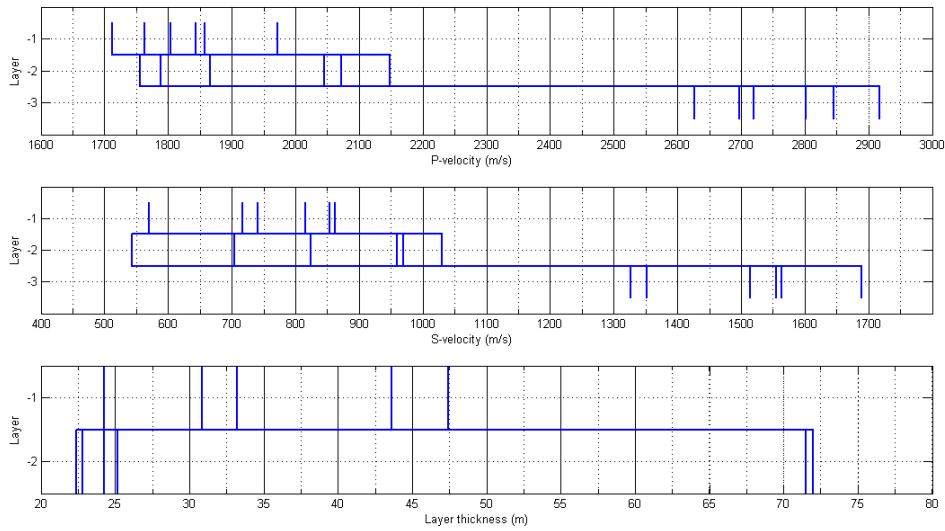


Figure 5.1: The best parameter set from the 6 best runs with the code based on Haskell's article

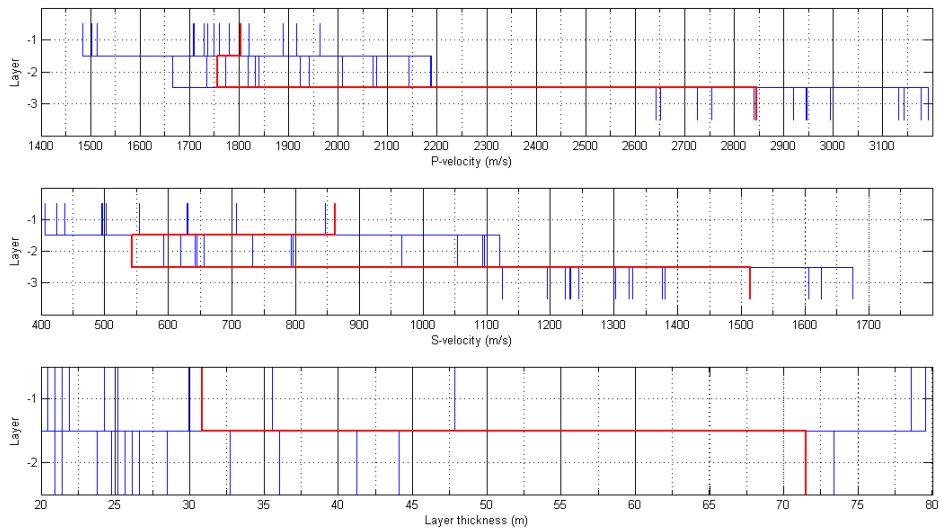


Figure 5.2: One of the "best" runs with the Haskell code as forward model



values for most of the parameter combinations chosen by the genetic algorithm. To avoid this problem the assumption that the inversion never will be able to find the true parameter combination is included when the Haskell forward model is used. The likelihood of getting all the parameters correct is slim because of both the statistic method used in the optimization and the fact that the discretization of the parameters might exclude the actual value. The assumption is implemented by giving the difference between the observed and generated data a new random value when it becomes zero.

From Figure 2.4 and Tabular A.3 we know the span of parameter values. In both Figure 5.1 and 5.2 it is clear that the results are randomly picked inside the limiting area. Within a single run the individuals will become more and more equal as time progresses. In Figure 5.2 it is evident that this has not happened. All the best individuals from the different runs are also scattered over a wide band of values, which is what was expected since the frequencies are too high for the program to work optimally.

### 5.1.2 Code made by Caiti

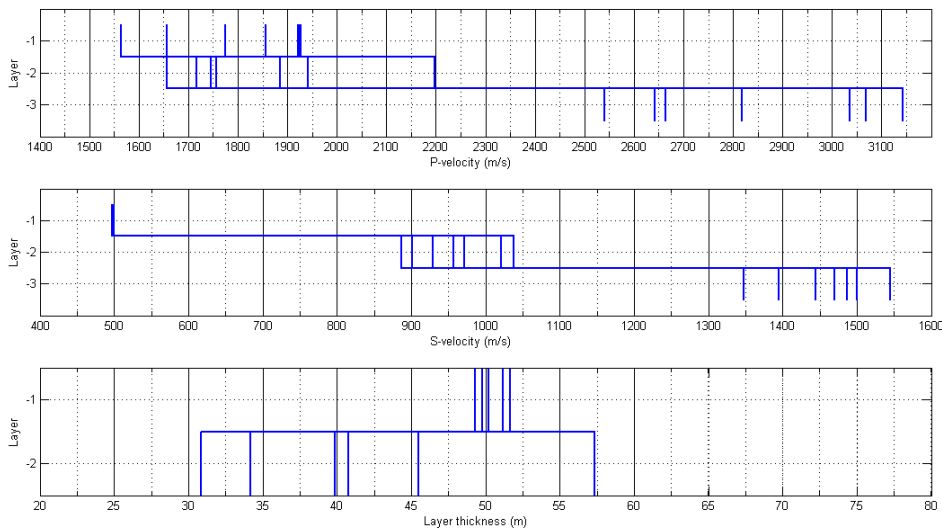


Figure 5.3: The best parameter set from the 10 best runs with Caiti's code as forward model

The Shear speed in Figure 5.4 is a good example of how the results look after a good run. In the top layer all the individuals has the same value. In the second layer

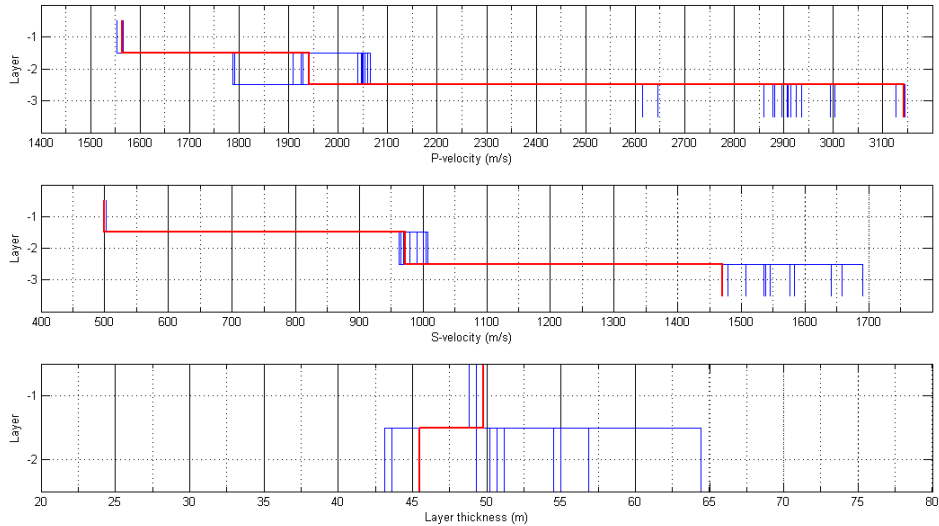


Figure 5.4: The best run with the Caiti code as forward model

the values are a bit scattered, but still close. While the third and final layer gets even more scattered values, but they are not spanning the entire search area. The reason for this is the top layer will give the biggest contributions, and of course the calculations in the forward model will put more emphasis on the top layers to mimic this effect. This makes the parameters of the top layers easier to find than the layers further down.

The best individual of the 10 best runs, Figure 5.3, also shows the same effect of limiting the possible parameter values, and estimating the top layers better than the bottom ones.

The time spent on the inversion with the Caiti code as the forward model is only 5.5 min, which means it is the fastest of the forward models tested.

Figure 5.4 has a good estimation of the compressional speed, but Figure 5.3 suggests that the compressional speed is hard to estimate with the Caiti code.

To test the use of the dispersion method to estimate the P-speed a series of runs where the S-speed is assumed known is performed. By comparing the results in Figure 5.5 with the limits in Figure 2.4 and Tabular A.3 it is evident that the P-speed estimations span the entire search area. This suggests that the Caiti code will not be able to estimate the P-speed correctly. This is logical when we remember that the surface waves have a low dependence on the compressional speed.

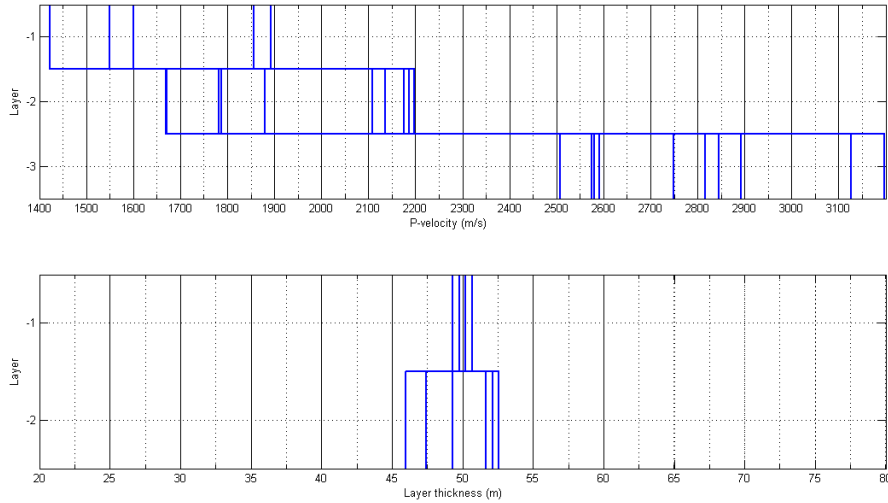


Figure 5.5: The 10 best individuals when only the thickness and P-speed is estimated with the Caiti code as forward model

## 5.2 Ray tracing

A few limitations are made to make the implementation and runtime as smooth as possible. As mentioned earlier there is a limitation on the number of layers given by the model, and the layer thickness has also been chosen as known to be able to reduce the runtime from one week to 9 hours. PlaneRay is made to look at what happens when we have a non constant sound speed profile. But to make the number of choices a minimum, a constant sound speed profile is used in this implementation. The number of reflections, number of receivers and their range is also set in advance after a set of trial and error runs. The emphasis when choosing was to find the combination which gave the lowest runtime together with a high enough resolution in the results.

Table 5.1: Geological model used to generate the synthetic data with ray tracing as forward model

	Depth	P-speed	S-speed	Density
	[m]	[m/s]	[m/s]	[kg/m <sup>3</sup> ]
<b>Layer 0</b>	0	1500	0	1000
<b>Layer 1</b>	50	1600	0	1600
<b>Layer 2</b>	100	2800	1500	2200

Table 5.2: The limitation of the inversion with PlaneRay as forward model

	<b>P-speed</b>		<b>S-speed</b>	
	[m/s]	Number of bits	[m/s]	Number of bits
Layer 1	1400-2000	512 (9 bit)	0-600	256 (8 bit)
Layer 2	2650-3200	256 (8 bit)	1210-1810	256 (8 bit)

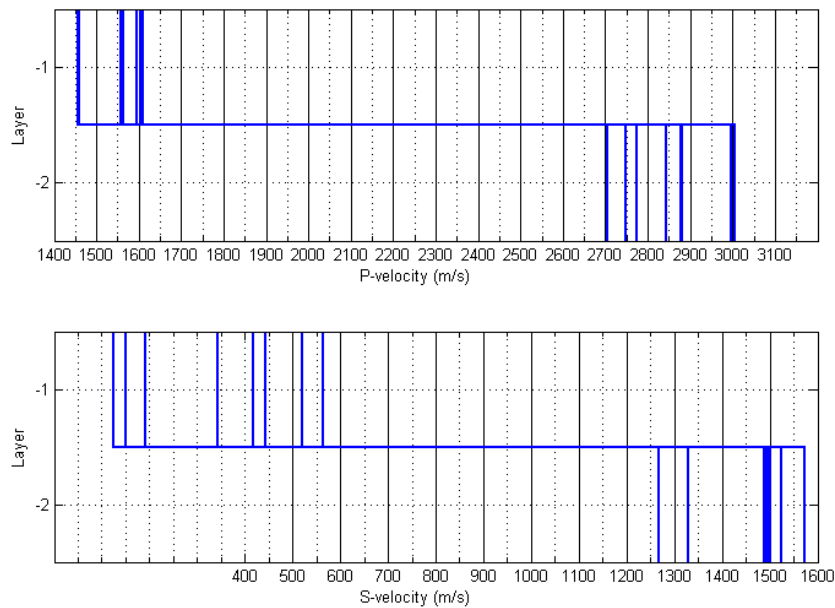


Figure 5.6: The best individual from the 8 best inversion runs with PlaneRay as forward model

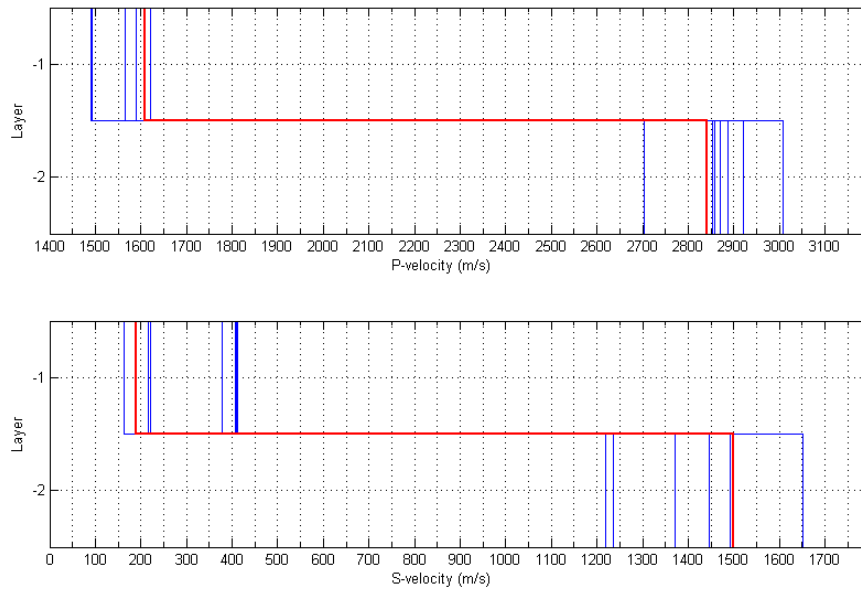


Figure 5.7: The best of all the runs using PlaneRay as the forward model

Comparing Figure 5.6 and 5.7 with Table 5.2 it is obvious that the results from the inversion is in a more limited band of values than the limiting values we start the inversion with.

It's important to remember that the shear speed of the top layer is not part of the calculations, and therefore can not be estimated. But the inversion program has no possibility to only estimate the values of some of the layers, and it has to be part of the optimization and the other steps of the inversion even if the forward model does not use it.

## 5.3 OSIRIS

The most obvious result obtained from looking at the best individual from the 10 best runs in Figure 5.8 is that the top and bottom layers seem to be estimated quite well, while the mid layer result is a bit more uncertain.

Figure 5.9 tells a slightly different story. Here it seems that the bottom layer causes the biggest estimation problem, while the two top layers seem to be estimated well. This might suggest that a larger number of runs might be needed, and that the

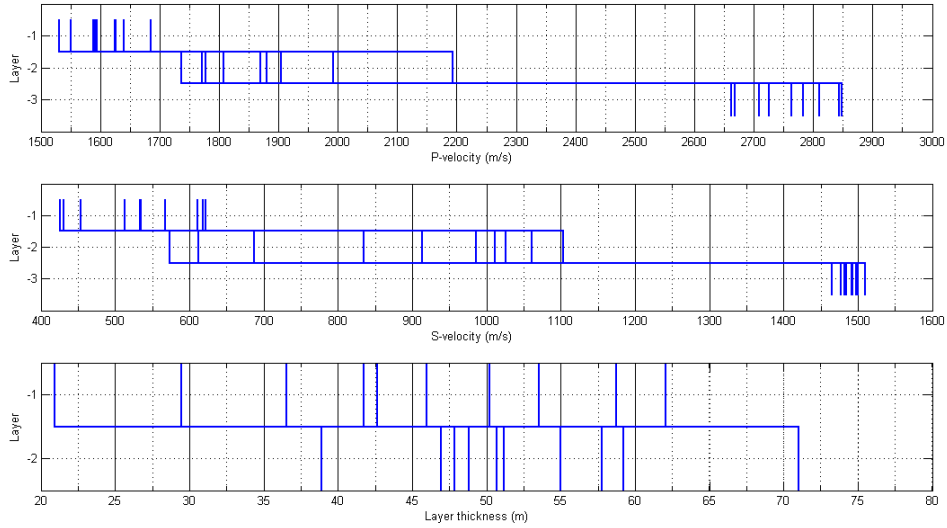


Figure 5.8: The best parameter set from the 10 best runs with OSIRIS as forward model

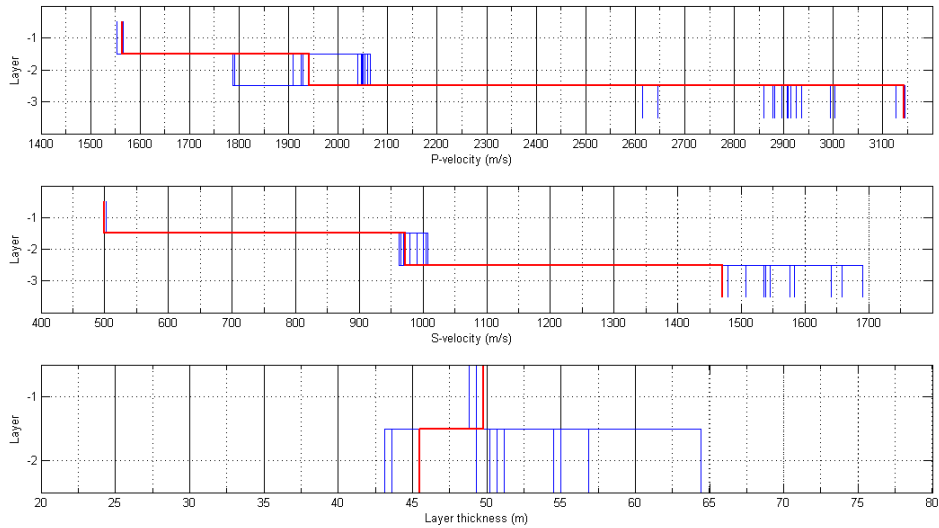


Figure 5.9: The best run with OSIRIS as forward model

parameter values should be decided after looking at several runs. Looking at more than one of the best individuals in several runs together might also be useful.

One run of the inversion with OSIRIS as forward model takes 2.5 hours.

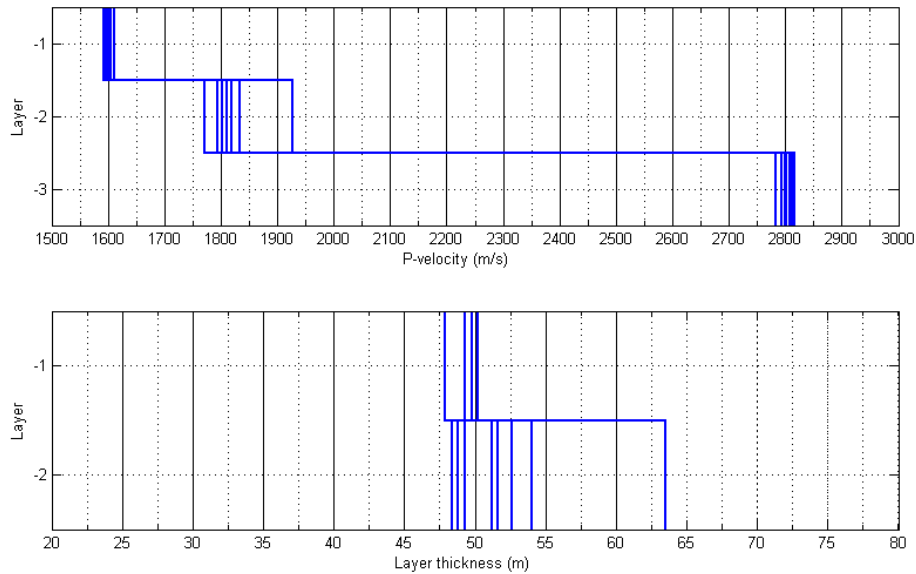


Figure 5.10: The 10 best individuals when only the thickness and P-speed is estimated with OSIRIS as forward model

In Figure 5.10 the shear speed is assumed to be known. Comparing Figure 5.8 and 5.10 suggests that estimating fewer parameters at once makes it easier to estimate the parameters correct. Both the top and bottom layer is estimated perfectly, and the estimated parameter values of the mid layer are spread over such a limited number of values that deciding on the correct one should be possible.

## 5.4 5 layers

Since the Matlab GUI made for the program has the option of choosing the number of layers in the geological model the forward models have to be tested with more layers. Since the ray tracing program only works with 3 layers, the Caiti code and OSIRIS are the only forward models tested.

### 5.4.1 Caiti

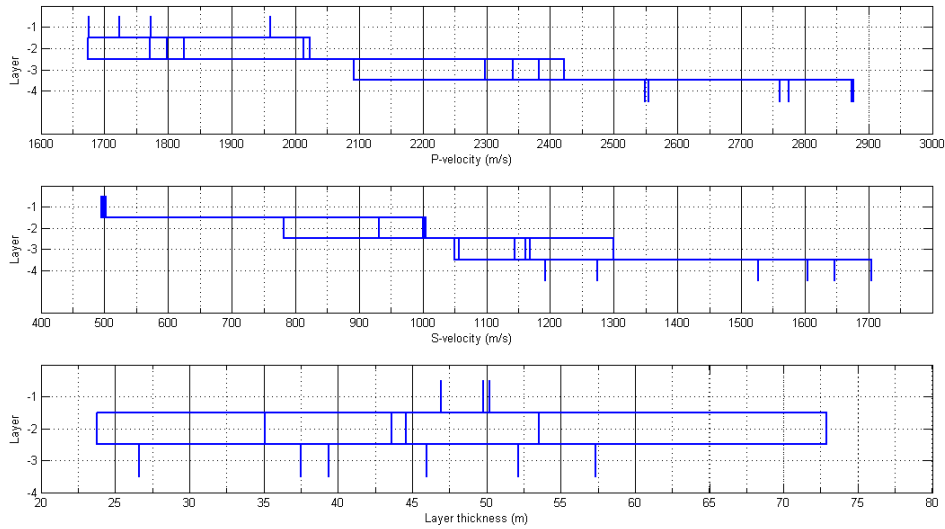


Figure 5.11: The best parameter set from the 10 best runs with Caiti's code as forward model with 5 layers

Comparing Figure 5.11 and Figure 5.3 it seems that the calculations have become more complex, and it is harder to estimate the parameters when the numbers of layers increase. But in the same way as earlier the S-speed is estimated correct for the top layer, which supports the fact that the top layer is the most important factor in the surface wave.

Figure 5.12 shows the same effects which have been discussed earlier. A fact which has not been discussed earlier is the well estimated thickness of the top layer with the dispersion model as forward model. The reason for this is that the surface wave will decrease exponentially with increasing distance from the interface [4]. This means that the contributions from deep layers will be insignificant to the dispersion, and will have a small significance in the calculations, and be hard to estimate correctly.

It is important to notice that the inversion still only takes 5.5 min with the additional layer.



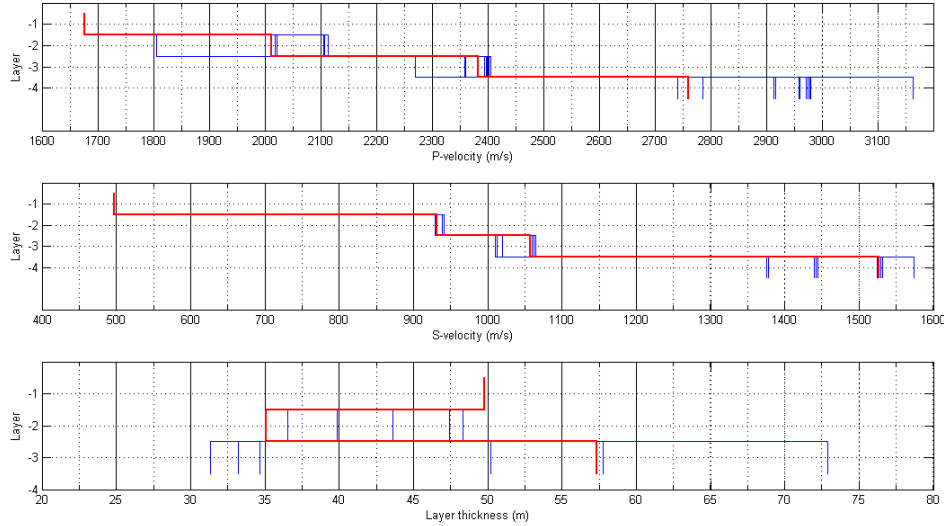


Figure 5.12: The best run with the Caiti code as forward model with 5 layers

## 5.4.2 OSIRIS

It is hard to draw any conclusions from Figure 5.13 and 5.14, but what can be said is that it obviously becomes harder to estimate the parameters with a higher number of layers. Figure 5.14 shows that it is possible to find approximately the parameter values, but with a number of runs giving such a wide span of results as Figure 5.13 it is almost impossible to be certain that the inversion is correct.

Figure 5.8 and 5.13 suggests that the layer thickness is hard to estimate with OSIRIS as forward model. But the top layer is estimated well in Figure 5.9 and 5.14 which means that the best runs usually will estimate the thickness quite well. The challenge then is to decide which model is the best when real data is used.

Adding extra layers increases the runtime to 3 hours for the inversion with OSIRIS as forward model, which is not a lot, but 10 runs with 30 min extra means 5 more hours' runtime.

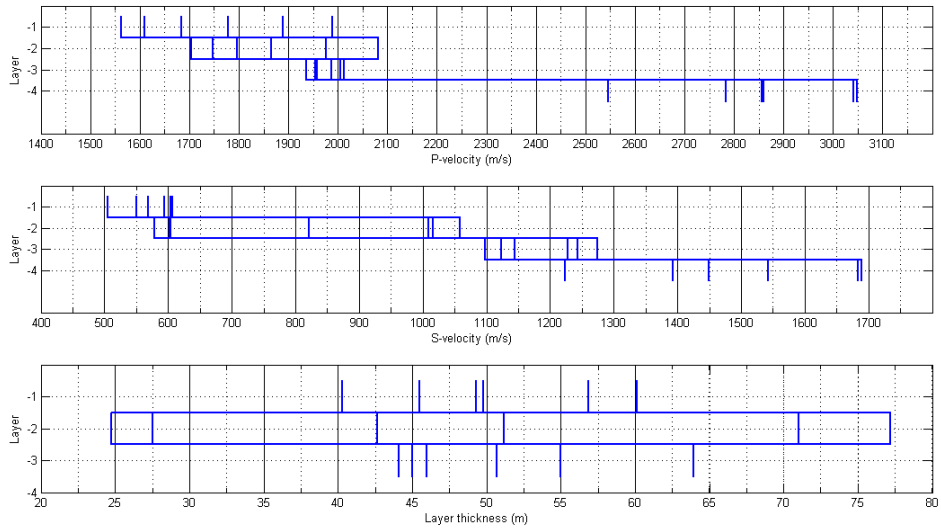


Figure 5.13: The best parameter set from the 10 best runs with OSIRIS as forward model with 5 layers

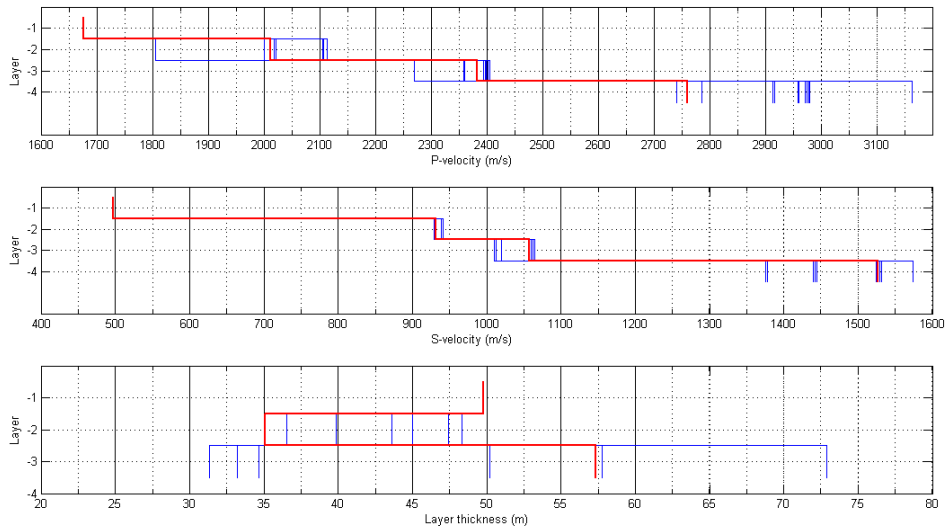


Figure 5.14: The best run with OSIRIS as forward model with 5 layers

# Chapter 6

## Conclusion

All the forward models have strengths and weaknesses, and this makes it hard to give absolute recommendations on which to use. The set of measured data will limit the choices when real data is used, but since this thesis has focused on synthetic data the limitations based on source-receiver layout are not taken into consideration.

The PlaneRay program made by Jens Hovem is still on a concept base. It is made to show that it is possible to make a range independent ray tracing program for the water column. Calculation time and a possibility of having several layers have not been the prime concerns in making the program. This is understandable, but if PlaneRay is going to be used in a seismic inversion these issues has to be addressed. Shortening the calculation time should be possible for a good programmer, and adding a program to calculate the reflection coefficient for a variable number of layers should also be a manageable task. With these improvements the PlaneRay program might be a good ray tracing forward model. But as long as it takes 9 hours to estimate three parameters will PlaneRay not be an option as a ray tracing forward model.

The two additional working forward models, OSIRIS and the Caiti code, seem to serve different purposes. When the Caiti code is used as forward model the shear speed is estimated well, and the inversion is really fast, but if the compressional speed is needed in addition to the shear speed a different forward model has to be used. The OSIRIS program seems to estimate both the compressional and shear speed reasonably well, even though none of the estimations are exceptionally good. 2-3 hours is not an unreasonable time to spend on one run of the inversion, but the fact that one run is not enough to be sure a parameter is estimated right makes the entire process a bit to slow.

Estimation of the layer thickness seems to be a bit of a problem area for both the

Caiti code and OSIRIS. The top layer is well estimated by the Caiti code, but the bottom layer seems to be a bit harder to estimate. The OSIRIS results span the entire search area, which suggests that the Caiti code is the preferred model if one needs the layer thickness.

One solution to get a more certain result from the inversion might be to run the inversion with two different forward models. Then it is very important that the measured data contains data which can be compared with the synthetic data calculated from both the forward models. The Caiti model estimates the shear speed well, and by using this estimate of the shear speed in a run with OSIRIS as forward model a good estimation of the compressional speed can also be obtained.

## **Future work**

The forward models are not tested together with real measured data. To be certain they perform equally well with real data is important to implement a possibility of testing real data as well in the future. Real data is noisy, and might give different results from the synthetic data, and it is therefore important to test the inversion with real data.

Using statistical methods to determine which parameter value to choose after the inversion might also be implemented. Gerstoft suggests a method for this in [13]. This is to ensure a higher probability of picking the right value when we use real data containing noise.

# Bibliography

- [1] Fredrik Helland. Seismisk inversjon ved bruk av genetiske algoritmer. Master's thesis, NTNU, 2006.
- [2] Peter Gerstoft. Inversion of seismoacoustic data using genetic algorithms and *a posteriori* probability distributions. *J. Acoust. Soc. Am.*, 95, 1994.
- [3] David A. Coley. *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific Publishing Co. Pte. Ltd., 1999.
- [4] Jens M. Hovem. *Marine Acoustics, Part 2*. NTNU, 2005.
- [5] Keiiti Aki and Paul G. Richards. *Quantitative Seismology*. University Science Books, 2002.
- [6] Norman A. Haskell. The dispersion of surface waves on multi-layered media. *Bullet. Seismolog. Soc. of America*, 43, 1953.
- [7] Jens M. Hovem. *Marine Acoustics, Part 1*. NTNU, 2005.
- [8] Hefeng Dong Jens M. Hovem and Xiukun Li. A forward model for geoacoustic inversion based on ray tracing and plane-wave reflection coefficients. *Acoustic Sensing Techniques for the Shallow Water Environment*, 2006.
- [9] Lars Ivar Leivestad. Shear wave velocity inversion from the group velocities of seismic surface waves. Master's thesis, NTNU, 2006.
- [10] John W. Dunkin. Computation of modal solutions in layered, elastic media at high frequencies. *Bulletin of the Seismological Society of America*, 55, 1965.
- [11] T. H. Watson. A note on fast computation of rayleigh wave dispersion in the multilayered elastic half-space. *Bulletin of the Seismological Society of America*, 60, 1970.

- [12] Tuncay Akal Andrea Caiti and Robert D. Stoll. Estimation of shear wave velocity in shallow marine sediments. *IEEE Journal of Oceanic Engineering*, 19, 1994.
- [13] Peter Gerstoft and Christoph F. Mecklenbäuker. Ocean acoustic inversion with estimation of *a posteriori* probability distributions. *J. Acoust. Soc. Am.*, 1998.

# Appendix A

## Variables

### A.1 The geological models

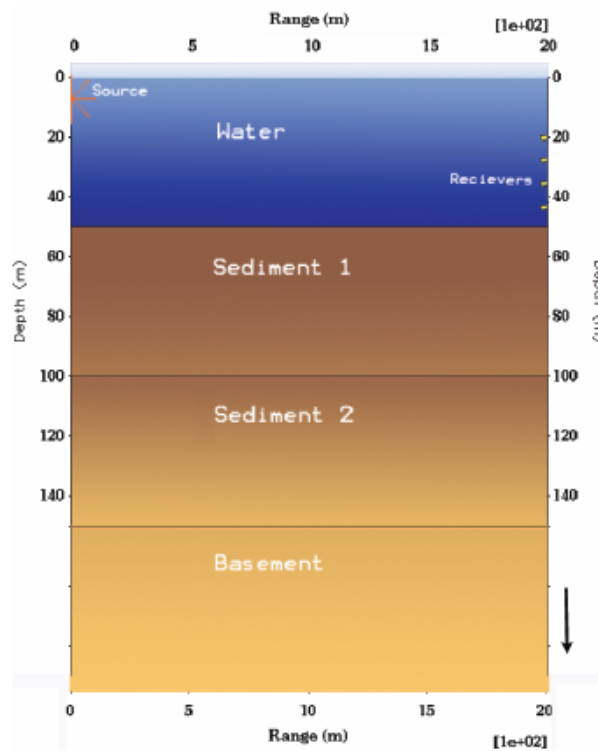


Figure A.1: Geological model

Table A.1: 4 layer geological model used to find the synthetic data  
 Data used in all models Used in OSIRIS

	Depth	P-speed	S-speed	Density	P-att.	S-att.
	[m]	[m/s]	[m/s]	[kg/m <sup>3</sup> ]	[dB/λ]	[dB/λ]
<b>Layer 0</b>	0	1500	0	1000	.01	...
<b>Layer 1</b>	50	1600	500	1600	.5	.8
<b>Layer 2</b>	100	1800	1000	2000	.5	.8
<b>Layer 3</b>	150	2800	1500	2200	.5	.8

Table A.2: 5 layer geological model used to find the synthetic data.  
 Data used in all models Used in OSIRIS

	Depth	P-speed	S-speed	Density	P-att.	S-att.
	[m]	[m/s]	[m/s]	[kg/m <sup>3</sup> ]	[dB/λ]	[dB/λ]
<b>Layer 0</b>	0	1500	0	1000	.01	...
<b>Layer 1</b>	50	1600	500	1600	.5	.8
<b>Layer 2</b>	100	1800	1000	2000	.5	.8
<b>Layer 3</b>	150	2200	1200	2100	.5	.8
<b>Layer 4</b>	200	2800	1500	2200	.5	.8

Table A.3: The discrete representation with 4 layers

	P-speed		S-speed		Layer thickness	
	[m/s]	Number of bits	[m/s]	Number of bits	[m]	Number of bits
Layer 1	1400-2000	512 (9 bit)	400-900	512 (9 bit)	20-80	128 (7 bit)
Layer 2	1650-2200	256 (8 bit)	520-1120	256 (8 bit)	20-80	128 (7 bit)
Layer 3	2500-3200	256 (8 bit)	1110-1710	256 (8 bit)	...	...

Table A.4: The discrete representation with 5 layers

	P-speed		S-speed		Layer thickness	
	[m/s]	Number of bits	[m/s]	Number of bits	[m]	Number of bits
Layer 1	1400-2000	512 (9 bit)	400-900	512 (9 bit)	20-80	128 (7 bit)
Layer 2	1650-2200	256 (8 bit)	520-1120	256 (8 bit)	20-80	128 (7 bit)
Layer 3	1900-2600	256 (8 bit)	700-1300	256 (8 bit)	20-80	128 (7 bit)
Layer 4	2500-3200	256 (8 bit)	1110-1710	256 (8 bit)	...	...



## A.2 Source - receiver geometries

### OSIRIS

Parameter	Value
Source type	Point source
Source depth	7 m
Distance to receivers	2000 m
Depth of first receiver	20 m
Number of vertical receivers	4
Distance between the receivers	6 m
<b>Time domain</b>	
Trace length	8.192 s
Sample time	0.002 s
Max center frequency of source wavelet	35,7 Hz
<b>Frequency domain</b>	
Lower frequency	0 Hz
Upper frequency	125 Hz
Delta f ("Frequency step")	0.12207 Hz

### PlaneRay

Parameter	Value
Source type	Ricker source
Source depth	25 m
Depth of receivers	25 m
Distance to first receiver	100 m
Distance between receivers	500 m
Number of horizontal receivers	6

## A.3 CASE I geological model in Haskell's article

	Layerthickness [m]	Density [kg/m <sup>3</sup> ]	P-speed [m/s]	S-speed [m/s]
<b>Layer 1</b>	13 600	2 700	6 140	3 390
<b>Layer 2</b>	11 850	2 700	5 500	3 180
<b>Layer 3</b>	inf	3 000	8 260	4 650



# Appendix B

## Deriving the Thomson-Haskell method

Some information about each layer is needed to be able to calculate the dispersion curve. These variables have to be stated for all of the  $n$  layers. The variables needed for the  $m$ 'th layer is

$\rho_m$  = density

$d_m$  = layer thickness

$v_{pm}$  = compressional speed

$v_{sm}$  = shear wave speed

The four variables which has boundary conditions connected to them are

$u$  = displacement component in the x direction

$w$  = displacement component in the z direction

$\sigma$  = normal pressure

$\tau$  = shear pressure

These four variables can be expressed by the derivated of the general solution of the pressure and shear wave equation.

Continuity of the two displacement components are achieved by ensuring the speed components  $dv/dt$  and  $dw/dt$  are continuous. Which give the following equations<sup>1</sup>

$$\begin{aligned} \dot{u}/c = & -(v_{pm}/c)^2[(\phi'_m + \phi''_m) \cos(kr_{v_{pm}}z) - i(\phi'_m - \phi''_m) \sin(kr_{v_{pm}}z)] \\ & - \gamma_m r_{v_{sm}}[(\Psi'_m - \Psi''_m) \cos kr_{v_{sm}}z - i(\Psi'_m + \Psi''_m) \sin kr_{v_{pm}}z], \end{aligned} \quad (\text{B.1})$$

$$\begin{aligned} \dot{w}/c = & -(v_{pm}/c)^2 r_{v_{pm}}[-i(\phi'_m + \phi''_m) \sin kr_{v_{pm}}z + (\phi'_m - \phi''_m) \cos kr_{v_{pm}}z] \\ & + \gamma_m[-i(\Psi'_m - \Psi''_m) \sin kr_{v_{sm}}z + (\Psi'_m + \Psi''_m) \cos kr_{v_{sm}}z], \end{aligned} \quad (\text{B.2})$$

$$\begin{aligned} \sigma = & -\rho_m v_{pm}^2 (\gamma_m - 1)[(\phi'_m + \phi''_m) \cos kr_{v_{pm}}z - i(\phi'_m - \phi''_m) \sin kr_{v_{pm}}z] \\ & - \rho_m c^2 \gamma_m^2 r_{v_{sm}}[(\Psi'_m - \Psi''_m) \cos kr_{v_{sm}}z \\ & - i(\Psi'_m + \Psi''_m) \sin kr_{v_{pm}}z], \end{aligned} \quad (\text{B.3})$$

$$\begin{aligned} \tau = & \rho_m v_{pm}^2 \gamma_m r_{v_{pm}}[-i(\phi'_m + \phi''_m) \sin kr_{v_{pm}}z + (\phi'_m - \phi''_m) \cos kr_{v_{pm}}z] \\ & - \rho_m c^2 \gamma_m (\gamma_m - 1)[-i(\Psi'_m - \Psi''_m) \sin kr_{v_{sm}}z \\ & + (\Psi'_m + \Psi''_m) \cos kr_{v_{sm}}z], \end{aligned} \quad (\text{B.4})$$

where  $\phi'_m$  and  $\phi''_m$  is the magnitude of the solution of the compressional wave equation, and  $\Psi'_m$  and  $\Psi''_m$  is the magnitude of the shear wave equation solution.  $r_{v_{pm}}$ ,  $r_{v_{sm}}$  and  $\gamma_m$  are constants calculated from the given variables for each layer

$$r_{v_{pm}} = \begin{cases} +\sqrt{(c/v_{pm})^2 - 1} & \text{if } c > v_{pm} \\ -i\sqrt{1 - (c/v_{pm})^2} & c < v_{pm} \end{cases} \quad (\text{B.5})$$

$$r_{v_{sm}} = \begin{cases} +\sqrt{(c/v_{sm})^2 - 1} & \text{if } c > v_{sm} \\ -i\sqrt{1 - (c/v_{sm})^2} & c < v_{sm} \end{cases} \quad (\text{B.6})$$

$$\gamma_m = 2(v_{sm}/c)^2. \quad (\text{B.7})$$

By looking at an interface  $m - 1$ , which means at  $z = 0$ , we can find the  $E_m$  matrix

---

<sup>1</sup> $\dot{u}/c$  and  $\dot{w}/c$  is used in [6] because they are dimensionless

$$(\dot{u}_{m-1}/c, \dot{w}_{m-1}/c, \sigma_{m-1}, \tau_{m-1}) = E_m(\phi'_m + \phi''_m, \phi'_m - \phi''_m, \Psi'_m - \Psi''_m, \Psi'_m + \Psi''_m) \quad (\text{B.8})$$

$$E_m = \begin{bmatrix} -(v_{pm}/c)^2 & 0 & -\gamma_m r_{vsm} & 0 \\ 0 & -(v_{pm}/c)^2 r_{vpm} & 0 & \gamma_m \\ -\rho_m v_{pm}^2 (\gamma_m - 1) & 0 & -\rho_m c^2 \gamma_m^2 r_{vsm} & 0 \\ 0 & \rho_m v_{pm}^2 \gamma_m r_{vpm} & 0 & -\rho_m c^2 \gamma_m (\gamma_m - 1) \end{bmatrix} \quad (\text{B.9})$$

Substituting  $z = d_m$  in Equation B.1 to B.4 gives the values at the  $m$ 'th interface. From some rewritings the following linear connection between the values  $\dot{u}/c$ ,  $\dot{w}/c$ ,  $\sigma$  and  $\tau$  at the top and bottom of a layer  $m$  emerges

$$(\dot{u}_m/c, \dot{w}_m/c, \sigma_m, \tau_m) = D_m E_m^{-1} (\dot{u}_{m-1}/c, \dot{w}_{m-1}/c, \sigma_{m-1}, \tau_{m-1}) \quad (\text{B.10})$$

Calling  $a_m = D_m E_m^{-1}$  gives the following

$$\begin{aligned} (a_m)_{11} &= \gamma_m \cos P_m - (\gamma_m - 1) \cos Q_m \\ (a_m)_{12} &= i[(\gamma_m - 1)r_{vpm}^{-1} \sin P_m + \gamma_m r_{vsm} \sin Q_m] \\ (a_m)_{13} &= -(\rho_m c^2)^{-1} (\cos P_m - \cos Q_m) \\ (a_m)_{14} &= i(\rho_m c^2)^{-1} (r_{vpm}^{-1} \sin P_m + r_{vsm} \sin Q_m) \\ (a_m)_{21} &= -i[\gamma_m r_{vpm} \sin P_m + (\gamma_m - 1)r_{vsm}^{-1} \sin Q_m] \\ (a_m)_{22} &= -(\gamma_m - 1) \cos P_m + \gamma_m \cos Q_m \\ (a_m)_{23} &= i(\rho_m c^2)^{-1} (r_{vpm} \sin P_m + r_{vsm}^{-1} \sin Q_m) \\ (a_m)_{24} &= (a_m)_{13} \\ (a_m)_{31} &= \rho_m c^2 \gamma_m (\gamma_m - 1) (\cos P_m - \cos Q_m) \\ (a_m)_{32} &= i\rho_m c^2 [(\gamma_m - 1)^2 r_{vpm}^{-1} \sin P_m + \gamma_m^2 r_{vsm} \sin Q_m] \\ (a_m)_{33} &= (a_m)_{22} \\ (a_m)_{34} &= (a_m)_{12} \\ (a_m)_{41} &= i\rho_m c^2 [\gamma_m^2 r_{vpm} \sin P_m + (\gamma_m - 1)^2 r_{vsm}^{-1} \sin Q_m] \\ (a_m)_{42} &= (a_m)_{31} \\ (a_m)_{43} &= (a_m)_{21} \\ (a_m)_{44} &= (a_m)_{11}, \end{aligned}$$

where  $P_m = kr_{vpm} d_m$  and  $Q_m = kr_{vsm} d_m$ .

From the boundary conditions we know that at the top of layer  $m$ ,  $\dot{u}/c$ ,  $\dot{w}/c$ ,  $\sigma$  and  $\tau$  has to be the same values as at the bottom of layer  $m - 1$  [6]. By using this in equation B.10 do we get the following equation for the  $n$ 'th layer

$$(\phi'_n + \phi''_n, \phi'_n - \phi''_n, \Psi'_n - \Psi''_n, \Psi'_n + \Psi''_n) = E_n^{-1} a_{n-1} a_{n-2} \cdots a_1 (\dot{u}_0/c, \dot{w}_0/c, \sigma_0, \tau_0). \quad (\text{B.11})$$

This is a general equation, but we are only interested in the case where there is no pressure over the free surface, which means that  $\sigma_0 = \tau_0 = 0$ , and there will in addition be no reflections from the bottom layer,  $\phi''_n = \Psi''_n = 0$ . By adding this, and setting  $A = a_{n-1} a_{n-2} \cdots a_1$ , a few calculations give

$$-(\dot{u}_0/\dot{w}_0) = K/L = M/N, \quad (\text{B.12})$$

where

$$\begin{aligned} K &= \gamma_n r_{v_{pn}} A_{12} + (\gamma_n - 1) A_{22} - r_{v_{pn}} A_{32}/\rho_n c^2 + A_{42}/\rho_n c^2 \\ L &= \gamma_n r_{v_{pn}} A_{11} + (\gamma_n - 1) A_{21} - r_{v_{pn}} A_{31}/\rho_n c^2 + A_{41}/\rho_n c^2 \\ M &= -(\gamma_n - 1) A_{12} + \gamma_n r_{v_{sn}} A_{22} + A_{32}/\rho_n c^2 + r_{v_{sn}} A_{42}/\rho_n c^2 \\ N &= -(\gamma_n - 1) A_{11} + \gamma_n r_{v_{sn}} A_{21} + A_{31}/\rho_n c^2 + r_{v_{sn}} A_{41}/\rho_n c^2 \end{aligned} \quad (\text{B.13})$$

## Fluid layers

Fluid layers are not a prioritized subject in Haskell's article [6]. Shear speed will not exist in fluid layers, and the calculation method used to find the  $a$ -matrix can not be used. Haskell introduces an other calculation method which avoids the problem, and gets a new  $a$ -matrix

$$\mathbf{a}_m = \begin{bmatrix} 0 & -ir_{v_{pm}}^{-1} \sin P_m & -(\rho_m c^2)^{-1} \cos P_m & 0 \\ 0 & \cos P_m & ir_{v_{pm}} (\rho_m c^2)^{-1} \sin P_m & 0 \\ 0 & i\rho_m c^2 r_{v_{pm}}^{-1} \sin P_m & \cos P_m & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{B.14})$$

Equation B.12 can not be solved since  $L$  and  $N$  become zero. Haskell gives no solution to this problem. My initial solution was to do the calculations preceding Equation B.12 in [6]. Then one gets a new equation

$$-(\dot{u}_0/\dot{w}_0) = K = M. \tag{B.15}$$

But this equation can not be correct because the parts of the solution with the large magnitudes do not get cancelled as they are supposed to be. We end up with magnitudes of the order  $10^2 - 10^{102}$ , and we are supposed to get a result of order  $10^{-10} - 10^{-1}$ . Something is wrong, but I have not been able to find where the error resides.





## Appendix C

### Flow diagrams of the inversion code

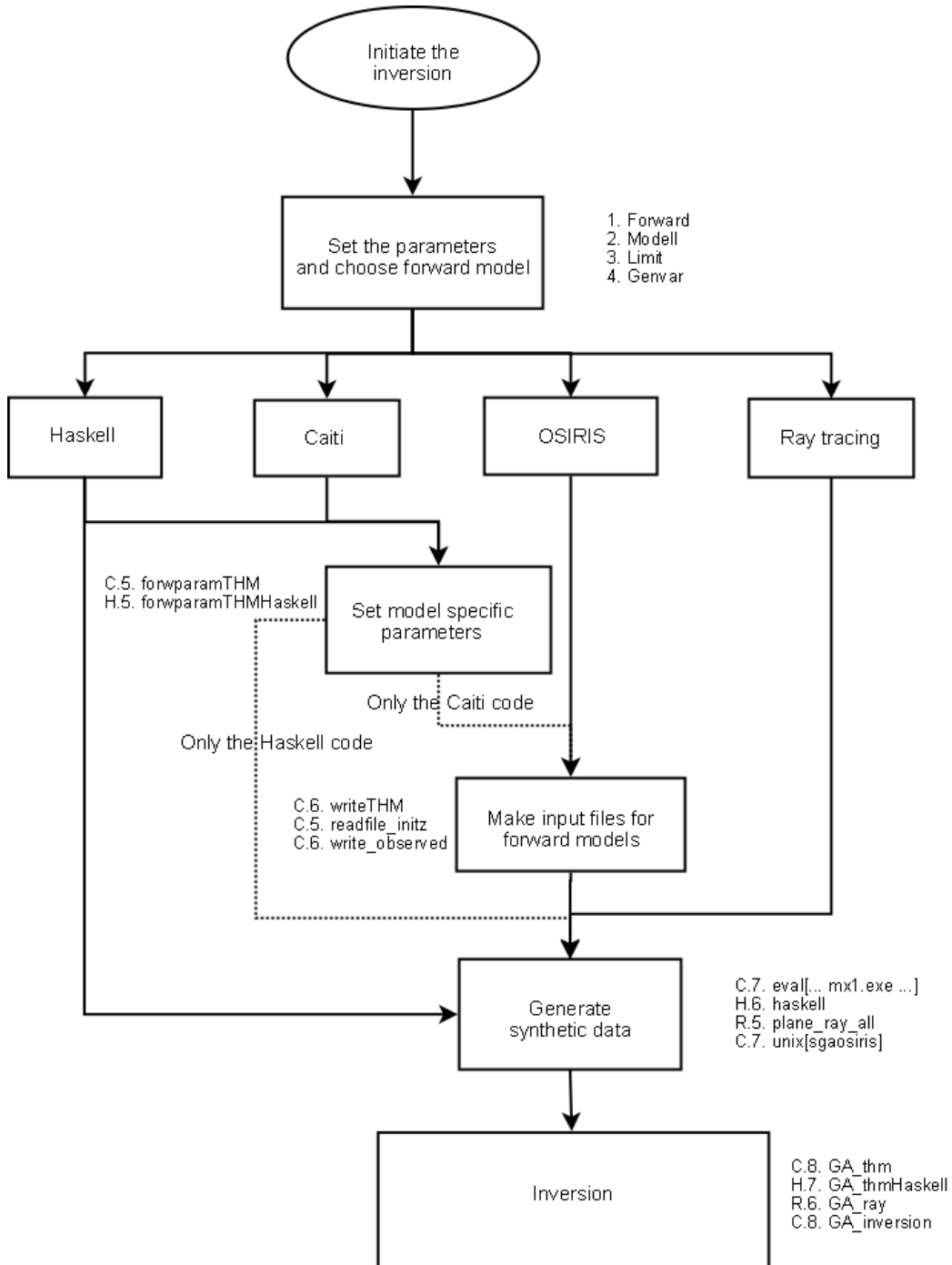


Figure C.1: The initialization of the seismic inversion

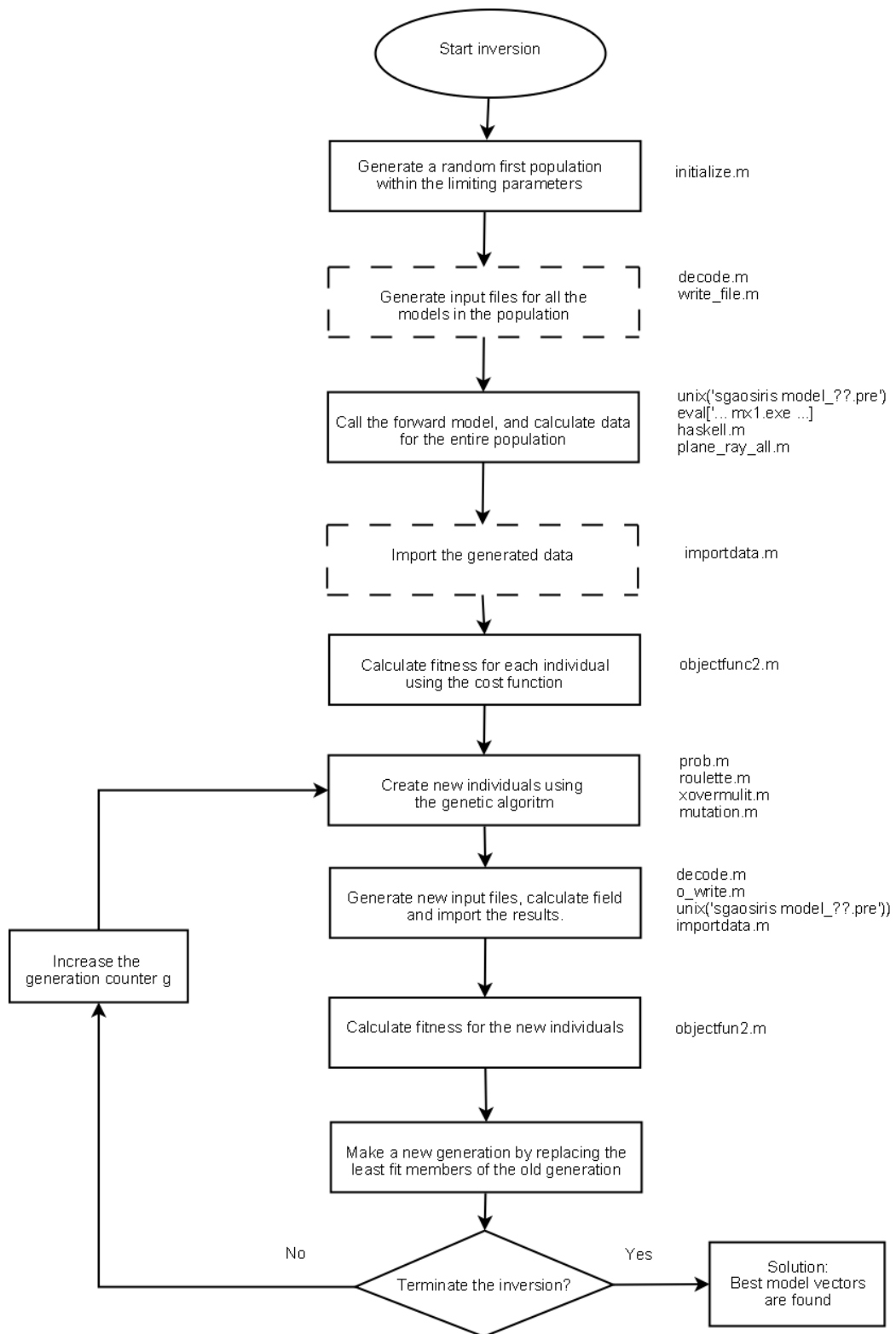


Figure C.2: The seismic inversion